Novel Biobjective Clustering (BiGC) based on Cooperative Game Theory

Vikas K. Garg, Y. Narahari, *Fellow, IEEE,* and M. Narasimha Murty

Abstract—We propose a new approach to clustering. Our idea is to map cluster formation to coalition formation in cooperative games, and to use the Shapley value of the patterns to identify clusters and cluster representatives. We show that the underlying game is convex and this leads to an efficient biobjective clustering algorithm which we call BiGC. The algorithm yields high quality clustering with respect to average point-to-center distance (potential) as well as average intra-cluster point-to-point distance (scatter). We demonstrate the superiority of BiGC over state-of-the-art clustering algorithms (including the center based and the multi-objective techniques) through a detailed experimentation using standard cluster validity criteria on several benchmark datasets. We also show that BiGC satisfies key clustering properties such as order independence, scale invariance, and richness.

1 Introduction

Clustering is the unsupervised assignment of data points to subsets such that points within a subset are more similar to *each other* than points from other subsets. Clustering is a very well studied problem in data mining, machine learning, and related disciplines primarily because of its wide applicability in domains so diverse as climate (Steinbach et. al [46]), microarray data analysis (Jiang et. al [25]), data streams (Zhang et. al [56], Chen and Tu [10]), privacy preserving mining (Kabir, Wang, and Bertino [26], Vaidya and Clifton [48]), and subsequence mining (Wang et. al, [51]), etc. Besides, clustering has also been used in solving extremely large scale problems, see for example, Li et al. [33]. Clustering also acts as a precursor to many data processing tasks including classification (Jain, Murty and Flynn [23]).

Typically, the different clustering techniques strive to achieve optimal clusters with respect to some objective function, for instance, the k-means based algorithms seek to minimize the average squared distance between each point and its closest cluster center. In this work, we focus on two key objectives, *potential* and *scatter*, simultaneously: we seek to minimize a) the average distance between each point and its closest center (potential), and b) the average intra-cluster point-to-point distance (scatter). We emphasize that incorporating this gestalt or collective behavior of points within each cluster is fundamental to the very notion of clustering. Our work is strongly motivated by what has been variously described in the literature as the *context-sensitive information* or *coherency* (Bulo' and Pelillo [4]): clustering should be done not just on the basis of distance between a pair of points but also on their relationship to other data points.

In this paper, we propose a promising new approach to clustering based on cooperative game theory. Our idea is to map cluster formation to coalition formation, using Shapley value, in an appropriately defined convex game setting. Shapley value is a fair solution concept in that it divides the collective or total value of the game among the players according to their marginal contributions in achieving that collective value. We strive to make best use of this intrinsic property of marginal contribution based *fairness* for efficient clustering.

In his work on unification of clustering, [28], Kleinberg considered three desirable properties (scale invariance, richness, and consistency) and proved an impossibility theorem, showing that no clustering algorithm satisfies all of these properties simultaneously. In this paper, we introduce order independence as another desirable property, and provide necessary and sufficient conditions for order independence. Our algorithm for clustering, BiGC, satisfies scale invariance, richness, and order independence.

1.1 Motivation

Clustering is the assignment of data points to subsets such that points within a subset are more similar to *each other* than points from other subsets. According to Backer and Jain [3], "in cluster analysis, a group of objects is split into a number of more or less homogeneous subgroups on the basis of an often subjectively chosen measure of similarity (i.e., chosen subjectively based on its ability to create "interesting" clusters), such that *the similarity between objects within a subgroup is larger than the similarity between objects belonging to different subgroups*". Similar views are echoed in other works on clustering, e.g., Xu and Wunsch [52]. We believe that most of the existing popular algorithms do not completely reflect the intrinsic notion of a cluster, since they try to minimize the distance of every point from its closest cluster representative alone, while overlooking the importance of other points in the same cluster. Although this approach succeeds in

- *Vikas K. Garg is with Toyota Technological Institute at Chicago (TTI-C). E-mail: vkg@ttic.edu, montsgarg@gmail.com*
- *Y. Narahari and M. Narasimha Murty are with Department of Computer Science and Automation (CSA), Indian Institute of Science (IISc), Bangalore. E-mail: {hari, mnm}@csa.iisc.ernet.in*

optimizing the average distance between a point and its closest cluster center, it conspicuously fails to capture what has been described by Michalski and others as the "context-sensitive" information [36]: clustering should be done not just on the basis of distance between a pair of points, A and B, but also on the relationship of A and B to other data points. Therefore, there is a need for an algorithm that gives an optimal solution in keeping both the point-to-center and point-to-point distances, within a cluster, to a minimum. We emphasize that incorporating the gestalt or collective behavior of points within the same cluster is fundamental to the very notion of clustering, and this provides the motivation for our work.

In addition, it is more intuitive to characterize similarity between different points as compared to the distance between them since the distance measure may not necessarily be scale invariant. Further, as described later, there are certain intrinsic properties that are crucial in the context of clustering. The Shapley value is an important solution concept, from cooperative game theory, that satisfies these properties and thereby captures key natural criteria for clustering.

1.2 Related Work

The machine learning and pattern recognition literature abounds in algorithms on clustering. The techniques such as Partitioning Around Medoids (PAM) [27] are based on vector quantization. The density estimation based models such as Gaussian Mixture Density Decomposition (GMDD) [57], information theory based models such as Bregman Divergence based clustering [5], graph theory based models such as Normalized Cut [35], [45] and Correlation Clustering [6], [30], neural networks based models such as Self-Organizing Map (SOM) [29], kernel based models such as Support Vector Clustering (SVC) ([7], [53]) and Maximum Margin Clustering [34] and data visualization based models such as Principal Component Analysis (PCA) [22] have received considerable attention from the research community. Miscellaneous other techniques such as Evolutionary Clustering [9], Projected Clustering ([54], [37]), Subspace Clustering [50], and Ensemble Clustering [17], etc. have also been proposed. Some work has been done on clustering using evolutionary games [31] and quantum games [32]. We refer the reader to [52] and [23] for an extensive overview of the different clustering methods.

Multi-objective Optimization based algorithms for clustering (Handl and Knowles [20], Suresh et al. [47]) have gained prominence recently. A game theoretic approach, based on Nash equilibrium, has been proposed (Gupta and Ranganathan [18]) to simultaneously optimize the compaction and equi-partitioning of spatial data.

The Leader algorithm [14] is a prototype incremental algorithm that dynamically assigns each incoming point to the nearest cluster. For a known value of k, the k-means algorithm and its variants [2], [21], [39], based on vector quantization are the most popular clustering algorithms. For the rest of this paper, by k-means, unless specified otherwise, we refer to a very specific and widely used k-means implementation: the Lloyd's algorithm [39], which aims to find a clustering with a good k-means cost.

1.3 Contributions and Outline

In this paper, we make the following contributions:

- We formulate the problem of clustering as a *transferable utility (TU) cooperative game* among the data points and show that the underlying characteristic form game is *convex*.
- We propose a novel approach, BiGC, for clustering the data points based on their Shapley values and the convexity of the proposed game theoretic model. BiGC provides high quality clustering with respect to both the *potential* (average distance between each point and its closest center) and the *scatter* (average intra-cluster point-to-point distance). To the best of our knowledge, this is the first approach to clustering that considers these two objective functions simultaneously.
- We demonstrate the efficacy of our approach through detailed experimentation. BiGC is compared with the popular k-means (and its state-of-the-art variants, k-means++ and HSI), Leader, Agglomerative Linkage, and several Multi-objective Optimization algorithms; the results of our experiments clearly show that BiGC provides a superior quality of clustering.
- We also show that BiGC satisfies certain desirable clustering properties such as *scale invariance*, *order independence*, and *richness*.

The rest of this paper is organized as follows. In §2, a succinct background encompassing important concepts from cooperative game theory is provided. Our Shapley value based clustering paradigm is presented in §3, along with BiGC that computes a closed form expression for Shapley value using the convexity of the underlying game. A detailed analysis of the experimental results is carried out in §4. We then characterize the ordering effects, and discuss the satisfiability of key clustering properties such as scale invariance, richness, and order independence by BiGC in §5. Finally, we present a summary of our work and highlight some directions for future work in §6.

2 Preliminaries

A cooperative game with transferable utility (TU) [38] is defined as the pair (N, v) where $N = \{1, 2, ..., n\}$ is the set of players and $v : 2^N \rightarrow \mathbb{R}$ is a mapping with $v(\phi) = 0$. The mapping v is called the characteristic function or the value function. Given any subset S of N, $v(S)$ is often called the value or the worth of the coalition S and represents the total transferable utility that can be achieved by the players in S, without help from the players in $N \setminus S$. The set of players N is called the grand coalition and $v(N)$ is called the value of the grand coalition. In the sequel, we use the phrases cooperative game, coalitional game , and TU game interchangeably.

A cooperative game can be analyzed using a *solution concept*, which provides a method of dividing the total value of the game among individual players. We describe below two important solution concepts, namely the *core* and the *Shapley value*.

2.1 The Core

A payoff allocation $x = (x_1, x_2, ..., x_n)$ denotes a vector in $\mathbb{R}^n$ with x_i representing the utility of player i where $i \in N$. The payoff allocation x is said to be *coalitionally rational* if $\sum_{i \in C} x_i \geq v(C), \ \ \forall C \subseteq N$. Finally, the payoff allocation x is said to be *collectively rational* if $\sum_{i \in N} x_i = v(N)$. The core of a TU game (N, v) is the collection of all payoff allocations that are coalitionally rational and collectively rational. It can be shown that every payoff allocation lying in the core of a game (N, v) is *stable* in the sense that no player will benefit by unilaterally deviating from a given payoff allocation in the core. The elements of the core are therefore potential payoff allocations that could result when rational players interact and negotiate among themselves.

2.2 The Shapley Value

The Shapley value is a solution concept that provides a unique expected payoff allocation for a given coalitional game (N, v). It describes an effective approach to the fair allocation of gains obtained by cooperation among the players of a cooperative game. Since some players may contribute more to the total value than others, an important requirement is to distribute the gains fairly among the players. The concept of Shapley value, which was developed axiomatically by Lloyd Shapley, takes into account the relative importance of each player to the game in deciding the payoff to be allocated to the players. We denote by

$$\phi(N, v) = (\phi_1(N, v), \phi_2(N, v), \ldots, \phi_n(N, v))$$

the Shapley value of the TU game (N, v). Mathematically, the Shapley value, $\phi_i(N, v)$, of a player i, $\forall v \in \mathbb{R}^{2^n - 1}$, is given by,

$$\phi_i(N, v) = \sum_{C \subseteq N - i} \frac{|C|!(n - |C| - 1)!}{n!} \{v(C \cup \{i\}) - v(C)\}$$

where $\phi_i(N, v)$ is the expected payoff to player i and $N - i$ denotes $N \setminus \{i\}$.

The Shapley value is the unique mapping that satisfies three key properties: linearity, symmetry, and carrier property [38]. The Shapley value of a player accurately reflects the bargaining power of the player and the marginal value the player brings to the game.

2.3 Convex Games

A cooperative game (N, v) is *a convex game* [44] if

$$v(C) + v(D) \leq v(C \cup D) + v(C \cap D), \ \ \forall C, D \subseteq N$$

Equivalently, a TU game (N, v) is said to be convex if for every player i, the *marginal contribution* of i to larger coalitions is larger. In other words,

$$v(C \cup \{i\}) - v(C) \leq v(D \cup \{i\}) - v(D), \\ \forall C \subseteq D \subseteq N - \{i\}, i \in N$$

where the marginal contribution $m(S, j)$ of player j in a coalition S is given by,

$$m(S, j) = v(S \cup \{j\}) - v(S), \ \ S \subseteq N, j \in N, j \notin S.$$

A very important property is that if a TU game (N, v) is convex, then the core of the game is non-empty and moreover, the Shapley value belongs to the core.

2.4 Shapley Value of Convex Games

Consider a permutation π of players in the game. Then, for any of a possible $|N|!$ such permutations, the *initial segments* of the ordering are given by

$$T_{\pi,r} = \{i \in N : \pi(i) \leq r\}, \ \ r \in \{1, ..., |N|\}$$

where $T_{\pi,0} = \{\}$ and $T_{\pi,|N|} = N$. Note that $\pi(i)$ refers to the position of the player i in the permutation π. To determine the core for a particular ordering π, we solve the equations

$$x_i^{\pi}(T_{\pi,r}) = v(T_{\pi,r}), \ \ r \in \{1, ..., |N|\}.$$

The solution to these equations defines a payoff vector x^{π} with elements given by

$$x_i^{\pi} = v(T_{\pi,\pi(i)}) - v(T_{\pi,\pi(i)-1}), \ \ \forall i = 1, 2, ..., |N|.$$

In fact, the payoff vectors x^{π} precisely represent the extreme points of the core in convex games. Moreover, it is known [44] that the Shapley value for a convex game is the center of gravity of x^{π}. Thus, if Π is the set of all permutations of N, then the Shapley value of player i can be computed as

$$\phi_i = \frac{1}{|N|!} \sum_{\pi \in \Pi} x_i^{\pi} \tag{1}$$

3 Shapley Value based Clustering

A central idea of this work is to map cluster formation to coalition formation in an appropriately defined TU game. The usage of Shapley value for clustering is in many ways natural as shown by the interpretation of the axioms of Shapley value in the context of clustering:

- **Symmetry** (*order independence*): For a game (N, v) and a permutation π on N, this axiom asserts that

  $$\sum_{i \in N} \phi_i(N, v) = \phi_{\pi(i)}(N, \pi v)$$

  Symmetry is extremely significant for achieving *order independence*, another desirable clustering property. Informally, we want the algorithms to yield the same final clustering across different runs, irrespective of the sequence in which data points are provided as input.

- **Preservation of Carrier** (*outlier detection*): For any game (N, v) such that $v(S \bigcup \{i\}) = v(S) \ \forall S \subseteq N$, this axiom asserts that $\phi_i(N, v) = 0$. This property implies that if a point does not contribute to the overall worth of a cluster, then it gets no incentive to become a member of that cluster. This takes care of the density issues since the *outliers* or the points in sparse regions are well separated from the points in the high density regions.

- **Additivity or Aggregation** (*scale invariance*): For any two games, (N, v) and (N, w),

  $$\phi_i(N, v + w) = \phi_i(N, v) + \phi_i(N, w), \text{ where}$$

  $$(v + w)(S) = v(S) + w(S)$$

  Additivity implies the *linearity* property: if the payoff function v is scaled by a real number α, then the

Shapley value is also scaled by the same factor. That is, $\phi_i(N, \alpha v) = \alpha\phi_i(N, v)$. Linearity is essential for achieving *scale invariance* with respect to the value function.

- **Pareto Optimality** (*gestalt behavior*): For any game (N, v), $\sum_{i \in N} \phi_i(N, v) = v(N)$. As an implication of this property, the overall worth that results because of the presence of every point in the dataset is distributed entirely among all the data points, and this characterizes the gestalt behavior.

In fact, Shapley value is the unique solution concept that satisfies these axioms simultaneously [38], and hence provides a well-motivated and compelling approach for clustering.

3.1 Cooperative Game Model

Consider a dataset $X = \{x_1, x_2, ..., x_n\}$ of n input instances. Given the dataset X, define a function, $d : X \times X \rightarrow \Re^+ \cup \{0\}$, where $d(x_i, x_j)\ \forall x_i, x_j \in X$ indicates the distance between x_i and x_j, with $d(x_i, x_i) = 0$; d can be any distance metric such as the Euclidean distance, for instance, depending on the application domain. Let $f^{'} : \Re^+ \cup \{0\} \rightarrow [0, 1)$ be a monotonically non-decreasing dissimilarity function over d such that $f^{'}(0) = 0$. Define a corresponding similarity mapping, $f : \Re^+ \cup \{0\} \rightarrow (0, 1]$, such that $f(a) = 1 - f^{'}(a)$. The problem of clustering can be viewed as grouping together those points which are less dissimilar as given by $f^{'}$ or equivalently, more similar as indicated by f.

We set up a cooperative game (N, v) among the input data points in the following way. In this setting, each of the n points corresponds to a player in this game, thereby $|N| = n$. Every player interacts with other players and tries to form a coalition or cluster with them, in order to maximize its value[1]. Now, we assign $v(\{x_i\}) = 0$, for all x_i such that x_i is not a member of any coalition. Further, define for a coalition T,

$$v(T) = \frac{1}{2} \sum_{\substack{x_i, x_j \in T \\ x_i \neq x_j}} f(d(x_i, x_j)) \tag{2}$$

We emphasize the relevance of defining the value function $v(.)$ for a coalition in this way. Our approach computes the total worth of a coalition as the sum of pairwise similarities between the points. Note that this formulation elegantly captures the notion of clustering in its natural form: points within a cluster are similar to *each other*.

3.2 Convexity of the Game

Theorem 1. *Define the value of an individual point x_i, $v(\{x_i\}) = 0\ \forall i \in \{1, 2, ..., n\}$, and that of a coalition T of n data points, $v(T) = \frac{1}{2} \sum_{\substack{x_i, x_j \in T \\ x_i \neq x_j}} f(d(x_i, x_j))$, where f is a similarity function. In this setting, the cooperative game (N, v) is a convex game.*

Proof. Consider any two coalitions C and D, $C \subseteq D \subseteq X \setminus \{x_p\}$, where $x_p \in X$. Then,
$v(D \cup \{x_p\}) - v(C \cup \{x_p\})$

$$= \frac{1}{2} \sum_{\substack{x_i, x_j \in D \\ x_i \neq x_j}} f(d(x_i, x_j)) + \sum_{x_i \in D} f(d(x_i, x_p)) - \frac{1}{2} \sum_{\substack{x_i, x_j \in C \\ x_i \neq x_j}} f(d(x_i, x_j)) - \sum_{x_i \in C} f(d(x_i, x_p))$$

$$= \frac{1}{2} \sum_{\substack{x_i, x_j \in D \setminus C \\ x_i \neq x_j}} f(d(x_i, x_j)) + \sum_{\substack{x_i \in D \setminus C \\ x_j \in C}} f(d(x_i, x_j)) + \sum_{x_i \in D \setminus C} f(d(x_i, x_p))$$

$$= v(D) - v(C) + \sum_{x_i \in D \setminus C} f(d(x_i, x_p))$$

$$\geq v(D) - v(C) \quad (\text{since } f : \Re^+ \cup \{0\} \rightarrow (0, 1])$$

□

Next we show that the points which are close to each other have nearly equal Shapley values.

Theorem 2. *Any two points x_i, x_t, such that $d(x_i, x_t) \leq \epsilon$, where $\epsilon \rightarrow 0$, in the convex game setting of §3.1 have almost equal Shapley values.*

Proof. As explained in §2.4, the Shapley value of a point x_i is given by,

$$\phi_i = \frac{1}{n!} \sum_{\pi \in \Pi} \left[v(T_{\pi, \pi(i)}) - v(T_{\pi, \pi(i)-1}) \right]$$

$$= \frac{1}{n!} \sum_{\pi \in \Pi} [\sum_{\substack{\pi(p) \leq \pi(i) \\ \pi(q) < \pi(p)}} f(d(x_p, x_q)) - \sum_{\substack{\pi(p) \leq \pi(i)-1 \\ \pi(q) < \pi(p)}} f(d(x_p, x_q))]$$

$$= \frac{1}{n!} \sum_{\pi \in \Pi} \sum_{\pi(p) < \pi(i)} f(d(x_i, x_p))$$

$$= \frac{1}{n!} \sum_{\pi \in \Pi} \sum_{\pi(p) < \pi(i)} [1 - f^{'}(d(x_i, x_p))]$$

$$= \frac{1}{n!} \sum_{\pi \in \Pi} [\pi(i) - 1] - \frac{1}{n!} \sum_{\pi \in \Pi} \sum_{\pi(p) < \pi(i)} f^{'}(d(x_i, x_p))$$

The first term on the right is a sum that is invariant for each point over all permutations. The second term can be expressed as, $D(i)$

$$= \frac{1}{n!} \sum_{\pi \in \Pi} \sum_{\substack{\pi(p) < \pi(i) \\ d(x_i, x_p) \leq \epsilon}} f^{'}(d(x_i, x_p)) + \frac{1}{n!} \sum_{\pi \in \Pi} \sum_{\substack{\pi(p) < \pi(i) \\ d(x_i, x_p) > \epsilon}} f^{'}(d(x_i, x_p))$$

It follows immediately using the definition of $f^{'}$ from §3.1, for x_t, $t \in \{1, 2, \ldots, n\}, t \neq i$ such that $d(x_i, x_t) \leq \epsilon \rightarrow 0$, we have $f^{'}(d(x_i, x_t)) \rightarrow 0$, and $f^{'}(d(x_i, x_p)) \rightarrow f^{'}(d(x_t, x_p))$, thereby implying $D(t) \rightarrow D(i)$. □

Note that Theorem 2 does not say anything about points that are far apart from each other. In particular, it does not forbid points, away from each other, from having similar Shapley value.

1. Note that the grand coalition has the maximum overall worth of all coalitions; however, Shapley value depends on the "average increase in worth" across all valid subsets rather than the overall worth. This is important in order to ensure that an appropriate number of clusters is obtained instead of just one.

3.3 The BiGC Algorithm

Algorithm 1 outlines our approach to clustering. BiGC takes as input a threshold parameter of similarity, δ, in addition to the dataset to be clustered. First, the Shapley value of each player is computed. Then, the algorithm iteratively chooses, from amongst the points not yet clustered, the point x_t with the current highest Shapley value as a new cluster center, and assigns all those points that are at least δ-similar to x_t, to the same cluster as x_t[2]. That is to say, we use δ as a threshold to ensure a) that nearby points, which tend to have almost equal Shapley value, are assigned to the same cluster, and b) the different cluster centers are reasonably far apart for input to the k-means algorithm. This ensures that the initial cluster centers thus chosen, while still accounting for the gestalt behavior, are well apart to provide an excellent seeding for centroid based algorithms like k-means[3], k-medoids, ISODATA, Genetic k-means, and fuzzy c-means [14].

For example, BiGC can be used in conjunction with the k-medoid algorithm by using the initial cluster centers, based on the Shapley value, as the initial medoids. Likewise, BiGC can be used with DBSCAN, a density based clustering algorithm [8], by choosing the point with the highest Shapley value as the initial "core" point, and so on. Each of the center based algorithms typically strives to optimize a particular objective function; to be consistent with the objectives of this work, we focus on the k-means algorithm that minimizes potential or the average point-to-center distance. Furthermore, we note that Algorithm 1 can be easily adapted to discard outliers by adding a step wherein all those clusters that are assigned fewer points than a minimum predefined number are discarded. The working of BiGC can be easily understood by applying Algorithm 1 to a simple one-dimensional data set of 10 points (see Fig. 1 and follow the step-by-step description given therein).

3.4 Exact Computation of Shapley Value in Quadratic Time

The exact computation of Shapley values for n players, in general, is computationally a hard problem since it involves taking the average over all the $n!$ permutation orderings. However, as mentioned earlier, the Shapley value for a convex game is the center of gravity of the extreme points of the non-empty core. For our choice of value function, $v(T)$, we can efficiently compute the Shapley value using the convexity result in Theorem 1, as shown next.

Theorem 3. *The Shapley value, for each player x_i in the convex game setting of §3.1, is given by*

$$\phi_i = \frac{1}{2} \sum_{\substack{x_j \in X \\ j \neq i}} f(d(x_i, x_j))$$

2. The points with almost equal Shapley value can be conceptualized as belonging to the same "class" but not necessarily the same cluster. Hence, taking a cue from Theorem 2, our heuristic ensures that only sufficiently close points from each class (as determined by δ) are clustered together.

3. If case of k-means, an example of a rather simplistic but bad heuristic would be to feed the top k points, based on Shapley value, to the algorithm since many points in close vicinity to each other would be taken as cluster centers.

Algorithm 1 BiGC

Input: data $X = \{x_1, x_2, ..., x_n\}$ and threshold similarity $\delta \in (0, 1]$
Output: Set of clusters

for $i = 1$ **to** n **do**
  $\phi_i = \frac{1}{2} \sum_{\substack{x_j \in X \\ j \neq i}} f(d(x_i, x_j))$;
end for
Initialize $Q = X$, $K = \emptyset$;
repeat
  $t = \underset{i:x_i \in Q}{\operatorname{argmax}}\ \phi_i$;
  $K = K \cup \{x_t\}$;
  $P_t = \{x_i \in Q : f(d(x_t, x_i)) \geq \delta\}$;
  $Q = Q \setminus P_t$;
until $Q = \emptyset$;
Run any centroid based algorithm (such as the k-means) with the cluster centers in K;

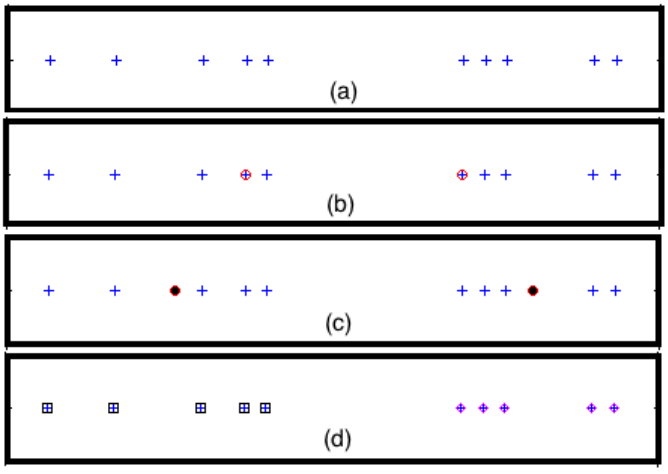

Fig. 1. Illustration of BiGC using a simple example: (a) sample one-dimensional data points (pluses), (b) initial cluster centers (circles) based on Shapley value and δ, (c) final cluster centers (filled circles) on running a center based algorithm (k-means here), and (d) two output clusters (rectangles and diamonds).

Proof. Since the underlying game is convex, therefore, the Shapley value of x_i can be computed using (1) as

$$\phi_i = \frac{1}{n!} \sum_{\pi \in \Pi} \left[v(T_{\pi,\pi(i)}) - v(T_{\pi,\pi(i)-1}) \right]$$

which can be re-written, using (2), as

$$\phi_i = \frac{1}{n!} \sum_{\pi \in \Pi} \sum_{\pi(j) < \pi(i)} f(d(x_i, x_j))$$

$$= \frac{1}{n!} \Big[\sum_{\substack{\pi(i)=1 \\ \pi(j)<\pi(i) \\ \pi \in \Pi}} f(d(x_i, x_j)) + \sum_{\substack{\pi(i)=2 \\ \pi(j)<\pi(i) \\ \pi \in \Pi}} f(d(x_i, x_j)) + \ldots + \sum_{\substack{\pi(i)=n \\ \pi(j)<\pi(i) \\ \pi \in \Pi}} f(d(x_i, x_j)) \Big]$$

Now, the total number of permutations for a fixed index i is $(n-1)!$; and, therefore summing over all such permutations, every player other

than x_i occurs exactly $\frac{(n-1)!}{(n-1)}$ times in each of the preceding positions, $\{1, 2, \ldots, (i-1)\}$. Then, the result follows from

$$\phi_i = \frac{(n-1)!}{(n-1)n!}[1 + 2 + \ldots + (n-1)] \sum_{\substack{x_j \in X \\ j \neq i}} f(d(x_i, x_j))]$$

□

We emphasize that this intuitively elegant closed form expression for computing the Shapley value is a direct consequence of the choice of our value function that BiGC exploits, as shown in Algorithm 1, to compute the Shapley value exactly in quadratic time.

3.5 Computing Shapley value Approximately in Linear Time

BiGC computes exactly the Shapley value in $O(n^2)$ time. The time complexity can be further reduced if we approximate the Shapley value by averaging marginal contributions over only p random permutations, where $p << n!$ (since the Shapley value of a convex game is represented by the center of gravity of the core). Then, the error resulting from this approximation can be bounded according to the concentration result proved in the following lemma.

Lemma 1. *Let $\Phi(p) = (\phi_1(p), \phi_2(p), \ldots, \phi_n(p))$ denote the empirical Shapley values, of n data points, computed using p permutations. Then, for some constants ϵ, c, and c_1, such that $\epsilon \geq 0$ and $c, c_1 > 0$,*

$$P(|\Phi(p) - \boldsymbol{E}(\Phi(p))| \geq \epsilon) \leq c_1 e^{-cp\epsilon^2}$$

Proof. Define $S = \sum_{i=1}^{p} Y_i$, where $Y_1, Y_2, \ldots, Y_p$ denote p independent random permutations of length n, corresponding to p n-dimensional points, randomly chosen from the boundary of the convex polyhedron. Clearly, S is a random variable. Now, applying Hoeffding-Chernoff's inequality, we can find constants c_1, c_2, and t, $0 \leq t \leq p\boldsymbol{E}(S)$, and $c_1, c_2 > 0$, such that

$$P(|S - \boldsymbol{E}(S)| \geq t) \leq c_1 e^{-\frac{c_2 t^2}{p\boldsymbol{E}(S)}}$$

$$\Rightarrow P(|S - \boldsymbol{E}(S)| \geq p\epsilon) \leq c_1 e^{-\frac{c_2 p\epsilon^2}{\boldsymbol{E}(S)}} \quad \text{(substituting } t = p\epsilon\text{)}$$

$$\Rightarrow P(\frac{1}{p}|S - \boldsymbol{E}(S)| \geq \epsilon) \leq c_1 e^{-\frac{c_2 p\epsilon^2}{\boldsymbol{E}(S)}}$$

$$\Rightarrow P(|\Phi(p) - \boldsymbol{E}(\Phi(p))| \geq \epsilon) \leq c_1 e^{-\frac{c_2 p\epsilon^2}{\mathbf{E}(S)}}$$

$$\Rightarrow P(|\Phi(p) - \boldsymbol{E}(\Phi(p))| \geq \epsilon) \leq c_1 e^{-cp\epsilon^2} \quad \text{(since } \Phi(p) = \frac{S}{p}\text{)}$$

□

4 Experimental Results

We carried out extensive experimentation to compare BiGC with state-of-the-art algorithms on several real benchmark datasets. We present a detailed analysis of our results in this section.

4.1 Evaluation Methodology

To be consistent with the objectives of this work, we measured the quality of clustering in terms of the following two parameters,

- $\alpha = \frac{1}{n} \sum_{x_i \in X} ||x_i - x^k||^2$, where x^k is the representative of cluster, $C_k \in C$, to which x_i is assigned.
- $\beta = \frac{1}{|C|} \sum_{\substack{x_i, x_j \in C_k \\ C_k \in C}} \frac{||x_i - x_j||^2}{|C_k|(|C_k| - 1)}$

where C is the set of clusters to which $x_i \in X = \{x_1, x_2, \ldots, x_n\}$ is assigned. The potential, α, quantifies the deviation of data points from the representative element while the scatter, β, captures the spread among different elements assigned to the same cluster. Clearly, the lower the values of α and β, the higher the quality of clustering. Furthermore, β also satisfies the desirable *loss conformity* requirement of a good clustering quality measure [1]. Moreover, we adopted the following standard cluster validity criteria:

(A) *Rand Statistic*: The Rand Index (RI) [41] estimates the quality of clustering with respect to the true (known) classes of the data. It measures the percentage of correct decisions made by the algorithm. Mathematically, $RI = \frac{TP + TN}{TP + FP + FN + TN}$, where TP, FP, FN, and TN are the number of true positives, false positives, false negatives, and true negatives respectively.

(B) *F-measure*: RI weights false positives and false negatives equally, which may be undesirable in certain applications. The F-measure [42][4] addresses this issue by weighting recall using a parameter $W \geq 0$. Mathematically, $F_W = \frac{(W^2 + 1)PR}{W^2 P + R}$, where the *precision* $P = \frac{TP}{TP + FP}$, and the *recall* $R = \frac{TP}{TP + FN}$.

(C) *Silhouette Width*: The Silhouette Index (SI) [43] is a validation technique, for model selection in cluster analyses, that considers both intra-cluster and inter-cluster distances to evaluate clustering. Mathematically, $SI = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i - a_i}{\max(b_i, a_i)}$, where a_i denotes the average distance between i and all other points in the same cluster; b_i denotes the average distance between i and the points in the nearest other cluster.

The higher the values of RI, F-measure and SI, the better the quality of clustering. Furthermore since *social fairness* is the primary characteristic of a game theoretic optimization methodology, we used the Jain's Fairness Index (JFI) [24]

4. It is common in literature to consider just the case where $W = 1$, when the F-measure is simply the harmonic mean of precision and recall.

to measure the fairness quantitatively. Mathematically, for num objectives,

$$JFI = \frac{\left(\sum_{i=1}^{num} z_i\right)^2}{num * \sum_{i=1}^{num} z_i^2},$$

where z_i denotes the improvement in the i^{th} objective. A high JFI signifies that the algorithm optimizes the objectives with almost equal priority [18].

We conducted an experimental study on a number of real-world datasets: *Wine*, *Iris*, *Spam*, *Cloud*, and *Intrusion*[5]. We implemented code in Matlab without any optimizations, and averaged the results over 30 runs to account for statistical significance. We chose Euclidean distance as our distance metric d, and set the dissimilarity between any two data points x_i and x_j, $f^{'}(d(x_i, x_j))$, to $\frac{d(x_i,x_j)}{d_{max}+1}$ where d_{max} denotes the maximum distance between any two data points. We emphasize BiGC is generic in that any monotonically non-decreasing dissimilarity function $f^{'} : \Re^+ \cup \{0\} \rightarrow [0, 1)$ such that $f^{'}(0) = 0$ can be used to define a corresponding similarity mapping, $f : \Re^+ \cup \{0\} \rightarrow (0, 1]$, where $f(a) = 1 - f^{'}(a)$.

4.2 Comparison with Center based Algorithms

The Leader algorithm [14] dynamically assigns each incoming point to the nearest cluster obeying the pre-specified distance threshold. BiGC can be viewed as an optimal unification of the Leader and the center based algorithms since the similarity threshold can be viewed as an extension of the idea of the distance threshold. Furthermore, since BiGC may be viewed as a technique to choose k suitable initial cluster centers, for a fair evaluation, we also compared BiGC with the the well-established heuristics, Hochbaum-Shmoys initialization (HSI) [21] and the k-means++ algorithm [2]. Further, since the Leader algorithm does not take δ as an input parameter, we executed the Leader algorithm for different distance thresholds, across different orders, and observed the number of clusters. Then, we modulated δ to obtain almost the same number of clusters. Likewise, we varied δ for adjusting the BiGC algorithm to the number of clusters used in the k-means. Finally, we averaged the α and β values for a fixed number of clusters.

Table 1 shows the α and β values resulting from the different algorithms on the Spam dataset. BiGC clearly outperforms all the other algorithms on both the parameters. While the improvement in performance with respect to α by itself is remarkable (Fig. 2), the gains in terms of β are nothing short of spectacular, as evident from Fig. 3. Similar results were obtained with the Intrusion dataset (Table 2). To corroborate the efficacy of our approach, we also conducted experiments wherein only one iteration of k-means in BiGC was carried out on the initial centers obtained using Shapley value and similarity threshold. We compared the α and β values thus obtained with those of the k-means algorithm allowed to proceed till convergence or a maximum of 100 iterations. As indicated by Fig. 8, BiGC still outperforms the k-means algorithm. Further, to be sure that the initial cluster centers derived from Shapley value play a crucial role, we compared the average performance achieved using a single iteration of k-means, on randomly chosen centers, with that allowed the complete execution (Fig. 4). We observe that there is a wide gap between the α and the β values resulting from the two. This contrasts with BiGC where the quality of clustering obtained does not vary significantly with the number of iterations. Moreover, the observation that choosing the initial cluster centers, taking into consideration their "global" suitability as captured using the Shapley value, provides excellent clustering with respect to α as well justifies our emphasis on gestalt clustering.

To firmly establish the efficacy of our approach, we also compared BiGC with the state-of-the-art k-means++ initialization, which is guaranteed to find a solution that is $O(\log k)$-competitive to the optimal k-means solution with respect to α [2]. We observe, from the results shown in [2], that there is hardly anything to choose between BiGC and k-means++ in terms of α; in particular, both produce excellent quality of clustering. However, BiGC is a conspicuously better algorithm in terms of β, e.g., Fig. 5(a) shows the results of comparison on the cloud dataset. Similarly, as shown in Fig. 5(b), BiGC registers an improvement to the tune of 60-70% over k-means++ on Spam and Intrusion as well. Moreover, as shown in Fig. 6, BiGC outperforms k-means and k-means++ not only in terms of Rand Index but also F-measure. This can be attributed to the fact that BiGC, in general, has a much higher recall than these algorithms. Thus, overall, BiGC is a decisively better algorithm than the k-means++.

4.3 Comparison with Multi-objective Optimization Algorithms

The center based algorithms are intrinsically designed toward optimizing a single objective. To put things in a better perspective, we also compared BiGC with the (complete) Agglomerative Linkage algorithm [14], which takes into account all pairs of distances while making a decision to merge two clusters. In other words, one of the objectives that complete linkage tries to optimize is similar to β. As a result, the complete linkage algorithm performs better than other variants like single linkage in terms of β, but worse in terms of α[6]. This observation is corroborated by Fig. 9 ; in particular, BiGC performs better than complete linkage in terms of α, whereas, both algorithms have very close values of β. This further endorses the motivation of this work: BiGC bridges the gap in the quality of clustering with respect to both α (like center based algorithms) and β (like complete and average agglomerative linkage algorithms).

Finally, we also compared BiGC with MOCK [20] and several other state-of-the-art Multi-objective Optimization based techniques like PDE, MODE, DEMO, NSDE, and NSGA-II [47]. Fig. 10 shows the results in terms of the Silhouette Index on the Iris and Wine datasets. We observe that BiGC registers a much better value than the other algorithms

5. These datasets are publicly available as online archives at the UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/

6. For brevity, we omit the results of our experiments using single and average linkage algorithms.

thereby implying correct clustering of a considerably greater number of points.

4.4 Sensitivity Analysis

4.4.1 Selection of δ

We also verified experimentally the impact of varying the similarity threshold δ on the number of clusters obtained using BiGC. Fig. 7 (a) shows the results on the Spam dataset. As expected, there is an increase in the number of clusters with an increase in δ. We observe that beyond a certain value of δ (close to 0.98 in case of Spam), there is a progressively sharp increase in cluster count. Therefore, we can use this "knee" of the δ-C curve as a heuristic to find a reasonable number of clusters. In general, the exact δ-C curve obtained, might depend on the particular dataset, and alternative ways of determining the appropriate δ may have to be devised, similar in spirit to the work done on k-means [19].

4.4.2 Execution Time

Finally, we analyze the time taken by the BiGC algorithm. In our experiments, we also observed that although BiGC is of complexity $O(n^2)$, in practice, the actual time it takes is quite low since it tends to converge rapidly, due to an appropriate selection of initial cluster centers. On Spam, for instance, we observed that BiGC with one iteration of k-means generally took less time than k-means (except for the case where number of clusters was low), and this effect was more pronounced as the number of clusters was increased. This is understandable since computing the pairwise similarities is the predominantly time consuming step in BiGC, while the number of iterations has a marked influence on the computational cost of k-means. Similar behavior was observed with other datasets (Intrusion, Cloud, Wine), however for brevity, we omit a detailed analysis. Thus, for practical purposes, a single iteration of k-means on the initial centers obtained from BiGC yields excellent clustering in a reasonable time. Further, we found that if the similarity threshold (δ) is kept fixed, α does not increase significantly with a decline in the number of permutations, p (Wine dataset, Fig. 7 (b)), thereby supporting the result in Lemma 1. Similar observations were made with respect to β. This indicates we can further bring down the execution time of BiGC at a slight expense of the quality of clustering.

4.4.3 Fairness

We observe, from Fig. 5(c) and Fig. 9(c), that BiGC has a consistently high Jain's Fairness Index, close to 1, that exceeds those of agglomerative linkage and center based algorithms. This reaffirms the fact that BiGC is not "biased" towards either objective, and that it optimizes both α and β with almost equal priority. In summary, BiGC ensures social fairness across a wide range of clusters, and this justifies the usage of our game theoretic approach.

TABLE 1
Spam Dataset

algorithm	Clusters	Average α	Average β
Leader	10	153974	250783
k-means	10	36850	5619527
HSI	10	20380	230563
BiGC	10	17820	215172
Leader	25	110673	170557
k-means	25	33281	2248452
HSI	25	8127	55315
BiGC	25	4822	44826
Leader	35	86139	33248
k-means	35	32711	1607324
HSI	35	7240	41176
BiGC	35	3380	20567
Leader	50	73901	23151
k-means	50	32626	1125174
HSI	50	4662	22706
BiGC	50	1850	11154
Leader	100	65443	12598
k-means	100	14085	251347
HSI	100	2643	8751
BiGC	100	677	2779

TABLE 2
Network Intrusion Dataset

algorithm	Clusters	Average α	Average β
Leader	10	4.1139e+09	4.4772e+07
k-means	10	1.4836e+06	1.3116e+08
HSI	10	6.8091e+05	1.1371e+08
BiGC	10	7.2876e+05	3.5754e+07
Leader	25	4.1014e+09	5.8943e+06
k-means	25	8.9539e+05	3.1790e+07
HSI	25	1.9103e+05	2.6012e+07
BiGC	25	5.2724e+04	9.7346e+05
Leader	35	4.0931e+09	4.2186e+05
k-means	35	2.8318e+05	1.9058e+07
HSI	35	1.8785e+04	1.8581e+07
BiGC	35	2.0239e+04	2.7405e+05
Leader	50	4.0819e+09	3.1730e+05
k-means	50	2.2037e+05	1.3114e+07
HSI	50	1.5037e+04	1.1773e+07
BiGC	50	8.8181e+03	7.8463e+04
Leader	100	1.7105e+07	3.1332e+05
k-means	100	1.8994e+05	6.5027e+06
HSI	100	1.1770e+04	5.0348e+06
BiGC	100	1.7079e+03	1.7347e+04

5 Additional Key Properties Satisfied by BiGC

In his work on unification of clustering [28], Kleinberg considered three properties: scale-invariance, richness, and consistency and proved an impossibility result, showing that no clustering algorithm satisfies all of these properties simultaneously. Informally, scale invariance imposes the requirement that there be no built in length scale; richness that all partitions be achievable; and consistency that there be no change in partitions on reducing intra-cluster and enlarging inter-cluster distances. In his influential work [28], Kleinberg proved an impossibility theorem that no clustering algorithm satisfies scale invariance, richness, and consistency simultaneously. Subsequently, a working set of axioms has been proposed to advance the work on unification of clustering [55]. BiGC is optimal in that it satisfies both scale invariance and richness, as explained below.

- **Scale Invariance:** The Leader algorithm does not satisfy scale invariance since it decides the clusters based on a distance threshold, and thus incurs a fundamental distance scale. The k-means algorithm satisfies scale invariance since it assigns clusters to points depending only on their relative distances

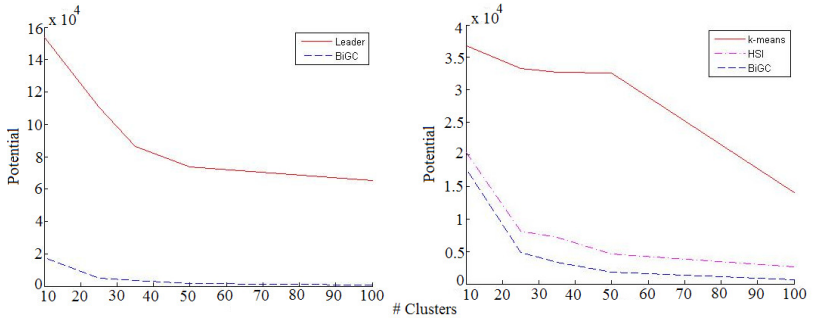

Fig. 2. $\alpha - C$ plot (*Spam Dataset*): (a) BiGC vs. Leader, (b) BiGC vs. k-means vs. HSI

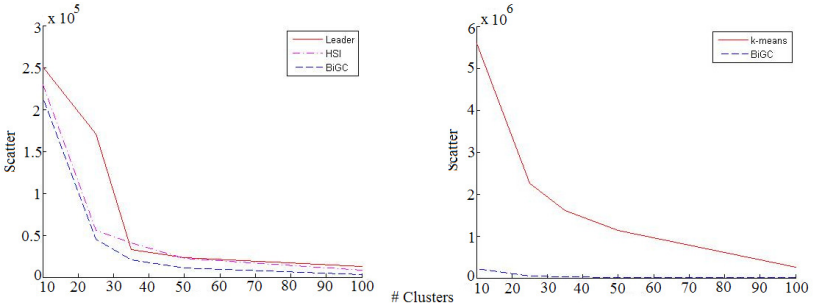

Fig. 3. $\beta - C$ plot (*Spam Dataset*): (a) BiGC vs. Leader vs. HSI, (b) BiGC vs. k-means

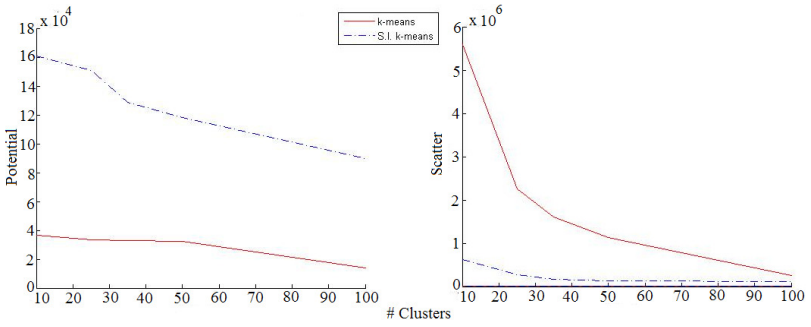

Fig. 4. *(Spam) k-means vs. single iteration of k-means*: (a) Potential (α), (b) Scatter (β)

to the k cluster centers, irrespective of the absolute distances. BiGC consists of two phases. In the first phase, clustering is done based on the similarity values, which are again relative (e.g. consider a similarity function $f(d(x_i, x_j)) = 1 - \frac{d(x_i, x_j)}{d_\tau}$ where $d_\tau > d_{max}$, with d_{max} denoting the maximum distance between any two points in the dataset). In the second phase, the k-means algorithm is used, which is scale invariant, as already mentioned. Thus, BiGC satisfies scale invariance.

- **Richness:** The Leader algorithm satisfies the richness property since we can always adjust the distances among points to generate any desired partition of the input dataset. For example, one of the ways to obtain a single cluster is to set all pairwise distances to some value less than the distance threshold, whereas to have each point assigned to a separate cluster, every pairwise distance may be set to some value greater than the distance threshold. The k-means algorithm satisfies the richness condition only if the value of k can be adjusted according to the desired partitions. However, since in general, k is a constant input provided to the k-means algorithm, we may not partition the input dataset into any number of clusters other than k, and this precludes the k-means algorithm from satisfying richness. Note that this restriction of k-means does not apply to the BiGC algorithm, since the number of clusters, k, is not provided as an input and is determined based on the Shapley values of points and similarities among them. Hence, BiGC satisfies the richness property.
- **Consistency:** The Leader and the k-means algorithms do not satisfy the consistency requirement. This follows directly from Kleinberg's result in [28], which states that there does not exist any centroid based clustering function that satisfies the consistency property. The k-means algorithm does not satisfy the consistency property since it uses the (k, g)-centroid clustering function: the underlying objective

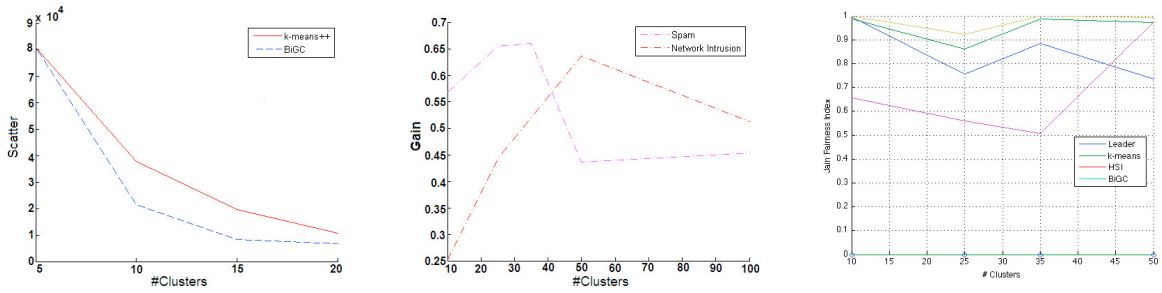

Fig. 5. (a) $\beta - C$ plot (Cloud Dataset): BiGC vs. k-means++, (b) Performance Gain $(= 1 - \frac{\text{BiGC } \beta}{\text{k-means++ } \beta})$ on Spam and Intrusion, and (c) Jain's Fairness Index comparison on Spam

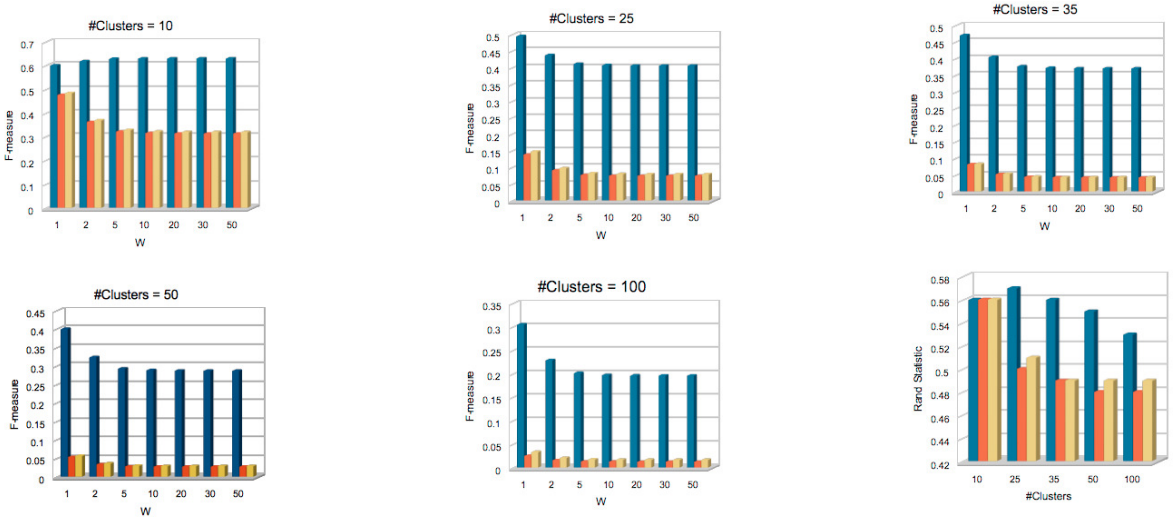

Fig. 6. Spam: BiGC (blue) vs. k-means (red) vs. k-means++ (yellow): (a)-(e) F-measure, (f) Rand statistic

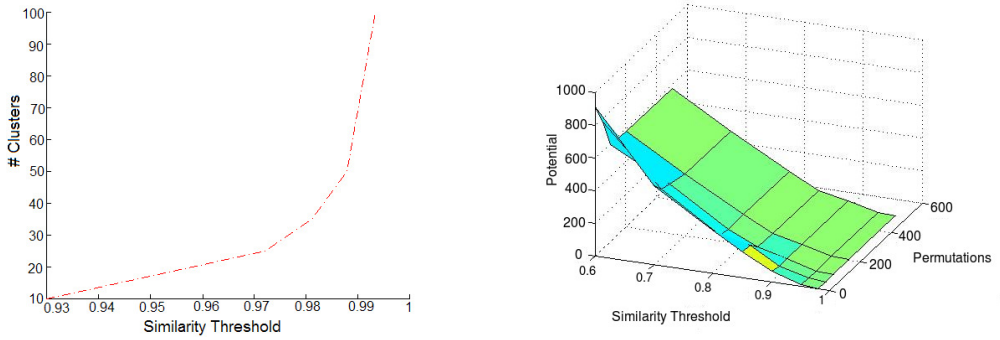

Fig. 7. (a) $\delta - C$ plot (Spam) (b) α does not vary much with permutations for a fixed δ (Wine)

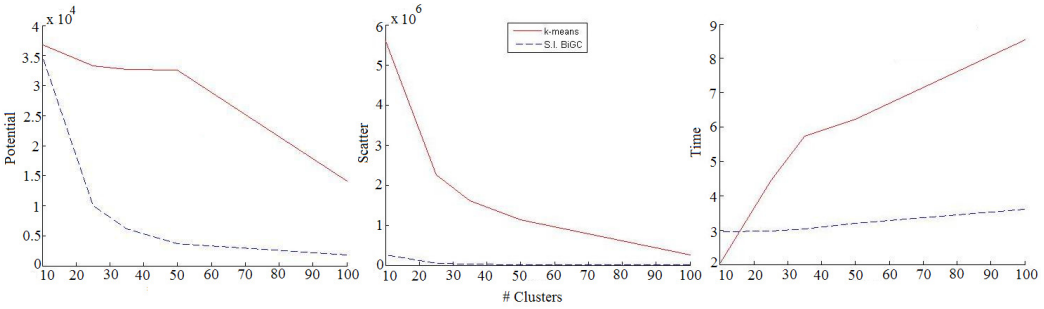

Fig. 8. *(Spam) k-means vs. BiGC with single iteration of k-means*: (a) α, (b) β, (c) Time (in sec)

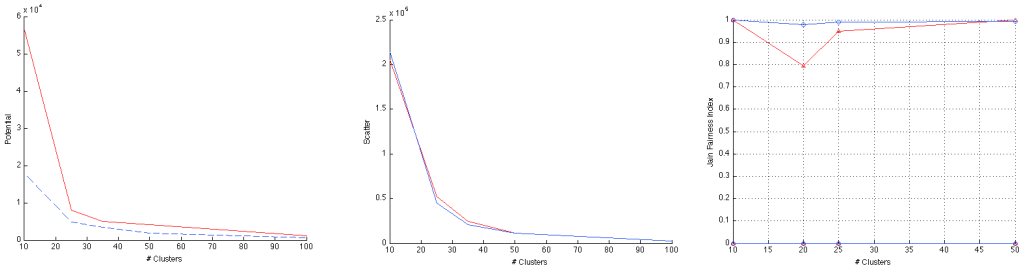

Fig. 9. (Spam) BiGC (blue) vs. Agglomerative Linkage (red): (a) $\alpha - C$, (b) $\beta - C$, (c) Jain's Fairness Index

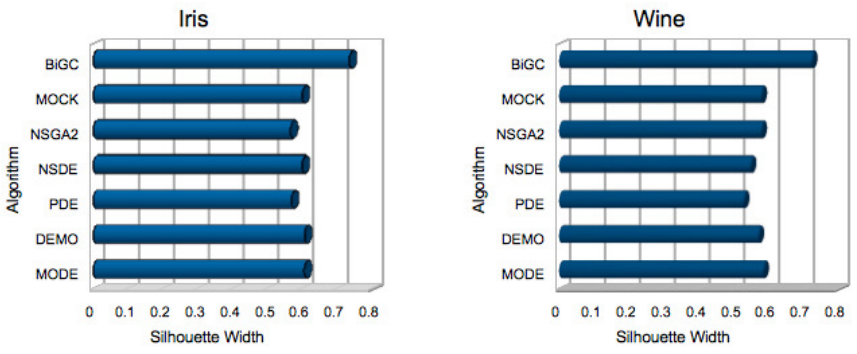

Fig. 10. BiGC vs. Multi-objective Optimization techniques: Silhouette Index comparison on Iris and Wine

function in k-means clustering can be expressed as $g(d) = d^2$ ([28], [55]). BiGC also does not satisfy consistency as a consequence of Kleinberg's impossibility theorem.

Order Independence: Order independence is another desirable fundamental property of clustering algorithms. A learner is *order sensitive* or *order dependent* if given a set of data objects, such an algorithm might return different clusterings based on the order of presentation of these objects. The objective of order independence in algorithms is to produce the same final clustering across different runs, irrespective of the sequence in which the input instances are presented[7]. Formally, this property can be defined as,

Order Independence. For any two permutation orderings $\pi, \pi' \in \Pi$, distance function d, and number of clusters k, we have for a partitioning function F[8],

$$F(d, k, \pi) = F(d, k, \pi')$$

Similar concepts have previously been proposed in a number of papers, under different names, for instance, Ackerman and Ben-David discuss such an axiom in the setting of clustering-quality measures under the name *isomorphism invariance* [1]. Puzicha et al. introduced *permutation invariance* in the setting of clustering objective functions [40]. It is important to note that *order independence* is significantly different from the *order consistency* axiom proposed by Bosagh Zadeh and Ben-David in [55]. Order

7. Note that order independence is not redundant since one can construct clustering functions that satisfy order independence but violate at least one other property. For example, a clustering function that computes the mean or average of a set of points does not satisfy scale invariance; the k-means objective function does not satisfy richness; and the clustering function used in BiGC does not satisfy consistency.

8. Note that F takes three arguments here instead of two propounded by Bosagh Zadeh and Ben-David in their highly influential work on unification of clustering, [55]; we have chosen to add an additional parameter, instead of introducing a radically different formalism, to be consistent with the work in [55]. It is readily seen that the formalism in [55] is directly amenable to the third parameter: $\pi = \pi'$ renders permutation immaterial.

consistency, which is demonstrated by clustering techniques such as the complete linkage algorithm, implies that for a given number of clusters, if the two different distance functions select the edges in the same order then the output clustering should be identical in both the cases. Order independence assumes great significance especially with a tremendous spurt in stream applications recently, see for instance, works by Cormode [11], Cao et. al [8], Guha et. al [16], and Cormode and Garofalakis [12].

It can be shown [49] that any order independent incremental algorithm must maintain a knowledge structure A of abstractions together with an operator $*$ defined on it, such that $(A, *)$ is a commutative monoid. Further, let g be a function defined as $g : A \times X \rightarrow A$, where $X = \{x_1, x_2, ..., x_n\}$ represents the set of input data instances and A represents the set of all valid memory structures. Then, as elaborated in [49], presence of a *dynamically complete set* X on (A, g) provides both a *necessary* and *sufficient* condition for order independence in any algorithm that takes X as an input and uses A and g. We refer the reader to [49] for a detailed exposition on characterizing order independence. It can be readily shown that BiGC is order independent: for any permutation ordering on the input instances, $\pi \in \Pi$, we may define an abstraction on i points, $T_{\pi,i} = \sum_{\substack{\pi(p)\leq\pi(i) \\ \pi(q)<\pi(p)}} f(d(x_p, x_q))$, and a function g such that $g(T_{\pi,i}, x_{i+1}) = T_{\pi,i} + \sum_{\pi(p)\leq\pi(i)} f(d(x_{i+1}, x_p))$, where x_{i+1} is the incoming input instance.

The Leader algorithm is known to be susceptible to ordering effects. On the other hand, the random selection of initial cluster centers precludes the k-means algorithm from being truly order-independent. We summarize the foregoing discussion in Table 3. It is easy to infer that BiGC stands out, since it satisfies three of the four properties. We emphasize that no other clustering algorithm can perform better since all the four properties can not be simultaneously satisfied, as a consequence of the impossibility theorem.

TABLE 3
Comparison between Leader, k-means, and BiGC

Property	Leader	k-means	BiGC
Scale Invariance	X	√	√
Richness	√	X	√
Consistency	X	X	X
Order Independence	X	X	√

Moreover, as Ackerman and Ben-David pointed out in [1], "an impossibility result is not an inherent feature of clustering, but rather, to a large extent, it is an artifact of the specific formalism used", thus, more properties need be considered to enhance the understanding of the general theory of clustering, and order independence is one such property.

6 Conclusion and Future Work

We proposed a novel approach BiGC, based on a cooperative game theoretic framework, with a view of obtaining a good clustering with respect to both the average point-to-center and the average intra-cluster point-to-point distance. BiGC elegantly captures the notion of gestalt or cohesive clustering and satisfies desirable clustering properties like scale invariance, order independence, and richness. Experimental comparisons with respect to different cluster validity criteria on several real benchmark datasets, with state-of-the-art center based and multi-objective algorithms, further substantiate the efficacy of BiGC.

In this work, we investigated the efficacy of BiGC using a particular similarity function. It would be interesting to analyze the impact of different dissimilarity measures on the quality of clustering. We also intend to examine more heuristics for determining a suitable similarity threshold δ for a given dataset, similar in spirit to the work done on k-means [19]. The extension of ideas presented in this work to the supervised setting is another interesting direction.

References

[1] M. Ackerman and S. Ben-David. Measures of Clustering Quality: A Working Set of Axioms for Clustering. *NIPS*, 2008.

[2] D. Arthur and S. Vassilvitskii. k-means++: The Advantages of Careful Seeding. *SODA*, pp. 1027–1035, 2007.

[3] E. Backer and A. Jain. A clustering performance measure based on fuzzy set decomposition. *PAMI*, 3(1), pp. 66–75, 1981.

[4] S. R. Bulo' and M. Pelillo. A Game-Theoretic Approach to Hypergraph Clustering. *NIPS*, pp. 1571–1579, 2009.

[5] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman Divergences. *JMLR*, 6, pp. 1705–1749, 2005.

[6] N. Bansal, A. Blum, and S. Chawla. Correlation Clustering. *Machine Learning Journal* (Special Issue on Theoretical Advances in Data Clustering), 56(1), pp. 86–113, 2004.

[7] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik. Support vector clustering. *JMLR*, 2, pp. 125–137, 2001.

[8] F. Cao, M. Ester, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. *SDM*, pp. 328–339, 2006.

[9] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. *KDD*, pp. 554–560, 2006.

[10] Y. Chen and L. Tu. Density-based clustering for real-time stream data. *KDD*, pp. 133–142, 2007.

[11] G. Cormode. Conquering the divide: Continuous clustering of distributed data streams. *ICDE*, pp. 1036–1045, 2007.

[12] G. Cormode and M. Garofalakis. Sketching Probabilistic Data Streams. *SIGMOD'07*, pp. 281–292, 2007.

[13] A. Cornuejols. Getting Order Independence in Incremental Learning. *ECML*, pp. 196–212, Springer-Verlag, 1993.

[14] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*, John Wiley and Sons, Second edition, 2000.

[15] D. Fisher, L. Xu, and N. Zard. Ordering effects in clustering. *ICML*, pp. 163–168, 1992.

[16] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering Data Streams: Theory and Practice. *TKDE*, 15(3), pp. 515–528, 2003.

[17] F. Gullo, A. Tagarelli, and S. Greco. Diversity-based Weighting Schemes for Clustering Ensembles. *SDM*, pp. 437–448, 2009.

[18] U. Gupta and N. Ranganathan. A Game Theoretic Approach for Simultaneous Compaction and Equipartitioning of Spatial Data Sets. *TKDE*, 22(4), pp. 465–478, 2010.

[19] G. Hamerly and C. Elkan. Learning the k in k-means. *NIPS*, pp. 281–288, 2003.

[20] J. Handl and J. Knowles. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1), pp. 56–76, 2007.

[21] D. S. Hochbaum and D. B. Shmoys. A Best Possible Heuristic for the k-center Problem. *Mathematics of Operations Research*, 10(2), pp. 180–184, 1985.

[22] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *PAMI*, 22(1), pp. 4–37, 2000.

[23] A. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3), pp. 264–323, 1999.

[24] R. Jain, D. Chiu, and W. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System. *DEC-TR-301*, Eastern Research Lab, Digital Equipment Corporation, 1984.

[25] D. Jiang, J. Pei, M. Ramanathan, C. Tang, and A. Zhang. Mining coherent gene clusters from gene-sample-time microarray data. *KDD*, pp. 430–439, 2004.

[26] M. E. Kabir, H. Wang, and E. Bertino. Efficient systematic clustering method for k-anonymization. *Acta Informatica*. 48(1), 2011.

[27] L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, 1990.

[28] J. Kleinberg. An Impossibility Theorem for Clustering. *NIPS*, 15, pp. 463–470, 2002.

[29] T. Kohonen. The self-organizing map. *Proc. of the IEEE*, 78(9), pp. 1464–1480, 1990.

[30] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering High Dimensional Data: A Survey on Subspace Clustering, Pattern-based Clustering, and Correlation Clustering. *TKDD*, 3 (1), pp. 1–58, 2009.

[31] Q. Li, Z. Chen, Y. He, and J.-p. Jiang. A novel clustering algorithm based upon games on evolving network. *Expert Systems with Applications*, 37(8), pp. 5621–5629, 2010.

[32] Q. Li, Y. He, and J.-p. Jiang. A novel clustering algorithm based on quantum games. *J. Phys. A: Math. Theor.*, 42, pp. 445303:1–16, 2009.

[33] W. Li, L. Jaroszewski, and A. Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, 17, pp. 282–283, 2001.

[34] Y.-F. Li, I. W. Tsang, J. T.-Y. Kwok, and Z.-H. Zhou. Tighter and Convex Maximum Margin Clustering. *JMLR - Proceedings Track*, 5, pp. 344–351, 2009.

[35] M. Meila and J. Shi. A Random Walks View of Spectral Segmentation. *AISTATS*, 2001.

[36] R. S. Michalski. Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and an algorithm for Partitioning Data into Conjunctive Concepts. *Journal of Policy Analysis and Information Systems*, 4(3), pp. 219–244, 1980.

[37] G. Moise, J. Sander, and M. Ester. Robust projected clustering. *Knowl. Inf. Syst.*, 14(3), pp. 273–298, 2008.

[38] R. B. Myerson. Game Theory: Analysis of Conflict. *Harvard University Press*, 1997.

[39] R. Ostrovsky, Y. Rabani, L. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the *k*-Means problem. *FOCS*, pp. 165–176, 2006.

[40] J. Puzicha, T. Hofmann, and J. Buhmann. Theory of Proximity Based Clustering: Structure Detection by Optimization. *Pattern Recognition*, 33(4), pp. 617–634, 2000.

[41] W. Rand. Objective criteria for the evaluation of clustering methods. *J. Amer. Statist. Assoc.*, 66(336), pp. 846–850, 1971.

[42] C. van Rijsbergen. *Information retrieval*, Butterworths, Second edition, 1979.

[43] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1), pp. 53–65, 1987.

[44] L. S. Shapley. Cores of convex games. *International Journal of Game Theory*, 1(1), pp. 11–26, 1971.

[45] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *PAMI*, 22(8), 2000.

[46] M. Steinbach, P.-N. Tan, V. Kumar, S. A. Klooster, and C. Potter. Discovery of climate indices using clustering. *KDD*, pp. 446-455, 2003.

[47] K. Suresh, D. Kundu, S. Ghosh, S. Das and A. Abraham. Data Clustering Using Multi-objective DE Algorithms. *Fundamenta Informaticae*, 21, pp. 1001–1024, 2009.

[48] J. Vaidya and C. Clifton. Privacy-Preserving k-Means Clustering over Vertically Partitioned Data. *SIGKDD '03*, 2003.

[49] V. K. Garg. Pragmatic Data Mining: Novel Paradigms for Tackling Key Challenges. Technical Report, Available from: *http://www.csa.iisc.ernet.in/TR/2009/11/thessamp.pdf*.

[50] H. Wang and J. Pei. Clustering by Pattern Similarity. *J. Comput. Sci. Technol.*, 23(4), pp. 481–496, 2008.

[51] J. Wang, Y. Zhang, L. Zhou, G. Karypis, and C. Aggarwal. Discriminating Subsequence Discovery for Sequence Clustering. *SDM*, 2007.

[52] R. Xu and D. Wunsch II. Survey of Clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), pp. 645–678, 2005.

[53] D. Yankov, E. Keogh, and K. Kan (2007). Locally Constrained Support Vector Clustering. *ICDM*, pp. 715–720, 2007.

[54] M. L. Yiu and N. Mamoulis. Iterative Projected Clustering by Subspace Mining. *TKDE*, 17(2), pp. 176–189, 2005.

[55] R. B. Zadeh and S. Ben-David. A Uniqueness Theorem for Clustering. *UAI*, 2009.

[56] P. Zhang, X. Zhu, J. Tan, and L. Guo. Classifier and Cluster Ensembling for Mining Concept Drifting Data Streams. *ICDM*, 2010.

[57] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao. Gaussian mixture density modeling, decomposition, and applications. *IEEE Trans. Image Process.*, 5(9), pp. 1293–1302, 1996.