On the Number of Clusters in Block Clustering Algorithms

Malika Charrad
National School of Computer Sciences, Tunisia
Conservatoire National des Arts et Métiers, France

Yves Lechevallier
INRIA Rocquencourt
Lechesnay, France

Mohamed Ben Ahmed
National School of Computer Sciences,
Tunisia

Gilbert Saporta
Conservatoire National des Arts
et Métiers, France

Abstract

One of the major problems in clustering is the need of specifying the optimal number of clusters in some clustering algorithms. Some block clustering algorithms suffer from the same limitation that the number of clusters needs to be specified by a human user. This problem has been subject of wide research. Numerous indices were proposed in order to find reasonable number of clusters. In this paper, we aim to extend the use of these indices to block clustering algorithms. Therefore, an examination of some indices for determining the number of clusters in CROKI2 algorithm is conducted on synthetic data sets. The purpose of the paper is to test the performance and ability of some indices to detect the proper number of clusters on rows and columns partitions obtained by a block clustering algorithm.

Introduction

Simultaneous clustering, usually designated by biclustering, co-clustering or block clustering, is an important technique in two way data analysis. The term was first introduced by Mirkin (Mirkin 1996) (recently by Cheng and Church in gene expression analysis), although the technique was originally introduced much earlier by J.Hartigan (Hartigan 1975). The goal of simultaneous clustering is to find submatrices, which are subgroups of rows and subgroups of columns that exhibit a high correlation. A number of algorithms that perform simultaneous clustering on rows and columns of a matrix have been proposed to date. They have practical importance in a wide variety of applications such as biology, data analysis, text mining and web mining. A wide range of different articles were published dealing with different kinds of algorithms and methods of simultaneous clustering. Comparisons of several biclustering algorithms can be found, e.g., in (Tanay, Sharan, and Shamir 2004), (Prelic et al. 2006), (Madeira and Oliveira 2004) or (Charrad et al. 2008). One of the major problems of simultaneous clustering algorithms, similarly to the simple clustering algorithms, is that the number of clusters must be supplied as a parameter. To overcome this problem, numerous strategies have been proposed for finding the right number of clusters. However, these strategies can only be applied with one way clustering algorithms and there is a lack of approaches to find the best number of clusters in block clustering algorithms. In this paper, we are interested by the problem of specifying the number of clusters on rows and columns in CROKI2 algorithm proposed in (Govaert 1983)(Govaert 1995)(Nadif and Govaert 2005). This paper is organized as follows. In the next section, we present the simultaneous clustering problem. Then in section 3 we present CROKI2 algorithm. In section 4 and section 5, we present a review of approaches based on relative criteria for cluster validity and some clustering validity indices proposed in the literature for evaluating the clustering results. Moreover, an experimental study based on some of these validity indices is presented in section 6 using synthetic data sets.

Simultaneous clustering problem

Given the data matrix A, with set of rows $X = (X_1, ..., X_n)$ and set of columns $Y = (Y_1, ..., Y_n)$, a_{ij}, $1 \leq i \leq n$ and $1 \leq j \leq n$ is the value in the data matrix A corresponding to row i and column j. Simultaneous clustering algorithms aim to identify a set of biclusters $B_k(I_k, J_k)$, where I_k is a subset of the rows X and J_k is a subset of the columns Y. I_k rows exhibit similar behavior across J_k columns, or vice versa and every bicluster B_k satisfies some criteria of homogeneity.

	Y_1	...	Y_j	...	Y_m
X_1	a_{11}	...	a_{1j}	...	a_{1m}
...	...	...	...	...	...
X_i	a_{i1}	...	a_{ij}	...	a_{im}
...	...	...	...	...	...
X_n	a_{n1}	...	a_{nj}	...	a_{nm}

Table 1: Data matrix

CROKI2 algorithm

CROKI2 algorithm is an adapted version of k-means based on the Chi-square distance. It is applied to contingency tables to identify a row partition P and a column partition Q that maximises χ^2 value of the new matrix obtained by grouping rows and columns. CROKI2 consists in applying K-means algorithm on rows and on columns alternatively

to construct a series of couples of partitions (P^n, Q^n) that optimizes χ^2 value of the new data matrix. Given a contingency table $A(X, Y)$, with set of rows X and set of columns Y, the aim of CROKI2 algorithm is to find a row partition $P = (P_1, ..., P_K)$ composed of K clusters and a column partition $Q = (Q_1, ..., Q_L)$ composed of L clusters that maximizes χ^2 value of the new contingency table (P, Q) obtained by grouping rows and columns in respectively K and L clusters. The criterion optimized by the algorithm is :

$$\chi^2(P, Q) = \sum_{k=1}^{K} \sum_{l=1}^{L} \frac{(f_{kl} - f_{k.} f_{.l})^2}{f_{k.} f_{.l}}$$

with

$$f_{kl} = \sum_{i \in P_k} \sum_{j \in Q_l} f_{ij}$$

$$f_{k.} = \sum_{l=1,L} f_{kl} = \sum_{i \in P_k} f_{i.}$$

$$f_{.l} = \sum_{k=1,K} f_{kl} = \sum_{j \in Q_l} f_{.j}$$

The new contingency table $T_1(P, Q)$ is defined by this expression:

$$T_1(k, l) = \sum_{i \in P_k} \sum_{j \in Q_l} a_{ij}$$

$k \in [1, ..., K]$ and $l \in [1, ..., L]$.
The author of the algorithm has shown that the maximization of $\chi^2(P, Q)$ can be carried out by the alternated maximization of $\chi^2(P, Y)$ and $\chi^2(X, Q)$ which guarantees the convergence. Inputs of Croki2 algorithm are: contingency table, number of clusters on rows and columns and number of runs. The different steps of Croki2 algorithm are the following :

1. Start from the initial position $(P^{(0)}, Q^{(0)})$.
2. Computation of $(P^{(n+1)}, Q^{(n+1)})$ starting from $(P^{(n)}, Q^{(n)})$.
   - Computation of $(P^{(n+1)}, Q^{(n)})$ starting from $(P^{(n)}, Q^{(n)})$ by applying kmeans on partition $P^{(n)}$.
   - Computation of $(P^{(n+1)}, Q^{(n+1)})$ starting from $(P^{(n+1)}, Q^{(n)})$ by applying kmeans on partition $Q^{(n)}$.
3. Iterate the step 2 until the convergence.

Cluster validation in clustering algorithms

While clustering algorithms are unsupervised learning processes, users are usually required to set some parameters for these algorithms. These parameters vary from one algorithm to another, but most clustering algorithms require a parameter that either directly or indirectly specifies the number of clusters. This parameter is typically either k, the number of clusters to return, or some other parameter that indirectly controls the number of clusters to return, such as an error threshold. Moreover, even if user has sufficient domain knowledge to know what a good clustering "looks" like, the result of clustering needs to be validated in most applications. The procedure for evaluating the results of a clustering algorithm is known under the term cluster validity. In general terms, there are three approaches to investigate cluster validity (Theodoridis and Koutroubas 1999). The first one is based on the choice of an external criterion. This implies that the results of a clustering algorithm are evaluated based on a pre-specified structure, which is imposed on a data set and reflects user intuition about the clustering structure of the data set. In other words, the results of classification of input data are compared with the results of classification of data not participating in the basic classification. The second approach is based on the choice of an internal criterion. In this case, only input data is used for the evaluation of classification quality. The internal criteria are based on some metrics which are based on data set and the clustering schema. The main disadvantage of these two methods is their computational complexity. Moreover, the indices related to these approaches aim at measuring the degree to which a data set confirms an a priori specified scheme. The third approach of clustering validity is based on the choice of a relative criterion. Here the basic idea is the comparison of the different clustering methods. One or more clustering algorithms are executed multiple times with different input parameters on the same data set. The aim of the relative criterion is to choose the best clustering schema from the different results. The basis of the comparison is the validity index. Several validity indices have been developed and introduced for each of the above approaches ((Halkidi, Vazirgiannis, and Batistakis 2000) and (Theodoridis and Koutroubas 1999)). In this paper, we focus only on indices proposed for the third approach.

Validity indices

In this section some validity indices are introduced. These indices are used for measuring the quality of a clustering result comparing to other ones which were created by other clustering algorithms, or by the same algorithms but using different parameter values. These indices are usually suitable for measuring crisp clustering. Crisp clustering means having non overlapping partitions.

Dunn's Validity Index

This index (Dunn 1974) is based on the idea of identifying the cluster sets that are compact and well separated. For any partition of clusters, where C_i represent the cluster *i* of such partition, the Dunn's validation index, D, could be calculated with the following formula:

$$D = min_{1 \leq i \prec j \leq K} \frac{d(C_i, C_j)}{max_{1 \leq k \leq K} d'(C_k)}$$

where K is the number of clusters, $d(C_i, C_j)$ is the distance between clusters C_i and C_j (intercluster distance) and $d'(C_k)$ is the intracluster distance of cluster C_k.In the case of contingency tables, the distance used is Chi-2 distance. The main goal of the measure is to maximise the intercluster distances and minimise the intracluster distances. Therefore, the number of clusters that maximises D is taken as the optimal number of clusters.

Davies-Bouldin Validity Index

The DB index (Davies and Bouldin 1979) is a function of the ratio of the sum of within-cluster scatter to between-cluster separation.

$$DB = \frac{1}{K}\sum_k max_{k \neq k'} \frac{S_n(c_k)+S_n(c_{k'})}{S(c_k,c_{k'})}$$

where K is the number of clusters, S_n is the average distance of all objects from the cluster C_k to their cluster centre c_k, $S(c_k, c_{k'})$ distance between clusters centres c_k and $c_{k'}$. In the case of contingency tables, the distance used is Chi-2 distance. Hence, the ratio is small if the clusters are compact and far from each other. Consequently, Davies-Bouldin index will have a small value for a good clustering.

Silhouette Index

The Silhouette validation technique (Rousseeuw 1987) calculates the silhouette width for each sample, average silhouette width for each cluster and overall average silhouette width for a total data set. The average silhouette width could be applied for evaluation of clustering validity and also could be used to decide how good is the number of selected clusters. To construct the silhouettes $S(i)$ the following formula is used:

$$S(i) = \frac{b(i) - a(i)}{maxa(i), b(i)}$$

where $a(i)$ is the average dissimilarity of i-object to all other objects in the same cluster and $b(i)$ is the minimum of average dissimilarity of i-object to all objects in other cluster (in the closest cluster). If silhouette value is close to 1, it means that sample is "well-clustered" and it was assigned to a very appropriate cluster. If silhouette value is about zero, it means that that sample could be assign to another closest cluster as well, and the sample lies equally far away from both clusters. If silhouette value is close to 1, it means that sample is "misclassified" and is merely somewhere in between the clusters. The overall average silhouette width for the entire plot is simply the average of the $S(i)$ for all objects in the whole dataset. The number of cluster with maximum overall average silhouette width is taken as the optimal number of the clusters.

C-Index

This index (Hubert and Levin 1976) is defined as follows:

$$C = \frac{S - S_{min}}{S_{max} - S_{min}}$$

where S is the sum of distances over all pairs of patterns from the same cluster. Let l be the number of those pairs. Then S_{min} is the sum of the l smallest distances if all pairs of patterns are considered (i.e. if the patterns can belong to different clusters). Similarly S_{max} is the sum of the l largest distances out of all pairs. Hence a small value of C indicates a good clustering.

Baker and Hubert index

Baker and Hubert index (BH) (Baker and Hubert 1975) is an adaptation of Goodman and Kruskal Gamma statistic (Goodman and Kruskal 1954).It is defined as:

$$BH(k) = \frac{S^+ - S^-}{S^+ + S^-}$$

where S^+ is the number of concordant quadruples, and S^- is the number of disconcordant quadruples. For this index all possible quadruples (q, r, s, t) of input parameters are considered. Let d(x,y) be the distance between the samples x and y. A quadruple is called concordant if one of the following two conditions is true:

- $d(q, r) < d(s, t)$, q and r are in the same cluster, and s and t are in different clusters.
- $d(q, r) > d(s, t)$, q and r are in different clusters, and s and t are in the same cluster.

By contrast, a quadruple is called disconcordant if one of following two conditions is true:

- $d(q, r) < d(s, t)$, q and r are in different clusters, and s and t are in the same cluster.
- $d(q, r) > d(s, t)$, q and r are in the same cluster, and s and t are in different clusters.

Obviously, a good clustering is one with many concordant and few disconcordant quadruples. Values of this index belong to [-1,1]. Large values of BH indicate a good clustering.

Krzanowski and Lai Index

KL index proposed by (Krzanowski and Lai 1988) is defined as

$$KL(k) = \left| \frac{DIFF(k)}{DIFF(k+1)} \right|$$

where $DIFF(k) = (k-1)^{2/p}W_{k-1} - k^{2/p}W_k$ and p is the number of variables. The number of clusters that maximizes KL(k) is the good number of clusters.

Calinski and Harabasz Index

The CH index (Calinsky and Harabsz 1974) is defined as:

$$CH(k) = \frac{B/(k-1)}{W/(n-k)}$$

where B is the sum of squares among the clusters, W is the sum of squares within the clusters, n is the number of data points and k is the number of clusters. In the case of groups of equal sizes, CH is generally a good criterion to indicate the correct number of groups. The best partition is indicated by the highest CH value.

Experimental results

CROKI2 algorithm uses k-means to cluster rows and columns. Therefore, the number of clusters needs to be specified by user. Once CROKI2 is applied to data set, we use all indices presented above to validate clustering alternatively

DataSet	K	L
Data Set 1	3	3
Data Set 2	4	4
Data Set 3	5	4
Data Set 4	6	3
Data Set 5	6	6

Table 2: DataSets table

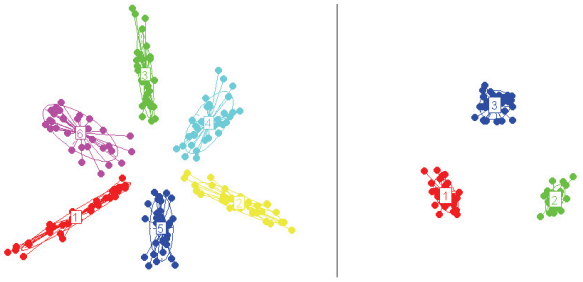

Figure 1: Projection of biclusters of Data set 4 on first and second principal axes obtained by a principal component analysis. Left are row-clusters and right column-clusters. Both clusters on rows and columns are well separated.

on rows and columns. For our study, we used 6 synthetic two-dimensional data sets. Every dataSet is composed of 200 rows and 100 columns generated around K clusters on rows and L clusters in columns. (See table 2) .

Each index is applied first to row partition then to the column partition. For each partition, the couple of clusters that correspond to the best value of the index is choosen as the good couple of clusters (see example in table 3).

Table 3: Results of application of indices on Data set 1

Indices	Row partition	Column partition	Best couple
Dunn	**(3,3)**	**(3,3)**	**(3,3)**
	(3,4)	(4,3)	
	(3,5)	(5,3)	
	(3,6)	(6,3)	
BH	**(3,3)**	**(3,3)**	**(3,3)**
	(3,5)	(5,3)	
	(3,6)	(6,3)	
HL	(6,2)	(4,6)	x
KL	(4,2)	(3,2)	x
DB	**(3,3)**	**(3,3)**	**(3,3)**
	(3,5)	(4,3)	
	(3,6)	(6,3)	
CH	**(3,3)**	**(3,3)**	**(3,3)**
	(3,4)	(4,3)	
	(3,5)	(6,3)	
Silhouette	**(3,3)**	**(3,3)**	**(3,3)**

Consequently, for each index, there are two sets of solutions, one set is for the row partition and the other set is for the column partition. For example, (3,3), (3,4), (3,5) and (3,6) are couples of clusters that maximize BH index when applied to row partition of Data set 1. When it is applied to the column partition, the maximum value of the index is obtained with couples (3,3), (4,3), (5,3) and (6,3) (figure 2).

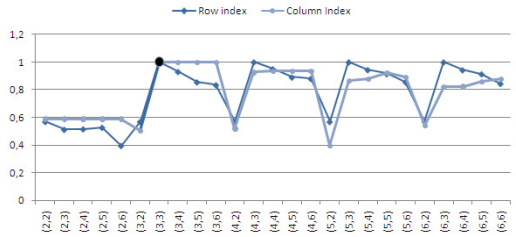

Figure 2: BH index calculated on row partition and column partition of Data Set 1

If one couple of clusters (K,L) belongs to the two sets of solutions, for example (3,3) in data set 1, then this is the good couple of clusters (see figures 2 and 3). In this case, the index is able to find the correct number of clusters on two dimensions.

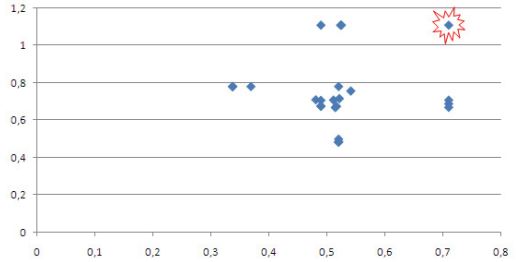

Figure 3: Cloud of data points with BH column index on axis Y and BH row index on axis X. There is only one solution (3,3) for Data Set 1 represented by the highlighted point.

When there is no couple of clusters (K, L) that maximizes simultaneously row Index and column index (fig. 4) i.e. there are many solutions for both row partition and column partition, the solution depend on the context and user preferences.

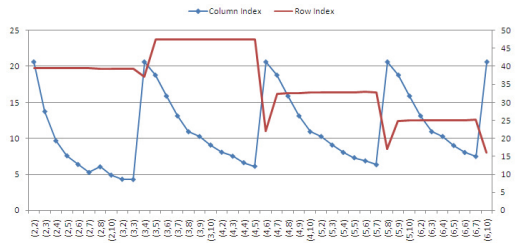

Figure 4: Index HL calculated on row partition and column partition of Data Set 3. There is no couple of clusters (K,L) that maximizes simultaneously row Index and column index. There are many solutions for both row partition and column partition.

In fact, we can choose couple of clusters that corresponds to best value of row index or column index. In the figure 5, (4,4) is the best solution for column partition and (4,5) is the best solution for row partition. However, (5,2) has better value of column index than (4,5) and better value of row index than (4,4). we propose to compute a weighted index :

$$GlobalIndex = \alpha * RowIndex + \beta * ColumnIndex$$

where $\alpha + \beta = 1$
In this case, α and β values depend on the relevancy of the row partition or the column partition for the user.

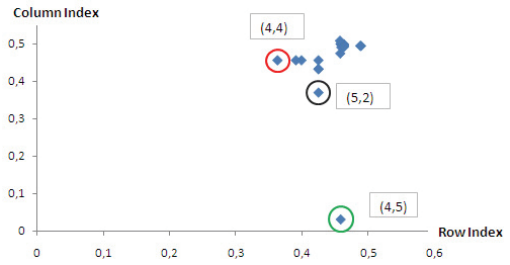

Figure 5: Cloud of data points with DB column index on axis Y and DB row index on axis X. There are many solutions for Data Set 3. (4,4) is the best solution for column partition and (4,5) is the best solution for row partition.

Table 4: Comparison of indices results on synthetic datasets

DataSets	DS1	DS2	DS3	DS4	DS5
Correct number of clusters	(3,3)	(4,4)	(5,4)	(6,3)	(6,6)
Dunn Index	**(3,3)**	**(4,4)**	**(5,4)**	(6,3) (9,6)	**(6,6)**
BH Index	**(3,3)**	**(4,4)**	**(5,4)**	**(6,3)**	**(6,6)**
HL index	x	x	x	x	x
KL index	x	x	x	x	x
DB Index	**(3,3)**	**(4,4)**	(4,4) (4,5)	(6,3) (3,4)	**(6,6)**
CH index	**(3,3)**	**(4,4)**	(4,4)	(6,3) (3,3)	**(6,6)**
Silhouette	**(3,3)**	**(4,4)**	(4,4)	(6,3) (2,3) (2,4)	**(6,6)**

The table 4 summarizes the results of the validity indices, for different clustering schemes of the above-mentioned data sets as resulting from the simultaneous clustering using CROKI2 with its input value (number of clusters on rows and columns), ranging between 2 and 12. When the number of clusters is the same on rows and columns, Dunn, BH, CH, DB and Sihouette indices are able to identify the best couple of clusters (K,L). But when the number of clusters on rows and columns is much different, only BH index is able to identify the correct number of clusters fitting the data set. (See table 4).

Conclusion and future work

In this paper, we proposed to extend the use of some indices used initially for classic clustering to biclustering algorithms, especially CROKI2 algorithm for contingency tables. Experimental results show that these indices are able to find correct number of clusters when applied to data sets with diagonal structure i.e data sets having the same number of clusters on rows and columns. This work can be improved by testing other indices on synthetic or real data sets with known partitions.

References

Baker, F., and Hubert, L. 1975. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association* 31–38.

Calinsky, R., and Harabsz, J. 1974. A dendrite method for cluster analysis. *Communications in statistics* 1–27.

Charrad, M.; Lechevallier, Y.; Saporta, G.; and Ahmed, M. B. 2008. Le bi-partitionnement : Etat de l'art sur les approches et les algorithmes. *Ecol'IA'08*.

Davies, D., and Bouldin, D. 1979. A cluster separation measure. *IEEE Trans. Pattern Anal. Machine Intell. 1 (4)* 224–227.

Dunn, J. 1974. Well separated clusters and optimal fuzzy partitions. *Journal Cybern.* 95–104.

Goodman, L., and Kruskal, W. 1954. Measures of association for cross-validation. *J. Am. Stat. Assoc. 49* 732–764.

Govaert, G. 1983. Classification croise. *Thse de doctorat d'tat, Paris*.

Govaert, G. 1995. Simultaneous clustering of rows and columns. *Control and Cybernetics* 437–458.

Halkidi, M.; Vazirgiannis, M.; and Batistakis, I. 2000. Quality scheme assessment in the clustering. *Process. In Proceedings of PKDD, Lyon, France* 79–132.

Hartigan, J. 1975. Clustering algorithms. *Wiley*.

Hubert, L., and Levin, J. 1976. A general statistical framework for assesing categorical clustering in free recall. *Psychological Bulletin* 1072–1080.

Krzanowski, W., and Lai, Y. 1988. A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics 44* 23–34.

Madeira, S., and Oliveira, A. 2004. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 24–45.

Mirkin, B. 1996. Mathematical classification and clustering. *Dordrecht: Kluwer*.

Nadif, M., and Govaert, G. 2005. Block clustering of contingency table and mixture model. *Intelligent Data Analysis IDA'2005, LNCS 3646, Springer-Verlag Berlin Heidelberg* 249–259.

Prelic, A.; Bleuler, S.; Zimmermann, P.; A.Wille; Bhlmann, P.; Gruissem, W.; Hennig, L.; Thiele, L.; and Zitzler, E. 2006. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics, 22(9)* 1122–1129.

Rousseeuw, P. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 53–65.

Tanay, A.; Sharan, R.; and Shamir, R. 2004. Biclustering algorithms: A survey. *In Handbook of Computational Molecular Biology, Edited by Srinivas Aluru, Chapman*.

Theodoridis, S., and Koutroubas, K. 1999. Pattern recognition. *Academic Press* 79–132.