

# K-sets<sup>+</sup>: a Linear-time Clustering Algorithm for Data Points with a Sparse Similarity Measure

Cheng-Shang Chang, Chia-Tai Chang, Duan-Shin Lee and Li-Heng Liou  
Institute of Communications Engineering  
National Tsing Hua University  
Hsinchu 30013, Taiwan, R.O.C.

Email: cschang@ee.nthu.edu.tw; s104064540@m104.nthu.edu.tw; lds@cs.nthu.edu.tw; dacapo1142@gmail.com

**Abstract**—In this paper, we first propose a new iterative algorithm, called the K-sets<sup>+</sup> algorithm for clustering data points in a semi-metric space, where the distance measure does not necessarily satisfy the triangular inequality. We show that the K-sets<sup>+</sup> algorithm converges in a finite number of iterations and it retains the same performance guarantee as the K-sets algorithm for clustering data points in a metric space. We then extend the applicability of the K-sets<sup>+</sup> algorithm from data points in a semi-metric space to data points that only have a symmetric similarity measure. Such an extension leads to great reduction of computational complexity. In particular, for an  $n \times n$  similarity matrix with  $m$  nonzero elements in the matrix, the computational complexity of the K-sets<sup>+</sup> algorithm is  $O((Kn + m)I)$ , where  $I$  is the number of iterations. The memory complexity to achieve that computational complexity is  $O(Kn + m)$ . As such, both the computational complexity and the memory complexity are linear in  $n$  when the  $n \times n$  similarity matrix is *sparse*, i.e.,  $m = O(n)$ . We also conduct various experiments to show the effectiveness of the K-sets<sup>+</sup> algorithm by using a synthetic dataset from the stochastic block model and a real network from the WonderNetwork website.

**keywords:** Clustering; community detection

## I. INTRODUCTION

The problem of clustering is of fundamental importance to data analysis and it has been studied extensively in the literature (see e.g., the books [1], [2] and the historical review papers [3], [4]). In such a problem, there is a set of data points and a similarity (or dissimilarity) measure that measures how similar two data points are. The objective of a clustering algorithm is to cluster the data points so that data points within the same cluster are similar to each other and data points in different clusters are dissimilar. Clustering is in general considered as an ill-posed problem and there are already many clustering algorithms proposed in the literature, including the hierarchical algorithm [1], [2], the K-means algorithm [4], [5], [6], the K-medoids algorithm [1], [7], [8], [9], the kernel and spectral clustering algorithms [10], [11], [12], [13], and the definition-based algorithms [14], [15], [16], [17]. However, clustering theories that justify the use of these clustering algorithms are still unsatisfactory.

Recently, a mathematical clustering theory was developed in [18] for clustering data points in a metric space. In that theory, clusters can be formally defined and stated in various equivalent forms. In addition to the definition of a cluster in

a metric space, the K-sets algorithm was proposed in [18] to cluster data points in a metric space. The key innovation of the K-sets algorithm in [18] is the triangular distance that measures the distance from a data point to a set (of data points) by using the triangular inequality. Like the K-means algorithm, the K-sets algorithm is an iterative algorithm that repeatedly assigns every data point to the closest set in terms of the triangular distance. It was shown in [18] that the K-sets algorithm converges in a finite number of iterations and outputs  $K$  disjoint sets such that any two sets of these  $K$  sets are two disjoint clusters when they are viewed in isolation.

The first contribution of this paper is to extend the clustering theory/algorithm in [18] to data points in a semi-metric space, where the distance measure does not necessarily satisfy the triangular inequality. Without the triangular inequality, the triangular distance in the K-sets algorithm is no longer non-negative and thus the K-sets algorithm may not converge at all. Even if it converges, there is no guarantee that the output of the K-sets algorithm are clusters. To tackle this technical challenge, we propose the K-sets<sup>+</sup> algorithm for clustering in a semi-metric space. In the K-sets<sup>+</sup> algorithm, we need to modify the original definition of the triangular distance so that the nonnegativity requirement of the triangular distance can be lifted. For this, we propose the adjusted triangular distance (in Definition 6) and show (in Theorem 7) that the K-sets<sup>+</sup> algorithm that repeatedly assigns every data point to the closest set in terms of the adjusted triangular distance converges in a finite number of iterations. Moreover, the K-sets<sup>+</sup> algorithm outputs  $K$  disjoint sets such that any two sets of these  $K$  sets are two disjoint clusters when they are viewed in isolation.

The second contribution of this paper is to further extend the applicability of the K-sets<sup>+</sup> algorithm from data points in a semi-metric space to data points that only have a symmetric similarity measure. A similarity measure is generally referred to as a bivariate function that measures how similar two data points are. We show there is a natural mapping from a symmetric similarity measure to a distance measure in a semi-metric space and the the K-sets<sup>+</sup> algorithm that uses this distance measure converges to the same partition as that using the original symmetric similarity measure. Such an extension leads to great reduction of computational complexity for the K-sets<sup>+</sup> algorithm. For an  $n \times n$  similarity matrix

with only  $m$  nonzero elements in the matrix, we show that the computational complexity of the K-sets<sup>+</sup> algorithm is  $O((Kn + m)I)$ , where  $I$  is the number of iterations. The memory complexity to achieve that computational complexity is  $O(Kn + m)$ . If the  $n \times n$  similarity matrix is *sparse*, i.e.,  $m = O(n)$ , then both the computational complexity and the memory complexity are linear in  $n$ .

To evaluate the performance of the K-sets<sup>+</sup> algorithm, we conduct two experiments: (i) community detection of signed networks generated by the stochastic block model, and (ii) clustering of a real network from the WonderNetwork website [19]. Our experiments show that the K-sets<sup>+</sup> algorithm is very effective in recovering the ground-truth edge signs even when the signs of a certain percentage of edges are flipped. For the real network of servers, the K-sets<sup>+</sup> algorithm yields various interesting observations from the clustering results obtained by using the geographic distance matrix and the latency matrix.

## II. CLUSTERING IN A SEMI-METRIC SPACE

In this paper, we consider the clustering problem for data points in a semi-metric space. Specifically, we consider a set of  $n$  data points,  $\Omega = \{x_1, x_2, \dots, x_n\}$  and a distance measure  $d(x, y)$  for any two points  $x$  and  $y$  in  $\Omega$ . The distance measure  $d(\cdot, \cdot)$  is assumed to be a *semi-metric* and it satisfies the following three properties:

- (D1) (Nonnegativity)  $d(x, y) \geq 0$ .
- (D2) (Null condition)  $d(x, x) = 0$ .
- (D3) (Symmetry)  $d(x, y) = d(y, x)$ .

The semi-metric assumption is weaker than the metric assumption in [18], where the distance measure is assumed to satisfy the triangular inequality. In [18], the K-sets algorithm was proposed for clustering data points in a metric space. One of the main contributions of this paper is to propose the K-sets<sup>+</sup> algorithm (as a generalization of the K-sets algorithm) for clustering data points in a *semi-metric* space. As both Euclidean spaces and metric spaces are special cases of semi-metric spaces, such a generalization allows us to unify the well-known K-means algorithm and the K-sets algorithm in [18].

### A. Semi-cohesion measure

Given a semi-metric  $d(\cdot, \cdot)$  for  $\Omega$ , we define the induced semi-cohesion measure as follows:

$$g(x, y) = \frac{1}{n} \sum_{z_2 \in \Omega} d(z_2, y) + \frac{1}{n} \sum_{z_1 \in \Omega} d(x, z_1) - \frac{1}{n^2} \sum_{z_2 \in \Omega} \sum_{z_1 \in \Omega} d(z_2, z_1) - d(x, y). \quad (1)$$

It is easy to verify that the induced semi-cohesion measure satisfies the following three properties:

- (C1) (Symmetry)  $g(x, y) = g(y, x)$  for all  $x, y \in \Omega$ .
- (C2) (Null condition) For all  $x \in \Omega$ ,  $\sum_{y \in \Omega} g(x, y) = 0$ .
- (C3) (Nonnegativity) For all  $x, y$  in  $\Omega$ ,

$$g(x, x) + g(y, y) \geq 2g(x, y). \quad (2)$$

Moreover, we have

$$d(x, y) = (g(x, x) + g(y, y))/2 - g(x, y). \quad (3)$$

Analogous to the argument in [18], one can easily show the following duality theorem.

*Theorem 1:* Consider a set of data points  $\Omega$ . For a semi-metric  $d(\cdot, \cdot)$  that satisfies (D1)–(D3), let

$$d^*(x, y) = \frac{1}{n} \sum_{z_2 \in \Omega} d(z_2, y) + \frac{1}{n} \sum_{z_1 \in \Omega} d(x, z_1) - \frac{1}{n^2} \sum_{z_2 \in \Omega} \sum_{z_1 \in \Omega} d(z_2, z_1) - d(x, y) \quad (4)$$

be the induced semi-cohesion measure of  $d(\cdot, \cdot)$ . On the other hand, for a semi-cohesion measure  $g(\cdot, \cdot)$  that satisfies (C1)–(C3), let

$$g^*(x, y) = (g(x, x) + g(y, y))/2 - g(x, y). \quad (5)$$

Then  $g^*(x, y)$  is a semi-metric that satisfies (D1)–(D3). Moreover,  $d^{**}(x, y) = d(x, y)$  and  $g^{**}(x, y) = g(x, y)$  for all  $x, y \in \Omega$ .

In view of the duality result, there is a one-to-one mapping between a semi-metric and a semi-cohesion measure. Thus, we will simply say data points are in a semi-metric space if there is either a semi-cohesion measure or a semi-metric associated with these data points.

### B. Clusters in a semi-metric space

In this section, we define what a cluster is for a set of data points in a semi-metric space.

*Definition 2: (Cluster)* Consider a set of  $n$  data points,  $\Omega = \{x_1, x_2, \dots, x_n\}$ , with a semi-cohesion measure  $g(\cdot, \cdot)$ . For two sets  $S_1$  and  $S_2$ , define

$$g(S_1, S_2) = \sum_{x \in S_1} \sum_{y \in S_2} g(x, y). \quad (6)$$

Two sets  $S_1$  and  $S_2$  are said to be *cohesive* (resp. *incohesive*) if  $g(S_1, S_2) \geq 0$  (resp.  $g(S_1, S_2) \leq 0$ ). A nonempty set  $S$  of  $\Omega$  is called a *cluster* (with respect to the semi-cohesion measure  $g(\cdot, \cdot)$ ) if

$$g(S, S) \geq 0. \quad (7)$$

Following the same argument in [18], one can also show a theorem for various equivalent statements for what a cluster is in a semi-metric space.

*Theorem 3:* Consider a set of  $n$  data points,  $\Omega = \{x_1, x_2, \dots, x_n\}$ , with a semi-cohesion measure  $g(\cdot, \cdot)$ . Let  $d(x, y) = (g(x, x) + g(y, y))/2 - g(x, y)$  be the induced semi-metric and

$$\bar{d}(S_1, S_2) = \frac{1}{|S_1| \times |S_2|} \sum_{x \in S_1} \sum_{y \in S_2} d(x, y). \quad (8)$$

be the average “distance” between two randomly selected points with one point in  $S_1$  and the other point in  $S_2$ . Consider a *nonempty* set  $S$  that is not equal to  $\Omega$ . Let  $S^c = \Omega \setminus S$  be the set of points that are not in  $S$ . The following statements are equivalent.

- (i) The set  $S$  is a cluster, i.e.,  $g(S, S) \geq 0$ .
- (ii) The set  $S^c$  is a cluster, i.e.,  $g(S^c, S^c) \geq 0$ .
- (iii) The two sets  $S$  and  $S^c$  are incohesive, i.e.,  $g(S, S^c) \leq 0$ .
- (iv) The set  $S$  is more cohesive to itself than to  $S^c$ , i.e.,  $g(S, S) \geq g(S, S^c)$ .
- (v)  $2\bar{d}(S, \Omega) - \bar{d}(\Omega, \Omega) - \bar{d}(S, S) \geq 0$ .
- (vi)  $2\bar{d}(S, S^c) - \bar{d}(S, S) - \bar{d}(S^c, S^c) \geq 0$ .

The condition for a cluster in Theorem 3(vi) is of particular importance as it allows us to characterize a cluster by using the average distance measures on the set  $S$  and its complement  $S^c$ . Such a condition will be used for proving our main result in Theorem 7.

### C. The K-sets<sup>+</sup> algorithm

Though the extensions of the duality result and the equivalent statements for clusters to semi-metric spaces are basically the same as those in [18], one problem arises when extending the K-sets algorithm to a semi-metric space. The key problem is that the *triangular distance* ( $\Delta$ -distance) defined in the K-sets algorithm (see Definition 4 below) might not be nonnegative in a semi-metric space.

**Definition 4: ( $\Delta$ -distance [18])** For a symmetric bivariate function  $g(\cdot, \cdot)$  on a set of data points  $\Omega = \{x_1, x_2, \dots, x_n\}$ , the  $\Delta$ -distance from a point  $x$  to a set  $S$ , denoted by  $\Delta(x, S)$ , is defined as follows:

$$\Delta(x, S) = g(x, x) - \frac{2}{|S|}g(x, S) + \frac{1}{|S|^2}g(S, S), \quad (9)$$

where  $g(S_1, S_2)$  is defined (6).

Note from (1) that the  $\Delta$ -distance from a point  $x$  to a set  $S$  in a semi-metric space can also be written as follows:

$$\Delta(x, S) = \frac{1}{|S|^2} \sum_{z_1 \in S} \sum_{z_2 \in S} (d(x, z_1) + d(x, z_2) - d(z_1, z_2)). \quad (10)$$

Now consider the data set of three points  $\Omega = \{x, y, z\}$  with the semi-metric  $d(\cdot, \cdot)$  in Table I. For  $S = \{y, z\}$ , one can easily compute from (10) that  $\Delta(x, S) = -1 < 0$ .

TABLE I

A DATA SET OF THREE POINTS  $\Omega = \{x, y, z\}$  WITH A SEMI-METRIC  $d(\cdot, \cdot)$ .

$d(\cdot, \cdot)$	$x$	$y$	$z$
$x$	0	1	1
$y$	1	0	6
$z$	1	6	0

Since the  $\Delta$ -distance might not be nonnegative in a semi-metric space, the proofs for the convergence and the performance guarantee of the K-sets algorithm in [18] are no longer valid. Fortunately, the  $\Delta$ -distance in a semi-metric space has the following (weaker) nonnegative property that will enable us to prove the performance guarantee of the K-sets<sup>+</sup> algorithm (defined in Algorithm 1 later) for clustering data points in a semi-metric space.

---

### ALGORITHM 1: The K-sets<sup>+</sup> Algorithm

---

**Input:** A data set  $\Omega = \{x_1, x_2, \dots, x_n\}$ , a *symmetric* matrix  $G = (g(\cdot, \cdot))$  and the number of sets  $K$ .

**Output:** A partition of sets  $\{S_1, S_2, \dots, S_K\}$ .

**(0)** Initially, choose arbitrarily  $K$  disjoint nonempty sets  $S_1, \dots, S_K$  as a partition of  $\Omega$ .

**(1)** **for**  $i = 1, 2, \dots, n$  **do**

Compute the adjusted  $\Delta$ -distance  $\Delta_a(x_i, S_k)$  for each set  $S_k$  by using (9) and (12). Find the set to which the point  $x_i$  is closest in terms of the adjusted  $\Delta$ -distance. Assign that point  $x_i$  to that set.

**end**

**(2)** Repeat from (1) until there is no further change.

---

**Proposition 5:** Consider a data set  $\Omega = \{x_1, x_2, \dots, x_n\}$  with a semi-metric  $d(\cdot, \cdot)$ . For any subset  $S$  of  $\Omega$ ,

$$\sum_{x \in S} \Delta(x, S) = |S|\bar{d}(S, S) \geq 0. \quad (11)$$

The proof of (11) in Proposition 5 follows directly from (10) and (8). To introduce the K-sets<sup>+</sup> algorithm, we first define the adjusted  $\Delta$ -distance in Definition 6 below.

**Definition 6: (Adjusted  $\Delta$ -distance)** The *adjusted  $\Delta$ -distance* from a point  $x$  to a set  $S$ , denoted by  $\Delta_a(x, S)$ , is defined as follows:

$$\Delta_a(x, S) = \begin{cases} \frac{|S|}{|S|+1} \Delta(x, S), & \text{if } x \notin S, \\ \frac{|S|}{|S|-1} \Delta(x, S), & \text{if } x \in S \text{ and } |S| > 1, \\ -\infty, & \text{if } x \in S \text{ and } |S| = 1. \end{cases} \quad (12)$$

Instead of using the  $\Delta$ -distance for the assignment of a data point in the K-sets algorithm, we use the adjusted  $\Delta$ -distance for the assignment in the K-sets<sup>+</sup> algorithm. We outline the K-sets<sup>+</sup> algorithm in Algorithm 1. Note that in Algorithm 1, the bivariate function  $g(\cdot, \cdot)$  is required to be symmetric, i.e.,  $g(x, y) = g(y, x)$ . If  $g(\cdot, \cdot)$  is not symmetric, one may consider using  $\hat{g}(x, y) = (g(x, y) + g(y, x))/2$ .

In the following theorem, we show the convergence and the performance guarantee of the K-sets<sup>+</sup> algorithm. The proof of Theorem 7 is given in Appendix A.

**Theorem 7:** For a data set  $\Omega = \{x_1, x_2, \dots, x_n\}$  with a *symmetric* matrix  $G = (g(\cdot, \cdot))$ , consider the clustering problem that finds a partition  $\{S_1, S_2, \dots, S_K\}$  of  $\Omega$  with a fixed  $K$  that maximizes the objective function  $\sum_{k=1}^K \frac{1}{|S_k|} g(S_k, S_k)$ .

- (i) The K-sets<sup>+</sup> algorithm in Algorithm 1 converges monotonically to a local optimum of the optimization problem in a finite number of iterations.
- (ii) Suppose that  $g(\cdot, \cdot)$  is a semi-cohesion measure. Let  $S_1, S_2, \dots, S_K$  be the  $K$  sets when the algorithm converges. Then for all  $i \neq j$ , the two sets  $S_i$  and  $S_j$  are two clusters if these two sets are viewed in isolation (by removing the data points not in  $S_i \cup S_j$  from  $\Omega$ ).

In particular, if  $K = 2$ , it then follows from Theorem 7(ii) that the K-sets<sup>+</sup> algorithm yields two clusters for data points in a semi-metric space.

### III. BEYOND SEMI-METRIC SPACES

#### A. Clustering with a symmetric similarity measure

In this section, we further extend the applicability of the K-sets<sup>+</sup> algorithm to the clustering problem with a symmetric similarity measure. A similarity measure is generally referred to as a bivariate function that measures how similar two data points are. The clustering problem with a similarity measure is to cluster data points so that similar data points are clustered together. For a symmetric similarity measure  $g(\cdot, \cdot)$ , we have shown in Theorem 7(i) that the K-sets<sup>+</sup> algorithm in Algorithm 1 converges monotonically to a local optimum of the optimization problem  $\sum_{k=1}^K \frac{1}{|S_k|} g(S_k, S_k)$  within a finite number of iterations. Thus, the K-sets<sup>+</sup> algorithm can be applied for clustering with a symmetric similarity measure. But what is the physical meaning of the sets returned by the K-sets<sup>+</sup> algorithm for such a symmetric similarity measure? In order to answer this question, we show there is a natural semi-cohesion measure from a symmetric similarity measure and the K-sets<sup>+</sup> algorithm that uses this semi-cohesion measure converges to the same partition as that using the original symmetric similarity measure (if they both use the same initial partition). As a direct consequence of Theorem 7(ii), any two sets returned by the K-sets<sup>+</sup> algorithm for such a symmetric similarity measure are clusters with respect to the semi-cohesion measure when they are viewed in isolation.

In Lemma 8 below, we first show how one can map a symmetric similarity measure to a semi-cohesion measure. The proof is given in Appendix B.

*Lemma 8:* For a symmetric similarity measure  $g(\cdot, \cdot)$ , let

$$\begin{aligned} \tilde{g}(x, y) &= g(x, y) - \frac{1}{n}g(x, \Omega) - \frac{1}{n}g(y, \Omega) \\ &+ \frac{1}{n^2}g(\Omega, \Omega) + \sigma\delta(x, y) - \frac{\sigma}{n}, \end{aligned} \quad (13)$$

where  $\delta(x, y)$  is the usual  $\delta$  function (that has value 1 if  $x = y$  and 0 otherwise), and  $\sigma$  is a constant that satisfies

$$\sigma \geq \max_{x \neq y} [g(x, y) - (g(x, x) + g(y, y))/2]. \quad (14)$$

Then the bivariate function  $\tilde{g}(\cdot, \cdot)$  in (14) is a *semi-cohesion measure* for  $\Omega$ , i.e., it satisfies (C1), (C2) and (C3).

In the following lemma, we further establish the connections for the  $\Delta$ -distance and the adjusted  $\Delta$ -distance between the original symmetric similarity measure  $g(\cdot, \cdot)$  and the semi-cohesion measure  $\tilde{g}(\cdot, \cdot)$  in (13). The proof is given in Appendix C.

*Lemma 9:* Let  $\Delta(x, S)$  (resp.  $\tilde{\Delta}(x, S)$ ) be the  $\Delta$ -distance from a point  $x$  to a set  $S$  with respect to  $g(\cdot, \cdot)$  (resp.  $\tilde{g}(\cdot, \cdot)$ ). Also, let  $\Delta_{\mathbf{a}}(x, S)$  (resp.  $\tilde{\Delta}_{\mathbf{a}}(x, S)$ ) be the adjusted  $\Delta$ -distance from a point  $x$  to a set  $S$  with respect to  $g(\cdot, \cdot)$  (resp.  $\tilde{g}(\cdot, \cdot)$ ). Then

$$\tilde{\Delta}(x, S) = \begin{cases} \Delta(x, S) + \sigma(1 - \frac{1}{|S|}), & \text{if } x \notin S, \\ \Delta(x, S) + \sigma(1 + \frac{1}{|S|}), & \text{if } x \in S. \end{cases} \quad (15)$$

Moreover,

$$\tilde{\Delta}_{\mathbf{a}}(x, S) = \Delta_{\mathbf{a}}(x, S) + \sigma. \quad (16)$$

It is easy to see that for any partition  $S_1, S_2, \dots, S_K$

$$\begin{aligned} &\sum_{k=1}^K \frac{1}{|S_k|} \tilde{g}(S_k, S_k) \\ &= \sum_{k=1}^K \frac{1}{|S_k|} g(S_k, S_k) - \frac{1}{n}g(\Omega, \Omega) + (K-1)\sigma. \end{aligned} \quad (17)$$

Thus, optimizing  $\sum_{k=1}^K \frac{1}{|S_k|} g(S_k, S_k)$  with respect to the symmetric similarity measure  $g(\cdot, \cdot)$  is equivalent to optimizing  $\sum_{k=1}^K \frac{1}{|S_k|} \tilde{g}(S_k, S_k)$  with respect to the semi-cohesion measure  $\tilde{g}(\cdot, \cdot)$ . Since

$$\tilde{\Delta}_{\mathbf{a}}(x, S) = \Delta_{\mathbf{a}}(x, S) + \sigma, \quad (18)$$

we conclude that for these two optimization problems the K-sets<sup>+</sup> algorithm converges to the same partition if they both use the same initial partition.

Note that the K-means algorithm needs the data points to be in a Euclidean space, the kernel K-means algorithm needs the data points to be mapped into some Euclidean space, and the K-sets algorithm needs the data points to be in a metric space. The result in Lemma 9 shows that the K-sets<sup>+</sup> algorithm lifts all the constraints on the data points and it can be operated merely by a symmetric similarity measure.

#### B. Computational complexity

In this section, we address the computational complexity and the memory complexity of the K-sets<sup>+</sup> algorithm. For an  $n \times n$  symmetric similarity matrix with only  $m$  nonzero elements in the matrix, we show that the computational complexity of the K-sets<sup>+</sup> algorithm is  $O((Kn+m)I)$ , where  $I$  is the number of iterations. The memory complexity to achieve that computational complexity is  $O(Kn+m)$ .

Note that the main computation overhead of the K-sets<sup>+</sup> algorithm is mainly for the computation of the adjusted  $\Delta$ -distance. In view of (9), we know that one needs to compute  $g(x, S)$  and  $\frac{1}{|S|^2}g(S, S)$  in order to compute  $\Delta(x, S)$ . Let

$$\bar{g}(S_1, S_2) = \frac{1}{|S_1||S_2|}g(S_1, S_2). \quad (19)$$

Our approach to reduce the computational complexity is to store  $\bar{g}(S_k, S_k)$ ,  $k = 1, 2, \dots, K$  and  $g(x_i, S_k)$  for  $i = 1, 2, \dots, n$  and  $k = 1, 2, \dots, K$ . Once these are stored in memory, one can compute the adjusted  $\Delta$ -distance  $\Delta_{\mathbf{a}}(x_i, S_k)$  in  $O(1)$  steps. Suppose that  $x_i$  is originally in the set  $S_1$  and it is reassigned to  $S_2$ . Then  $\bar{g}(S_2 \cup \{x_i\}, S_2 \cup \{x_i\})$  can be updated by computing

$$\begin{aligned} &\frac{|S_2|^2}{(|S_2|+1)^2} \bar{g}(S_2, S_2) + \frac{2|S_2|}{(|S_2|+1)^2} \bar{g}(\{x_i\}, S_2) \\ &+ \frac{1}{(|S_2|+1)^2} g(x_i, x_i). \end{aligned} \quad (20)$$

Also,  $\bar{g}(S_1 \setminus \{x_i\}, S_1 \setminus \{x_i\})$  can be updated by computing

$$\frac{|S_1|^2}{(|S_1| - 1)^2} \left( \bar{g}(S_1, S_1) - 2 \frac{1}{|S_1|} \bar{g}(\{x_i\}, S_1) \right) + \frac{1}{(|S_1| - 1)^2} g(x_i, x_i). \quad (21)$$

Such updates can be done in  $O(1)$  steps. On the other hand, let

$$\text{Nei}(i) = \{j : g(x_i, x_j) \neq 0\} \quad (22)$$

be the set of data points that are neighbors of  $x_i$ . Note that if  $g(x_i, x_i) \neq 0$ , then  $x_i$  is also in  $\text{Nei}(i)$ . When  $x_i$  is moved from  $S_1$  to  $S_2$ , we only need to update  $g(y, S_1)$  and  $g(y, S_2)$  for the data point  $y$  that is a neighbor of  $x_i$ . Specifically, For each node  $y \in \text{Nei}(i)$ , update

$$\begin{aligned} g(y, S_2) &\leftarrow g(y, S_2) + g(y, x_i), \\ g(y, S_1) &\leftarrow g(y, S_1) - g(y, x_i). \end{aligned}$$

Such updates can be done in  $O(\text{Nei}(i))$  steps. Let

$$m = \sum_{i=1}^n |\text{Nei}(i)| \quad (23)$$

be the total number of nonzero entries in the  $n \times n$  symmetric matrix  $G = (g(\cdot, \cdot))$ . Then the total number of updates for all the data points  $x_i$ ,  $i = 1, 2, \dots, n$ , can be done in  $O(m)$  steps. Since we need to compute the  $\Delta$ -distance for the  $K$  sets for each data point in the *for* loop in Algorithm 1, the computational complexity of the K-sets<sup>+</sup> algorithm of this implementation is thus  $O((Kn + m)I)$ , where  $I$  is the number of iterations in the *for* loop of Algorithm 1. Regarding the memory complexity, one can store the symmetric matrix  $G = (g(\cdot, \cdot))$  in the adjacency list form and that requires  $O(m)$  amount of memory. The memory requirement for storing  $S_k$ ,  $k = 1, 2, \dots, K$ ,  $\bar{g}(S_k, S_k)$ ,  $k = 1, 2, \dots, K$  and  $g(x_i, S_k)$  for  $i = 1, 2, \dots, n$  and  $k = 1, 2, \dots, K$  is  $O(Kn)$ . Thus, the overall memory complexity is  $O(Kn + m)$ .

#### IV. EXPERIMENTS

In this section, we evaluate the performance of the K-sets<sup>+</sup> algorithm by conducting two experiments: (i) community detection of signed networks generated by the stochastic block model in Section IV-A, and (ii) clustering of a real network from the WonderNetwork website [19] in Section IV-B.

##### A. Community detection of signed networks with two communities

In this section, we conduct experiments for the K-sets<sup>+</sup> algorithm by using the signed networks from the stochastic block model. We follow the procedure in [20] to generate the test networks. Each test network consists of  $n$  nodes and two ground-truth blocks, each with  $n/2$  nodes. There are three key parameters  $p_{in}$ ,  $p_{out}$ , and  $p$  for generating a test network. The parameter  $p_{in}$  is the probability that there is a *positive* edge between two nodes within the same block and  $p_{out}$  is the probability that there is a *negative* edge between two nodes in two different blocks. All edges are generated independently

according to  $p_{in}$  and  $p_{out}$ . After all the signed edges are generated, we then flip the sign of an edge independently with the crossover probability  $p$ .

In our experiments, the total number of nodes in the stochastic block model is  $n = 2000$  with 1000 nodes in each block. Let  $c = (n/2 - 1)p_{in} + np_{out}/2$  be the average degree of a node, and it is set to be 6, 8, and 10, respectively. Also, let  $c_{in} = np_{in}$  and  $c_{out} = np_{out}$ . The value of  $c_{in} - c_{out}$  is set to be 5 and that is used with the average degree  $c$  to uniquely determine  $p_{in}$  and  $p_{out}$ . The crossover probability  $p$  is in the range from 0.01 to 0.2 with a common step of 0.01. We generate 20 graphs for each  $p$  and  $c$ . We remove isolated nodes, and thus the exact numbers of nodes in the experiments might be less than 2000. We show the experimental results with each point averaged over 20 random graphs. The error bars represent the 95% confident intervals.

To test the K-sets<sup>+</sup> algorithm, we use the similarity matrix  $G$  with

$$G = A + 0.5A^2, \quad (24)$$

where  $A$  is the adjacency matrix of the signed network after randomly flipping the sign of an edge. Such a similarity matrix was suggested in [20] for community detection in signed networks as it allows us to “see” more than one step relationship between two nodes.

In Figure 1, we show our experimental results for edge accuracy (the percentage of edges that are correctly detected) as a function of the crossover probability  $p$ . As shown in Figure 1, the K-sets<sup>+</sup> algorithm performs very well. For  $c = 10$ , it can still recover almost 100% of the edges even when the crossover probability  $p$  is 0.1 and roughly 95% of the edges when the crossover probability  $p$  is 0.2. Also, increasing the average degree  $c$  in the stochastic block model also increases the edge accuracy for the K-sets<sup>+</sup> algorithm. This might be due to the fact that the tested signed networks with a larger average degree are more dense.

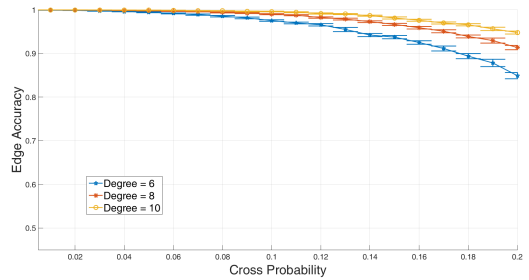


Fig. 1. Community detection of signed networks with two communities.

##### B. Clustering of a real network

In this section, we test the K-sets<sup>+</sup> algorithm on the real network from the WonderNetwork website [19]. In this dataset, there are 216 servers in different locations and the latency (measured by the round trip time) between any two servers of these 216 servers are recorded in real time. The dataset in our experiment is a snapshot on Sept. 24, 2016. For

this dataset, the triangular inequality is not always satisfied. For example, we notice that  $\text{latency}(\text{Adelaide}, \text{Athens})=250$ ,  $\text{latency}(\text{Athens}, \text{Albany})=138$ ,  $\text{latency}(\text{Adelaide}, \text{Albany})=400$ , and  $250 + 138 \leq 400$ . In addition to the latency data, the WonderNetwork website also provides the geographic location of each server. We then use the Haversine formula to compute the distance between any two servers. In the WonderNetwork dataset, the latency measure from location  $L_1$  to location  $L_2$  is slightly different from that from location  $L_2$  to location  $L_1$ . To ensure that the latency measure is a semi-metric, we simply symmetrize the latency matrix by taking the average of the latency measures from both directions. In our experiments, the number of clusters  $K$  is set to 5. We run 20 times of the K-sets<sup>+</sup> algorithm by using the distance matrix and the latency matrix, respectively. In each of the 20 trials, the initial partition is randomly selected. The output partition that has the best objective value from these 20 trials is selected. The results for the distance matrix and the latency matrix are shown in Figure 2(a) and (b), respectively. In Figure 2(a) and (b), the servers that are in the same cluster are marked with the same colored marker. In view of Figure 2(a), we can observe that almost all the servers are partitioned into densely packed clusters except for the servers in South America and Africa. On the other hand, as shown in Figure 2(b), the servers in South America and Africa are merged into other clusters. To shed more light on these interesting differences, we compare the findings in Figure 2(b) to the Submarine Cable Map [21] (which records the currently active submarine cables). We notice that there are many cables placed around South America, connecting to the Caribbean and then to the East Coast of the United States. These cables greatly reduce the latency from South America to North America and thus cause the servers in South America to be clustered with the servers in North America. Similarly, there are a few connected cables between Africa and Europe. Therefore, the servers in Africa and Europe are clustered together. Due to many directly connected cables from Dubai to the Singapore Strait, servers around India are not clustered with the other servers in Asia. In particular, there are two servers marked with green dots, located in Jakarta and Singapore, that are clustered with servers in India even though they are geographically closer to the East Asia. These outliers have low latency to communicate with those servers in India. Finally, there are three servers, two in Russia and one in Lahore, that have low latency to the servers in Europe and they are clustered with the servers in Europe.

## V. CONCLUSION

In this paper, we proposed the K-sets<sup>+</sup> algorithm for clustering data points in a semi-metric space and data points that only have a symmetric similarity measure. We showed that the K-sets<sup>+</sup> algorithm converges in a finite number of iterations and it retains the same performance guarantee as the K-sets algorithm in [18]. Moreover, both the computational complexity and the memory complexity are linear in  $n$  when the  $n \times n$  similarity matrix is *sparse*, i.e.,  $m = O(n)$ . To show the effectiveness of the K-sets<sup>+</sup> algorithm, we also

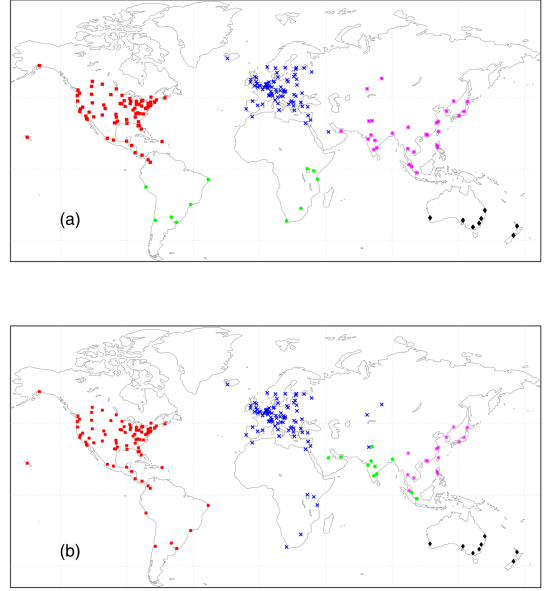


Fig. 2. Clustering for the WonderNetwork dataset: (a) the (geographic) distance matrix and (b) the latency matrix.

conducted various experiments by using a synthetic dataset from the stochastic block model and a real network from the WonderNetwork website [19].

## APPENDIX A

In this section, we prove Theorem 7.

(i) It suffices to show that if  $x$  is in a set  $S_1$  with  $|S_1| > 1$  and  $\Delta_a(x, S_2) < \Delta_a(x, S_1)$ , then move point  $x$  from  $S_1$  to  $S_2$  increases the value of the objective function. Let  $S_k$  (resp.  $S'_k$ ),  $k = 1, 2, \dots, K$ , be the partition before (resp. after) the change. Also let  $R$  (resp.  $R'$ ) be the value of the objective function before (resp. after) the change. Then

$$\begin{aligned} R' - R &= \frac{g(S_1 \setminus \{x\}, S_1 \setminus \{x\})}{|S_1| - 1} + \frac{g(S_2 \cup \{x\}, S_2 \cup \{x\})}{|S_2| + 1} \\ &\quad - \frac{g(S_1, S_1)}{|S_1|} - \frac{g(S_2, S_2)}{|S_2|}. \end{aligned}$$

Since

$$\begin{aligned} &g(S_2 \cup \{x\}, S_2 \cup \{x\}) \\ &= g(S_2, S_2) + 2g(x, S_2) + g(x, x), \end{aligned}$$

we have from (9) and (12) that

$$\begin{aligned} &\frac{g(S_2 \cup \{x\}, S_2 \cup \{x\})}{|S_2| + 1} - \frac{g(S_2, S_2)}{|S_2|} \\ &= \frac{2|S_2|g(x, S_2) + |S_2|g(x, x) - g(S_2, S_2)}{|S_2| \cdot (|S_2| + 1)} \\ &= g(x, x) - \frac{|S_2|}{|S_2| + 1} \Delta(x, S_2) \\ &= g(x, x) - \Delta_a(x, S_2). \end{aligned}$$

On the other hand, we note that

$$g(S_1 \setminus \{x\}, S_1 \setminus \{x\}) = g(S_1, S_1) - 2g(x, S_1) + g(x, x). \quad (25)$$

Using (25), (9) and (12) yields

$$\begin{aligned} & \frac{g(S_1 \setminus \{x\}, S_1 \setminus \{x\})}{|S_1| - 1} - \frac{g(S_1, S_1)}{|S_1|} \\ &= \frac{-2|S_1|g(x, S_1) + |S_1|g(x, x) + g(S_1, S_1)}{|S_1| \cdot (|S_1| - 1)} \\ &= -g(x, x) + \frac{|S_1|}{|S_1| - 1} \Delta(x, S_1) \\ &= -g(x, x) + \Delta_{\mathbf{a}}(x, S_1). \end{aligned}$$

Thus,

$$R' - R = \Delta_{\mathbf{a}}(x, S_1) - \Delta_{\mathbf{a}}(x, S_2) > 0.$$

As the objective value is non-increasing after a change of the partition, there is no loop in the algorithm. Since the number of partitions is finite, the algorithm thus converges in a finite number of steps (iterations).

(ii) Let  $d(\cdot, \cdot)$  be the induced semi-metric. In view of Theorem 3(vi), it suffices to show that for all  $i \neq j$

$$2\bar{d}(S_i, S_j) - \bar{d}(S_i, S_i) - \bar{d}(S_j, S_j) \geq 0. \quad (26)$$

If the set  $S_i$  contains a single element  $x$ , then

$$\bar{d}(S_i, S_i) = d(x, x) = 0.$$

Thus, the inequality in (26) holds trivially if  $|S_i| = |S_j| = 1$ .

Now suppose that  $\min(|S_i|, |S_j|) \geq 2$ . Without loss of generality, we assume that  $|S_i| \geq 2$ . When the K-sets<sup>+</sup> algorithm converges, we know that for any  $x \in S_i$ ,

$$\Delta_{\mathbf{a}}(x, S_i) \leq \Delta_{\mathbf{a}}(x, S_j).$$

Summing over  $x \in S_i$  yields

$$\sum_{x \in S_i} \Delta_{\mathbf{a}}(x, S_i) \leq \sum_{x \in S_i} \Delta_{\mathbf{a}}(x, S_j). \quad (27)$$

Note from (12) that for any  $x \in S_i$ ,

$$\Delta_{\mathbf{a}}(x, S_i) = \frac{|S_i|}{|S_i| - 1} \Delta(x, S_i),$$

and

$$\Delta_{\mathbf{a}}(x, S_j) = \frac{|S_j|}{|S_j| + 1} \Delta(x, S_j).$$

Thus, it follows from (27) that

$$\frac{|S_i|}{|S_i| - 1} \sum_{x \in S_i} \Delta(x, S_i) \leq \frac{|S_j|}{|S_j| + 1} \sum_{x \in S_i} \Delta(x, S_j). \quad (28)$$

Note from (10) that

$$\sum_{x \in S_i} \Delta(x, S_i) = |S_i| \bar{d}(S_i, S_i), \quad (29)$$

and that

$$\sum_{x \in S_i} \Delta(x, S_j) = |S_i| (2\bar{d}(S_i, S_j) - \bar{d}(S_j, S_j)). \quad (30)$$

Since  $d(\cdot, \cdot)$  is the induced semi-metric, we know from Proposition 5 that

$$\sum_{x \in S_i} \Delta(x, S_i) \geq 0.$$

Using this in (28) yields

$$\sum_{x \in S_i} \Delta(x, S_j) \geq 0.$$

Thus, we have from (28) that

$$\begin{aligned} \sum_{x \in S_i} \Delta(x, S_i) &\leq \frac{|S_i|}{|S_i| - 1} \sum_{x \in S_i} \Delta(x, S_i) \\ &\leq \frac{|S_j|}{|S_j| + 1} \sum_{x \in S_i} \Delta(x, S_j) \leq \sum_{x \in S_i} \Delta(x, S_j). \end{aligned} \quad (31)$$

That the inequality in (26) holds follows directly from (29), (30) and (31).

## APPENDIX B

In this section, we prove Lemma 8.

Since  $g(\cdot, \cdot)$  is symmetric, clearly  $\tilde{g}(\cdot, \cdot)$  is also symmetric. Thus, (C1) is satisfied trivially. To see that (C2) is satisfied, observe from (13) that

$$\begin{aligned} \sum_{y \in \Omega} \tilde{g}(x, y) &= g(x, \Omega) - g(x, \Omega) - \frac{1}{n} g(\Omega, \Omega) \\ &\quad + \frac{1}{n} g(\Omega, \Omega) + \sigma - \sigma = 0. \end{aligned} \quad (32)$$

To see that (C3) holds, we note that

$$\tilde{g}(x, x) = g(x, x) - \frac{2}{n} g(x, \Omega) + \frac{1}{n^2} g(\Omega, \Omega) + \frac{(n-1)}{n} \sigma, \quad (33)$$

and that

$$\tilde{g}(y, y) = g(y, y) - \frac{2}{n} g(y, \Omega) + \frac{1}{n^2} g(\Omega, \Omega) + \frac{(n-1)}{n} \sigma.$$

Thus, for  $x \neq y$ , we have from (14) that

$$\begin{aligned} &\tilde{g}(x, x) + \tilde{g}(y, y) - 2\tilde{g}(x, y) \\ &= g(x, x) + g(y, y) - 2g(x, y) + 2\sigma \geq 0. \end{aligned}$$

## APPENDIX C

In this section, we prove Lemma 9.

Note from Definition 4 and Definition 6 that

$$\tilde{\Delta}(x, S) = \tilde{g}(x, x) - \frac{2}{|S|} \tilde{g}(x, S) + \frac{1}{|S|^2} \tilde{g}(S, S), \quad (34)$$

and that

$$\tilde{\Delta}_{\mathbf{a}}(x, S) = \begin{cases} \frac{|S|}{|S|+1} \tilde{\Delta}(x, S), & \text{if } x \notin S, \\ \frac{|S|}{|S|-1} \tilde{\Delta}(x, S), & \text{if } x \in S \text{ and } |S| > 1, \\ -\infty, & \text{if } x \in S \text{ and } |S| = 1. \end{cases} \quad (35)$$

To show (15), we need to consider two cases: (i)  $x \in S$  and (ii)  $x \notin S$ . For both cases, we have from (13) that

$$\tilde{g}(x, x) = g(x, x) - \frac{2}{n} g(x, \Omega) + \frac{1}{n^2} g(\Omega, \Omega) + \frac{(n-1)}{n} \sigma, \quad (36)$$

and

$$\begin{aligned}\tilde{g}(S, S) &= g(S, S) - \frac{|S|}{n}g(S, \Omega) - \frac{|S|}{n}g(S, \Omega) \\ &+ \frac{|S|^2}{n^2}g(\Omega, \Omega) + \sigma|S| - \sigma\frac{|S|^2}{n}.\end{aligned}\quad (37)$$

Now we consider the first case that  $x \in S$ . In this case, note that for  $x \in S$

$$\begin{aligned}\tilde{g}(x, S) &= g(x, S) - \frac{|S|}{n}g(x, \Omega) - \frac{1}{n}g(S, \Omega) \\ &+ \frac{|S|}{n^2}g(\Omega, \Omega) + \sigma - \sigma\frac{|S|}{n}.\end{aligned}\quad (38)$$

Using (36), (38) and (37) in (34) yields

$$\begin{aligned}\tilde{\Delta}(x, S) &= \tilde{g}(x, x) - \frac{2}{|S|}\tilde{g}(x, S) + \frac{1}{|S|^2}\tilde{g}(S, S) \\ &= g(x, x) - \frac{2}{|S|}g(x, S) + \frac{1}{|S|^2}g(S, S) \\ &\quad + \sigma\left(1 - \frac{1}{|S|}\right) \\ &= \Delta(x, S) + \sigma\left(1 - \frac{1}{|S|}\right).\end{aligned}\quad (39)$$

From (35), it then follows that

$$\begin{aligned}\tilde{\Delta}_{\mathbf{a}}(x, S) &= \frac{|S|}{|S|-1}\tilde{\Delta}(x, S) \\ &= \frac{|S|}{|S|-1}\left(\Delta(x, S) + \sigma\left(1 - \frac{1}{|S|}\right)\right) \\ &= \Delta_{\mathbf{a}}(x, S) + \sigma.\end{aligned}\quad (40)$$

Now we consider the second case that  $x \notin S$ . In this case, note that for  $x \notin S$

$$\begin{aligned}\tilde{g}(x, S) &= g(x, S) - \frac{|S|}{n}g(x, \Omega) - \frac{1}{n}g(S, \Omega) \\ &+ \frac{|S|}{n^2}g(\Omega, \Omega) - \sigma\frac{|S|}{n}.\end{aligned}\quad (41)$$

Using (36), (41) and (37) in (34) yields

$$\begin{aligned}\tilde{\Delta}(x, S) &= \tilde{g}(x, x) - \frac{2}{|S|}\tilde{g}(x, S) + \frac{1}{|S|^2}\tilde{g}(S, S) \\ &= g(x, x) - \frac{2}{|S|}g(x, S) + \frac{1}{|S|^2}g(S, S) \\ &\quad + \sigma\left(1 + \frac{1}{|S|}\right) \\ &= \Delta(x, S) + \sigma\left(1 + \frac{1}{|S|}\right).\end{aligned}\quad (42)$$

From (35), it then follows that

$$\begin{aligned}\tilde{\Delta}_{\mathbf{a}}(x, S) &= \frac{|S|}{|S|+1}\tilde{\Delta}(x, S) \\ &= \frac{|S|}{|S|+1}\left(\Delta(x, S) + \sigma\left(1 + \frac{1}{|S|}\right)\right) \\ &= \Delta_{\mathbf{a}}(x, S) + \sigma.\end{aligned}\quad (43)$$

## REFERENCES

- [1] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Elsevier Academic press, USA, 2006.
- [2] A. Rajaraman, J. Leskovec, and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2012.
- [3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.
- [4] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [5] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [6] M. Agarwal, R. Jaiswal, and A. Pal, "k-means++ under approximation stability," in *Theory and Applications of Models of Computation*. Springer, 2013, pp. 84–95.
- [7] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.
- [8] M. Van der Laan, K. Pollard, and J. Bryan, "A new partitioning around medoids algorithm," *Journal of Statistical Computation and Simulation*, vol. 73, no. 8, pp. 575–584, 2003.
- [9] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.
- [10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [11] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [12] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 176–190, 2008.
- [13] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, "Spectral redemption in clustering sparse networks," *Proceedings of the National Academy of Sciences*, vol. 110, no. 52, pp. 20 935–20 940, 2013.
- [14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD*, vol. 96, 1996, pp. 226–231.
- [15] A. Cuevas, M. Febrero, and R. Fraiman, "Cluster analysis: a further approach based on density estimation," *Computational Statistics & Data Analysis*, vol. 36, no. 4, pp. 441–459, 2001.
- [16] M. Halkidi and M. Vazirgiannis, "A density-based cluster validity approach using multi-representatives," *Pattern Recognition Letters*, vol. 29, no. 6, pp. 773–786, 2008.
- [17] M.-F. Balcan, A. Blum, and A. Gupta, "Clustering under approximation stability," *Journal of the ACM (JACM)*, vol. 60, no. 2, p. 8, 2013.
- [18] C.-S. Chang, W. Liao, Y.-S. Chen, and L.-H. Liou, "A mathematical theory for clustering in metric spaces," *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 1, pp. 2–16, 2016.
- [19] "The wondernetwork dataset," <https://wondernetwork.com/>.
- [20] S.-M. Lu, L.-H. Liou, C.-S. Chang, and D.-S. Lee, "Community detection in signed networks: an error-correcting code approach," *submitted for publication*, 2016.
- [21] "The submarine cable map," <http://www.submarinecablemap.com/>.