Entropic-Genetic Clustering

Mihaela Breaban*, Henri Luchian*, Dan A. Simovici**

*Univerity of Jassy, Dept. of Computer Science, Jassy, Romania,
e-mail:pmihaela,hluchian@infoiasi.ro
**Univ. of Massachusetts Boston, Massachusetts 02125, USA,
e-mail:dsim@cs.umb.edu

Abstract. This paper addresses the clustering problem given the similarity matrix of a dataset. We define two distinct criteria with the aim of simultaneously minimizing the cut size and obtaining balanced clusters. The first criterion minimizes the similarity between objects belonging to different clusters and is an objective generally met in clustering. The second criterion is formulated with the aid of generalized entropy. The trade-off between these two objectives is explored using a multi-objective genetic algorithm with enhanced operators.

1 Introduction

This paper addresses the clustering problem given the similarity matrix of a dataset. A straightforward representation of the problem instance in this case is a weighted graph, having the objects as vertices and weighted edges expressing the similarity between objects. This leads to a graph clustering/partitioning problem which aims at identifying groups of strongly inter-connected vertices. A survey of graph clustering is presented in Schaeffer (2007).

A *similarity space* is a pair (S, w), where $w : S \times S \longrightarrow \mathbb{R}$ is a function such that $w(s, t) \geq 0$, $w(s, t) = w(t, s)$, and $w(s, s) = 1$. for every $s, t \in S$. A similarity space (S, w) can be regarded as a labelled graph $G = (S, E, w)$, referred to as the *similarity graph*, where the set of edges E is defined as $E = \{(s_i, s_j) \mid s_i, s_j \in S \text{ and } w(s_i, s_j) > 0\}$. If S is a finite set $S = \{s_1, \ldots, s_n\}$, the dissimilarity w is described by a symmetric matrix $W \in \mathbb{R}^{n \times n}$, where $w_{ij} = w(s_i, s_j)$ for $1 \leq i, j \leq n$.

A *k-way clustering of a finite similarity space* (S, w) is a partition $\kappa = \{C_1, \ldots, C_k\}$ of S. The sets $C_1, \ldots, C_k$ are the clusters of κ. We seek a k-way partition of S, κ such that the cut size (i.e. the sum of weights of edges between clusters in the similarity graph) is minimal, and $|C_p| \approx |C_q|$, for $1 \leq p, q \leq k$, which means that the sizes of the clusters are as equal as possible. Presentations of the state-of-the-art of graph clustering can be found in Fjällström (1998), Karypis and Kumar (1998).

The paper is structured as follows. Section 2 examines the two objectives which have to be optimized as stated in the problem definition. Section 3 provides a brief survey on the genetic algorithms for clustering with an emphasis on the multi-objective formulation; the representation and the operators we used are detailed. Section 4 presents experimental results.

2 Clustering as multi-objective optimization

Let $\kappa = \{C_1, \ldots, C_k\}$ a clustering of the objects of the set $S = \{s_1, \ldots, s_n\}$. The matrix $X \in \mathbb{R}^{n \times k}$ defined by $x_{ip} = 1$ if $s_i \in C_p$ and $x_{ip} = 0$ otherwise, represents the clustering κ. Note that each row of this matrix contains a single 1 and that the total number of 1 entries equals the number n of elements of the set S.

The matrix $Y = X'X \in \mathbb{R}^{k \times k}$ is given by

$$y_{pq} = \sum_{i=1}^{n} x'_{pi} x_{iq} = \sum_{i=1}^{n} x_{ip} x_{iq} \tag{1}$$

for $1 \leq p, q \leq k$. Since any two clusters C_p, C_q are disjoint, this a diagonal matrix. Its diagonal elements are $y_{pp} = |C_p|$ for $1 \leq p \leq k$.

Let $\mathcal{G} = (S, E, w)$ be the similarity graph of S. The symmetric matrix $W \in \mathbb{R}^{n \times n}$ is defined by $w_{ij} = w(s_i, s_j)$ if $i \neq j$ and $w_{ij} = 1$ if $i = j$, for $1 \leq i, j \leq n$.

Let $Z = X'WX \in \mathbb{R}^{k \times k}$. We have $z_{pq} = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ip} w_{ij} x_{jq}$ for $1 \leq i, j \leq n$. Therefore, for the distinct clusters C_p, C_q, z_{pq} is precisely the value of $\mathsf{cut}(C_p, C_q)$. Note also that $z_{pp} = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ip} w_{ij} x_{jp}$ equals the sum of the similarities between the objects of the cluster C_p. Clearly, to achieve maximal intra-clustering cohesion and minimal inter-clustering dissimilarity it is necessary that the trace of the matrix Z (that is, the sum of the diagonal elements of Z) to be maximal and the sum of the off-diagonal elements of Z to be minimal.

Since Z is a non-negative matrix, its norm $\parallel Z \parallel_1 = \sum_{p=1}^{k} \sum_{q=1}^{k} |z_{pq}|$ coincides with the sum of its elements. Moreover, $\parallel Z \parallel_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij}$ and is constant for a given similarity matrix W, regardless of the clustering X. Therefore, the total weight of the inter-cluster cuts equals $\parallel Z \parallel_1 - \mathsf{trace}(Z)$ and minimizing it is equivalent to maximizing the total within clusters similarity which is given as $\mathsf{trace}(Z) = \sum_{p=1}^{k} z_{pp}$.

We use a novel approach to insure that the clusters of κ are balanced. To this end, we use the generalized entropy of partitions of finite sets (see Simovici and Djeraba (2008)) introduced by Daróczy (1970) and by Havrda and Charvat (1967) and axiomatized by Simovici and Jaroszewicz (2002). The use of entropy is suggested by the fact that it is a natural instrument for evaluating the balancing quality of a probability distribution, and, therefore, the balancing quality of a partition of a finite set.

For a partition $\kappa = \{C_1, \ldots, C_k\}$ of a set S and a number $\beta > 1$, the β-entropy is defined by $\mathcal{H}_\beta(\kappa) = \frac{1}{1 - 2^{1-\beta}} \left(1 - \sum_{p=1}^{k} \frac{|C_p|}{|S|}^{\beta}\right)$. Note that $\lim_{\beta \to 1} \mathcal{H}_\beta(\kappa) = -\sum_{p=1}^{k} \frac{|B_p|}{|S|} \log_2 \frac{|B_p|}{|S|}$. In other words, the Shannon entropy is a limit case of the generalized entropy.

An important special case of the entropy is obtained for $\beta = 2$. We have $\mathcal{H}_2(\kappa) = 2\left(1 - \sum_{p=1}^{k} \frac{|C_p|}{|S|}^2\right)$ and this is the well-known *Gini index*, $\mathsf{gini}(\kappa)$ used frequently in statistics.

The largest value of $\mathcal{H}_\beta(\kappa)$ is obtained when κ consists of singletons, that is, when $k = n$ and $\kappa = \alpha_S = \{\{s_i\} \mid 1 \leq i \leq n\}$ and is $\mathcal{H}_\beta(\alpha_S) = \frac{1}{1 - 2^{1-\beta}} \left(1 - \frac{|1|}{|S|}^{\beta - 1}\right)$; the least value of $\mathcal{H}_\beta(\kappa)$ is obtained for $\kappa = \omega_S$ and equals 0. Maximization of the entropy can be used as a criterion for ensuring the uniformity of the cluster sizes. We will use the Gini index of κ because it presents certain computational advantages as shown next.

Theorem 2.1 *Let $\kappa = \{C_1, \ldots, C_k\}$ a clustering of the objects of the set $S = \{s_1, \ldots, s_n\}$ and let $X \in \mathbb{R}^{n\times k}$ be the characteristic matrix of the clustering. We have gini$(\kappa) = 2(1 -$ trace$(X'XX'X))$.*

Two objectives are used to find a balanced k-clustering κ:

(i) minimization of the total cut of the clustering partition, which amounts to minimization of

$$f_1(X) = \parallel Z \parallel_1 - \text{trace}(Z) = \parallel X'WX \parallel_1 - \text{trace}(X'WX) \tag{2}$$

(ii) maximization of cluster uniformity, which is equivalent to the maximization of the Gini index of κ, or to the minimization of

$$f_2(X) = \text{trace}(X'XX'X) \tag{3}$$

We seek X subjected to the conditions $x_{ip} \in \{0, 1\}$ for $1 \leq i \leq n$ and $1 \leq p \leq k$. Depending on the aspects we need to emphasize in the clustering we can use a convex combination of these criteria $\Phi_a(X) = af_1(X) + (1-a)f_2(X) = a(\parallel X'WX \parallel_1 - \text{trace}(X'WX)) + (1 - a)\text{trace}(X'XX'X)$, where $a \in [0, 1]$. To simultaneously minimize criteria f_1 and f_2, also a non-linear combination can be used:

$$\Psi(X) = \frac{f_1(X)}{n^2 - f_2(X)} = \frac{\parallel X'WX \parallel_1 - \text{trace}(X'WX)}{n^2 - \text{trace}(X'XX'X)}. \tag{4}$$

3 The clustering algorithm

We use a genetic algorithm (GA) to deal with the graph clustering problem. In Luchian et al. (1994) a new clustering encoding is proposed which considers only cluster representatives, allowing for simultaneous search of the optimum number of clusters and the optimum partition. The partition is constructed in a manner similar to k-means: the data items are assigned to clusters based on the proximity to the cluster representatives.

Multi-objective GAs are used in the optimization of several conflicting objectives. These algorithms optimize simultaneously several objectives and return a set of non-dominated solutions which approximate the Pareto front. For a problem involving m objectives denoted with $f_i, 1 \leq i \leq m$ which have to be minimized, a solution x is *dominated* by a solution x^* if $f_i(x^*) \leq f_i(x)$, for all i, $1 \leq i \leq m$ and there exists j such that $f_j(x^*) < f_j(x)$.

The Pareto optimal set of solutions X^* consists of all those solutions for which no improvement in an objective can be made without a simultaneous worsening in some other objective. In other words, the Pareto front consists of all solutions that are not dominated by any other solution.

The multi-objective scheme we use to tackle the graph clustering problem is PESA-II obtained by Corne et al. (2001). The algorithm maintains two populations of solutions. An *external population* stores mutually non-dominated clustering solutions, which correspond to different trade-offs between the two objectives. At each iteration an *internal population* is created by selecting chromosomes from the external population. This selection phase takes into account the distribution of solutions across the two objectives by maintaining a hypergrid of equally sized cells in the objective space. After selection, the crossover and mutation operators are applied within the internal population. The external population is updated by joining the two populations and eliminating the dominated solutions.

A solution of the partitioning problem is represented in our GA as a string of length n (the number of vertices in the graph), taking values in the set $\{1, \ldots, k\}$, where k is the number of clusters.

Initially a minimum spanning tree (MST) is constructed. Half of the population is initialized with candidate solutions created by repeating the following procedure: $k - 1$ edges are randomly removed from MST and the connected components are marked as individual clusters. The rest of the population is filled with chromosomes generated randomly.

The crossover operator computes the intersection of two partitions (individuals in the population) and merges clusters of the intersection to produce a new partition having k clusters. The decisions are made with regard to the two objectives to be optimized and therefore two distinct crossover operators are use. One operator aims at decreasing the cut size and therefore performs some iterations of the hierarchical agglomerative clustering algorithm using average linkage metric. The second operator merges iteratively the two smallest clusters aiming at balancing the clusters, until a number of k clusters is reached.

The mutation operator applies to a single partition and reallocates a randomly chosen vertex and its most similar adjacent vertices to a randomly chosen cluster. The number of adjacent vertices to be reallocated decreases during the run so that in final iterations only small perturbations are allowed.

The fitness functions used in our multi-objective genetic approach are based on the two objectives presented in Section 2 and are formulated for minimization. We maximize the entropy by minimizing the Gini index criterion 3 and minimize the average cut size as expressed by Equation (4).

4 Experiments

Experiments on synthetic datasets produced by a synthetic generator[1] was used to create five datasets, each one consisting of 1500 data items grouped into 3 clusters. Overlapping clusters are rejected and regenerated, until a valid set of clusters is found. The datasets are named as $n_1 - n_2 - n_3$ with n_p denoting the size of cluster p. The size of the internal population was set to 10. The maximum size for the external population containing non-dominated solutions was set to 500 but in our experiments it did not exceed 250 elements. The number of iterations was set to 10000.

Figure 1 presents the set of non-dominated solutions returned in the last iteration of the genetic algorithm. The fitness values corresponding to the two criteria to be optimized are normalized in range $[0, 1]$. The horizontal axis corresponds to Criterion (3) and the vertical axis corresponds to the average cut size. The solution closest to the real partition of the dataset is marked as a square; in this regard, the Adjusted Rand Index (ARI) (see Hubert (1985)) is used to evaluate the quality of the partitions. The partition corresponding to the best/minimum score computed as sum between the two objectives is marked as a triangle.

The shape of the Pareto front plotted for datasets of various degrees of uniformity is an indicator of the interaction between the two objectives. Because both objectives are formulated

[1] http://dbkgroup.org/handl/generators/generators.pdf

for minimization, the desirable position of a clustering is towards the southwestern corner of the diagram. The average cut size cannot be lowered indefinitely without severely affecting the balancing of the clusters. A gap is recorded for the criterion measuring the uniformity once the optimum solution (with regard to the true partition) is met. This gap is due to the dependency between the two objectives: the second criterion measuring the average cut size is built using both the cut size and the entropy (the first objective).

Note that the ARI takes values higher than 0.95 in all cases, which indicates a very close match to the real partition. Using a a convex combination of the two criteria we can identify a near-optimum solution if the final set of non-dominated solutions is normalized within the same range for both objectives.

To highlight the advantages of our multi-objective approach over other graph clustering methods, the well-known recursive partitioning algorithm METIS[2] is used, which delivers only perfectly-balanced clusters, even though in practice this may not be the best solution from the point of view of the cut size.

Table 1 presents comparative results. The ARI is reported for the solutions corresponding to: 1) the partitioning with the highest ARI value, 2) the best partitioning under the convex combination (average) over the two criteria normalized in range [0,1] and 3) the best balanced partitioning from the non-dominated set of solutions delivered by the genetic algorithm, which corresponds to clusters of equal size. Also, the ARI is reported for the partition computed with METIS.

Instance	best under ARI	best convex combination	best balanced	METIS
500-500-500	0.9999	0.9880	0.9999	0.9999
500-600-400	0.9909	0.9909	0.8111	0.8118
500-700-300	0.9625	0.9535	0.6588	0.6817
500-800-200	0.9839	0.9839	0.5764	0.5954
500-900-100	0.9950	0.9950	0.5615	0.5493

TAB. 1: Comparative Results

Our algorithm is comparable with METIS with regard to the quality of the balanced partitioning. However, a near-optimal match with the true partitioning of the dataset can be extracted from the final the set of non-dominated solutions in a unsupervised manner, using a convex combination of the two criteria we use. Furthermore, this set can be explored to extract the most convenient solution for the problem being solved.

Also Figure 1 shows that the non-linear criterion $\Psi(X)$ given by Equality (4) biases the search towards highly balanced clusters and can be successfully used when a perfectly balanced partition is desired. Its convex combination with the criterion measuring the balancing degree of the partitioning is necessary to retrieve the true partitioning.

[2] http://glaros.dtc.umn.edu/gkhome/

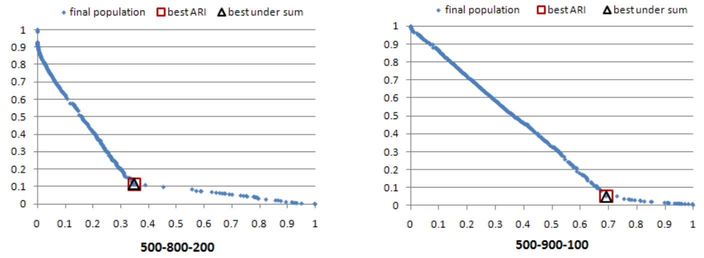

FIG. 1: The set of non-dominated solutions for various datasets.

References

Corne, D. W., N. R. Jerram, J. D. Knowles, and M. J. Oates (2001). Apesa-ii: regionbased selection in evolutionary multiobjective optimization. In *Proc. Genetic and Evolutionary Computation Conference*, pp. 283–290.

Daróczy, Z. (1970). Generalized information functions. *Information and Control 16*, 36–51.

Fjällström, P.-O. (1998). Algorithms for graph partitioning: A survey. *Linköping Electronic Articles in Computer and Information Science 3*.

Havrda, J. H. and F. Charvat (1967). Quantification methods of classification processes: Concepts of structural α-entropy. *Kybernetica 3*, 30–35.

Hubert, A. (1985). Comparing partitions. *Journal of Classification 2*, 193–198.

Karypis, G. and V. Kumar (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing 20*, 359–392.

Luchian, S., H. Luchian, and M. Petriuc (1994). Evolutionary automated classification. In *Proceedings of the First Congress on Evolutionary Computation*, pp. 585–588.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review, Elsevier I*, 27–64.

Simovici, D. A. and C. Djeraba (2008). *Mathematical Tools for Data Mining – Set Theory, Partial Orders, Combinatorics*. London: Springer-Verlag.

Simovici, D. A. and S. Jaroszewicz (2002). An axiomatization of partition entropy. *IEEE Transactions on Information Theory 48*, 2138–2142.

Résumé

Cet article traite le problème de classification à partir d'une matrice de similarité sur un ensemble de données. Nous définissons deux critères distincts pour obtenir des clusters équilibrés et bien separés. Le premier critère minimise la similarité entre les objets appartenant à différents groupes et constitue un objectif généralement atteint en matière de regroupement. Le deuxième critère est formulé avec l'aide de l'entropie généralisée. Le compromis entre ces deux objectifs est exploré en utilisant un algorithme génétique multi-objectifs avec opérateurs renforcés.