Fuzzy Systems and Data Mining VI
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA200724

Community Detection, Pattern Recognition, and Hypergraph-Based Learning: Approaches Using Metric Geometry and Persistent Homology

Dong Quan Ngoc NGUYEN [a,1], Lin XING [a], and Lizhen LIN [a]
[a] *Department of Applied and Computational Mathematics and Statistics,*
University of Notre Dame,
Notre Dame, Indiana 46556 USA

Abstract. Hypergraph data appear and are hidden in many places in the modern age. They are data structure that can be used to model many real data examples since their structures contain information about higher order relations among data points. One of the main contributions of our paper is to introduce a new topological structure to hypergraph data which bears a resemblance to a usual metric space structure. Using this new topological space structure of hypergraph data, we propose several approaches to study community detection problem, detecting persistent features arising from homological structure of hypergraph data. Also based on the topological space structure of hypergraph data introduced in our paper, we introduce a modified nearest neighbors methods which is a generalization of the classical nearest neighbors methods from machine learning. Our modified nearest neighbors methods have an advantage of being very flexible and applicable even for discrete structures as in hypergraphs. We then apply our modified nearest neighbors methods to study sign prediction problem in hypegraph data constructed using our method.

Keywords. Distance matrices; Hypergraphs; Metric geometry; Metric spaces; Nearest Neighborhoods; Persistent homology;

1. Introduction

One of the challenges in the modern age is to classify data arising from many resources; for example, following the rapid developments of several areas in mathematics, a large number of publications in mathematics creates a tremendous amount of data, which signifies useful information such as relationships (or collaborations) among authors and their publications, and their influences on development of mathematics. It is often the case that analyzing such data is not straightforward, and very difficult task because of the extremely fast growth of relations among data, and of data itself.

[1] Corresponding Author: Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, Indiana 46556 USA; E-mail: dongquan.ngoc.nguyen@nd.edu

In this paper, we propose several methods to analyze *hypergraph data*. Recall that a hypergraph X is a pair $(\mathscr{V}(X), \mathscr{E}(X))$, where $\mathscr{V}(X)$ is the set of data points (called vertices of X), and $\mathscr{E}(X)$ is a subset of the power set of $\mathscr{V}(X)$ which represents relations among data points. Each element in $\mathscr{E}(X)$ is called a *hyperedge*. Note that by abuse of notation, we sometime use the same symbol for X and its set of vertices.

A standard example of a hypergraph is a collaboration network in which the set of vertices consists of mathematicians, and a group of mathematicians (not necessarily only two) forms a hyperedge if they have at least one joint publication. Many real data can be modeled as a hypergraph. Applications of hypergraph data are diverse such as in protein function prediction (see [1]), and other areas (for example, see [2], [3], [4], [5]).

The aim of this paper is to propose several approaches to studying community detection, pattern recognition, and sign prediction problem. Our approaches use main tools from metric geometry (see, for example, [6]), combined with techniques from geometric and topological inference, to adapt classical techniques such as nearest neighbors methods into the hypergraph settings. More precise, for a given hypergraph data, we introduce a class of metrics modulo certain equivalence relations (for a precise definition, see Section 2) to equip such hypergraph with a metric space structure. Using these structures, we propose several approaches to detect features from hypergraphs; for example, only using distance matrix approach, we provide a way to approach to community detection problem. Based on the metric space structure, we apply tools from algebraic topology to propose a method for detecting *persistent features* arising from homological structures hidden in a hypergraph. Such approach has an advantage of visualization of the space structure of data which provides a visual insight into the topological structure of hypegraph data. Also based on the metric space structure of hypergraph data, we introduce a modified nearest neighbors method which is a generalization of the classical nearest neighbors method from machine learning. Using our modified nearest neighbors methods, we apply to sign prediction problem on hypegraph data constructed by our method.

One of the novel and main features in our paper is that we propose a new type of hypergraph data (which we coin the term "*congruence hypergraph data*") which are very easy to construct and implement, and very flexible for testing our theories.

The structure of our paper is as follows. In Section 2, we introduce several notions and our main metric on hypergraphs that will be used throughout the paper. In Section 3, we introduce congruence hypergraph data, and several methods for analyzing hypergraph data including the distance matrix approach, homology-based learning, and modified nearest neighbors methods. Several examples will be performed on congruence hypergraph data which we introduce in Subsection 3.2.

2. Basic notions

2.1. Metrics modulo equivalence relations

Let X be a set. An *equivalence relation*, denoted by $\cong$, on X is a subset of $X \times X$ such that the following conditions are true:

(i) (**Reflexivity**) $(a,a) \in \cong$ for every $a \in X$.
(ii) (**Symmetry**) $(a,b) \in \cong$ if and only if $(b,a) \in \cong$.
(iii) (**Transitivity**) if $(a,b) \in \cong$ and $(b,c) \in \cong$ then $(a,c) \in \cong$.

When $(a,b) \in \cong$, we say that a is $\cong$–equivalent to b. Throughout this paper, in order to signify this relation, we write $a \cong b$ whenever $(a,b) \in \cong$.

For a given high order network (which is another terminology for hypergraph data), one of the problems that we address in this paper is concerned with *distinguishing communities* in the network. It is clear that there are many examples of networks in which several communities are viewed as identical communities with respect to certain properties that one wants to know about these networks. So if we view a given high order network X as a hypergraph, in order to use a metric geometry approach to the community detection problem, it is natural to introduce a metric (or distance) on X modulo a certain equivalence relation which will be explicitly introduced depending on the structure of X. Before making it clear what exactly we mean by this point of view, using an example of high order collaboration network, we first introduce the notion of a metric modulo an equivalence relation.

Definition 2.1 *Let X be a set, and let $\cong$ be an equivalence relation on X. A mapping $d : X \times X \to \mathbb{R}$ is said to be a metric on X modulo the equivalence relation $\cong$ if the following conditions are satisfied:*

(i) $d(a,b) \geq 0$ *for all* $a,b \in X$.
(ii) $d(a,b) = 0$ *if and only if* $a \cong b$.
(iii) ***(Symmetry)*** $d(a,b) = d(b,a)$ *for all* $a,b \in X$.
(iv) ***(Triangle inequality)*** *for any* $a,b,c \in X$,

$$d(a,b) \leq d(a,c) + d(c,b).$$

A set X equipped with a metric modulo an equivalence relation $\cong$, say $d : X \times X \to \mathbb{R}$ is called a metric space modulo $\cong$. In notation, we write (X,d) to indicate this metric space modulo $\cong$.

2.2. Hypergraphs equipped with intrinsic properties

Let X be a set. In order to create a *hypergraph* structure on X, we view the set of all points in X as the set of vertices $\mathscr{V}(X)$, and one needs to identify the relations among points in X, which one can view as the set of hyperedges of X, denoted as $\mathscr{E}(X)$. A hyperedge having exactly ℓ vertices is called an ℓ-hyperedge. The way which one identifies hyperedges in X, signifies certain properties pertained to the set X that we want to study. For example, let X be a set of mathematicians. In order to study how collaborative the mathematicians in X are, we introduce a hypergraph structure on X as follows. The set of vertices of X simply consists of all mathematicians in X. A group of mathematicians, say $m_1, \ldots, m_\ell$ in X forms an ℓ-hyperedge if they have at least one joint publication. In this way, the set X becomes a hypergraph in which the construction of hyperedges signifies the collaboration among mathematicians in X.

In many real data examples, one is not only interested in the hypergraph structure of X, but also in knowing certain properties attached to such structure but hidden in the hyperedge data. For example, in the collaboration network just described, in order to study in which areas of mathematics the mathematicians in X have joint publications, we can associate to each hyperedge the *main area of mathematics* in which the joint publication of the hyperedge belongs. If an ℓ-hyperedge $\mathfrak{e}$ is formed out of the joint paper, say P, of ℓ mathematicians $m_1, \ldots, m_\ell$, and the paper P is mainly concerned about number theory, then one can define $\Gamma(\mathfrak{e}) =$ number theory. Hence one obtains a mapping Γ from the set of hyperedges of X to the set of all areas in mathematics. Studying such a map Γ provides insight into the relationships between joint publications of mathematicians in X and their contributions to certain fields in mathematics. Motivated by this example in mind, we introduce a notion of hypergraphs equipped with certain properties.

Definition 2.2 *Let $X = (\mathscr{V}(X), \mathscr{E}(X))$ be a hypergraph, and let $\mathscr{P}$ be a nonempty set. The hypergraph X is called a hypergraph equipped with properties $\mathscr{P}$ if there is a map $\Gamma : \mathscr{E}(X) \to \mathscr{P}$ which associates each hyperedge in X to a unique element in $\mathscr{P}$.*

In notation, we write $\{X, \mathscr{P}\}_\Gamma$ to indicate X is a hypergraph equipped with properties $\mathscr{P}$, where the subscript Γ is a map from $\mathscr{E}(X)$ to $\mathscr{P}$.

In this subsection, for each hypergraph equipped with properties $\mathscr{P}$, say $\{X, \mathscr{P}\}_\Gamma$, we introduce a metric space structure on X, which provides a way to distinguishing communities in X. We begin by defining a notion of neighborhood of a vertex which is more suitable for defining a metric on the hypergraph X.

Definition 2.3 *Let $X = (\mathscr{V}(X), \mathscr{E}(X))$ be a hypergraph. Let a be a vertex in X. The neighborhood of a, denoted by $\mathscr{N}(a)$, is defined by*

$$\mathscr{N}(a) = \{a\} \cup \{b \in \mathscr{V}(X) \mid \exists \mathfrak{e} \in \mathscr{E}(X) \text{ such that } \{a, b\} \subset \mathfrak{e}\ \}.$$

The next result is clear from the above definition.

Proposition 2.4 *Let X be a hypergraph whose set of vertices consists of $a_1, \ldots, a_n$. Then X can be decomposed into n neighborhoods, say $\mathscr{N}(a_1), \ldots, \mathscr{N}(a_n)$ of the form*

$$X = \mathscr{N}(a_1) \cup \cdots \cup \mathscr{N}(a_n).$$

Remark 2.5 *For the community detection problem, the proposition above plays a key role. Indeed, by communities in X, we mean neighborhoods of each vertex. And thus in order to point out differences among communities, we are interested in finding out the exact differences among the populations of hyperegdes with specific properties in $\mathscr{P}$, contained in these neighborhoods; more precisely, letting a property P range over the set $\mathscr{P}$, the differences between the neighborhoods of a_i and a_j are reflected in terms of the differences between the numbers of hyperedges contained in the neighborhoods of a_i and a_j whose values under the map Γ is exactly P, i.e., they share the same property P. Because of this observation and the proposition above, we want to study neighborhoods of vertices in X instead of the vertices themselves, and thus one views X as a space whose points are neighborhoods $\mathscr{N}(a_i)$. So each neighborhood is in fact viewed as a single point in the space X.*

The metric geometry approach we use in this paper is that we want to construct a metric d on such a space X which should incorporate information about the number of hyperedges in X with specific properties P. And once such a metric is established for the space X, two neighborhoods $\mathscr{N}(a_i), \mathscr{N}(a_j)$ (i.e., two points in X) are different if and only if $d(\mathscr{N}(a_i), \mathscr{N}(a_j))$ is nonzero. And this is our first method for distinguishing communities in hypergraphs.

Let $\{X, \mathscr{P}\}_\Gamma$ be a hypergraph equipped with properties $\mathscr{P}$. Suppose that the set of vertices in X consists of $a_1, \ldots, a_n$, and the largest size of hyperedges in X is ℓ. Let $1 \leq i \leq n$ be an integer. For each property $P \in \mathscr{P}$, let $\mathfrak{C}_P^1(a_i)$ be the number of 1-hyperedges in $\mathscr{N}(a_i)$ whose value under the map Γ is P. In a similar manner, let $\mathfrak{C}_P^2(a_i)$ be the number of 2-hyperedges in $\mathscr{N}(a_i)$ whose value under the map Γ is P. In general, for any integer $1 \leq m \leq \ell$, let $\mathfrak{C}_P^m(a_i)$ be the number of m-hyperedges in $\mathscr{N}(a_i)$ whose value under the map Γ is P. Thus one obtains a unique double sequence $((\mathfrak{C}_P^m(a_i))_{1 \leq m \leq \ell})_{P \in \mathscr{P}}$ of non-negative real numbers for each neighborhood $\mathscr{N}(a_i)$.

We introduce an equivalence relation on the space X which allows to identify certain points in X. Note that if two points, say $\mathscr{N}(a_i)$ and $\mathscr{N}(a_j)$ have the same double sequence $((\mathfrak{C}_P^m(a_i))_{1 \leq m \leq \ell})_{P \in \mathscr{P}} = ((\mathfrak{C}_P^m(a_j)_{1 \leq m \leq \ell})_{P \in \mathscr{P}}$, then it is natural to view both of them as *identical points* in X since their hyperdege structures are exactly the same with respect to the map Γ and the properties $\mathscr{P}$. Hence it is natural to define a binary relation on X as follows: *two points $\mathscr{N}(a_i)$ and $\mathscr{N}(a_j)$ are equivalent, denoted by $\mathscr{N}(a_i) \cong \mathscr{N}(a_j)$ if their associated sequences $((\mathfrak{C}_P^m(a_i))_{1 \leq m \leq \ell})_{P \in \mathscr{P}}$, $((\mathfrak{C}_P^m(a_j))_{1 \leq m \leq \ell})_{P \in \mathscr{P}}$ are identical, i.e., $\mathfrak{C}_P^m(a_i) = \mathfrak{C}_P^m(a_j)$ for all $1 \leq m \leq \ell$ and all $P \in \mathscr{P}$*. One obtains the following.

Proposition 2.6 *The binary relation "$\cong$" is an equivalence relation.*

For the rest of this paper, whenever we use the symbol $\cong$ on hypergraphs, we mean the equivalence relation "$\cong$" in the proposition above.

Now we define a mapping $\mathscr{D} : X \times X \to \mathbb{R}_{\geq 0}$ as follows. For $1 \leq i, j \leq n$, define

$$\mathscr{D}(\mathscr{N}(a_i), \mathscr{N}(a_j)) := \sum_{P \in \mathscr{P}} \sum_{m=1}^{\ell} |\mathfrak{C}_P^m(a_i) - \mathfrak{C}_P^m(a_j)|.$$

From the above definition, we obtain the following.

Theorem 2.7 *The mapping $\mathscr{D}$ defined above is a metric on X modulo the equivalence relation $\cong$.*

The proof of the above theorem will be given in the appendix.

Remark 2.8 *In [7, Definition 7, p.1060], Leontjeva et al. constructed a distance function (or metric) between hypergraphs which uses the sizes of hyperedges in hypergraphs, in contrast to our construction using the number of hyperedges of each size. Note that in [7], Leontjeva et al. claimed their metric is the metric in the usual sense which is not correct. It is in fact a metric modulo an equivalence relation.*

3. Analysis for hypergraph data

In this section, we propose several methods for analyzing hypergraph data. Instead of using real data examples as in most papers studying data structures in literature, we introduce in this paper a new type of data which is inspired from elementary number theory (or more precisely from the theory of congruences in number theory), and is extremely easy to construct. There are many advantages of using such data which can also be viewed as hypergraphs. For simplicity, we call such data *congruence hypergraph data*. Firstly, these data are very easy to construct, simply using congruences in the ring of integers $\mathbb{Z}$. Secondly, congruence hypergraph data are very *diverse* and *random*, which provide a reasonably fine data to immediately test theory without referring to other data resources which in turn take a huge amount of time to build. The *randomness* of congruence hypergraph data allows to justify with high probability that any theory used to successfully test on such data can be also applied to real data examples. Lastly, for congruence hypergraph data, we can easily control the size of data. On letting the data size go to infinity, one can detect patterns hidden in the data which are often not available and straightforward if the data size is only limited to be finite.

3.1. Congruence hypergraph data

We now describe congruence hypergraph data which relies on the theory of congruences in the ring of integers $\mathbb{Z}$.

Let n be a positive integer, and $\{m_1, \ldots, m_n\}$ be a collection of positive integers. Take n collections of integers, say $\{a_{i,1}, \ldots, a_{i,m_i}\}$ for each $1 \leq i \leq n$ such that

$$\{a_{i,1}, \ldots, a_{i,m_i}\} \bigcap \{a_{j,1}, \ldots, a_{j,m_j}\} = \emptyset$$

for any $i \neq j$.

Consider n sets of integers, say $V_i = \{a_{i,1}, \ldots, a_{i,m_i}\}$ for each $1 \leq i \leq n$ so that $\#V_i = m_i$. We want to introduce a hypergraph structure on each V_i, and thus the set $X = V_1 \cup V_2 \cup \cdots \cup V_n$ becomes a hypergraph which is a disjoint union of subhypergraphs V_i.

Now take an integer $1 \leq i \leq n$. Let s_i be an integer such that $2 \leq s_i \leq m_i$. We want to introduce a hypergraph structure on V_i such that the largest size of hyperedges in V_i is s_i.

Let $\{p_{2,i}, \ldots, p_{s_i,i}\}$ be a sequence of integers such that the $p_{j,i}$ are ≥ 2 and not necessarily distinct. Correspondingly we choose a sequence of finite sets of integers $\{S_{2,i}, \ldots, S_{s_i,i}\}$ for each $1 \leq i \leq r$.

Let k be an integer such that $2 \leq k \leq s_i$. A k-tuple of integers $\{\alpha_1, \ldots, \alpha_k\}$ in V_i forms a k-hyperedge if the following conditions are satisfied:

(i) $\alpha_s - \alpha_r \equiv 0 \pmod{p_{k.i}}$ for any $1 \leq s, r \leq k$.
(ii) $\alpha_s \pmod{p_{k,i}}$ belongs in $S_{k,i}$ for any $1 \leq s \leq k$.

So we have obtained a subhypergraph structure for each of the V_i, and thus $X = \cup_{i=1}^{n} V_i$ is a hypergraph which splits into disjoint subhypergraphs. Note that X has exactly $m_1 + m_2 + \cdots + m_n$ vertices.

3.2. Main example

The hypergrah data we use to test our proposed methods in this paper is motivated from the construction of congruence hypergraph data in Subsection 3.1. We now describe two hypergraphs that we use throughout this work.

3.2.1. First example

Let $X = \{1, \ldots, 1000\}$. We introduce a hypergraph structure on X as follows. A pair $\{a, b\}$ in X forms a 2-hyperedge if and only if either $a, b \equiv 1 \pmod 2$ or $a, b \equiv 0 \pmod 2$. In other words, a, b have the same parity. Now for each $3 \leq n \leq 9$, an n-tuple $\{a_1, \ldots, a_n\}$ forms an n-hyperedge if and only if

$$a_i \pmod n = \begin{cases} 0 & \text{if } n \equiv 0 \pmod 3 \\ 1 & \text{if } n \equiv 1 \pmod 3 \\ 2 & \text{if } n \equiv 2 \pmod 3 \end{cases}$$

for every $1 \leq i \leq n$.

Since this data is about integers, we are interested in properties regarding integers such as divisibility. For this reason, we study, for example, the divisibility by 11 of each vertex in a hyperedge in X. So it is natural to define a map $\Gamma : \mathscr{E}(X) \to \{0, 1\}$ by letting, for each n-hyperedge $\{a_1, \ldots, a_n\}$ in $\mathscr{E}(X)$,

$$\Gamma(\{a_1, \ldots, a_n\}) = 1$$

if

$$a_i \equiv 0 \pmod{11} \tag{1}$$

for every $1 \leq i \leq n$, and

$$\Gamma(\{a_1, \ldots, a_n\}) = 0$$

if condition (1) is not satisfied.

The hypergraph X above has very large number of hyperedges. Up to our knowledge, comparing with real data examples in literature, the hypergraph data X above contains the *largest* number of hyperedges which is very suitable for testing theories. For example, the number of hyperedges in the neighborhood (or community) of the vertex 1 is approximately 2.3685×10^{11}.

3.2.2. Second example

Let X be a set of integers obtained by randomly selecting 5000 positive integers. The sizes of hyperedges range from 2 to 9. We randomly select 8 integers, say $\{\alpha_2, \ldots, \alpha_9\}$, such that any two of them have no common divisors. The set of vertices X are sorted in increasing order. For each $2 \leq n \leq 9$, we divide X into n subsets. The first subset, say X_1, contains vertices a in X such that $\min(X) \leq a \leq p_1$, where $\min(X)$ is the minimum value of X and p_1 is the $1/n$-th percentile of X. For each $2 \leq j \leq n$, the j-th subset, say X_j, contains vertices a in X such that $p_j < a \leq p_{j+1}$, where p_j is the j/n-th percentile of X. An n-tuple $\{a_1, \ldots, a_n\}$ forms an n-hyperedge if and only if

$$a_j \equiv 1 \pmod{\alpha_n} \text{ and } a_j \in X_j$$

for every $1 \leq j \leq n$.

Then we randomly select an odd prime number β that does not divide any elements in $\{\alpha_2, \ldots, \alpha_9\}$. The congruence classes modulo β are divided into two sets, the first of which consists of $\{-(\beta-1)/2, -(\beta-1)/2+1, \ldots, -1, 0\}$, and the second of which consists of $\{1, 2, \ldots, (\beta-1)/2\}$. Set

$$S_\beta^- = \{-(\beta-1)/2, -(\beta-1)/2+1, \ldots, -1, 0\},$$

and

$$S_\beta^+ = \{1, 2, \ldots, (\beta-1)/2\}.$$

The properties of hyperedges are defined as follows. For each n-hyperedge $\{a_1, \ldots, a_n\}$ in $\mathscr{E}(X)$,

$$\Gamma(\{a_1, \ldots, a_n\}) = -1$$

if every a_i modulo β belongs to S_β^-, and

$$\Gamma(\{a_1, \ldots, a_n\}) = 1$$

otherwise.

3.3. Using patterns from the distance matrices to recognize patterns in hypergraph data

In this subsection, we describe a simple but useful approach to detecting communities in hypergraphs. Using this approach, one can identify which communities in a hypergraph are the same with respect to the equivalence relation "$\cong$" and the metric $\mathscr{D}$ in Subsection 2.2. On the other hand, one can also find patterns among vertices whose neighborhoods (i.e., communities) are identified as the same.

Let $\{X, \mathscr{P}\}_\Gamma$ be a hypergraph equipped with properties $\mathscr{P}$. Assume that the set of vertices in X consists of $a_1, \ldots, a_n$. Hence there are exactly n neighborhoods (or communities), say $\mathscr{N}(a_1), \ldots, \mathscr{N}(a_n)$ which as remarked in Remark 2.5 can be viewed as points in the space X. Using the metric $\mathscr{D}$, we equipped X with a metric space structure in which each community $\mathscr{N}(a_i)$ is a point of the metric space X. Since there are exactly n points $\mathscr{N}(a_1), \ldots, \mathscr{N}(a_n)$ in the metric space, one obtains the *distance matrix* of the finite metric space X, say $\mathscr{M}_X$ of dimensions $n \times n$ of the form

$$\mathscr{M}_X = (\mathscr{D}(\mathscr{N}(a_i), \mathscr{N}(a_j)))_{1 \leq i,j \leq n},$$

where the (i, j)-entry in this matrix is the value $\mathscr{D}(\mathscr{N}(a_i), \mathscr{N}(a_j))$.

In order to identify which communities are the same in the hypergraph X, we identify all zero entries in $\mathscr{M}_X$ except the diagonal. More precisely, let $1 \leq i \leq n$, and consider the i-th column in $\mathscr{M}_X$. Define

$$Z_i = \{1 \leq j \leq n \mid j \neq i \text{ and } \mathscr{D}(\mathscr{N}(a_i), \mathscr{N}(a_j)) = 0\}.$$

Then the set Z_i consists of all vertices j whose communities $\mathscr{N}(a_j)$ are considered to be the same as the community $\mathscr{N}(a_i)$. It is often the case that one can find patterns to describe Z_i.

We use the hypergraph data in section 3.2.1. In this case X is a hypergraph whose vertices are $1, 2, \ldots, 1000$. Thus the distance matrix $\mathscr{M}_X$ is of dimensions 1000×1000. For example, considering the 1st column of $\mathscr{M}_X$, we see that Z_1 contains exactly the following vertices: 29, 43, 71, 85, 113, 155, 169, 211, 239, 253, 281, 295, 323, 365, 379, 421, 449, 463, 491, 505, 533, 575, 589, 631, 659, 673, 701, 715, 743, 785, 799, 841, 869, 883, 911, 925, 953, 995. And thus the communities (or neighborhoods) of these vertices are viewed as the same as that of the vertex 1.

From the list of vertices in Z_1, one can recognize the patterns shared by the vertices in Z_1. Indeed all vertices j in Z_1 satisfy the following four conditions: (i) $j \not\equiv 0 \pmod 3$; (ii) $j \not\equiv 2 \pmod 5$; (iii) $j \equiv \pm 1 \pmod 4$; and (iv) $j \equiv 1 \pmod 7$.

From the distance matrix $\mathscr{M}_X$, one also can identify the set of all distinct communities in X consisting of the neighborhoods of 1, 2, 3, 4, 5, 6, 7, 8, 12, 15, 22, 27,36, 57, 127, and 162 such that every community in X is equal to exactly one of these neighborhoods.

3.4. Homology-based learning using the metric $\mathscr{D}$

In this subsection, we use the *persistent homology* of filtrations of simplicial complexes arising from a finite metric space modulo an equivalence relation $\cong X$ to study the community detection problem. For simplicity, in this subsection, we simply call X a metric space instead of a metric space modulo $\cong$. Let $\mathscr{V} = \{a_1, \ldots, a_n\}$ be a finite set. A *simplicial complex* X with vertex set $\mathscr{V}$ is a set of finite subsets of $\mathscr{V}$ satisfying the following conditions:

(i) every element in $\mathscr{V}$ belongs to X;
(ii) if $\tau \in X$ and $\sigma \subseteq \tau$, then $\sigma \in X$.

The elements of X are called the *simplices* of X. If a simplex σ has exactly $k+1$ elements, the *dimension* of σ is k, and we call σ a k-simplex.

To each simplicial complex X one can associate a unique sequence of *homology groups* $(H_k(X))_{k \geq 0}$ which contains information about topological and geometric properties of X. (See, for example, [8] or [9] for a notion of homology groups and their properties.)

Now we describe how to use homology groups to identify distinct communities in hypergraphs. Let $\{X, \mathscr{P}\}_\Gamma$ be a hypergraph equipped with properties $\mathscr{P}$, and suppose that the set of vertices of X consists of the vertices $a_1, \ldots, a_n$. We equip X with the metric $\mathscr{D}$ in Subsection 2.2.

We introduce a method to attach to the finite metric space X a collection of simplexes which one in turn can obtain the corresponding *persistent homology sequences* and their *barcodes* (see [10], [11], [12], [13], [14] for persistent homology and barcodes.) We first recall a notion of Vietoris–Rips complex.

Definition 3.1 *Let $\varepsilon > 0$, and let X be a finite metric space with metric $\mathscr{D}$. The Vietoris–Rip complex, denoted by $\mathscr{V}\mathscr{R}(X, \varepsilon)$ is defined by the following condition: a $k+1$-tuple $\{x_0, \ldots, x_k\}$ forms a k-simplex in $\mathscr{V}\mathscr{R}(X, \varepsilon)$ if and only if $\mathscr{D}(x_i, x_j) \leq \varepsilon$ for all i, j.*

Let h be a sufficiently large positive integer, and let $1 \leq n_1 < n_2 < \cdots < n_h = n$ be a collection of positive integers. For each $1 \leq k \leq h$, define

$$X_k = \{a_1, \ldots, a_{n_k}\}.$$

Note that $X_k \subset X$ for all $1 \leq k \leq h$, and thus each X_k is a metric space with the same metric $\mathscr{D}$. We have a filtration of metric spaces

$$X_1 \subset X_2 \subset \cdots \subset X_h = X.$$

For each finite metric space X_k, one obtains a filtration of Vietoris–Rip complexes $\mathscr{V}\mathscr{R}(X_k)$ from which one obtains the *barcode* containing the topological and geometric information about X_k. The key observation using homology-based learning is that when k ranges from 1 to h, the barcodes of dimension 0 will stabilize to have exactly m bars, which signifies that *there are exactly m distinct communities* in the hypergraph X. Furthermore when considering the barcodes of dimension 1, they will stabilize to have very similar forms when k approaches to h.

We illustrate the above method by testing this theory on sub-hypergraph datasets of the congruence hypergraphs defined in the first and second examples in Subsections 3.2.1 and 3.2.2. Note that for computing barcodes, we use the package **TDAstats** in R (see [15]). For the first example, let $h = 3$, and for each $0 \leq k \leq h$, set

$$X_k = \{1, \ldots, (2k+1)100\},$$

and $X_3 = \{1, \ldots, 700\}$. One obtains exactly 4 barcodes, each of which corresponds to exactly one X_k.

In the barcode of X_0 (see Fig. 1a), we observe that the barcodes of dimension 0 (the blue barcodes) have exactly 13 bars; so there are 13 distinct communities in X_0. For the barcodes of X_1, X_2, X_3 (see Fig. 1b, 1c, 1d), we note that all barcodes of dimension 0 have exactly 15 bars (which is stabilized), and thus since X_3 is the last finite metric space in the filtration

$$X_0 \subset X_1 \subset X_2 \subset X_3,$$

we deduce that there are exactly 15 distinct communities. This result agrees with the one in Subsection 3.3.

On the other hand, note that in the barcode of dimension 0 of X_0, there are 13 bars, and in the barcode of dimension 0 of X_1, there are 15 bars. Since $X_0 = \{1, \ldots, 100\}$, and $X_1 = \{1, \ldots, 300\}$, we conclude that out of 15 distinct communities in X_3, 13 of them are communities of vertices in X_0, and 2 of them belong to the communities in X_1.

Note that one can also study barcodes of dimension 1 of the metric spaces X_k, and observe that the barcodes of dimension 1 of X_1, X_2 and X_3 have exactly 5 important bars, and the remaining bars are *noises*. All these 5 bars have similar patterns although the number of vertices in X_k changes when k varies from 0 to 3. Using persistent homology, one can also realize geometric properties of each X_k, for example, how *connected* these spaces are.

For the second example, we choose the 8 integers $\alpha_2, \ldots, \alpha_9$ to be 3, 4, 5, 7, 11, 13, 17, 19, respectively, and the prime number $\beta = 23$. We randomly select 700 integers from $\{1,\ldots,1000\}$ and sorted in increasing order, denoted as Y_3. Let $h = 3$, for each $0 \leq k < h$, set Y_k to be the first $(2k+1)100$ integers in Y_3. Thus we have a filtration of metric spaces

$$Y_0 \subset Y_1 \subset Y_2 \subset Y_3.$$

Fig. 2 presents the 4 barcodes, each of which corresponds to one Y_k. Note that the barcodes of Y_0 to Y_3 have similar patterns in both of dimension 0 and dimension 1. As the number of vertices increases, the barcodes become stabilized.

An important remark is that when comparing the barcodes of the X and Y, for example, in dimension 0, the barcodes of the Y is very connected, which indicates that communities in Y are closely related to each other. This can be seen by observing that each integer can fall into congruence classes of different moduli 3, 4, 5, 7, 11, 13, 17, 19. In contrast, in order to define hyperedges of X in the first example, the conditions depend on certain congruence classes modulo 3, and thus the communities of X are decomposed into distinct connected components which are related to congruence classes modulo 3 in some way.

3.5. Hypergraph-based learning using the metric $\mathscr{D}$ and nearest neighborhoods

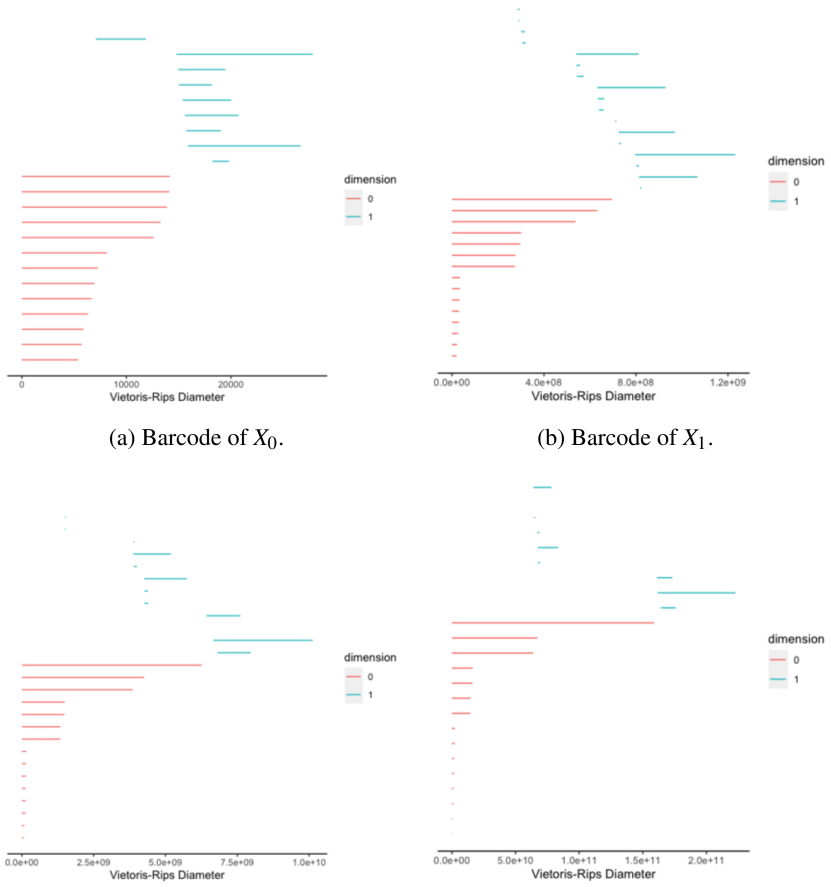
In hypergraph-based learning, for a given hypergraph, the aim is to find the correct labels for the unlabeled vertices of the test set in the hypergraph under the assumption that one knows the correct labels for the training set. In this subsection, we introduce a modification of the nearest neighbors methods to learn the objective function for a hypergraph. (for the classical nearest neighbors methods, see, for example, in [16].)

Let $\{X, \mathscr{P}\}_\Gamma$ be a hypergraph equipped with properties $\mathscr{P}$. We equipped X with the metric $\mathscr{D}$ in Subsection 2.2 so that X becomes a finite metric space under the metric $\mathscr{D}$. Suppose that the set of vertices in X consists of $a_1,\ldots,a_n$. Let $f : \mathscr{V}(X) \to \{-1,1\}$ be the objective function of labels to be learned such that it sends each vertex to exactly one of the values -1 or 1. The values of f are also called *signs* of vertices. Let $T = \{(\alpha_i,\beta_i) \mid 1 \leq i \leq m\}$ for some positive integer $1 \leq m < n$. Here the α_i are vertices in X, and $\beta_i \in \{-1,1\}$ are the correct label of α_i, i.e., $\beta_i = f(\alpha_i)$ for each $1 \leq i \leq m$. Our aim is to find all values of $\mathscr{V}(X) \setminus \{\alpha_1,\ldots,\alpha_m\}$ under the objective function f, based on the training set T. The set $\mathscr{V}(X) \setminus \{\alpha_1,\ldots,\alpha_m\}$ is called the *test set*. For this purpose, we use the modified nearest neighbors to find a *predictive model* f_{NN} for f. Fix a positive integer $k \geq 1$. For each vertex a in X, we define the following two sets attached to k, denoted as $\mathrm{kNN}_1(a)$ and $\mathrm{kNN}_{\mathrm{all}}(a)$ as follows.

(i) $\mathrm{kNN}_1(a)$ is the set of k-th *nearest neighbors* of a in the training set T according to the metric $\mathscr{D}$. Note that if there are more than one vertex, say x,y in T such that $\mathscr{D}(a,x) = \mathscr{D}(a,y)$ and x,y are k-th nearest neighbors, then one picks up randomly exactly one such vertex to include in $\mathrm{kNN}_1(a)$.

(ii) $\mathrm{kNN}_{\mathrm{all}}(a)$ is the set of k-th *nearest neighbors* of a in the training set T according to the metric $\mathscr{D}$. Note that in this set, one includes all vertices x in T such that x is a k-th nearest neighbor of a.

Using the above two sets $\mathrm{kNN}_1(\cdot)$ and $\mathrm{kNN}_{\mathrm{all}}(\cdot)$, we propose two predictive models for f, denoted as f_{kNN_1} and $f_{\mathrm{kNN}_{\mathrm{all}}}$, respectively. We define



(a) Barcode of X_0.

(b) Barcode of X_1.

(c) Barcode of X_2.

(d) Barcode of X_3.

Figure 1. Barcodes of the first example.

(i) $f_{\mathrm{kNN}_1}(a) = \operatorname{sign}\left(\sum_{\alpha \in \mathrm{kNN}_1(a)} f(\alpha)\right)$ for each a in the test set.

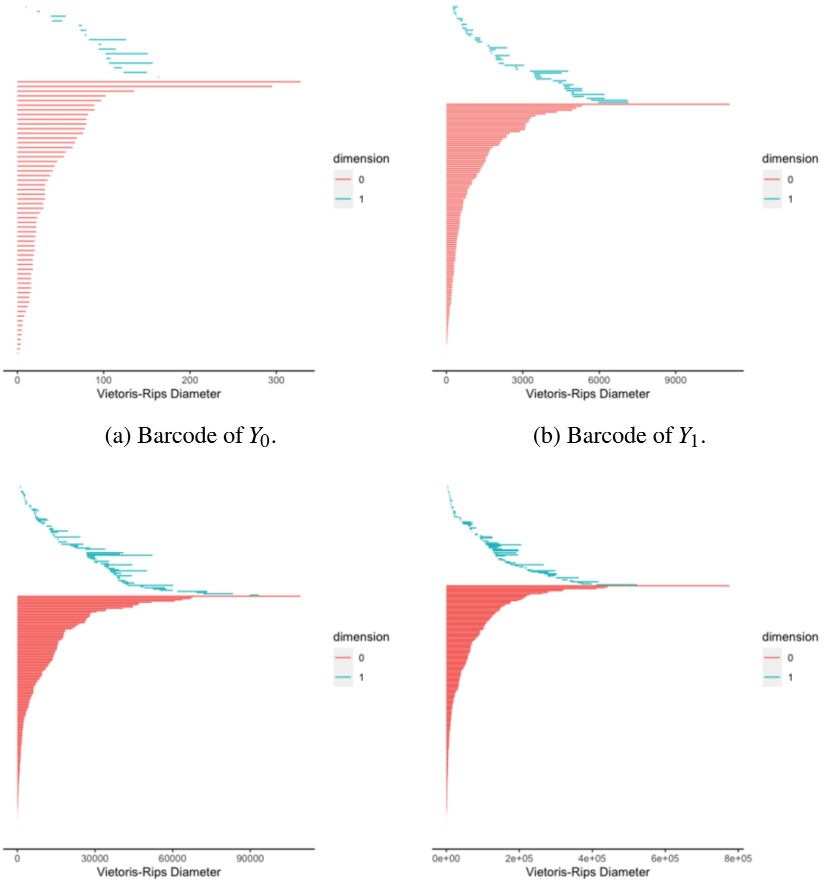
(ii) $f_{\mathrm{kNN}_{\mathrm{all}}}(a) = \operatorname{sign}\left(\sum_{\alpha \in \mathrm{kNN}_{\mathrm{all}}(a)} f(\alpha)\right)$ for each a in the test set.

Here the sign function is defined by

$$\operatorname{sign}(a) = \begin{cases} 1 & \text{if } a \geq 0 \\ -1 & \text{if } a < 0 \end{cases}$$

We illustrate our method by testing on the hypergraph datasets defined in Subsections 3.2.1 and 3.2.2. Here we define the objective function $f : \mathscr{V}(X) \to \{-1, 1\}$ as follows: $f(a) = 1$ if $a \equiv 0, 1 \pmod 3$, and $f(a) = -1$ if $a \equiv -1 \pmod 3$.

Table 1 contains the results of kNN using the congruence hypergraph defined in the first example, and we set $X_{2000} = \{1, \ldots, 2000\}$. The value of k for kNN are set to be 1 to 5. In each time, we randomly select 70% vertices from X to be the training set, and we repeat the computation 10 times for each k. Each element in the table presents an



(a) Barcode of Y_0.

(b) Barcode of Y_1.

(c) Barcode of Y_2.

(d) Barcode of Y_3.

Figure 2. Barcodes of the second example.

error rate which is computed by the percentage of incorrect predictions. According to the average error rates in Table 1, we obtain the smallest average error rate 0.2841 at $k = 3$ when using kNN$_{\text{all}}$ method and 0.2641 at $k = 2$ when using kNN$_1$ method. Figure 3 presents the curve comparison for the predicted and true signs for the method kNN$_{\text{all}}$. In this figure, the error rates of $f_{\text{kNN}_{\text{all}}}$ are from the ninth row in Table 1. According to the figure, most of vertices with label 1 are predicted correctly. One of the reasons that cause this result is that the number of vertices with positive sign are much larger then the number of vertices with negative sign according to the way we define the objective function.

Table 2 contains the results of kNN using the congruence hypergraph defined in the second example. We randomly select 5000 vertices from $\{1,\ldots,8000\}$, the values of $\{\alpha_2,\ldots,\alpha_9\}$ and β are the same as described in section 3.4. Using the kNN$_{\text{all}}$ method, the smallest average error rate is 0.3463 at $k = 5$. Using the kNN$_1$ method, the smallest average error rate is 0.3419 at $k = 2$. According to the results in Table 1 and 2, the kNN$_{\text{all}}$ method performs slightly better then kNN$_1$.

Table 1. Error rates of KNN using the first example

	Error rate of $f_{\mathrm{kNN}_{\mathrm{all}}}$					**Error rate of f_{kNN_1}**				
Error rate	**K=1**	**K=2**	**K=3**	**K=4**	**K=5**	**K=1**	**K=2**	**K=3**	**K=4**	**K=5**
1	0.3263	0.3200	0.3200	0.3200	0.3200	0.4762	0.2700	0.4012	0.2800	0.3775
2	0.3187	0.4613	0.1887	0.3925	0.4463	0.4712	0.2837	0.4712	0.2700	0.4225
3	0.2800	0.3163	0.2987	0.3888	0.3050	0.5100	0.2913	0.4975	0.3337	0.4525
4	0.3762	0.3225	0.3313	0.1850	0.1775	0.4087	0.2213	0.2813	0.1900	0.3075
5	0.3013	0.2925	0.2925	0.2925	0.3850	0.4225	0.2650	0.3812	0.2450	0.3800
6	0.4400	0.2538	0.3938	0.3775	0.4150	0.4437	0.2163	0.3888	0.2875	0.3938
7	0.1562	0.2675	0.2675	0.4225	0.3063	0.4587	0.2562	0.3975	0.2850	0.3800
8	0.3137	0.2762	0.2712	0.2712	0.2850	0.4625	0.2450	0.4050	0.2750	0.3938
9	0.4250	0.2638	0.1125	0.2438	0.2825	0.4675	0.2312	0.4663	0.2937	0.4287
10	0.4350	0.3900	0.3650	0.3275	0.3550	0.5075	0.2688	0.4525	0.3363	0.4300
Average	0.3372	0.3164	0.2841	0.3221	0.3278	0.4626	0.2641	0.4098	0.2866	0.3946

Table 2. Error rates of KNN using the second example

	Error rate of $f_{\mathrm{kNN}_{\mathrm{all}}}$					**Error rate of f_{kNN_1}**				
Error rate	**K=1**	**K=2**	**K=3**	**K=4**	**K=5**	**K=1**	**K=2**	**K=3**	**K=4**	**K=5**
1	0.3527	0.3447	0.3447	0.3440	0.3433	0.3713	0.3347	0.3687	0.3260	0.3500
2	0.3567	0.3413	0.3487	0.3347	0.3360	0.3493	0.3367	0.3660	0.3453	0.3527
3	0.3793	0.3693	0.3633	0.3433	0.3480	0.3913	0.3473	0.3933	0.3413	0.3540
4	0.3733	0.3580	0.3540	0.3500	0.3460	0.3707	0.3487	0.3813	0.3433	0.3580
5	0.4000	0.3660	0.3607	0.3627	0.3593	0.3960	0.3540	0.3807	0.3573	0.3780
6	0.3853	0.3580	0.3653	0.3453	0.3460	0.3873	0.3380	0.3893	0.3447	0.3680
7	0.3520	0.3347	0.3373	0.3493	0.3453	0.3680	0.3333	0.3720	0.3420	0.3520
8	0.3773	0.3587	0.3687	0.3547	0.3540	0.3627	0.3513	0.3847	0.3500	0.3567
9	0.3707	0.3500	0.3653	0.3453	0.3467	0.3807	0.3527	0.3800	0.3493	0.3640
10	0.3633	0.3427	0.3473	0.3360	0.3380	0.3707	0.3227	0.3613	0.3333	0.3540
Average	0.3715	0.3523	0.3555	0.3465	0.3463	0.3748	0.3419	0.3777	0.3460	0.3587

4. Conclusions

Our main contributions in this paper can be summarized as follows:

(i) Introducing a natural metric space structure modulo certain equivalence relations on a general hypergraph data which bears a resemblance to a usual metric space structure;

(ii) Using the metric space modulo certain equivalence relation structure introduced, we emphasize that this topological space structure on hypegraphs is very natural and suitable for studying several problems in machine learning;

(iii) Proposing a distance matrix approach using the metric space structure introduced in this paper to study community detection problem in hypegraphs;

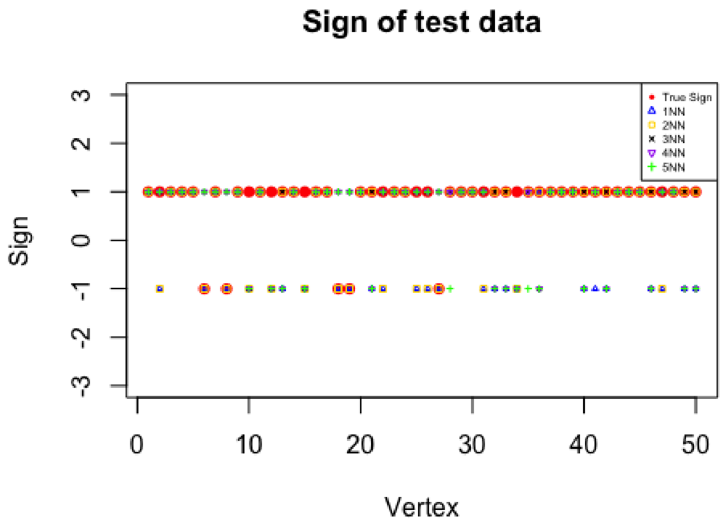


Figure 3. Predicted and true signs of test data for $\mathrm{kNN}_{\mathrm{all}}$.

(iv) Proposing a modified homology-based learning to study topological structures of hypergraphs which in turn can be used to detect persistent homological features; this method can also be used to study community detection problem;

(v) Proposing modified nearest neighbors methods for studying sign prediction problem on general hypergraph data; such methods have advantages that they can be applied even to hypergraphs which do not contain an embedding into a Euclidean space, or do not carry a Euclidean space structure.

(vi) One of our main contributions is to propose a new way to construct hypergraph data which are very easy to implement and test theories from machine learning which we coin the term "*congruence hypergraph data*".

(vii) Experimental analysis are performed on congruence hypergraph data which are simulated by our methods.

5. Acknowledgements

Lizhen Lin would like to acknowledge the support of NSF grant DMS CAREER 1654579.

6. Appendix

In this Appendix, we give a proof of Theorem 2.7

For the sake of simplicity, let $\alpha_i = \mathscr{N}(a_i)$ for each $1 \le i \le n$.

Suppose that $\mathscr{D}(\alpha_i, \alpha_j) = 0$ for some $1 \le i, j \le n$. By definition, we know that

$$\mathscr{D}(\alpha_i,\alpha_j)=\sum_{P\in\mathscr{P}}\sum_{m=1}^{\ell}|\mathfrak{C}_P^m(a_i)-\mathfrak{C}_P^m(a_j)|=0,$$

which implies that

$$\mathfrak{C}_P^m(a_i)-\mathfrak{C}_P^m(a_j)=0$$

for all $m\geq 1$ and $P\in\mathscr{P}$. Thus $\mathfrak{C}_P^m(a_i)=\mathfrak{C}_P^m(a_j)$ for all $m\geq 1$ and $P\in\mathscr{P}$, and hence $\alpha_i=\mathscr{N}(a_i)\cong\alpha_j=\mathscr{N}(a_j)$.

It is obvious that $\mathscr{D}(\alpha_i,\alpha_j)=\mathscr{D}(\alpha_j,\alpha_i)$ for all $1\leq i,j\leq n$, which proves that $\mathscr{D}$ is symmetric.

We now show that $\mathscr{D}$ satisfies the triangle inequality. Indeed, we see that

$$\begin{aligned}\mathscr{D}(\alpha_i,\alpha_j)&=\sum_{P\in\mathscr{P}}\sum_{m=1}^{\ell}|\mathfrak{C}_P^m(a_i)-\mathfrak{C}_P^m(a_j)|\\&=\sum_{P\in\mathscr{P}}\sum_{m=1}^{\ell}|((\mathfrak{C}_P^m(a_i)-\mathfrak{C}_P^m(a_k))+(\mathfrak{C}_P^m(a_k)-\mathfrak{C}_Q^m(a_j)|\\&\leq\sum_{P\in\mathscr{P}}\sum_{m=1}^{\ell}|\mathfrak{C}_P^m(a_i)-\mathfrak{C}_P^m(a_k)|+\sum_{P\in\mathscr{P}}\sum_{m=1}^{\ell}|\mathfrak{C}_P^m(a_k)-\mathfrak{C}_P^m(a_j)|\\&=\mathscr{D}(\alpha_i,\alpha_k)+\mathscr{D}(\alpha_k,\alpha_j)\end{aligned}$$

for any $1\leq i,j,k\leq n$. Thus $\mathscr{D}$ satisfies the triangle inequality, and therefore $\mathscr{D}$ is a metric on X modulo the equivalent relation $\cong$.

References

[1] Tian Z, Hwang T, Kuang R. A hypergraph-based learning algorithm for classifying gene expression and arrayCGH data with prior knowledge. Bioinformatics. 2009 Nov 1;25(21):2831–8.

[2] Levene M, Poulovassilis A. An object-oriented data model formalised through hypergraphs. Data & Knowledge Engineering. 1991 May 1;6(3):205-24.

[3] Goertzel B. Patterns, hypergraphs and embodied general intelligence. In: The 2006 IEEE international joint conference on neural network proceedings; 2006 Jul 16; pp. 451–458.

[4] Klamt S, Haus UU, Theis F. Hypergraphs and cellular networks. PLoS Comput Biol. 2009 May 29;5(5):e1000385.

[5] Kok S, Domingos P. Learning Markov logic network structure via hypergraph lifting. In: Proceedings of the 26th annual international conference on machine learning; 2009 Jun 14; p. 505-512.

[6] Burago D, Burago Y, Ivanov S. A course in metric geometry. American Mathematical Soc.; 2001.

[7] Leontjeva A, Konstantin T, Vilo J, Tamkivi T. Fraud Detection: Methods of Analysis for Hypergraph Data. In: The 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; August 2012; pp. 1060–1064.

[8] Edelsbrunner H, Harer J. Computational topology: an introduction. American Mathematical Soc.; 2010.

[9] Zhu X. Persistent homology: An introduction and a new text representation for natural language processing. In: IJCAI; 2013 Aug 3; p. 1953-1959.

[10] Boissonnat JD, Chazal F, Yvinec M. Geometric and topological inference. Cambridge University Press; 2018 Sep 27.

[11] Carlsson G. Topology and data. Bulletin of the American Mathematical Society. 2009;46(2):255-308.

[12] Carlsson G, Zomorodian A, Collins A, Guibas LJ. Persistence barcodes for shapes. International Journal of Shape Modeling. 2005 Dec;11(02):149-87.

[13] Collins A, Zomorodian A, Carlsson G, Guibas LJ. A barcode shape descriptor for curve point cloud data. Computers & Graphics. 2004 Dec 1;28(6):881-94.

[14] Zomorodian A, Carlsson G. Computing persistent homology. Discrete & Computational Geometry. 2005 Feb 1;33(2):249-74.

[15] Wadhwa RR, Williamson DF, Dhawan A, Scott JG. TDAstats: R pipeline for computing persistent homology in topological data analysis. Journal of open source software. 2018 Aug 8;3(28):860.

[16] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning. New York: Springer series in statistics; 2001.