Consistent Biclustering for Feature Selections and Supervised Classifications

Antonio Mucherino

IRISA, INRIA and University of Rennes 1

Classification Day 1 **CD1**
November 10^{th}, 2016

Outline

Outline

IRISA
UNIVERSITE DE RENNES1

Samples, features and vectors

Samples can be represented by vectors.

$$\mathbf{v} = \{v^1, v^2, v^3, \dots, v^m\}$$

The generic component v^i of the vector $\mathbf{v}$ represent the i^{th} feature of the sample.

For example, a feature can be:

- the expression of a gene
- the measure of a chemical component
- a pixel of a matrix representing an image
- ...

Biclustering

A set of samples can be represented by a set of vectors:

$$\{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$$

or by a matrix containing all their features:

$$A = \begin{pmatrix} a_1^1 & a_2^1 & \ldots & a_n^1 \\ a_1^2 & a_2^2 & \ldots & a_n^2 \\ a_1^3 & a_2^3 & \ldots & a_n^3 \\ \ldots & \ldots & \ldots & \ldots \\ a_1^m & a_2^m & \ldots & a_n^m \end{pmatrix}$$

- Each column of the matrix represents a *sample*.
- Each row of the matrix represent a *feature*.

Biclustering

Definition

Biclusters are sub-matrices of the matrix A.

Biclusters having different properties can be of interest:

- biclusters with constant values;
- biclusters with constant row or column values;
- biclusters with “coherent” values.

Definition

A partition in biclusters of A is referred to as **biclustering**.

The problem of finding a biclustering of a given set of data A can be formulated as an optimization problem.

Some notations

The matrix A represents our set of data.

- j, the generic index for the samples (n in total)
- i, the generic index for the features (m in total)
- a^j, the j^{th} column of the matrix A, i.e. the j^{th} sample
- a_i, the i^{th} row of the matrix A, i.e. the i^{th} feature
- a_i^j, the generic element of the matrix A, representing the i^{th} feature of the j^{th} sample

Some notations

- S_r, class of samples (we can suppose it's a set of indices "j")
- $B_S = \{S_1, S_2, \ldots, S_k\}$, classification of the samples
- F_r, class of features (we can suppose it's a set of indices "i")
- $B_F = \{F_1, F_2, \ldots, F_k\}$, classification of the features
- $A[F_r, S_r]$ is a submatrix of A representing the **bicluster** (S_r, F_r)
- $\mathbb{B} = \{(F_1, S_1), (F_2, S_2), \ldots, (F_k, S_k)\}$ is a **biclustering** consisting of k biclusters.

Outline

IRISA
UNIVERSITE DE RENNES1

Constructing biclusterings

Given a classification for the *samples*:

$$B_S = \{S_1, S_2, \ldots, S_k\},$$

we can define a classification B_F for the *features* by imposing:

$$i \in F_{\hat{r}} \quad \Longleftrightarrow \quad \forall \xi \neq \hat{r} \quad \frac{1}{|S_{\hat{r}}|} \sum_{j \in S_{\hat{r}}} a_i^j > \frac{1}{|S_{\hat{\xi}}|} \sum_{j \in S_{\hat{\xi}}} a_i^j.$$

A **biclustering** can be generated by combining B_F and B_S.

Constructing biclusterings

Let A be a training set.

$$A = \begin{pmatrix} a_1^1 & a_2^1 & a_3^1 & a_4^1 & a_5^1 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 & a_5^2 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 & a_5^3 \\ a_1^4 & a_2^4 & a_3^4 & a_4^4 & a_5^4 \\ a_1^5 & a_2^5 & a_3^5 & a_4^5 & a_5^5 \end{pmatrix}$$

We employ a supervised technique for constructing

- a classification B_F for the features in A;
- and hence, a biclustering $\mathbb{B}$ for the matrix A.

Constructing biclusterings

- What are the samples for which these features are mostly expressed?
- What is the class S_r of samples for which these features are mostly expressed?
- Let us suppose the **blue class** is mostly expressed by these features.

$$A = \begin{pmatrix} a_1^1 & a_2^1 & a_3^1 & a_4^1 & a_5^1 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 & a_5^2 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 & a_5^3 \\ a_1^4 & a_2^4 & a_3^4 & a_4^4 & a_5^4 \\ a_1^5 & a_2^5 & a_3^5 & a_4^5 & a_5^5 \end{pmatrix}$$

Constructing biclusterings

- What are the samples for which these features are mostly expressed?
- What is the class S_r of samples for which these features are mostly expressed?
- Let us suppose the **blue class** is mostly expressed by these features.

$$A = \begin{pmatrix} a_1^1 & a_2^1 & a_3^1 & a_4^1 & a_5^1 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 & a_5^2 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 & a_5^3 \\ a_1^4 & a_2^4 & a_3^4 & a_4^4 & a_5^4 \\ a_1^5 & a_2^5 & a_3^5 & a_4^5 & a_5^5 \end{pmatrix}$$

Constructing biclusterings

- What are the samples for which these features are mostly expressed?
- What is the class S_r of samples for which these features are mostly expressed?
- Let us suppose now the **red class** is mostly expressed by these features.

$$A = \begin{pmatrix} a_1^1 & a_2^1 & a_3^1 & a_4^1 & a_5^1 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 & a_5^2 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 & a_5^3 \\ a_1^4 & a_2^4 & a_3^4 & a_4^4 & a_5^4 \\ a_1^5 & a_2^5 & a_3^5 & a_4^5 & a_5^5 \end{pmatrix}$$

Constructing biclusterings

- What are the samples for which these features are mostly expressed?
- What is the class S_r of samples for which these features are mostly expressed?
- Let us suppose now the **red class** is mostly expressed by these features.

$$A = \begin{pmatrix} a_1^1 & a_2^1 & a_3^1 & a_4^1 & a_5^1 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 & a_5^2 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 & a_5^3 \\ a_1^4 & a_2^4 & a_3^4 & a_4^4 & a_5^4 \\ a_1^5 & a_2^5 & a_3^5 & a_4^5 & a_5^5 \end{pmatrix}$$

Constructing biclusterings

By continuing and properly sorting the rows and the columns of A, we can identify a *biclustering* of A.

$$A = \begin{pmatrix} \mathbf{a}_1^1 & \mathbf{a}_2^1 & \mathbf{a}_3^1 & \mathbf{a}_4^1 & \mathbf{a}_5^1 \\ \mathbf{a}_1^2 & \mathbf{a}_2^2 & \mathbf{a}_3^2 & \mathbf{a}_4^2 & \mathbf{a}_5^2 \\ \mathbf{a}_1^3 & \mathbf{a}_2^3 & \mathbf{a}_3^3 & \mathbf{a}_4^3 & \mathbf{a}_5^3 \\ \mathbf{a}_1^4 & \mathbf{a}_2^4 & \mathbf{a}_3^4 & \mathbf{a}_4^4 & \mathbf{a}_5^4 \\ \mathbf{a}_1^5 & \mathbf{a}_2^5 & \mathbf{a}_3^5 & \mathbf{a}_4^5 & \mathbf{a}_5^5 \end{pmatrix}$$

Constructing biclusterings

By continuing and properly sorting the rows and the columns of A, we can identify a *biclustering* of A.

$$A = \begin{pmatrix} a_1^1 & a_2^1 & a_3^1 & a_4^1 & a_5^1 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 & a_5^3 \\ a_1^4 & a_2^4 & a_3^4 & a_4^4 & a_5^4 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 & a_5^2 \\ a_1^5 & a_2^5 & a_3^5 & a_4^5 & a_5^5 \end{pmatrix}$$

Constructing biclusterings

By continuing and properly sorting the rows and the columns of A, we can identify a *biclustering* of A.

$$A = \begin{pmatrix} \mathbf{a}_1^1 & \mathbf{a}_2^1 & \mathbf{a}_4^1 & \mathbf{a}_3^1 & \mathbf{a}_5^1 \\ \mathbf{a}_1^3 & \mathbf{a}_2^3 & \mathbf{a}_4^3 & \mathbf{a}_3^3 & \mathbf{a}_5^3 \\ \mathbf{a}_1^4 & \mathbf{a}_2^4 & \mathbf{a}_4^4 & \mathbf{a}_3^4 & \mathbf{a}_5^4 \\ \mathbf{a}_1^2 & \mathbf{a}_2^2 & \mathbf{a}_4^2 & \mathbf{a}_3^2 & \mathbf{a}_5^2 \\ \mathbf{a}_1^5 & \mathbf{a}_2^5 & \mathbf{a}_4^5 & \mathbf{a}_3^5 & \mathbf{a}_5^5 \end{pmatrix}$$

IRISA
UNIVERSITE DE RENNES1

Performing supervised classifications

What is the point in constructing a biclustering?

The obtained classification of the features can be used for identifying classifications of samples that do not belong to the training set A.

Training set — *Validation set*

$$\begin{pmatrix} a_1^1 & a_2^1 & a_4^1 & a_3^1 & a_5^1 \\ a_1^3 & a_2^3 & a_4^3 & a_3^3 & a_5^3 \\ a_1^4 & a_2^4 & a_4^4 & a_3^4 & a_5^4 \\ a_1^2 & a_2^2 & a_4^2 & a_3^2 & a_5^2 \\ a_1^5 & a_2^5 & a_4^5 & a_3^5 & a_5^5 \end{pmatrix} \quad \begin{pmatrix} b_1^1 & b_2^1 & b_3^1 & b_4^1 & b_5^1 \\ b_1^2 & b_2^2 & b_3^2 & b_4^2 & b_5^2 \\ b_1^4 & b_2^4 & b_3^4 & b_4^4 & b_5^4 \\ b_1^3 & b_2^3 & b_3^3 & b_4^3 & b_5^3 \\ b_1^5 & b_2^5 & b_3^5 & b_4^5 & b_5^5 \end{pmatrix}$$

Performing supervised classifications

What is the point in constructing a biclustering?

The obtained classification of the features can be used for identifying classifications of samples that do not belong to the training set A.

Training set *Validation set*

$$\begin{pmatrix} a_1^1 & a_2^1 & a_4^1 & a_3^1 & a_5^1 \\ a_1^3 & a_2^3 & a_4^3 & a_3^3 & a_5^3 \\ a_1^4 & a_2^4 & a_4^4 & a_3^4 & a_5^4 \\ a_1^2 & a_2^2 & a_4^2 & a_3^2 & a_5^2 \\ a_1^5 & a_2^5 & a_4^5 & a_3^5 & a_5^5 \end{pmatrix} \quad \begin{pmatrix} b_1^1 & b_2^1 & b_3^1 & b_4^1 & b_5^1 \\ b_1^2 & b_2^2 & b_3^2 & b_4^2 & b_5^2 \\ b_1^4 & b_2^4 & b_3^4 & b_4^4 & b_5^4 \\ b_1^3 & b_2^3 & b_3^3 & b_4^3 & b_5^3 \\ b_1^5 & b_2^5 & b_3^5 & b_4^5 & b_5^5 \end{pmatrix}$$

Performing supervised classifications

What is the point in constructing a biclustering?

The obtained classification of the features can be used for identifying classifications of samples that do not belong to the training set A.

Training set   *Validation set*

$$\begin{pmatrix} a_1^1 & a_2^1 & a_4^1 & a_3^1 & a_5^1 \\ a_1^3 & a_2^3 & a_4^3 & a_3^3 & a_5^3 \\ a_1^4 & a_2^4 & a_4^4 & a_3^4 & a_5^4 \\ a_1^2 & a_2^2 & a_4^2 & a_3^2 & a_5^2 \\ a_1^5 & a_2^5 & a_4^5 & a_3^5 & a_5^5 \end{pmatrix} \qquad \begin{pmatrix} b_1^1 & b_2^1 & b_3^1 & b_4^1 & b_5^1 \\ b_1^2 & b_2^2 & b_3^2 & b_4^2 & b_5^2 \\ b_1^4 & b_2^4 & b_3^4 & b_4^4 & b_5^4 \\ b_1^3 & b_2^3 & b_3^3 & b_4^3 & b_5^3 \\ b_1^5 & b_2^5 & b_3^5 & b_4^5 & b_5^5 \end{pmatrix}$$

Performing supervised classifications

What is the point in constructing a biclustering?

The obtained classification of the features can be used for identifying classifications of samples that do not belong to the training set A.

Training set *Validation set*

$$\begin{pmatrix} a_1^1 & a_2^1 & a_4^1 & a_3^1 & a_5^1 \\ a_1^3 & a_2^3 & a_4^3 & a_3^3 & a_5^3 \\ a_1^4 & a_2^4 & a_4^4 & a_3^4 & a_5^4 \\ a_1^2 & a_2^2 & a_4^2 & a_3^2 & a_5^2 \\ a_1^5 & a_2^5 & a_4^5 & a_3^5 & a_5^5 \end{pmatrix} \quad \begin{pmatrix} b_1^1 & b_2^1 & b_3^1 & b_4^1 & b_5^1 \\ b_1^2 & b_2^2 & b_3^2 & b_4^2 & b_5^2 \\ b_1^4 & b_2^4 & b_3^4 & b_4^4 & b_5^4 \\ b_1^3 & b_2^3 & b_3^3 & b_4^3 & b_5^3 \\ b_1^5 & b_2^5 & b_3^5 & b_4^5 & b_5^5 \end{pmatrix}$$

Outline

Verifying the classification technique on A

The given procedure is able to perform the following steps:

$$B_S \longrightarrow B_F \longrightarrow \hat{B}_S.$$

Definition

If $B_S \equiv \hat{B}_S$, then the corresponding biclustering $\mathbb{B}$ is **consistent**.

Note that:

- supervised classifications should be performed only when the found biclustering is consistent;
- the verification on the consistency of the validation set can give clues about the correctness of the classifications.

Consistent biclusterings and real data

Validation sets from real-life applications usually do not admit consistent biclusterings.

How to overcome this problem?

A maximal subset of features can be selected so that the corresponding biclustering is consistent.

Why to maximize the number of selected features?

This is done in order to preserve information in the set of data.

What if our data are noisy?

We introduce the concepts of α-consistent and β-consistent biclusterings.

A feature selection problem (*consistency*)

Find the maximal subset of features such that the corresponding *biclustering* is *consistent*:

$$\max_x \left(f(x) = \sum_{i=1}^{m} x_i \right)$$

subject, $\forall \hat{r}, \xi \in \{1, 2, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}}$, to:

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}$$

where:

- x_i, binary decision variable, it is 0 if the i^{th} feature is not selected;
- f_{ir}, binary parameter, it is 1 if the i^{th} feature belongs to the r^{th} bicluster.

A feature selection problem (α-consistency)

And if the data are noisy?

Feature selection for α-consistent biclustering:

$$\max_x \left(f(x) = \sum_{i=1}^{m} x_i \right)$$

subject, $\forall \hat{r}, \xi \in \{1, 2, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}}$, to:

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \alpha_j + \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}$$

where $\alpha_j > 0$.

A feature selection problem (β-*consistency*)

And if the data are noisy?

Feature selection for β-consistent biclustering:

$$\max_x \left(f(x) = \sum_{i=1}^{m} x_i \right)$$

subject, $\forall \hat{r}, \xi \in \{1, 2, \ldots, k\}, \hat{r} \neq \xi, j \in S_{\hat{r}}$, to:

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \beta_j \times \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}$$

where $\beta_j > 1$.

It is supposed here that all a_{ij}'s are positive.

Problem complexity

Feature selection for

- consistent biclustering,
- α-consistent biclustering, and
- β-consistent biclustering

is NP-hard.

- Exponential complexity for exact methods;
- Two heuristic approaches were previously proposed in the literature;
- We presented a new VNS-based heuristic.

Outline

Analyzing the problem constraints

What do the denominators of the constraints represent?

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} > \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i}$$

- $y_{\hat{r}} = \sum_{i=1}^{m} f_{i\hat{r}} x_i$ is the number of selected features in $(F_{\hat{r}}, S_{\hat{r}})$;
- $y_{\xi} = \sum_{i=1}^{m} f_{i\xi} x_i$ is the number of selected features in (F_{ξ}, S_{ξ}).

What if we substitute the denominators with real variables?

Analyzing the problem constraints

Let us introduce new decision variables:

- y_r, $r \in \{1, 2, \ldots, k\}$, real.

The constraints of the optimization problem can be rewritten as:

$$\frac{1}{y_{\hat{r}}} \sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i > \frac{1}{y_{\xi}} \sum_{i=1}^{m} a_{ij} f_{i\xi} x_i .$$

For example, we can suppose that 700 features are selected in the first bicluster, while 300 features are selected in the second one.

Analyzing the problem constraints

Let us introduce new decision variables:

- y_r, $r \in \{1, 2, \ldots, k\}$, real.

The constraints of the optimization problem can be rewritten as:

$$\frac{1}{700} \sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i > \frac{1}{300} \sum_{i=1}^{m} a_{ij} f_{i\xi} x_i .$$

For example, we can suppose that 700 features are selected in the first bicluster, while 300 features are selected in the second one.

Analyzing the problem constraints

Let us introduce new decision variables:

- y_r, $r \in \{1, 2, \ldots, k\}$, real.

The constraints of the optimization problem can be rewritten as:

$$\frac{1}{7}\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i > \frac{1}{3}\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i.$$

We can of course modify the values while keeping the proportions ...700,300,..., as well as 7,3,...

Analyzing the problem constraints

Let us introduce new decision variables:

- y_r, $r \in \{1, 2, \ldots, k\}$, real.

The constraints of the optimization problem can be rewritten as:

$$\frac{1}{0.7} \sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i > \frac{1}{0.3} \sum_{i=1}^{m} a_{ij} f_{i\xi} x_i .$$

Each variable y_r represents the percentage of selected features in the biclusters $(F_{\hat{r}}, S_{\hat{r}})$ and (F_ξ, S_ξ).

Introducing a new penalty function

We introduce the following function:

$$c(x,\hat{r},\xi) = \sum_{j \in S_{\hat{r}}} \left| \frac{\sum_{i=1}^{m} a_{ij} f_{i\xi} x_i}{\sum_{i=1}^{m} f_{i\xi} x_i} - \frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i}{\sum_{i=1}^{m} f_{i\hat{r}} x_i} \right|_{+},$$

where:

- $|\cdot|_{+}$ represents the function which returns its argument if it is positive, and it returns 0 otherwise;
- c depends upon x and on one pair of biclusters for which $\hat{r} \neq \xi$.

The value of this function is strictly positive if and only if at least one of the constraints is not satisfied.

Bilevel program

Our bilevel reformulation:

$$\min_{y} \left(g(\mathbf{x}, \mathbf{y}) = \sum_{r=1}^{k} \left[(1 - \mathbf{y}_r) + \sum_{\xi=1:\xi\neq r}^{k} c(x, r, \xi) \right] \right)$$

subject to:

$$\mathbf{x} = \arg\max_{x} \left(f(x) = \sum_{i=1}^{m} x_i \right)$$

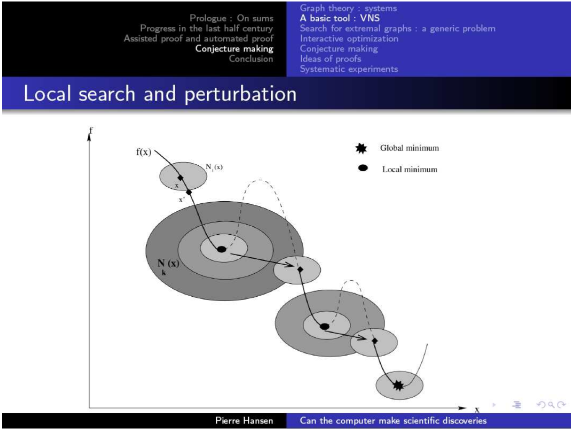
$$\text{subject to} \begin{cases} \sum_{i=1}^{m} f_{ir} x_i = \lfloor y_r \sum_{i=1}^{m} f_{ir} \rfloor & \forall r \in \{1, \dots, k\} \\ \frac{1}{y_{\hat{r}}} \sum_{i=1}^{m} a_{ij} f_{i\hat{r}} x_i > \frac{1}{y_\xi} \sum_{i=1}^{m} a_{ij} f_{i\xi} x_i, \end{cases}$$

$$\sum_{r=1}^{k} y_r \leq 1.$$

The same reformulation can be applied for α and β-consistency.

Outline

Variable Neighborhood Search (VNS)



from EURO10 lecture given by **Pierre Hansen**, GERAD, Montreal.

A heuristic algorithm

We developed a **VNS**-based heuristic for optimizing the outer function:

$$g(x, y)$$

- we perform a random search on the variables y_r
- the variables x_i are computed, at each iteration of the heuristic, by solving exactly the inner problem;
- therefore, the search domain of our heuristic corresponds to the k variables y_r only, where k is the number of biclusters.

A heuristic algorithm

let $iter = 0$;
let $x_i = 1, \forall i \in \{1, 2, \ldots, m\}$;
let $y_r = \sum_i f_{ir}/m, \forall r \in \{1, 2, \ldots, k\}$;
let $range = starting\_range$;
while $(g(x, y) > 0$ and $range \leq max\_range)$ **do**
  let $iter = iter + 1$;
  solve the inner optimization problem (linear);
  if $(g(x, y) > 0)$ **then**
    increase *range*;
    if $(g(x, y)$ has improved) **then**
      let $range = starting\_range$;
    end if
    let r' = random in $\{1, 2, \ldots, k\}$;
    choose randomly $y_{r'}$ in $[y_{r'} - range, y_{r'} + range]$;
    let r'' = random in $\{1, 2, \ldots, k\}$ such that $r' \neq r''$;
    set $y_{r''}$ so that $\sum_r y_r = 1$;
  end if
end while

An implementation in AMPL+CPLEX

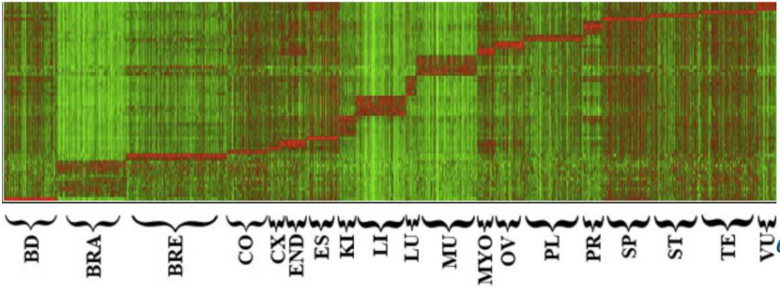
```
... ...

repeat while (stop < 1)
{
   printf "solving ... (";
   for {k in 1..r}
   {
      printf "%lf ",nf[k];
   };
   printf ") [tot = %d]\n",nf_tot;

   # solving the inner problem with CPLEX
   solve;

   display numb_features;

   let count := 0;
   for {i in 1..m}
   {
      if (x[i] >= 0.5) then
      {
         let count := count + 1;
      };
   };
   display count;

   # checking the consistency

   let cons := 0;
   let err := 0;
```

Outline

Analyzing microarray data

Microarrays in biology are used for studying the expression of genes under different conditions.

Microarray data can be represented by a matrix, for which we can identify a suitable biclustering:

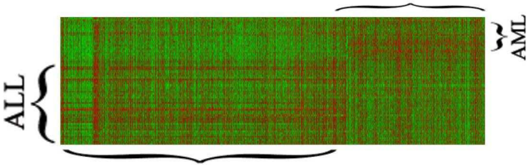


Human Gene Expression (HuGE) Index data set.

ALL-AML dataset

Consists of samples from patients diagnosed with two different diseases:

- *acute lymphoblastic leukemia* (ALL);
- *acute myeloid leukemia* (AML).



- *Training set*: 38 samples: 27 ALL and 11 AML;
- *Validation set*: 34 samples: 20 ALL and 14 AML;
- *Number of features*: 7129.

Computational experiments

Applying the heuristic algorithm for finding consistent, α-consistent and β-consistent biclusterings.

α	$f(x)$	*err*
0	7081	2
10	7076	2
20	7075	2
30	7072	2
40	7068	2
50	7061	1
60	7046	1
70	6954	1

β *	$f(x)$	*err*
1.001	7011	2
1.002	6984	2
1.003	6946	1
1.004	6702	1
1.005	5914	1
1.006	5072	1
1.007	4524	0
1.008	3932	0

err is the number of errors obtained while classifying the samples of the validation set accordingly to the found biclustering.

* data are here scaled to avoid the presence of negative a_{ij} values.
Total number of features: 7129.

Outline

UMR IRISA
UNIVERSITE DE RENNES1

Predicting problematic wine fermentations

Set of data obtained from a winery in Chile's Maipo Valley, which is the result of 24 measurements of industrial vinifications of Cabernet sauvignon.

- normal fermentations (9)
- slow fermentations (10)
- stuck fermentations (5)

Training set:

- Number of compounds: 30;
- Number of measurements per compound: 8 (before 150 hours);
- Total number of features: 240.

Predicting problematic wine fermentations

Applying the heuristic algorithm for finding consistent, α-consistent and β-consistent biclusterings.

α	0.00	0.10	0.20	0.40	0.50	0.70	1.00
$f(x)$	192	192	192	190	190	170	149

β	1.00	1.01	1.02	1.04	1.05	1.07	1.10
$f(x)$	192	191	186	186	180	177	165

The analysis showed that:

- important compounds: *sugar* and *lactic*, *malic*, *succinic*, and *tartaric* organic acids;
- not important compounds: *arginine*, *proline*, *glutamic acid*, *glutamine*, and *treonine*.

Some bibliography

- A. Mucherino, L. Liberti, *A VNS-based Heuristic for Feature Selection in Data Mining*. In: “Hybrid Meta-Heuristics”, Studies in Computational Intelligence **434**, E-G. Talbi (Ed.), 353–368, 2013.
- A. Mucherino, *Extending the Definition of β-Consistent Biclustering for Feature Selection*, IEEE Conference Proceedings, Federated Conference on Computer Science and Information Systems (FedCSIS11), Workshop on Computational Optimization (WCO11), Szczecin, Poland, 269–274, 2011.
- S. Busygin, O.A. Prokopyev, P.M. Pardalos, *Feature Selection for Consistent Biclustering via Fractional 0–1 Programming*, Journal of Combinatorial Optimization **10**, 7–21, 2005.

UMR IRISA
UNIVERSITE DE RENNES1

To be continued . . .

Thanks!