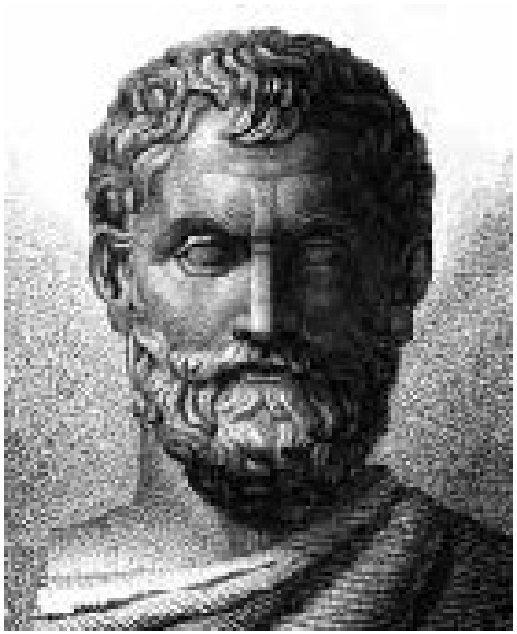
Clustering and Search Techniques in Large Scale Networks

Panos M. Pardalos
Distinguished Professor,
Center For Applied Optimization,
Industrial and Systems Engineering, University of Florida,
Florida, USA.
http://www.ise.ufl.edu/pardalos/
National Research University Higher School of Economics,
Laboratory of Algorithms and Technologies for Network Analysis,
Nizhny Novgorod, Russia
http://nnov.hse.ru/en/latna/



Thales of Miletus (624 BC -546 BC) one of the Seven Sages of Greece and the **first data analyst**!

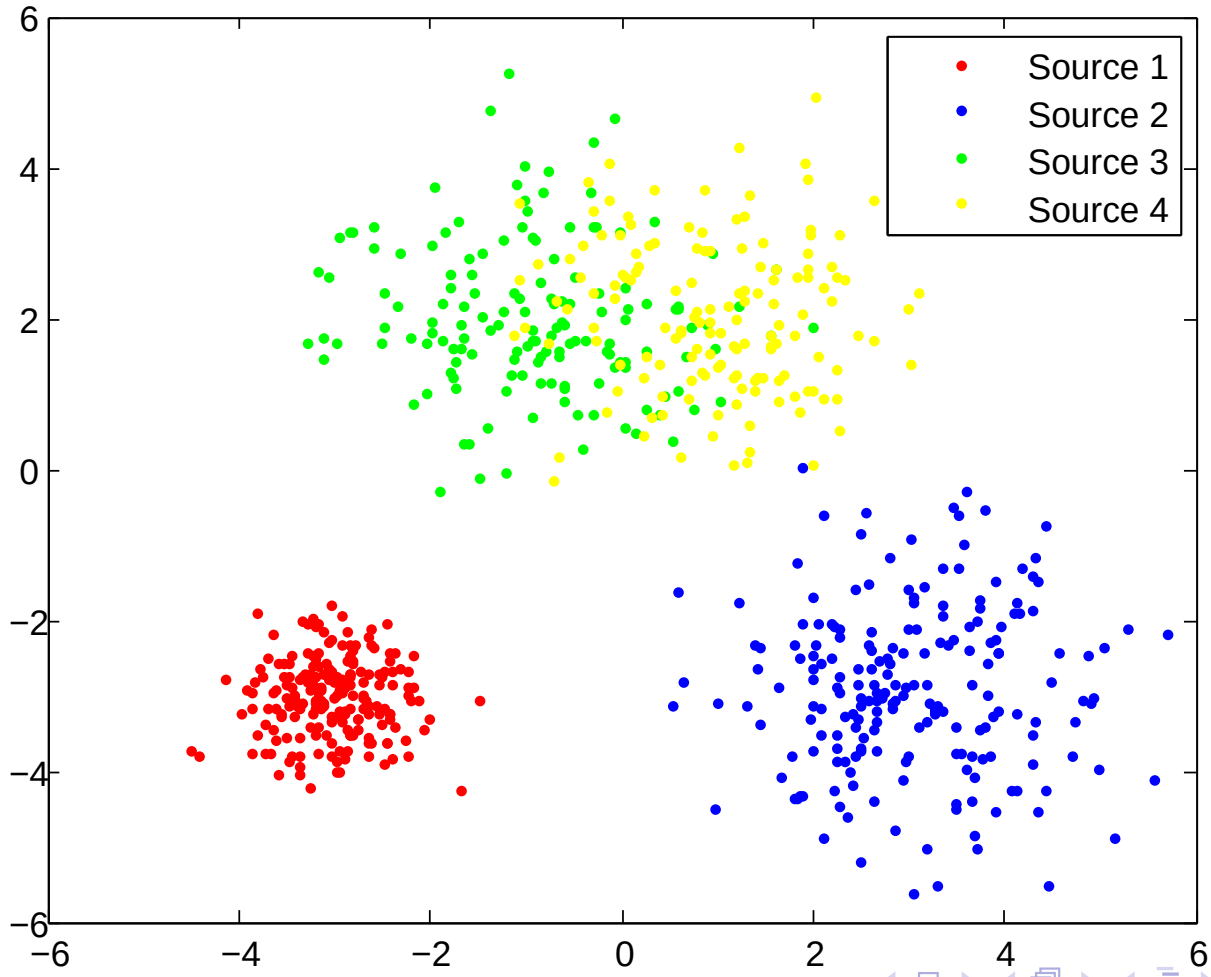
Section 1

Introduction

Multisource Data

- Imagine there is a number of sources delivering samples to a data set.
- Each source generates samples as m-dimensional vectors from some probabilistic distribution with certain mean and variance.
- Then each source creates a cloud of samples, while the clouds can be distinguishable from each other.

Example of Multisource Data

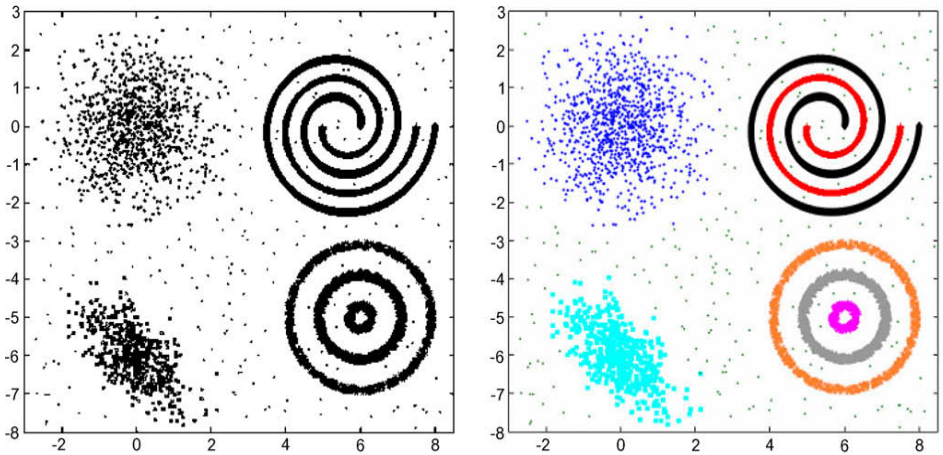


- Different sources may correspond to different conditions under which the samples are formed.
  - In biomedical applications, it may be different diseases, or distinct types of the same disease (such as two types of cancer).
  - In textual mining applications, the sources may correspond to different topics on which the text samples are written.
  - In marketing applications, they may correspond to different groups of customers with specific preferences.
  - In social networks, they may correspond to difference groups of people (from same school or have same interest)
  - ...

- One useful starting point to understanding such data sets is to look for their subsets which are more closely related to each other than they are to other members of the collection.
- Such subsets are called clusters.
- Therefore, the goal of clustering is to discover the natural grouping(s) of a set of patterns, points, or objects

Definition of Clustering

- Given a representation of n objects, find K groups based on a measure of similarity such that the similarities between objects in the same group are high while the similarities between objects in different groups are low
- clusters can differ in terms of their shape, size, and density.



(a) Input data

(b) Desired clustering

Section 2

Major Techniques

Overview of Clustering Algorithms

- Clustering algorithms can be broadly divided into two groups: Hierarchy Clustering and Partitional Clustering
  - Hierarchy clustering algorithms recursively find nested clusters either in agglomerative mode or divisive(top-down) mode
    - Input to a hierarchical algorithm is an $n \times n$ similarity matrix, where n is the number of objects to be clustered.
  - Partitional clustering algorithms find all the clusters simultaneously as a partition of the data
    - a partitional algorithm can use either an $n \times d$ pattern matrix, where n objects are embedded in a d-dimensional feature space, or an $n \times n$ similarity matrix.
- The minimum $k-$clustering problem is the **combinatorial optimization problem**

Hierarchical Clustering

- Most hierarchical clustering algorithms are variants of the single-linkage (Sneath and Sokal 1973) and complete-linkage (King 1967), and minimum-variance (Ward 1963; Murtagh 1984) algorithms.
- The single-linkage and complete-linkage algorithms are most popular.
- Obviously, the output of hierarchical clustering is not a unique partition of the samples but the structure of the entire process.
- It is visualized as a so-called dendrogram depicting the tree of splits/unions.

Hierarchical Clustering

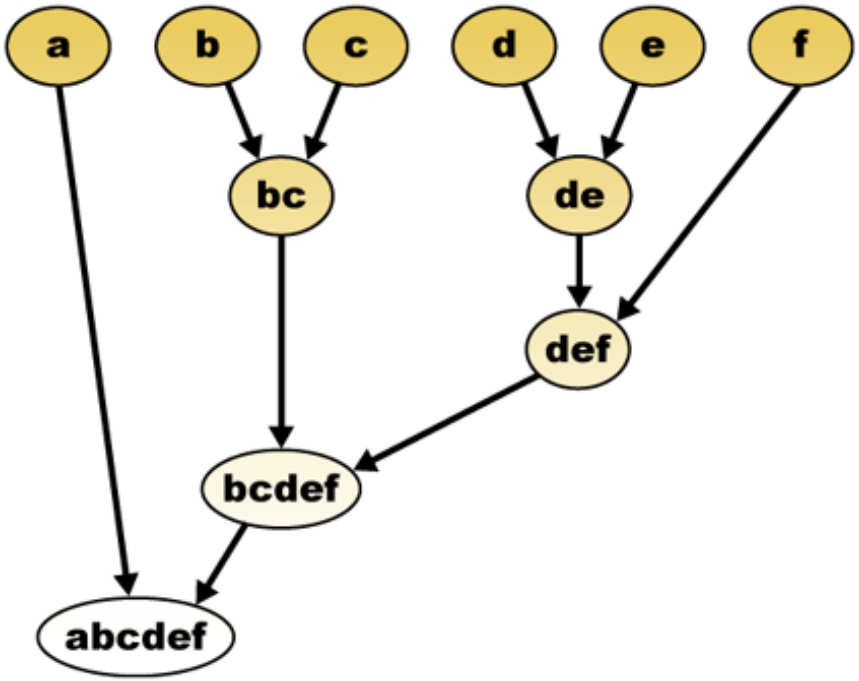


Figure: Example of Dendrogram

Hierarchical Clustering

- In the single-link method, the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from the two clusters (one pattern from the first cluster, the other from the second).
- In the complete-link algorithm, the distance between two clusters is the maximum of all pairwise distances between patterns in the two clusters
- The complete-link algorithm produces tightly bound or com-pact clusters (Baeza-Yates 1992)
- The single-link algorithm, by contrast, suffers from a chaining effect (Nagy 1968).

Hierarchical Clustering

Make each sample its own cluster;

while More than one cluster **do**

Find most similar pair of clusters;

Merge the pair into a parent cluster;

end while

- The output of the algorithm is a nested hierarchy of graphs which can be cut at a desired dissimilarity level forming a clustering identified by
  - simply connected components in the corresponding graph for single-linkage clustering
  - completely connected components in the corresponding graph for complete-linkage clustering

Complexity of Hierarchical Clustering

- The time complexity of hierarchical agglomerative algorithm is $O(n^2 \log n)$ (Kurita 1991). It is possible to obtain single-link clusters using an MST of the data, which can be constructed in $O(n \log^2 n)$ time for two-dimensional data (Choudhury and Murty 1990)
- The space complexity of agglomerative algorithms is $O(n^2)$. This is because a similarity matrix of size $n \times n$ has to be stored. It is possible to compute the entries of this matrix based on need instead of storing them (this would increase the algorithm's time complexity (Anderberg 1973)

References


SNEATH,P.H.A.AND SOKAL, R. R.: *Numerical Taxonomy.* Freeman, London, 1973


KING, B.: *Step-wise clustering procedures* J. Am. Stat. Assoc. 69, 86 101. 1967


WARD,J.H.JR.: *Hierarchical grouping to optimize an objective function.* J. Am. Stat. Assoc. 58, 236 244. 1963


MURTAGH, F.: *A survey of recent advances in hierarchical clustering algorithms which use cluster centers.* Comput. J. 26, 354 359. 1984


BAEZA-YATES, R. A.: *Introduction to data structures and algorithms related to information retrieval.* In Information Retrieval: Data Structures and Algorithms,W.B. Frakes and R. Baeza-Yates, Eds. PrenticeHall, Inc., Upper Saddle River, NJ, 1327. 1992


NAGY, G. *State of the art in pattern recognition.* Proc. IEEE 56, 836 862. 1968


KURITA, T. *An efficient agglomerative clustering algorithm using a heap.* Pattern Recogn. 24, 3 (1991), 205209. 1991


MURTY,M.N.ANDKRISHNA, G. *A computationally efficient technique for data clustering.* Pattern Recogn. 12, 153158. 1980


ANDERBERG, M. R. *Cluster Analysis for Applications.* Academic Press, Inc., New York, NY. 1973

Partitional algorithms

- Partitional algorithms are preferred in pattern recognition due to the nature of available data
- The most popular and the simplest partitional algorithm is K-means
- K-means has a rich and diverse history as it was independently discovered in different scientific fields by Steinhaus (1956), Lloyd (proposed in 1957, pub-lished in 1982), Ball and Hall (1965), and MacQueen (1967)
- Even though K-means was first proposed over 50 years ago, it is still one of the most widely used algorithms for clustering.

K-means

Let $X = \{x_i\}, i = 1, \ldots, n$ be the set of n d-dimensional points to be clustered into a set of k clusters, $S = \{s_j, j = 1, \ldots, k\}$. K-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. Let μ_j be the mean of cluster c_j. The squared error between μ_j and the points in cluster c_k is defined as

$$J(s_j) = \sum_{x_i \in s_j} \|x_i - \mu_j\|^2$$

The goal of K-means is to minimize the sum of the squared error over all K clusters

$$J(S) = \sum_{j=1}^{k} \sum_{x_i \in s_j} \|x_i - \mu_j\|^2$$

Global Optimization for K-Means

- To argue for the hardness of the problem, let us re pre sent it in a form standard for optimization theory.
- Let a_{ij} be a $0-1$ variable showing if i-th sample is in j-th cluster. Then the problem is

$$\min_{a,c} \sum_{i=1}^{k} \sum_{j=1}^{n} a_{ij} |x_i - c_j|$$

$$s.t. \sum_{i=1}^{k} s_{ij} = 1, i = 1 \ldots n$$

- Obviously, it is a nonlinear programming problem involving $0-1$ variables, so it is expected to be hard

- Minimizing this objective function is known to be an NP-hard problem (even for $K = 2$) (Drineas et al., 1999).
- Thus K-means, which is a greedy algorithm, can only converge to a local minimum, even though recent study has shown with a large probability K-means could converge to the global optimum when clusters are well separated (Meila, 2006)
- K-means starts with an initial partition with K clusters and assign patterns to clusters so as to reduce the squared error. Since the squared error always decreases with an increase in the number of clusters K (with $J(S) = 0$ when $K = n$), it can be minimized only for a fixed number of clusters.

Lloyd's algorithm

The following iterative refinement heuristic is known in mathematics as *Lloyd's algorithm* Generate class $s_i; i = 1, \ldots k$ randomly;

while $s_k : k = 1 \ldots K$ are not stable **do**

Compute class centroid $c_k = \frac{\sum_{j \in s_k} x_j}{|s_k|} : k = 1 \ldots K$;

Re-classify each sample j to the class k with $\min \|x_j - c_k\|$;

end while

Complexity of K-Means

- Its time complexity is $O(nkl)$, where n is the number of patterns, k is the number of clusters, and l is the number of iterations taken by the algorithm to converge. Typically, k and l are fixed in advance and so the algorithm has linear time complexity in the size of the data set(Day 1992).
- Its space complexity is $O(k + n)$.It requires additional space to store the data matrix. It is possible to store the data matrix in a secondary memory and access each pattern based on need. However, this scheme requires a huge access time because of the iterative nature of the algorithm, and as a consequence processing time increases enormously.

Other Popular Clustering Algorithms and their complexity

Clustering Algorithm	Time Complexity	Space Complexity
leader	$O(kn)$	O(k)
ISODATA	$O(nkl)$	O(k)
shortest spanning path	$O(n^2)$	$O(n)$

References


Drineas, P., Frieze, A., Kannan, R., Vempala, S., Vinay, V.: *Clustering large graphs via the singular value decomposition.* Machine Learn. 56 (13), 933, 199


Meila, Marina: *The uniqueness of a good optimum fork-means.* Proc. 23rd Internat. Conf. Machine Learning, pp. 625632. 2006


DAY, W. H. E.: *Complexity theory: An introduction for practitioners of classification.* Clustering and Classification,P. Arabie and L. Hubert, Eds. World Scientific Publishing Co., Inc., River Edge, NJ. 1992

Section 3

Graph Clustering

Definition

- A graph G is a pair of sets G = (V , E). V is the set of vertices and the number of vertices $n = |V|$ is the order of the graph. The set E contains the edges of the graph
- Graph clustering is the task of grouping the vertices of the graph into clusters taking into consideration the edge structure of the graph in such a way that there should be many edges within each cluster and relatively few between the clusters.
  - In loosest sense, a graph cluster is a connected component
  - in strictest sense, it is a maximal clique of a graph
- In some of the clustering literature, a cluster in a graph is called a community

A Graph of Facebook Network

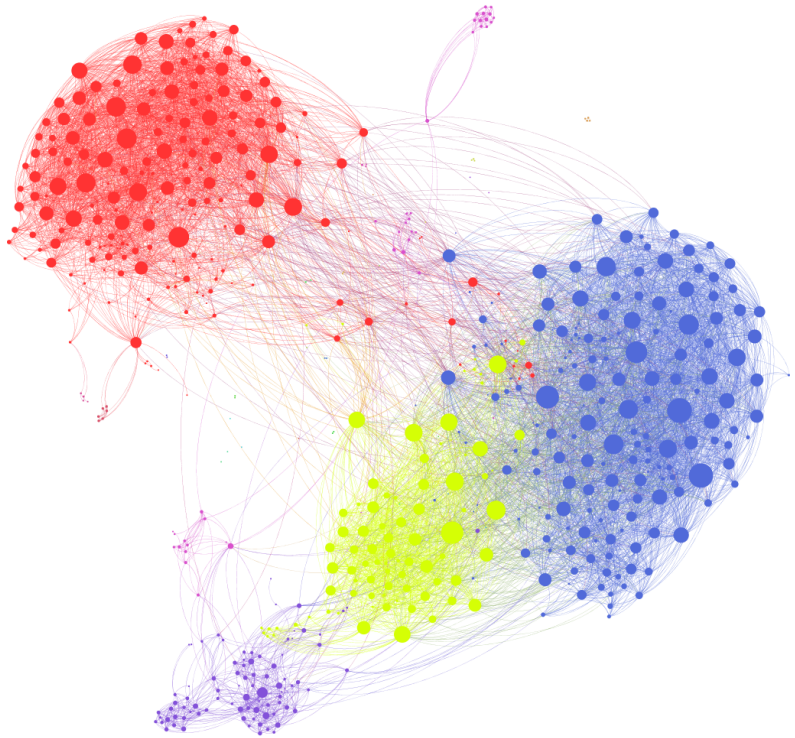


Figure : A Graph of Facebook Network

Generation models for clustered graphs

- Uniform random model (Gilbert 1959):
  - each of the $\binom{n}{2}$ possible edges is included in the graph with probability p, considering each pair of vertices independently
  - The degree distribution is Poissonian
  - No dense clusters can be expected
- Planted *l* partition model
  - A graph is generated with $n = k$ vertices that are partitioned into groups each with k vertice
  - Two probability parameters p and $q < p$ are used to construct the edge set
- Caveman graphs, RMAT generation etc.
- Fuzzy graphs??

Hierarchical Clustering for graph

- Hierarchical clustering is useful if the graph structure itself is hierarchical
- A single cluster can naturally be composed further to obtain a more fine-grained clustering or alternatively merged with another cluster to obtain a coarser division into clusters.
- It depends on the application and the input data whether it makes sense to compute a hierarchy of clusterings or a flat clustering
  - In a flat clustering, each cluster is defined by vertex subset $C \subset V$ as the subgraph induced by C
  - In a hierarchical clustering, each level of the clustering hierarchy defines a different subset

Graph Terminology

- Pairwise affinity (Similarity): $w_{nm} = e^{-\frac{\|s_n - s_m\|}{\sigma^2}}$
- Node volume (degree): $D_n = \sum_{m=1}^{N} w_{nm}$
- Volume of a set(cluster): $Vol(C) = \sum_{n \in C} D_n$
- Cut between 2 sets: $Cut(C_1, C_2) = \sum_{n \in C_1} \sum_{m \in C_2} w_{nm}$

Cuts

- Minimal bipartition Cut:

$$\min\ Cut(C_1, C_2)$$

- Normalized minimal bipartition Cut:

$$\min\ \frac{Cut(C_1, C_2)}{Vol(C_1)} + \frac{Cut(C_2, C_1)}{Vol(C_2)}$$

$$=\min\ \left(\frac{1}{Vol(C_1)} + \frac{1}{Vol(C_2)}\right) Cut(C_1, C_2)$$

- The optimization problem for normalized cuts is intractable (an **NP-hard** problem)
- Hence we resort to spectral clustering and approximation algorithms.

Graph Laplacian

- A graph can be represented as an adjacency matrix A where
$$A_{u,v} = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{Otherwise} \end{cases}$$
- Degree Matrix
$$D = \begin{pmatrix} deg(v_1) & 0 & 0 & \dots & 0 & 0 \\ 0 & deg(v_2) & 0 & \dots & 0 & 0 \\ 0 & 0 & deg(v_3) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & deg(v_3) & 0 \\ 0 & 0 & 0 & \dots & 0 & deg(v_3) \end{pmatrix}$$
- The unnormalized graph laplacian is defined as $L = D - A$
- The normalized graph laplacian is $L_{norm_1} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$

Properties of the Laplacian

- For every vector $f \in \mathbb{R}^n$, $f'Lf = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$
- L is symmetric and positive definite
- 0 is an eigenvalue of the Laplacian, with the constant vector as a corresponding eigenvector.
- L has n non-negative eigenvalues. $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$
- The multiplicity of the eigenvalue 0 gives the number of connected components in the graph.

Applications of Graph Clustering

- Data Transformation: convert feature-vector data into graph format
- Information networks and usage information: a tool for analysis, modelling and prediction of the function, usage and evolution of the network
- Database systems: determine a clustering for a large database in one scan using a limited memory buffer
- Biological and sociological networks: gene expression, protein interactions, epidemic spreading, voter behavior, etc.

Open Problems in Graph Clustering

- Parameter selection: how is the user to determine the parameter values to give as input to the clustering algorithm,
- Scalability: how does the runtime and memory consumption of the algorithm behave for massive input graphs, and
- Evaluation: how to decide which of several clusterings is the best

References


E.N. Gilbert: *Random graphs.* Annals of Mathematical Statistics 30 (4) 11411144. 1959


A. Condon, R.M. Karp: *Algorithms for graph partitioning on the planted partition model,* Random Structures & Algorithms 18 (2) 116140.2001


R. Wilson, X. Bai, E.R. Hancock: *Graph clustering using symmetric polynomials and local linear embedding.* British Machine Vision Conference, 2003


P.S. Bradley, U.M. Fayyad, C. Reina: *Scaling clustering algorithms to large databases.* Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD, ACM, New York, NY, USA, 1998

Section 4

Biclustering

- Biclustering is a methodology allowing for feature set and test set clustering (supervised or unsupervised) simultaneously.
- It finds clusters of samples possessing similar characteristics together with features creating these similarities.
- The required consistency of sample and feature classification gives biclustering an advantage over other methodologies treating samples and features of a dataset separately of each other.

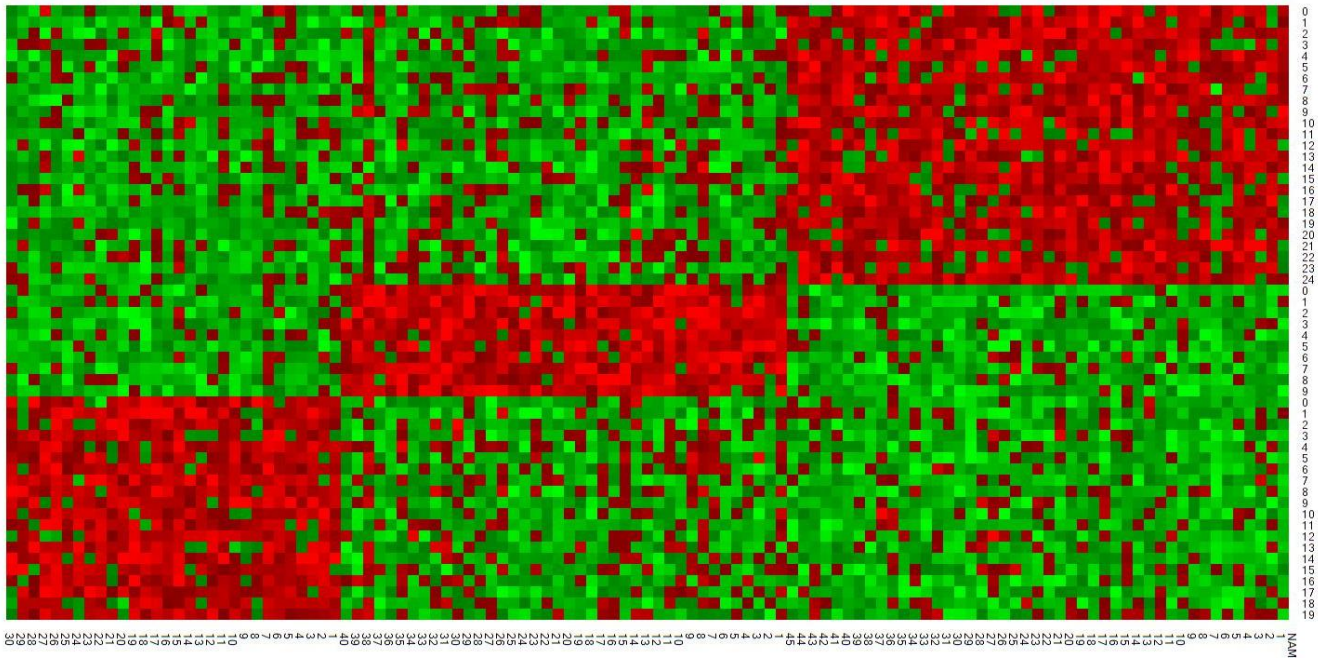


Figure : Partitioning of samples and features into 3 classes

Survey on Biclustering Methodologies

- **Direct Clustering (Hartigan)**
- The algorithm begins with the entire data as a single block and then iteratively nds the row and column split of every block into two pieces. The splits are made so that the total variance in the blocks is minimize d.
- The whole partitioning procedure can be represented in a hierarchical manner by trees.
- Drawback: this method does NOT optimize a global objective function.

Survey on Biclustering Methodologies

- **Cheng & Churchs algorithm**
- The algorithm constructs one bicluster at a time using a statistical criterion a low mean squared residue (the variance of the set of all elements in the bicluster, plus the mean row variance and the mean column variance).
- Once a bicluster is created, its entries are replaced by random numbers, and the procedure is repeated iteratively.

Survey on Biclustering Methodologies

- **Graph Bipartitioning**
- Define a bipartite graph $G(F, S, E)$, where F is the set of data set features, S is the set of data set samples, and E are weighted edges such that the weight $E_{ij} = a_{ij}$ for the edge connecting $i \in F$ with $j \in S$. The biclustering corresponds to partitioning of the graph into bicliques.

Survey on Biclustering Methodologies

- Given vertex subsets V_1 and V_2, define

$$cut(V1, V2) = \sum_{i \in V_1} \sum_{j \in V_2} a_{ij}$$

- and for k vertex subsets $V_1, V_2, \ldots, V_k$,

$$cut(V_1, V_2, \ldots, V_k) = \sum_{i<j} cut(V_i, V_J)$$

Survey on Biclustering Methodologies

- Biclustering may be performed as

$$\min_{V_1, V_2, \ldots V_K} = cut(V_1, V_2, \ldots V_k)$$

  on G or with some modification of the definition of cut to favor balanced clusters.
- This problem is **NP-hard**, but spectral heuristics show good performance **[Dhillon]**

Biclustering: Applications

- Biological and Medical:
  - Microarray data analysis
  - Analysis of drug activity, Liu and Wang (2003)
  - Analysis of nutritional data, Lazzeroni et al. (2000)

Biclustering: Applications

- **Text Mining**: Dhillon (2001, 2003)
- **Marketing**: Gaul and Schader (1996)
- **Dimensionality Reduction in Databases**: Agrawal et al.(1998)
- Others:
  - electoral data - Hartigan (1972)
  - currency exchange - Lazzeroni et al. (2000)

Biclustering: Surveys

- S. Madeira, A.L. Oliveira, Biclustering Algorithms for Biological Data Analysis: A Survey, 2004.
- A. Tanay, R. Sharan, R. Shamir, Biclustering Algorithms: A Survey, 2004.
- S. Busygin, O.A. Prokopyev, and P.M. Pardalos, Biclustering in Data Mining, to appear in C&OR, 2007.

Data Representation

- 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 the value a_{ij} is the expression of i-th feature in j-th sample.

Definition

- Data set of n samples and m features is a matrix

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

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

- We consider classification of the samples into classes

$$S_1, S_2, \ldots, S_r, S_k \subseteq \{1 \ldots n\}, k = 1 \ldots r,$$

$$S1 \cup S2 \cup \ldots \cup S_r = \{1 \ldots n\},$$

$$S_k \cap S_l = \phi, k, l = 1 \ldots r, k \neq l.$$

- This classification should be done so that samples from the same class share certain common properties. Correspondingly, a feature i may be assigned to one of the feature classes

$$F_1, F_2, \ldots, F_r, F_k \subseteq \{1 \ldots m\}, k = 1 \ldots r,$$

$$F_1 \cup F_2 \cup \ldots \cup F_r = \{1 \ldots m\},$$

$$F_k \cup F_l = \phi, k, l = 1 \ldots r, k \neq l,$$

in such a way that features of the class F_k are responsible for creating the class of samples S_k.

Definition

- This may mean for microarray data, for example, strong up-regulation of certain genes under a cancer condition of a particular type (whose samples constitute one class of the data set). Such a simultaneous classification of samples and features is called **biclustering (or co-clustering)**.

Definition

A biclustering of a data set is a collection of pairs of sample and feature subsets

$$B = ((S_1, F_1), (S_2, F_2), \ldots, (S_r, \ldots, F_r))$$

such that the collection $(S_1, S_2, \ldots, S_r)$ forms a partition of the set of samples, and the collection $(F_1, F2, \ldots, F_r)$ forms a partition of the set of features.

Our Approach: Intuition

- Let us distribute features among the classes of training set such that each feature belongs to the class where its average expression among the training samples is highest.

$$(S_1^0, S_2^0, \ldots, S_r^0) \rightarrow (F_1, F_2, \ldots, F_r)$$

- Now, if we transpose the matrix, take the feature classification as given, and re-classify the training samples according to highest average expression values in feature classes...

$$(F_1, F_2, \ldots, F_r) \rightarrow (S_1^1, S_2^1, \ldots, S_r^1)$$

Intuition Behind Biclustering

- Will we obtain the same training set classification?

$$?(S_1^0, S_2^0, \ldots, S_r^0) = (S_1^1, S_2^1, \ldots, S_r^1)?$$

- If yes, we will say that we obtained a **consistent biclustering**

Consistent Biclustering

- Let each sample be already assigned somehow to one of the classes $S_1, S_2, \ldots, S_r$. Introduce a 01 matrix $S = (s_{jk})_{n \times r}$ such that $s_{jk} = 1$ if $j \in Sk$, and $s_{jk} = 0$ otherwise.
- The sample class *centroids* can be computed as the matrix $C = (c_{ik})_{m \times k}$:

$$C = AS(S^T S)^{-1}, (c_{ik} = \frac{\sum_{j \in S_k} a_{ij}}{|S_k|})$$

whose k-th column represents the centroid of the class S_k.

Consistent Biclustering

- Consider a row i of the matrix C . Each value in it gives us the average expression of the i-th feature in one of the sample classes. As we want to identify the checkerboard pattern in the data, we have to assign the feature to the class where it is most expressed. So, let us classify the i -th feature to the class $\hat{k}$ with the maximal value $c_{i\hat{k}}$:

$$i \in F_{\hat{k}} \Rightarrow \forall k = 1 \dots r, k \neq \hat{k} : c_{i\hat{k}} > c_{ik}$$

Consistent Biclustering

- Using the classification of all features into classes $F_1, F_2, \ldots, F_r$, let us construct a classification of samples using the same principle of maximal average expression. We construct a 01 matrix $F = (f_{ik})_{m \times k}$ such that $f_{ik} = 1$ if $i \in F_k$ and $f_i k = 0$ otherwise. Then, the feature class centroids can be computed in form of matrix $D = (d_{jk})_{n \times r}$:

$$D = A^T F (F^T F)^{-1}, (d_{jk} = \frac{\sum_{j \in F_k} a_{ij}}{|F_k|})$$

whose k-th column represents the centroid of the class F_k.

- The condition on sample classification we need to verify is

$$j \in S_{\hat{k}} \Rightarrow \forall k = 1 \ldots r, k \neq \hat{k} : d_{j\hat{k}} > d_{jk}$$

Consistent Biclustering

Definition

A biclustering B will be called **consistent** if the folloing relations hold:

$$i \in F_{\hat{k}} \Rightarrow \forall k = 1 \ldots r, k \neq \hat{k} : c_{i\hat{k}} > cik$$

$$j \in S_{\hat{k}} \Rightarrow \forall j = 1 \ldots r, k \neq \hat{k} : d_{j\hat{k}} > d_{jk}$$

Consistent Biclustering

Definition

A data set is biclustering-admitting if some consistent biclustering for it exists.

Definition

The data set will be 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.

Consistent Biclustering

- **A consistent biclustering implies separability of the classes by convex cones.**

Theorm

Let B be a consistent biclustering. Then there exist convex cones $P_1, P_2, \ldots, P_r \in \mathbb{R}^m$ such that all samples from S_k belong to the cone P_k and no other sample belongs to it, $k = 1 \ldots r$. Similarly, there exist convex cones $Q_1, Q_2, \ldots, Q_r \in \mathbb{R}^n$ such that all features from F_k belong to the cone Q_k and no other feature belongs to it, $k = 1 \ldots r$.

Separation by Cones

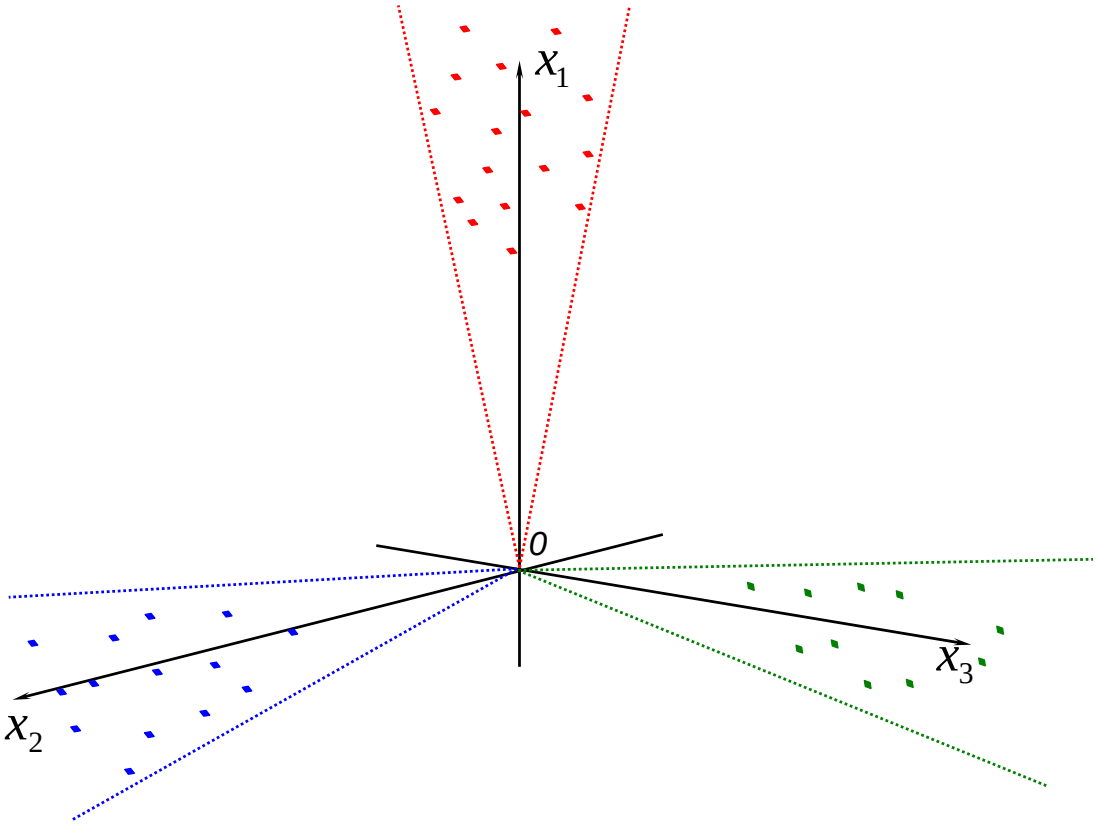


Figure : 3 classes are separated in 3D space

Conic Separability

Proof

Let P_k be the conic hull of the samples of S_k. Suppose $\hat{j} \in S_l, l \neq k$, belongs to P_k. Then

$$a_{\cdot\hat{j}} = \sum_{j \in S_k} \gamma_j a_{\cdot\hat{j}},$$

where $\gamma_j \geq 0$. Biclustering consistency implies that $d_{\hat{j}l} > d_{\hat{j}k}$, that is

$$\frac{\sum_{i \in F_l} a_{i\hat{j}}}{|F_l|} > \frac{\sum_{i \in F_k} a_{i\hat{j}}}{|F_k|}$$

Conic Separability

Proof(cont'd)

Plugging the conic representation of $a_{\hat{ij}}$, we can pbtain

$$\sum_{j \in S_k} \gamma_j d_{jl} > \sum_{j \in S_k} \gamma_j d_{jk},$$

that contradicts to $d_{jl} < d_{jk}$ (also implied by biclustering consistency).

Similarly, we can show that the formulated conic separability holds for feature classes.

$\alpha-$ and $\beta-$ Consistent Biclustering

Definition A biclustering B will be called $\alpha-$ consistent if the following relations hold:

$$i \in F_{\hat{k}} \Rightarrow \forall k = 1 \ldots r, k \neq \hat{k} : c_{i\hat{k}} > \alpha_i^F + c_{ik}$$
$$j \in S_{\hat{k}} \Rightarrow \forall k = 1 \ldots r, k \neq \hat{k} : d_{i\hat{k}} > \alpha_i^S + d_{ik}$$

where α is a vector of $\alpha_j^S \geq 0$ and $\alpha_i^F \geq 0$

$\alpha-$ and $\beta-$ Consistent Biclustering

Definition A biclustering B will be called $\beta-$ consistent if the following relations hold:

$$i \in F_{\hat{k}} \Rightarrow \forall k = 1 \ldots r, k \neq \hat{k} : c_{i\hat{k}} > \beta_i^F c_{ik}$$
$$j \in S_{\hat{k}} \Rightarrow \forall k = 1 \ldots r, k \neq \hat{k} : d_{i\hat{k}} > \beta_i^S d_{ik}$$

where β is a vector of $\beta_j^S \geq 1$ and $\beta_i^F \geq 1$

$\alpha-$ and $\beta-$ Consistent Biclustering

- If a biclustering B is α-consistent then it is consistent.
- If a biclustering B is β-consistent and $c_{ik} \geq 0$ and $d_{jk} \geq 0, \forall i, j, k$, then it is consistent.
- Both allow selecting the most representative subset of features and/or samples.

Supervised Biclustering

- One of the most important problems for real-life data mining applications is **supervised classification** of test samples on the basis of information provided by training data.
- A **supervised classification** method consists of two routines, first of which derives classification criteria while processing the training samples, and the second one applies these criteria to the test samples.

Supervised Biclustering

- In genomic and proteomic data analysis, as well as in other data mining applications, where only a small subset of features is expected to be relevant to the classification of interest, the classification criteria should involve dimensionality reduction and feature selection.
- We handle such a task utilizing the notion of consistent biclustering. Namely, we select a subset of features of the original data set in such a way that the obtained subset of data becomes conditionally biclustering-admitting with respect to the given classification of training samples.

Fractional $0-1$ Programming Formulation

- Formally, let us introduce a vector of $0-1$ variables $x = (x_i)_{i=1\ldots m}$ and consider the i-th feature selected if $x_i = 1$
- The condition of biclustering consistency, when only the selected features are used, becomes

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{k}} x_i}{\sum_{i=1}^{m} f_{i\hat{k}} x_i} > \frac{\sum_{i=1}^{m} a_{ij} f_{ik} x_i}{\sum_{i=1}^{m} f_{ik} x_i}, \forall j \in S_{\hat{k}}, \hat{k}, k = 1 \ldots r. \hat{k} \neq k.$$

Fractional $0-1$ Programming Formulation

- We will use the fractional relations as constraints of an optimization problem selecting the feature set. It may incorporate various objective functions over x , 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**. In this case, the objective function is

$$\max \sum_{i=1}^{m} x_i$$

Fractional $0-1$ Programming Formulation

- One of the possible fractional $0-1$ formulations based on β-consistent biclustering criterion (suitable for microarrays):

$$\max_{x \in \mathbb{B}^n} \sum_{i=1}^{m} x_i,$$

s.t.

$$\frac{\sum_{i=1}^{m} a_{ij} f_{i\hat{k}} x_i}{\sum_{i=1}^{m} f_{i\hat{k}} x_i} > \beta_j^S \frac{\sum_{i=1}^{m} a_{ij} f_{ik} x_i}{\sum_{i=1}^{m} f_{ik} x_i}, \forall j \in S_{\hat{k}}, \hat{k}, k = 1 \ldots r. \hat{k} \neq k.$$

Fractional 0 – 1 Programming Formulation

- Generally, in the framework of fractional 01 programming we consider problems, where we optimize a multiple-ratio fractional 01 function subject to a set of linear constraints.
- We have **a new** class of fractional 01 programming problems, where fractional terms are not in the objective function, but in constraints, i.e. we optimize a linear objective function subject to fractional constraints.
- **How to solve fractionally constrained** 01 **programming problem?**

Linear Mixed $0-1$ Formulation

- We can reduce our problem to a linear mixed $0-1$ programming problem applying the approach similar to the one used to linearize problems with fractional $0-1$ objective function.

T.-H. Wu: *A note on a global approach for general* $0-1$ *fractional programming.* European J. Oper. Res. 101 (1997) 220223.

Linear Mixed $0-1$ Formulation

Theorem

A polynomial mixed 01 term $z = xy$, where x is a 01 variable, and y is a continuous variable, can be represented by the following linear inequalities:

$(1) z \leq Ux$;

$(2) z \leq y + L(x-1)$;

$(3) z \geq y + U(x-1)$;

$(4) z \geq Lx$,

where U and L are upper and lower bounds of variable y, i.e. $L \leq y \leq U$

Linear Mixed $0-1$ Formulation

- To linearize the fractional 01 program we need to introduce new variable y_k

$$\mathbf{y}_k = \frac{1}{\sum_{l=1}^{m} f_{lk} x_l}, k = 1, \ldots, r$$

Linear Mixed $0-1$ Formulation

- In terms of the new variables fractional constraints are replaced by

$$\sum_{i=1}^{m} a_{ij} f_{i\hat{k}} x_i y_{\hat{k}} \geq \beta_j^S \sum_{i=1}^{m} a_{ij} f_{ik} x_i y_k$$

Linear Mixed $0-1$ Formulation

- Next, observe that the term $x_i y_k$ is present if and only if $f_{ik} = 1$, i.e., $i \in F_k$. So, there are totally only m of such products, and hence we can introduce m variables $z_i = x_i y_k, i \in F_k$:

$$z_i = \frac{x_i}{\sum_{l=1}^{m} f_{lk} x_l}, i \in F_k$$

Linear Mixed $0-1$ Formulation

- In terms of z_i we have the following constraints:

$$\sum_{i=1}^{m} f_{ik} z_i = 1, k = 1 \ldots r.$$

$$\sum_{i=1}^{m} a_{ij} f_{i\hat{k}} z_i \geq \beta_j^S \sum_{i=1}^{m} a_{ij} f_{ik} z_i \forall j \in S_{\hat{k}}, \hat{k}, k = 1 \ldots r, \hat{k} \neq k.$$

$$y_k - z_i \leq 1 - x_i, z_i \leq y_k, z_i \leq x_i, z_i \geq 0, i \in F_k$$

Supervised Biclustering

- Unfortunately, while the linearization works nicely for small-size problems, it often creates instances, where the gap between the integer programming and the linear programming relaxation optimum solutions is very big for larger problems. As a consequence, the instance **can not be solved** in a reasonable time even with the best techniques implemented in modern integer programming solvers.
- **HuGE Index Data set: about 7000 features**
- **ALL vs. AML Data Set: about 7000 features**
- **GBM vs. AO data set: about 12000 features**

Heuristic

- If we know that no more than m_k features can be selected for class F_k, then we can impose

$$x_i \leq m_k z_i, x_i \geq z_i, i \in F_k$$

Heuristic

Assign $m_k := |F_k|, k = 1 \ldots r$;
LOOP: Solve the mixed $0-1$ programming formulation using the inequalities

$$x_i \leq M_k z_i, x_i \geq z_i, i \in F_k$$

instead of $y_k - z_i \leq 1 - x_i, z_i \leq Y_k, z_i \leq x_i, z_i \geq 0, i \in F_k$;
If $m_k = \sum_{i=1}^{m} f_{ik} x_i$ for all $k = 1 \ldots r$, EXIT;
$m_k := \sum_{i=1}^{m} f_{ik} x_i$ for all $k = 1 \ldots r$;
Go to LOOP

Supervised Biclustering

- After the feature selection is done, we perform classification of test samples according to the following procedure.
- If $b = (b_i)_{i=1\ldots m}$ is a test sample, we assign it to the class $F_{\hat{k}}$ satisfying

$$\frac{\sum_{i=1}^{m} b_i f_{i\hat{k}} x_i}{\sum_{i=1}^{m} f_{i\hat{k}} x_i} > \frac{\sum_{i=1}^{m} b_i f_{ik} x_i}{\sum_{i=1}^{m} f_{ik} x_i}, k = 1 \ldots r, \hat{k} \neq k.$$

HuGE index data set: Feature Selection

- A computational experiment that we conducted was on **feature selection for consistent biclustering** of the Human Gene Expression (*HuGE*) Index data set. The purpose of the HuGE project is to provide a comprehensive database of gene expressions in normal tissues of different parts of human body and to highlight similarities and differences among the organ systems.
- The number of selected features (genes) is 6889 (out of 7070).

HuGE index data set: Feature Selection

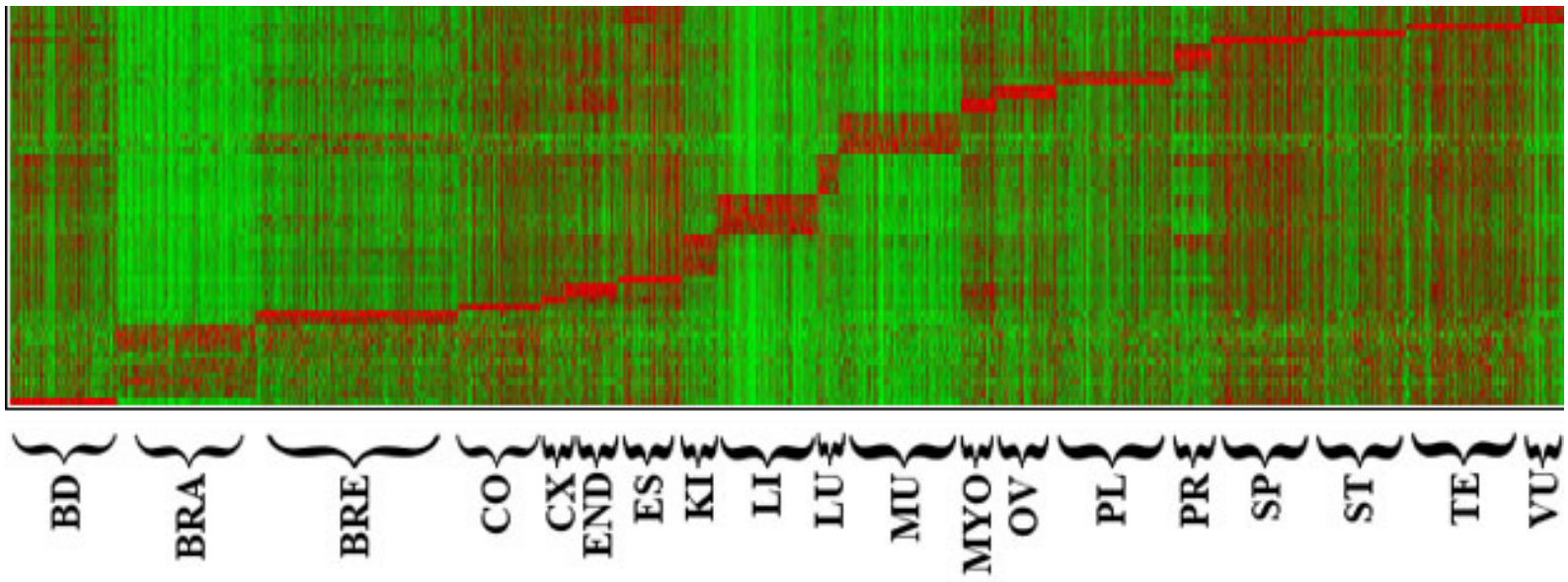


Figure : HuGe Index heatmap

ALL vs. AML data set

- T. Golub at al. (1999) considered a dataset containing 47 samples from *ALL* patients and 25 samples from *AML* patients. The dataset was obtained with Affymetrix GeneChips.
- Our biclustering algorithm selected 3439 features for class *ALL* and 3242 features for class *AML*. The subsequent classification contained **only one error**: the *AML*-sample 66 was classified into the ALL class.
- The SVM approach delivers up to 5 classification errors depending on how the parameters of the method are tuned. The perfect classification was obtained only with one specific set of values of the parameters.

ALL vs. AML data set

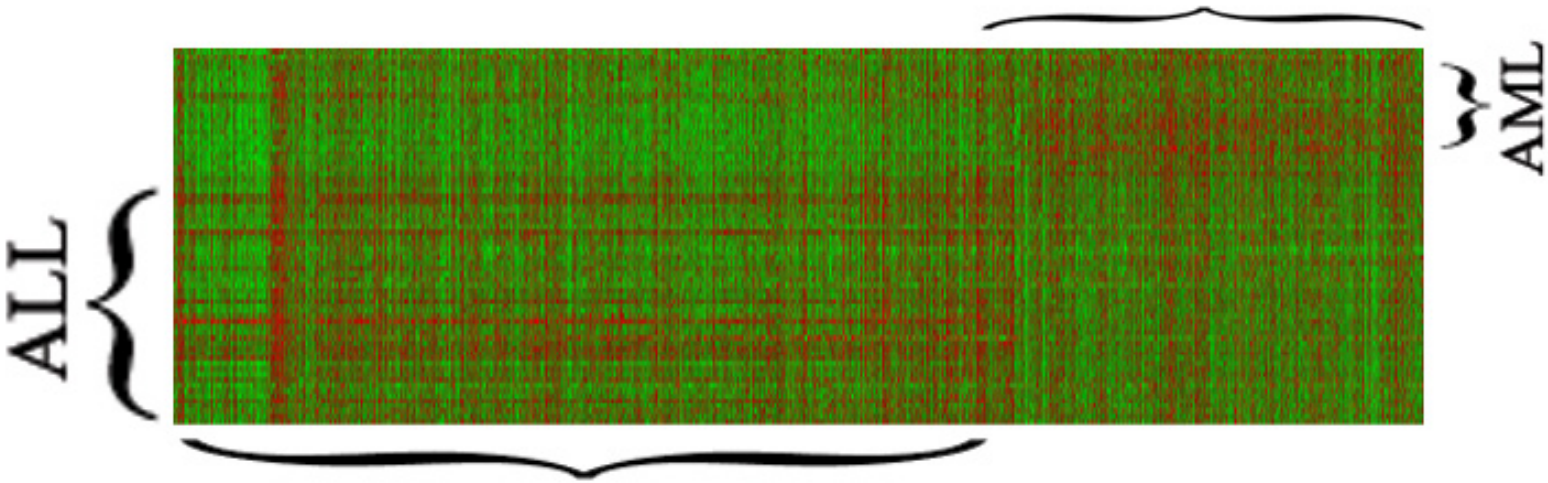


Figure : ALL vs. AML heatmap

GBM vs. AO data set

- The algorithm selected 3875 features for the class GBM and 2398 features for the class AO. The obtained classification contained only 4 errors: two GBM samples (Brain_NG_1 and Brain_NG_2) were classified into the AO class and two AO samples (Brain_NO_14 and Brain_NO_8) were classified into the GBM class.

GBM vs. AO data set

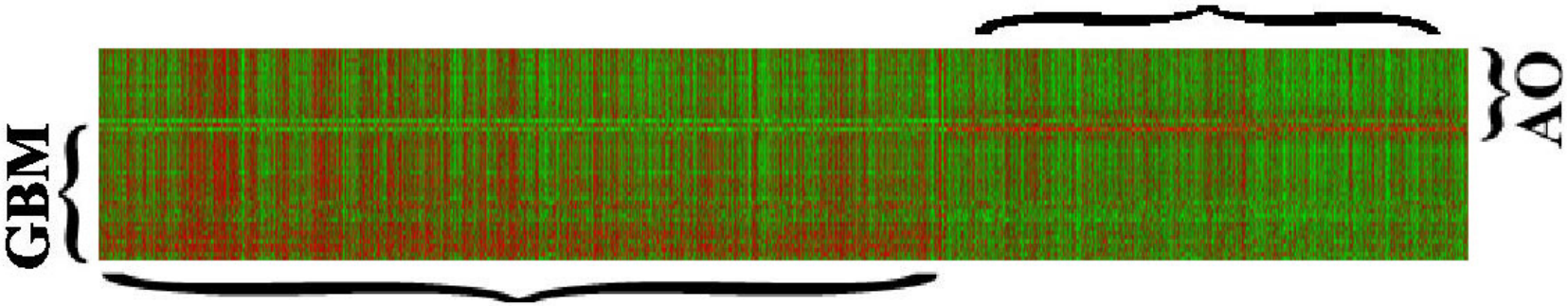
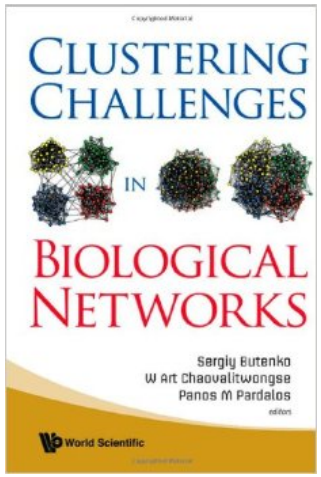


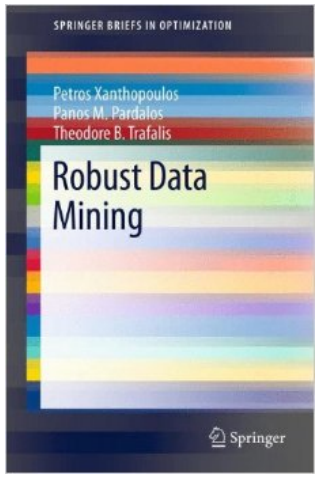
Figure : GBM vs. AO heatmap

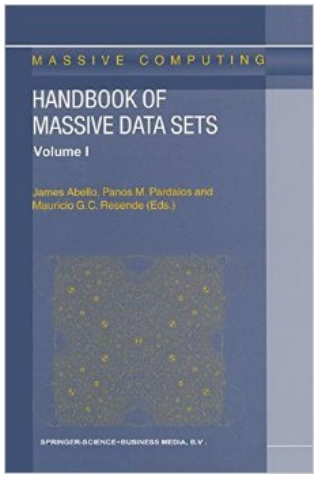
References

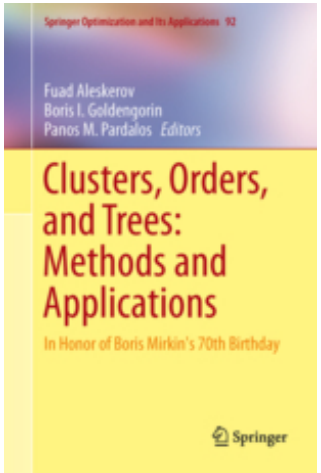

S. Busygin, P. Pardalos, O. Prokopyev: *Feature selection for consistent biclustering via fractional 01 programming.* Journal of Combinatorial Optimization, Vol. 10/1 (2005), pp. 721.

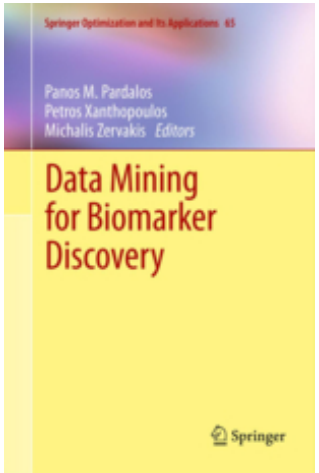

P.M. Pardalos, S. Busygin, O.A. Prokopyev: *On Biclustering with Feature Selection for Microarray Data Sets* BIOMAT 2005 International Symposium on Mathematical and Computational Biology, R. Mondaini (ed.), World Scientific (2006), pp. 367378.

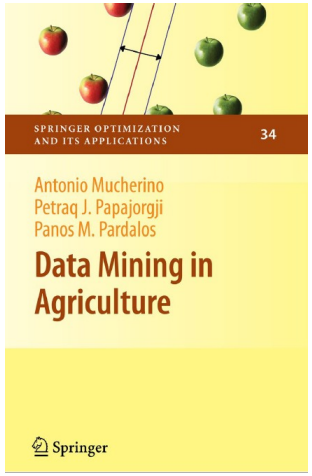

CLUSTERING CHALLENGES IN BIOLOGICAL NETWORKS
Sergiy Butenko
W Art Chaovalitwongse
Panos M Pardalos
World Scientific


SPRINGER BRIEFS IN OPTIMIZATION
Petros Xanthopoulos
Panos M. Pardalos
Theodore B. Trafalis
Robust Data Mining
Springer


MASSIVE COMPUTING
HANDBOOK OF MASSIVE DATA SETS
Volume I
James Abello, Panos M. Pardalos and Mauricio G.C. Resende (Eds.)


Springer Optimization and Its Applications 92
Fuad Aleskerov
Boris I. Goldengorin
Panos M. Pardalos Editors
Clusters, Orders, and Trees: Methods and Applications
In Honor of Boris Mirkin's 70th Birthday
Springer


Springer Optimization and Its Applications 65
Panos M. Pardalos
Petros Xanthopoulos
Michalis Zervakis Editors
Data Mining for Biomarker Discovery
Springer


SPRINGER OPTIMIZATION AND ITS APPLICATIONS 34
Antonio Mucherino
Petraq J. Papajorgji
Panos M. Pardalos
Data Mining in Agriculture
Springer