Entropy-based graph construction methods for unsupervised data structure detection

Thesis for the Degree of
Doctor of Philosophy in Mathematical Engineering

Leandro Fabio Ariza Jiménez

Advisors:
Olga Lucía Quintero Montoya, PhD
Nicolás Pinel Peláez, PhD

Department of Mathematical Sciences
School of Sciences
Universidad EAFIT
Colombia

To my wife and children

Abstract

Data recorded from real-world phenomena or situations mostly consist of unlabeled observations. A current major challenge is to find simple representations that best uncover and portray the underlying structure of this kind of data in order to extract useful information or knowledge. In this regard, we attempt in this thesis to investigate whether the construction and use of graph-based models can offer a viable alternative to conventional unsupervised algorithms in the problem of data structure detection. To this end, we propose methods to characterize data objects within a given data set in terms of an entropy-based feature that is computed using together similarity measures and the binary entropy function, which in turn is based on Shannon's entropy. By analyzing such features, we can build a graph-based model upon the data set and whose nodes are samples of the data set, representing neighborhoods, that are identified in an unsupervised fashion. In this way, the problem of finding a hidden structure on the feature space where data objects are represented is converted into community detection in complex networks. In particular, the developed models are capable of helping or detecting by themselves the internal structure of high-dimensional and unlabeled data such as financial and banking data, metagenomic data, and spike-waveform-related data. Based on benchmarking data, we evidence that the proposed graph-based models are useful and effective for biological applications such as metagenomic binning and neuronal spike sorting, wherein it is required to organize data into unknown and meaningful groups. Our experimental results also suggest that our methods perform comparably to or better than some state-of-art methods in the aforementioned biological applications.

Acknowledgment

First and foremost, I am especially grateful to my wife and children for their extensive support and encouragement throughout my PhD studies. Together we faced the ups and downs of this stage of our lives and we were able to overcome this challenge. I also want to mention my parents, my parents-in-law, my younger brother, and brothers- and sisters-in-law for the help we receive from each of them.

I want to thank my advisors, Professors Olga Lucía Quintero and Nicolás Pinel, for their help and patience in guiding me through the preparation and completion of my doctorate. All the internal and external professors in the doctoral program also deserve my gratitude for their teachings. Among them, I would like to acknowledge Professor Andrés Sicard. His selfless contribution during the development of this thesis is unmatched. My gratitude also goes out to Professor John Hopcroft for the opportunities he provided me in the first years of my PhD.

Finally, I would like to thank my colleagues in the PhD program and at the Mathematical Modelling Research Group, especially Juan Guillermo Paniagua, Juan David Palacios, and Alejandro Montoya.

Contents

List of Figures

Chapter 1

Introduction

Contents

1.1 Problem statement

Data recorded from real-world phenomena or situations mostly consist of unlabeled observations (Chawla and Karakoulas, 2005), i.e. meaningful information about each piece of data is not available, since labeled data are difficult or expensive to obtain (Gokcay and Principe, 2002). Moreover, advances in sensing and storage technology have contributed to generate both high-volume and high-dimensional real-world data sets (Jain, 2010).

For instance, real-world data can consist of records describing the financial behavior of all the customers of a bank, during a given period of time, based on which the financial institution wants to estimate the risk and profitability profile for each client (Ariza-Jiménez et al., 2019). In metagenomics, which is the study of microbial communities sampled directly from their natural environment without prior culturing (Wooley and Ye, 2009), genomic data is obtained by means of modern sequencing technologies and analyzed without any prior information about the taxonomical identity of the microorganisms that are present in the sequenced sample (Ariza-Jiménez et al., 2018; Girotto et al., 2017).

A current major challenge is to find simple representations that best uncover and portray the underlying structure of real-world data. To this end, unsupervised methods can be used to learn the inner structure from data. However, due to the provenance and nature of this kind of data, conventional approaches proposed to accomplish this task rely on methods that make assumptions about data distribution (e.g., center-based clustering (Gan et al., 2007) and kernel density estimation (Geng and Hu, 2012)) and require the reduction of the original dimensionality of the input data (e.g., via principal component analysis (Manly

and Alberto, 2016) or *t*-Stochastic Neighbor Embedding (van der Maaten, 2014)).

In response to the above problems, this thesis attempts to investigate whether the construction and use of graph-based representations and models can offer a viable alternative to conventional unsupervised algorithms in the problem of understanding data. Particularly, we propose an approach for uncovering the underlying structure of data by modeling the local neighborhood relationships between the observations within a data set. To this end, we will look into the development of a framework based on graph theory and entropy-based measures to learn and model the inner structure of multidimensional and unlabeled data.

1.2 Goal of the thesis

The main goal of this thesis is to develop a framework based on graph theory and entropy measures to learn, model, and represent the underlying structure of multidimensional and unlabeled data, while the information retrieved from the data is maximized.

To reach the above goal we aim to:

- Study, implement, and evaluate methods for measuring the entropy of multidimensional data sets and graphs with potential application in the construction of graph-based data representations.

- Propose, implement, and evaluate an unsupervised entropy-based graph construction method to learn, model, and represent the inner structure of high-dimensional data.

- Propose, implement, and evaluate an unsupervised graph clustering method that exploits the graph-based data representation in order to organize the original data into unknown groups and uncover the underlying natural structure of data.

- Apply and evaluate the developed methods in practical problems and applications related with the unsupervised structure detection in high-dimensional and unlabeled data.

1.3 Structure of the thesis

This dissertation is organized into seven chapters. Apart from this introduction, the thesis consists of the following:

Chapter 2: Background

In this chapter, a relevant background about the discovering of natural groupings in data is discussed. We provide an overview of concepts, methods, and approaches regarding data clustering and graph clustering, which are two unsupervised learning techniques of central importance for this thesis. A literature review on the development of graph-based representations for multidimensional data is also provided since most of our contributions are related to this problem.

Chapter 3: On data exploration and visualization

This chapter looks into aspects related to data exploration and visualization. It presents an early graph-based method developed to gain insight into the inner structure of multivariate data through the visualization of fuzzy clustering results. The content of this chapter is based on the following publications:

- "*Memberships Networks for High-Dimensional Fuzzy Clustering Visualization*". L. Ariza-Jiménez, L.F. Villa, O.L. Quintero. Applied Computer Sciences in Engineering. WEA 2019. Communications in Computer and Information Science, vol 1052 (Ariza-Jiménez et al., 2020a).
- "*Extracted information quality, a comparative study in high and low dimensions*". L. Ariza-Jiménez, L.F. Villa, N. Pinel, O.L. Quintero. International Journal of Business Intelligence and Data Mining (Ariza-Jiménez et al., 2020b). Article accepted for publication.

Chapter 4: Entropy-based features and graph-based data models

In this chapter, we address the problem of constructing graph-based models for multidimensional data sets to facilitate the uncovering of their inner structure. In particular, we present an initial method that articulates the extraction of entropy-based features of objects within a data set with the identification of prototype-like objects based on which a compact graph-based model is built. We also present experiments conducted on synthetic data sets and data sets with biological origin to study the performance of the proposed method. These last experiments are related to real-world applications, such as metagenomic binning and neuronal spike sorting, in which data is organized into unknown and meaningful groups. The content of this chapter is based on the following publications:

- "*Unsupervised fuzzy binning of metagenomic sequence fragments on three-dimensional Barnes-Hut* t-*Stochastic Neighbor Embeddings*". L. Ariza-Jiménez, O.L. Quintero, N. Pinel. Proceedings of the 2018 40th International Conference of the IEEE Engineering in Medicine and Biology Society (Ariza-Jiménez et al., 2018).

- "*An Entropy-Based Graph Construction Method for Representing and Clustering Biological Data*". L. Ariza-Jiménez, N. Pinel, L.F. Villa, O.L. Quintero. Proceedings of the 2019 8th Latin American Conference on Biomedical Engineering (Ariza-Jiménez et al., 2020a).

- "*Standardized Approaches for Assessing Metagenomic Contig Binning Performance from Barnes-Hut* t-*Stochastic Neighbor Embeddings*". J. Ceballos, L. Ariza-Jiménez, N. Pinel. Proceedings of the 2019 8th Latin American Conference on Biomedical Engineering (Ceballos et al., 2020)

Chapter 5: Entropy-based node strength and graph-based data representations

In this chapter, we propose a restructuring of the former method of obtaining graph-based models for multi-dimensional data sets. The modifications introduced derive mainly from the use of secondary similarity metrics based on the concept of shared nearest neighbors between data objects rather than conventional measures. Diverse experiments are presented to study the performance of this alternative approach on both synthetic and real-world data sets. Besides this, a comparison between the restructured method and the former method is provided in the context of the aforementioned biological applications.

Chapter 6: Further aspects of the graph-based models

This chapter is dedicated to exploring some properties of the developed graph-based models that can be exploited to improve and complement the unsupervised detection of structure in data. Additionally, an analysis of the time complexity of some of the algorithms developed in this thesis is presented.

Chapter 7: Conclusions and future work

The conclusions, contributions, and directions of future work of the thesis are presented in this chapter.

Appendix A: Experiments on the natural language processing of Coronavirus literature

This appendix presents additional experiments and an application on the natural language processing of coronavirus-related scholarly articles.

Bibliography

Ariza-Jiménez, L., Pinel, N., Villa, L. F., and Quintero, O. L. (2020a). An Entropy-Based Graph Construction Method for Representing and Clustering Biological Data. In *VIII Latin American Conference on Biomedical Engineering and XLII National Conference on Biomedical Engineering*, pages 315–321, Cham. Springer International Publishing.

Ariza-Jiménez, L., Quintero, O., and Pinel, N. (2018). Unsupervised fuzzy binning of metagenomic sequence fragments on three-dimensional Barnes-Hut t-Stochastic Neighbor Embeddings. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1315–1318.

Ariza-Jiménez, L., Villa, L. F., Pinel, N., and Quintero, O. L. (2020b). Extracted information quality, a comparative study in high and low dimensions. *International Journal of Business Intelligence and Data Mining*, X(X):X.

Ariza-Jiménez, L., Villa, L. F., and Quintero, O. L. (2019). Memberships Networks for High-Dimensional Fuzzy Clustering Visualization. In *Applied Computer Sciences in Engineering*, pages 263–273, Cham. Springer International Publishing.

Ceballos, J., Ariza-Jiménez, L., and Pinel, N. (2020). Standardized Approaches for Assessing Metagenomic Contig Binning Performance from Barnes-Hut t-Stochastic Neighbor Embeddings. In *VIII Latin American Conference on Biomedical Engineering and XLII National Conference on Biomedical Engineering*, pages 761–768, Cham. Springer International Publishing.

Chawla, N. V. and Karakoulas, G. (2005). Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research*, 23:331–366.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Geng, X. and Hu, G. (2012). Unsupervised feature selection by kernel density estimation in wavelet-based spike sorting. *Biomedical Signal Processing and Control*, 7(2):112–117.

Girotto, S., Comin, M., and Pizzi, C. (2017). Binning metagenomic reads with probabilistic sequence signatures based on spaced seeds. *2017 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–8.

Gokcay, E. and Principe, J. C. (2002). Information theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):158–171.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666.

Manly, B. F. and Alberto, J. A. N. (2016). *Multivariate Statistical Methods: A Primer*. CRC Press, 4th edition.

van der Maaten, L. (2014). Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245.

Wooley, J. C. and Ye, Y. (2009). Metagenomics: Facts and artifacts, and computational challenges. *Journal of Computer Science and Technology*, 25(1):71–81.

Chapter 2

Background

Contents

2.1 Introduction

In this chapter, we discuss some fundamental concepts and methods used in this thesis, concerning the discovering of natural groupings in data, along with some previous work on obtaining graph-based representations for multidimensional data.

This chapter is structured as follows. Section 2.2 briefly introduces the concepts of learning from data and the unsupervised learning paradigm around which this thesis develops. Section 2.3 and Section 2.4 provide an overview of concepts, methods, and approaches regarding data clustering and graph clustering, respectively, which are two unsupervised learning techniques of central importance for this thesis. Section 2.5 presents a review of the literature on the development of graph-based representations for multidimensional data.

2.2 Learning from data

Learning from data is an approach used to solve problems whose solution is not possible by analytic means, but where data are available and based on them an empirical solution can be constructed (Abu-Mostafa et al., 2012).

Because the premise of this approach is versatile enough to be applied in diverse settings,

several schemes of learning have been developed. Among them, *supervised learning*, which is essentially a learning-by-examples approach (Hastie et al., 2009), is probably the most widely known. This thesis, by contrast, deals with the problem of discovering the underlying structure of input data, without relying on external information that can guide the learning process. In particular, this kind of learning-from-data approach is known as *unsupervised learning*.

So under the unsupervised learning scheme, algorithms must learn the underlying relationships or features from the available data on its own. In this regard, unsupervised learning algorithms that can be found in the literature revolve around one of the following general goals: 1) to reduce the data dimensionality; 2) to approximate the probability distribution of data; 3) or to discover natural structure in data.Since this thesis is concerned with this last goal, our coverage here will develop in that direction.

2.3 Data clustering

Data clustering is perhaps the most representative unsupervised learning technique. This technique comprehends unsupervised methods that aim to spontaneously find patterns and structure in input data (Abu-Mostafa et al., 2012), with little or no ground truth. To this purpose, such algorithms separate data into multiple groups, or *clusters*, based on an intrinsic characteristic or similarity (Jain, 2010). Hence, each resulting group is a collection of similar data items and those in different groups are less similar to each other.

Suppose $\mathbf{X}$ is a collection of data, or simply a *dataset*, made up of data objects, which in turn are a collection of measurements, features, or attributes. More formally, let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ be a data set of N t-dimensional data objects, where $\mathbf{x}_i = \left(x_{i1}, x_{i2}, \ldots, x_{it}\right) \in \mathbb{R}^t$. The similarity between data objects can be measured based on different distances functions, e.g., the Euclidean distance, which is here defined for two data objects $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathbf{X}$, as

$$\begin{aligned} D &: \mathbf{X} \times \mathbf{X} \to [0, \infty) \\ D\left(\mathbf{x}_i, \mathbf{x}_j\right) &\equiv \left(\sum_{l=1}^{t} \left(x_{il} - x_{jl}\right)^2 \right)^{1/2} . \end{aligned} \tag{2.1}$$

Then, the degree of similarity between the same pair of data objects $\mathbf{x}_i$ and $\mathbf{x}_j$, can be calculated, for instance, based on the widely used Gaussian neighborhood function

$$\begin{aligned} S &: \mathbf{X} \times \mathbf{X} \to [0, 1] \\ S\left(\mathbf{x}_i, \mathbf{x}_j\right) &= \exp\left(-\frac{D^2\left(\mathbf{x}_i, \mathbf{x}_j\right)}{2\sigma^2} \right), \end{aligned} \tag{2.2}$$

where σ is the function scale parameter. Other distance and similarity functions commonly used in clustering algorithms can be found elsewhere (see e.g., Xu and Wunsch (2010)).

Clustering approaches can be categorized in different ways. As in Akman et al. (2019), here we consider two broad forms of clustering, namely *hard clustering* and *soft clustering*. Both hard and soft clustering share the same goal: to group the data objects in $\mathbf{X}$ into a set of K clusters, $C = \{C_1, C_2, \ldots, C_K\}$, according to predefined criteria. However, they differ in a fundamental aspect. In hard clustering, each data object in $\mathbf{X}$ is assigned to a single cluster, while in soft clustering, a data object can belong to more than one cluster to some extent. Fuzzy clustering is another name for soft clustering, since, as its name suggests, it is based on notions of the fuzzy sets proposed by Zadeh (1965).

The clustering structure C obtained by either a hard or fuzzy clustering algorithm can be represented by a $K \times N$ matrix $U = [u_{ij}]$ (Gan et al., 2007). The matrix U, referred to as the *partition matrix*, is defined in such a way that it can reflect the characteristic features of each of the above resulting clustering structures. The entry u_{ij} describes how the data point $\mathbf{x}_j$ is assigned to the cluster C_i. In the hard clustering setting, u_{ij} equals one if $\mathbf{x}_j$ belongs to C_i and zero otherwise, thus $u_{ij} \in \{0, 1\}$. On the other hand, in fuzzy clustering, u_{ij} takes values between 0 and 1, that is, $u_{ij} \in [0, 1]$, to represent the grade of membership of $\mathbf{x}_j$ in cluster C_i. Regardless of which type of clustering is applied, the matrix U satisfies the following additional conditions:

1. $\sum_{j=1}^{N} u_{ij} > 0, \quad \text{for} \quad i = 1, \ldots, K;$

2. $\sum_{i=1}^{K} u_{ij} = 1, \quad \text{for} \quad j = 1, \ldots, N.$

In other words, no empty clusters are allowed in both hard and soft clustering, according to the first condition. On the other hand, the second condition indicates that clusters in hard clustering are disjoint, while it compels each data object, in soft clustering, to distribute all its membership among every single cluster.

Two traditional, and still effective, algorithms for hard and soft clustering are K-means and Fuzzy c-Means (FCM), respectively. Although they belong to opposite categories, they operate based on the common idea that clusters within data can be represented using single data objects. Such representative data objects, which are the centers of the clusters, are used to seek an optimal partition of the data by minimizing intra-cluster variance (Dougherty, 2013). Clustering algorithms following the above approach are also known as center-based algorithms and tend to work well on data sets having compact and globular clusters (Gan et al., 2007). K-means and FCM are popular because they are easy to implement and work well for many practical problems, even in large data sets (see e.g., Girotto et al. (2016); Quintero Montoya et al. (2015)). However, both algorithms have several major drawbacks. For instance, the user must provide the number of clusters in which the data set will be divided, the clustering results depend on the initial

working conditions, and there is no guarantee that both algorithms converge to the global optimum (Dougherty, 2013). To overcome one of the aforementioned drawbacks, a preliminary method such as the Subtractive algorithm (Chiu, 1994) can be used to estimate the cluster centers within a given data set (e.g., see Ariza-Jiménez et al. (2018)). In particular, FCM is considered the fuzzy counterpart of K-means, and it uses an additional input parameter m that controls the "fuzziness" of the resulting clusters (Xu and Wunsch, 2010). Additional details about these clustering algorithms and many others can be found elsewhere (see e.g., Gan et al. (2007)).

2.4 Graph clustering

It is customary to associate clustering with the process of structuring data objects into groups in a t-dimensional feature space. However, the unsupervised task of clustering can also be extended to graphs, which are mathematical models able to represent interactions or relationships between entities in real-world situations.

Before proceeding, we introduce some basic definitions and notations about graphs that will be used in the following chapters. A *graph* $G = (V, E)$ is formed by a set V of *vertices* and a set E of *edges*, each connecting pairs of vertices. Here both sets V and E are expected to be finite, so we denote by $V = \{v_1, v_2, \ldots, v_b\}$ the vertex set of G. Unless otherwise stated, the edge set E is defined as $E \subseteq [V]^2$, a set of two-element subsets of V. Therefore, E consists of unordered pairs of the form $\{v_i, v_j\}$, where $v_i, v_j \in V$. In this particular case, edges are considered to have no direction and thus the graph G is called *undirected*. Conversely, a graph is said to be *directed* if $E \subseteq V \times V$. In a directed graph, also known as *digraph*, each element of E is then an ordered pair (v_i, v_j), which represents an edge directed from $v_i \in V$ to $v_j \in V$. Furthermore, a graph is said to be *weighted* if a numerical value is assigned to each edge or to each vertex, otherwise it is considered *unweighted*. The vertices which are joined by an edge are called the *ends* of the edge, and they are said to be *incident* with the latter, and *vice versa*. The *degree* of the vertex v_i, denoted by $\deg(v_i)$, is equal to the number of edges of G incident with v_i.

Graph, vertex, and edge are terms habitually used in graph theory (Diestel, 2017). To refer to each of the above elements, the trio of network, node, and link is of widespread use in network science (Newman, 2018). Although both groups of terms tend to be interchanged in the scientific literature, there is a fundamental difference between the two terminologies (Barabási, Albert-László and Pósfai, 2016). The second trio is often used to refer to real-world systems, such as social networks, telecommunication networks, road networks, biological networks, and so on. On the other hand, the first trio is used to refer to the mathematical representation of the aforementioned networks. In the following chapters we will mainly use the term *graph*.

Graph clustering aims to group the vertices of a graph following certain criteria, without any prior knowledge about them, to gain insight into the graph's inner structure (Bedi and Sharma, 2016). In the context of network science, these groups of nodes are known as *communities* and the graph clustering task is called *community detection* (Newman, 2018). Although community detection is a central problem in network analysis, there is no general agreement on the definition of what a graph cluster or community is (Fortunato and Hric, 2016). Thus, it is a widespread practice that methods proposed for community detection address this problem by following its own explicitly or implicitly definition of community (Coscia et al., 2011). A traditional definition, for instance, derives from a property observed on real-world networks, namely *community structure* (Girvan and Newman, 2002; Newman and Girvan, 2004). According to this property, networks are organized into communities having sparse connections between them, while within each community, nodes are densely connected. Clearly, the above definition only considers the topology of network links, but there are also node-centered definitions. Nodes forming a community can share some common properties or play similar roles within the interacting phenomenon that is being represented by the graph (Fortunato, 2010). Therefore, nodes can be organized into communities based on their similarity concerning some reference property.

Communities can be classified in various ways. For instance, communities can be either disjoint or overlapping subsets of vertices. In disjoint communities, vertices belong to a single community, whereas in overlapping communities, a vertex can belong completely to more than one community (Javed et al., 2018). Consequently, the problem of community detection, in the disjoint setting, consists in organizing a graph $G = (V, E)$ into a set of K non-empty and mutually exclusive groups, $\{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_K\}$, such that (Chakraborty et al., 2017)

$$|\mathscr{C}_1| + |\mathscr{C}_2| + \ldots + |\mathscr{C}_K| = |V|,$$

where $|\cdot|$ denotes the size of a given set. On the other hand, detecting overlapping communities implies structuring the vertices into non-empty groups, $\{\mathscr{F}_1, \mathscr{F}_2, \ldots, \mathscr{F}_K\}$, such that

$$|\mathscr{F}_1| + |\mathscr{F}_2| + \ldots + |\mathscr{F}_K| \geq |V|.$$

A required property of a community, irrespective of its type, is *connectedness* (Fortunato, 2010). This means that for any pair of nodes u and w in the community ω, there is a sequence of links $\{u, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, w\}$ starting at $v_0 = u$ and ending at $v_{k+1} = w$, where $v_i \in \omega$, with $i = 0, 1, \ldots, k+1$. In particular, the sequence described above is called the *path* from u to w.

Different algorithms and methods for community detection, or graph clustering, have been proposed in the literature. Spectral clustering methods (Zelnik-manor and Perona, 2004; Von Luxburg, 2007), for instance, encompasses approaches to detect clusters based on the spectral properties of the graph, hence its name. In these methods, the vertex set of the graph is projected into the space spanned by eigenvectors of certain matrices able to rep-

resent the graph's structure. Then, the set is clustered using conventional methods like the K-means algorithm. Another group of algorithms are those that perform random walks on graphs such as Markov Clustering (Enright et al., 2002), Walktrap (Pons and Latapy, 2006), and Infomap (Rosvall and Bergstrom, 2008). These methods, in particular, seek to identify communities based on the intuition that random walkers moving from node to node, along the edges, will tend to spend long periods within densely connected clusters of nodes. Furthermore, some algorithms that attempt to organize the graph nodes into communities based on modularity optimization (Clauset et al., 2004; Newman, 2004; Blondel et al., 2008; Mehrle et al., 2015). In particular, modularity, proposed by (Newman and Girvan, 2004), is a measure of the quality of a particular division of a network into a set of disjoint communities. So the above methods operate based on the assumption that good community structures are those with high values of modularity. We refer to the works of Fortunato (2010); Coscia et al. (2011); Harenberg et al. (2014); Fortunato and Hric (2016); Bedi and Sharma (2016); and Javed et al. (2018) for recent reviews on community detection algorithms.

2.5 On graph-based data representations

The seminal work by Von Luxburg (2007) reports the use of graphs to represent data when there is no information available other than similarities between data objects. In such scenario, r-neighborhood graphs and k-nearest neighbor (kNN) graphs, which Von Luxburg calls *similarity graphs*, can be used to model the local neighborhood relationships between the data objects. According to Kang et al. (2017), some open issues related to these approaches include the selection of a proper neighbor number k or radius r, the selection of an appropriate similarity metric to measure the similarity among data points, and overcoming the adverse effect of noise and outliers.

Laskaris and Zafeiriou (2008) suggested the concept of fuzzy connectivity graph, which it is built based on the cluster centers obtained by means of the FCM algorithm, and showed that it captures rich topological information regarding the structure of an unlabeled data set. However, this approach heavily depends on running the soft clustering algorithm in an over-clustering fashion.

With regard to the kNN graphs, Liu et al. (2010) improved their construction by applying an anchor-based approach to capture the data structure. In particular, the anchors could be randomly sampled data objects or K-means clustering centers obtained based on the input data set. Then, the similarity graph construction is performed based on the obtained anchors, rather than the whole data observations. Recent anchor generation strategies based on the K-means algorithm can be found elsewhere (Zhu et al., 2017).

A method to visualize high-dimensional data set as a landscape was presented by Zhang

and Chen (2013). This method uses complete similarity graphs, in which every data object is connected to each other by an edge whose weight is proportional to the distance between data objects, to run an algorithm that projects an input high-dimensional data set onto a two-dimensional space to obtain a landscape-like visualization.

Zhen et al. (2014) presented a novel scheme to construct a similarity graph, where the similarity computation among different data points depends not only on their pairwise distances but also on mutually linear representation relationships. In particular, the proposed scheme, called locally linear representation, encodes each data point using a collection of data points that not only produce the minimal reconstruction error but also are close to the objective point. According to Zhen et al., this scheme makes the process robust to noises and outliers, and avoids selecting inter-subspaces points to represent the objective point to a large extent.

Gan (2014) adopted three strategies, namely power-law adjustment, nearest neighbor, and threshold filtration, to obtain user similarity graphs from pairwise user similarity scores calculated on historical data. The proposed graph construction strategies feature low-complexity operations like applying a power-law function to the user similarity scores as well as ranking and thresholding relationships between pairs of users, in order to emphasize strong relationships or remove weak relationships between users.

In the context of archaeological research, Östborn and Gerding (2014) introduced the notion of "general similarity networks" as a flexible framework in which all kinds of similarity relations, established among continuous and discrete attributes, can be used as proxies for causal or social relationships between archaeological contexts and define links between them.

Cao et al. (2015) proposed a method based on random walks to construct reliable similarity graphs. This approach takes as input a previously obtained raw kNN graph that represents the data set and the definitive neighbors of each data object are determined by the probability of the random walk. According to Cao et al., since the high-order transition probabilities carry complex relationships among data, the graph obtained by the proposed method is able to reflect the structure of the data.

While addressing the problem of feature selection in high-dimensional data, Zhang et al. (2015) developed a method to learn the data similarity matrix by optimally re-assigning the neighbors for each data point based on local distances or dissimilarities, in order to construct an effective similarity graph representation of the input data.

Vogt (2015) presented a model to find the underlying structure in biomedical data sets. The model is based on ranking and comparing the neighbors of every data objects to find overlaps of the same close neighbors. The model takes as input pairwise distances between the data objects and produces a weighted adjacency matrix as output, in which every entry represents the number of top ranked neighbor overlaps. Then, the graph structure of the

adjacency matrix can be used for structure detection and grouping of data points, as well as for visualization of the input data.

In order to improve the performance of graph-based clustering methods, which heavily relies on the goodness of a data similarity graph, Kang et al. (2017) developed an adaptive model that can automatically extract similarity information among data points, and simultaneously preserve the global and local manifold structures hidden in the data. In the local scenario, instead of constructing the data similarity matrix based on a deterministic neighborhood relationship, Kang et al. proposed to adaptively learn the aforementioned matrix from the data by solving an optimization problem. The global structure is exploited based on the self-expressive property of data in which each data point can be approximated as a linear combination of all the other points.

Regarding the joint use of graphs and entropy in the characterization of high-dimensional data sets, an early work is due to Hero et al. (2002). In that work, entropic graphs are proposed as an unparameterized and efficient way to estimate the entropy of high-dimensional data. In particular, an entropic graph is any graph whose normalized total weight, i.e. the sum of the edge lengths, is a consistent estimator of the Rényi entropy. Examples of entropic graphs are the Minimum Spanning Tree (Diestel, 2017) and the kNN graph. Based on entropic graph methods, Costa and Hero (2003) and Costa and Hero (2004) introduced an approach to simultaneously estimate both the intrinsic dimension and intrinsic entropy of random data sets lying on manifolds. In addition, Van Gemert et al. (2006) use entropic graphs to compute an unparametric similarity between image features for object recognition. Later, Zhang et al. (2010) developed an unsupervised technique that learns a low-dimensional representation of a high-dimensional data set that preserves the local geometry in the original data. To this end, an initial kNN graph is constructed based on given raw data and then a graph-based learning process ruled by an objective function that incorporates an entropy regularization term is performed to obtain a two-dimensional projection of the input data set.

The above literature review suggests that works using similarity graphs to represent the structure of data are broadly divided into two classes. Works constructing the similarity graphs without considering entropy measures and those using this kind of graphs and entropy for characterizing high-dimensional data sets. As a result, the incorporation of entropy related measures into the process of constructing graph-based representations for multidimensional and unlabeled data sets remains an open problem, and thus contributions and improvements are possible in this respect.

The fact that the structure of data should dictate the process of getting accurate and compact data representations, without human intervention, and the close relationship between this process and the process of learning from the data themselves, motivate us to consider that entropy measures can consolidate our capacity of obtaining better information from unlabeled data. To this end, entropy measures can be used as criteria for build-

ing graph-based representations, with the potential of inheriting information simplification properties leading to compact models, and the capacity of obtaining clustering structures able to reflect the underlying natural structure of data.

Bibliography

Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. (2012). *Learning From Data*. AML-Book.

Akman, O., Comar, T., Hrozencik, D., and Gonzales, J. (2019). Data Clustering and Self-Organizing Maps in Biology. In Robeva, R. and Macauley, M., editors, *Algebraic and Combinatorial Computational Biology*, MSE/Mathematics in Science and Engineering, pages 351–374. Academic Press.

Ariza-Jiménez, L., Quintero, O., and Pinel, N. (2018). Unsupervised fuzzy binning of metagenomic sequence fragments on three-dimensional Barnes-Hut t-Stochastic Neighbor Embeddings. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1315–1318.

Barabási, Albert-László and Pósfai, M. (2016). *Network science*. Cambridge University Press, Cambridge.

Bedi, P. and Sharma, C. (2016). Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3):115–135.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Cao, J., Ling, B. W.-K., Woo, W.-L., and Yang, Z. (2015). k-NN graph construction based on markov random walk. In *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pages 343–346. IEEE.

Chakraborty, T., Dalmia, A., Mukherjee, A., and Ganguly, N. (2017). Metrics for Community Analysis. *ACM Computing Surveys*, 50(4):1–37.

Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3):267–278.

Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, 70(6):066111.

Coscia, M., Giannotti, F., and Pedreschi, D. (2011). A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, 4(5):512–546.

Costa, J. A. and Hero, A. O. (2003). Entropic graphs for manifold learning. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, pages 316–320. IEEE.

Costa, J. A. and Hero, A. O. (2004). Geodesic Entropic Graphs for Dimension and Entropy Estimation in Manifold Learning. *IEEE Transactions on Signal Processing*, 52(8):2210–2221.

Diestel, R. (2017). *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Dougherty, G. (2013). *Pattern Recognition and Classification*, volume 53. Springer New York, New York, NY.

Enright, A. J., Van Dongen, S., and Ouzounis, C. A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5):75–174.

Fortunato, S. and Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports*, 659:1–44.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Gan, M. (2014). Walking on a User Similarity Network towards Personalized Recommendations. *PLoS ONE*, 9(12):1–27.

Girotto, S., Pizzi, C., and Comin, M. (2016). MetaProb: Accurate metagenomic reads binning based on probabilistic sequence signatures. *Bioinformatics*, 32(17):i567–i575.

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826.

Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., Padmanabhan, K., and Samatova, N. (2014). Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):426–439.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, 2 edition.

Hero, A. O., Ma, B., Michel, O. J., and Gorman, J. (2002). Applications of entropic spanning graphs. *IEEE Signal Processing Magazine*, 19(5):85–95.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666.

Javed, M. A., Younis, M. S., Latif, S., Qadir, J., and Baig, A. (2018). Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*, 108:87–111.

Kang, Z., Peng, C., and Cheng, Q. (2017). Clustering with Adaptive Manifold Structure Learning. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 79–82.

Laskaris, N. A. and Zafeiriou, S. P. (2008). Beyond FCM: Graph-theoretic post-processing algorithms for learning and representing the data structure. *Pattern Recognition*, 41(8):2630–2644.

Liu, W., He, J., and Chang, S.-F. (2010). Large Graph Construction for Scalable Semi-Supervised Learning. In *Proceedings of the 27th International Conference on Machine Learning*, pages 679–689. Omnipress.

Mehrle, D., Strosser, A., and Harkin, A. (2015). Walk-modularity and community structure in networks. *Network Science*, 3(3):348–360.

Newman, M. (2018). *Networks*, volume 1. Oxford University Press.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 69(6 2):1–5.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 69(2):1–15.

Östborn, P. and Gerding, H. (2014). Network analysis of archaeological data: A systematic approach. *Journal of Archaeological Science*, 46(1):75–88.

Pons, P. and Latapy, M. (2006). Computing Communities in Large Networks Using Random Walks. *Journal of Graph Algorithms and Applications*, 10(2):191–218.

Quintero Montoya, O. L., Villa, L. F., Muñoz, S., Arenas, A. C. R., and Bastidas, M. (2015). Information retrieval on documents methodology based on entropy filtering methodologies. *International Journal of Business Intelligence and Data Mining*, 10(3):280.

Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences of the United States of America*, 105(4):1118–23.

Van Gemert, J. C., Burghouts, G. J., Seinstra, F. J., and Geusebroek, J. M. (2006). Color invariant object recognition using entropic graphs. *International Journal of Imaging Systems and Technology*, 16(5):148–153.

Vogt, J. E. (2015). Unsupervised Structure Detection in Biomedical Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(4):753–760.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Xu, R. and Wunsch, D. C. (2010). Clustering algorithms in biomedical research: a review. *IEEE reviews in biomedical engineering*, 3:120–54.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.

Zelnik-manor, L. and Perona, P. (2004). Self-tuning spectral clustering. *Advances in Neural Information Processing Systems 17*, 2:1601–1608.

Zhang, H. and Chen, X. (2013). Network-based clustering and embedding for high-dimensional data visualization. *Proceedings - 13th International Conference on Computer-Aided Design and Computer Graphics, CAD/Graphics 2013*, pages 290–297.

Zhang, L., Qiao, L., and Chen, S. (2010). Graph-optimized locality preserving projections. *Pattern Recognition*, 43(6):1993–2002.

Zhang, Y., Shen, Y., Wang, H., Zhang, Y., and Jiang, X. (2015). On Secure Wireless Communications for Service Oriented Computing. *IEEE Transactions on Services Computing*, 1374(c):1–12.

Zhen, L., Yi, Z., Peng, X., and Peng, D. (2014). Locally linear representation for image clustering. *Electronics Letters*, 50(13):942–943.

Zhu, W., Nie, F., and Li, X. (2017). Fast Spectral Clustering with efficient large graph construction. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2492–2496.

CHAPTER 3

On data exploration and visualization

Contents

3.1 Introduction

An essential mechanism for data exploration and understanding is the search and identification of the data underlying structure. To this end, it is customary to deploy methods able to discover the natural groupings of objects within data based on unsupervised learning notions. This approach, widely known as clustering (see Section 2.3), has been extensively used over time in various scientific fields, and it remains as a relevant tool to deal with the expanding amount of data that modern society generates.

Two essential features characterize clustering. First, the groupings of objects, i.e., the clusters, are discovered without any *a priori* knowledge. Second, objects are organized into groups according to their degree of similarity or distance. A common assumption is that data to be clustered consist of multidimensional objects whose attributes are continuous, categorical, etc. Then, similar objects are grouped and dissimilar objects are kept apart. However, data can also consist of objects not described by their multidimensional attributes, but by their pairwise interactions taking the form of networks or graphs (see Section 2.4). In such situations, where objects are represented as nodes and their interactions as links between them, the clustering problem consists in grouping the nodes according to a similarity criterion computed based on the topology of the network links or other characteristics of the network.

Practical problems and applications whose solutions depend on clustering data not only

can benefit from organizing data into unknown groups, but also from understanding how groups are constituted and related to each other. In the case of clustering multidimensional objects, this refers, for instance, to knowing how groups are situated and positioned relative to each other in a multidimensional feature space. On the other hand, for clustering on networks, this can refer to knowing how communities of nodes are interacting with each other based on the network topology.

In the case of network data, achieving this level of data understanding is straightforward because networks can be visualized by using layout algorithms that automatically arrange the nodes and links in an aesthetically pleasing way (see e.g., Gibson et al. (2013); Hu and Shi (2015) for a recent survey of these algorithms). However, in the case of multidimensional data, carrying out a visual exploration to generate hypotheses regarding the structure of the data under analysis is a non-trivial task that has to deal with the likely high-dimensionality of the input data.

An indirect and simple solution to visually explore the structure of multidimensional data is projecting the data to low-dimensional feature spaces (e.g., either two or three-dimensional). Traditional methods to obtain such spaces are principal component analysis (PCA), multidimensional scaling (MDS), and self-organizing maps (SOM); or more recent ones like the family of methods based on stochastic neighbor embeddings (Hinton and Roweis, 2003; van der Maaten and Hinton, 2008; van der Maaten, 2014). However, because of the data loss that inevitably results from the dimensionality reduction, an accurate representation of the data structure is not guaranteed, thereby compromising the interpretability of the results.

Motivated by the inherent ease with which networks can provide visual information, in this chapter, we develop the idea of constructing graph-based representations for fuzzy clusterings on a given data set. Rather than provide a representation based on the data themselves, fuzzy cluster memberships are used to build a representation that facilitates the exploration and understanding of multivariate data structures. Although several works have addressed the problem of visualizing this kind of complex clustering structures (Abonyi and Babuska, 2004; Berthold et al., 2005; Feil et al., 2007; Höppner and Klawonn, 2006; Sato-Ilic and Ilic, 2016; Sharko and Grinstein, 2009; Zhou et al., 2017), a graph-based representation has not been proposed yet, to the best of our knowledge.

The rest of this chapter is organized as follows. Section 3.2 introduces the proposed graph-based method for representing fuzzy clusterings structures in multivariate data. Section 3.3 describes the experimental work performed on real-world data sets and the results obtained. The concluding remarks and directions for future research are presented in Section 3.4. In particular, this chapter is based on Ariza-Jiménez et al. (2018) and (Ariza-Jiménez et al., 2020a).

3.2 Fuzzy membership networks

In this section, we present the *fuzzy membership networks*, which are undirected weighted networks able to represent soft clusterings on a given data set. The proposed approach relies entirely on the entries of the fuzzy partition matrix that represents a particular fuzzy clustering, rather than the original data set themselves, to build a graph-based representation for data exploration and visualization.

Suppose that we are given a data set with N multidimensional objects, $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^{N}$, where $\mathbf{x}_j \in \mathbb{R}^t$. Let $C = \{C_1, C_2, \ldots, C_K\}$ be a clustering of $\mathbf{X}$ provided by a center-based soft clustering algorithm (e.g., the FCM algorithm), and let $U = [u_{ij}]$ be the corresponding K-by-N fuzzy partition matrix of the resulting clustering. In addition, let $\mathbf{c}_i$ be the center of the partition C_i with $i = 1, 2, \ldots, K$.

An undirected weighted membership network, $G_U = (V, E)$, that represents the above fuzzy clustering can be constructed as follows:

1. Consider each data object $\mathbf{x}_j \in \mathbf{X}$ and each cluster center $\mathbf{c}_i$ as a node of G_U, i.e. let $V = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \cup \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$.

2. Link each data object $\mathbf{x}_j \in \mathbf{X}$ with every cluster center $\mathbf{c}_i$ to represent the belonging of $\mathbf{x}_j$ to more than one cluster in the set C, i.e. let $E = \{\{\mathbf{x}_j, \mathbf{c}_i\}, i = 1, 2, \ldots, K; j = 1, 2, \ldots, N\}$.

3. Associate to each link $\{\mathbf{x}_j, \mathbf{c}_i\}$ a weight $\omega(\{\mathbf{x}_j, \mathbf{c}_i\})$ equal to the degree of membership of $\mathbf{x}_j$ in the cluster C_i, i.e. let $\omega(\{\mathbf{x}_j, \mathbf{c}_i\}) = u_{ij}$.

In particular, a mathematical property of the obtained membership network is that it allows computing a cluster validity index for fuzzy clusterings, called partition coefficient (PC) (Wang and Zhang, 2007), as $PC = (2|V|)^{-1}\mathrm{Tr}(W^2)$, where W is the symmetric $|V|$-by-$|V|$ matrix obtained when all of the weight edges of the graph G_U are recorded in a single matrix.

To exemplify how our methods works, suppose that eight multidimensional data objects were clustered into two groups using a soft clustering algorithm, according to the following fuzzy partition matrix U:

$$U = \begin{bmatrix} 0.8 & 0.9 & 0.9 & 0.7 & 0.2 & 0.3 & 0.5 & 0.1 \\ 0.2 & 0.1 & 0.1 & 0.3 & 0.8 & 0.7 & 0.5 & 0.9 \end{bmatrix} \tag{3.1}$$

As is shown in Figure 3.1, the membership network that represents this fuzzy clustering has as many data objects as nodes, plus two additional nodes, nodes 9 and 10, which represent the centers of the fuzzy clusters. Data objects are connected to the cluster-center nodes

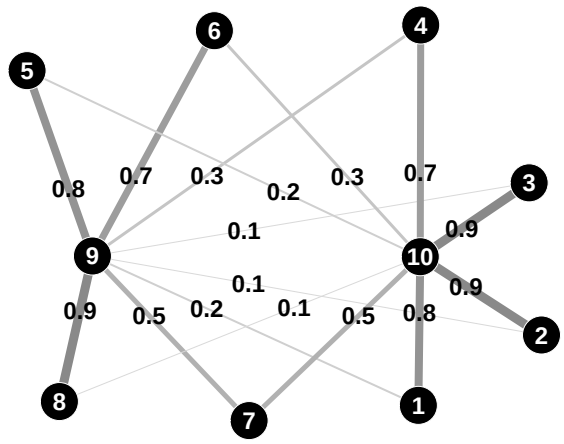


Figure 3.1: Representing a given fuzzy clustering result as an undirected weighted membership network using the proposed method.

using links having weights given by the entries of the matrix U. For clarity, the color and the thickness of the links between nodes and cluster centers is proportional to the degree of membership of each data object to the different clusters. Darker and thicker links indicate a higher degree of membership.

Once a fuzzy network membership is built, it can be visualized using a layout algorithm that automatically arranges the nodes and links in an aesthetically pleasing way. The OpenOrd algorithm (Martin et al., 2011) is used here for this task, since it is an algorithm suitable for drawing undirected weighted networks, and it can provide layouts for large-scale real-world networks wherein clusters can be easily distinguished.

3.3 Experimental results

In this section, we perform experiments on real-world data sets to demonstrate the usefulness of our approach. As our purpose here is mainly to present a data exploration and visualization method, we arbitrary use the traditional FCM algorithm to obtain fuzzy clusterings from such data sets. Moreover, the number of clusters K to detect utilizing the algorithm was manually set depending on the study case, whereas the fuzzification parameter was set as $m = 2$.

3.3.1 Iris flower dataset

The proposed method is first demonstrated using the widely known Iris flower data set. This multidimensional data set consists of four morphological measurements taken on 50 samples from three different species of Iris flowers: *Iris setosa*, *I. versicolor*, and *I. virginica*.

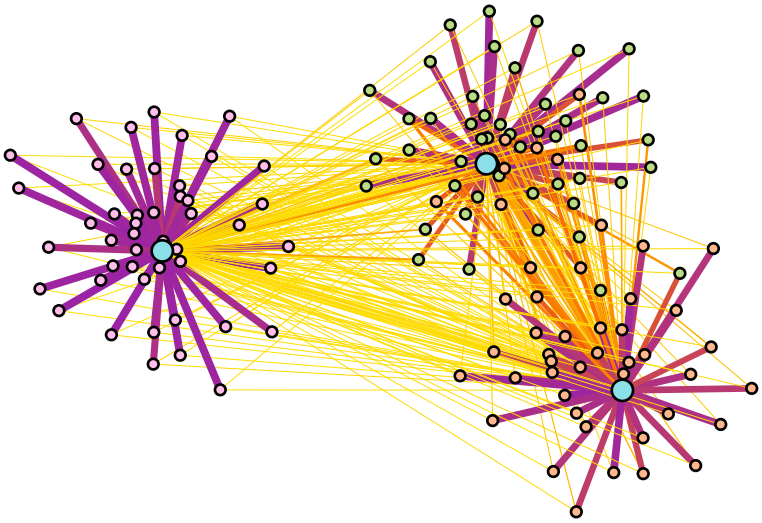


Figure 3.2: Membership network representing a 3-cluster fuzzy structure discovered by the FCM algorithm on the Iris flower data set.

Three soft clusters for the Iris data set were initially obtained using FCM. Figure 3.2 shows the membership network constructed based on the fuzzy partition matrix that represented the resulting clustering. Nodes are colored according to the real classification of the data objects. Three groups of nodes representing the 3-cluster structure are visible in this network-based representation. Groups at the right side of the figure represent two clusters that contain *I. virginica* and *I. versicolor* samples, while the remaining group represents a cluster with *I. setosa* samples. Each group of nodes has an anchor node that represents the cluster center, and for clarity purposes, they are colored in blue and their size is larger than the nodes representing data objects. Darker and thicker links indicate a higher degree of membership. As expected, intra-cluster links have a greater membership weight in comparison with extra-cluster links when clusters are well separated from each other. However, as the boundary between the clusters of *I. virginica* and *I. versicolor* samples is ambiguous, some nodes in this boundary belong to both clusters with no well-defined degree of membership, and thus they are linked to both cluster center-nodes with darker and thicker connections. In particular, the above observation indicates that the fuzzy membership network can reflect a known fact about the Iris data set: all the samples of *I. setosa* are linearly separable from the samples of *I. virginica* and *I. versicolor*, whereas the latter are not linearly separable from each other.

3.3.2 Bank customer datasets

We now turn our attention to two data sets, with an unknown underlying structure, that describe the financial behavior of 18,583 customers of a local bank during a particular

window of time. Both data sets were introduced in Ariza-Jiménez et al. (2020b), where conventional approaches for data exploration were used to find structure in unlabeled data. In that work, clustering algorithms were employed to partition the feature spaces into a given number of clusters, and the quality of the resulting clustering structures was evaluated using internal validity indices. Histograms were computed in addition to describe and then analyze the size of the partitions obtained by each algorithm. Two- and three-dimensional Barnes-Hut Stochastic Neighbor Embeddings (BH-SNE) of the bank customer data were also obtained to provide the means for data and clustering visualization, as well as to carry out a comparative study about the quality of the information that can be extracted from low- and high-dimensional feature spaces.

Regarding the aforementioned data sets, the first one describes each customer using four variables that characterize its transactions with other customers. The number of individual transactions between clients amounted to 397,571. For each customer, the above set of variables corresponds to records such as the total number of transactions in which the customer received or made payments, and the total amount of money received or deposited. The second data set consists of ten variables describing the financial statements of each customer.

We particularly considered these data sets to test the functionality of the fuzzy membership networks in the exploration of real-world data with unknown structure. For this purpose, we arbitrary clustered both data sets into ten groups of customers using the FCM algorithm. Figures 3.3 and 3.4 show the resulting network-based representation for each soft clustering. Nodes that represent cluster centers are depicted larger than the nodes representing data objects for clarity.

Data exploration based on organizing both data sets into ten arbitrarily soft clusters and using the network-based approach to visualize both clustering results demonstrate how different are their corresponding underlying structures. The clustering structure provided by FCM on the first data set consists of both large and predominant groups, and small groups (Fig. 3.3), while the same algorithm partitioned the second data set into regular size clusters (Fig. 3.4). Furthermore, larger groups discovered in the first data set are more closely connected in comparison with the smaller groups (Fig. 3.3), which could indicate that the former groups represent data objects lying in large regions with almost uniform data density, while the latter groups consist of data objects that are outliers. A subsequent examination of these groups of outliers indicated that they correspond to clients who participated in numerous and large transactions, and thus these clients could be of interest in receiving special banking services related to financial transactions. On the other hand, the well-connected groups of nodes discovered in the second data set (Fig. 3.4) may also consist of data objects sharing the same feature as the larger groups in the first dataset (Fig. 3.3). However, this result could also be interpreted as there is no evidence of the existence of natural groups in the second data set, and thus, the FCM algorithm only performed seg-

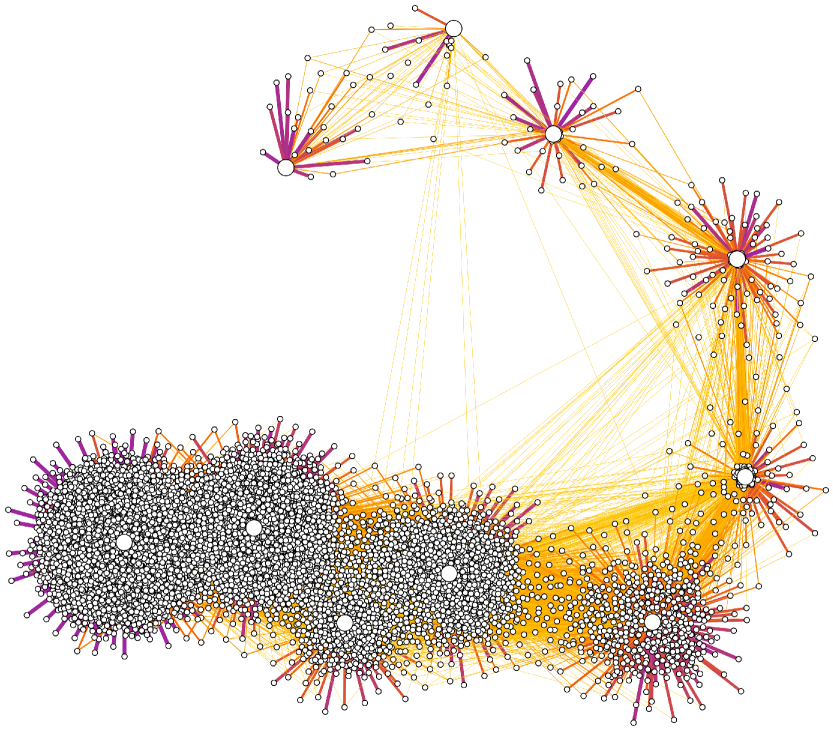


Figure 3.3: Membership network representing a 10-cluster fuzzy structure discovered by the FCM algorithm on the first bank customer dataset.

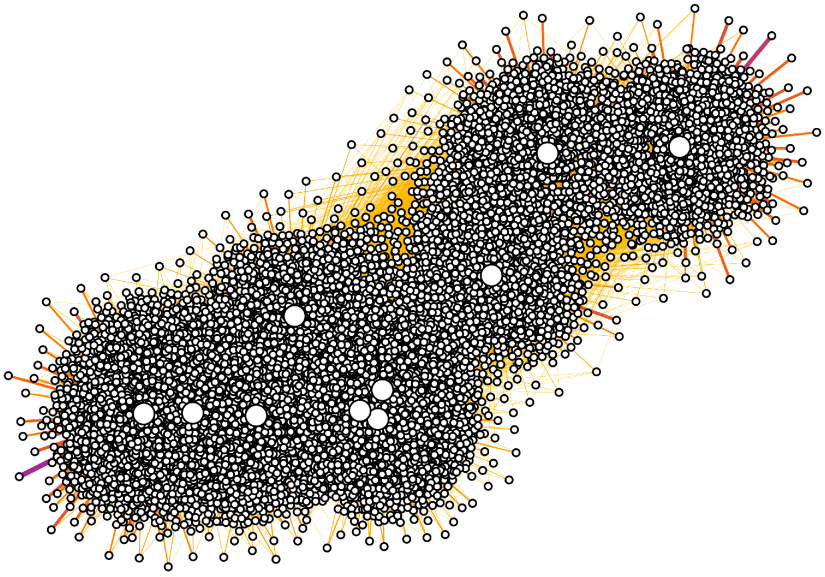


Figure 3.4: Membership network representing a 10-cluster fuzzy structure discovered by the FCM algorithm on the second bank customer dataset.

mentation of the bank customers. Finally, the closeness between the three cluster center nodes in the middle of Fig. 3.4 suggests that a cluster has been wrongly split into three clusters, as FCM necessarily must partition the data into the given number of clusters.

3.4 Discussion

In this chapter, we reported a network-based data exploration and visualization method to facilitate the understanding of the uncertainties present in the data. For this purpose, fuzzy clustering structures imposed on multidimensional data are translated from fuzzy partition matrices into undirected weighted membership networks. This simple network-based method enables us to understand visually how elements (data objects and clusters) involved in this kind of complex data clustering structures interact with each other, without relying on a visualization of the input data themselves.

We tested our approach on the widely known Iris flower data set as well as on bank customer data sets. In the first case, the fuzzy membership networks can reflect the well-known clustering structure of the Iris data set. In the second case, our approach provided valuable insight into the nature of the data sets under study without depending on dimensionality reduction methods, unlike the approach followed by Ariza-Jiménez et al. (2020a), which heavily depends on such methods to study the same data sets.

Although membership networks performed well in our experiments, there is still room for improvements. In particular, the membership network representing a given fuzzy clustering has as many nodes as data objects. While this is not inconvenient for small data sets (see Fig. 3.2), for larger data sets the depicted nodes tend to accumulate around the center nodes and thus the membership interaction between data objects and clusters, represented by the weighted edges, could not be clearly seen (see Fig. 3.4). To overcome this shortcoming, we plan to investigate whether a more compact graph-based data representation is possible in comparison with the membership networks. In this regard, we consider that it is feasible to follow a data clustering approach for that purpose. Indeed, data clustering can be used as the basis of a data compression strategy where data is organized and summarized employing cluster prototypes. Based on the reported results, we also realize the convenience of using fuzzy memberships derived from soft clustering structures to develop this kind of compact representation.

Bibliography

Abonyi, J. and Babuska, R. (2004). FUZZSAM - Visualization of fuzzy clustering results by modified Sammon mapping. *IEEE International Conference on Fuzzy Systems*, 1:365–370.

Ariza-Jiménez, L., Pinel, N., Villa, L. F., and Quintero, O. L. (2020a). An Entropy-Based Graph Construction Method for Representing and Clustering Biological Data. In *VIII Latin American Conference on Biomedical Engineering and XLII National Conference on Biomedical Engineering*, pages 315–321, Cham. Springer International Publishing.

Ariza-Jiménez, L., Quintero, O., and Pinel, N. (2018). Unsupervised fuzzy binning of metagenomic sequence fragments on three-dimensional Barnes-Hut t-Stochastic Neighbor Embeddings. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1315–1318.

Ariza-Jiménez, L., Villa, L. F., Pinel, N., and Quintero, O. L. (2020b). Extracted information quality, a comparative study in high and low dimensions. *International Journal of Business Intelligence and Data Mining*, X(X):X.

Berthold, M. R., Wiswedel, B., and Patterson, D. E. (2005). Interactive exploration of fuzzy clusters using neighborgrams. *Fuzzy Sets and Systems*, 149(1):21–37.

Feil, B., Balasko, B., and Abonyi, J. (2007). Visualization of fuzzy clusters by fuzzy Sammon mapping projection: Application to the analysis of phase space trajectories. *Soft Computing*, 11(5):479–488.

Gibson, H., Faith, J., and Vickers, P. (2013). A survey of two-dimensional graph layout techniques for information visualisation. *Information Visualization*, 12(3-4):324–357.

Hinton, G. E. and Roweis, S. T. (2003). Stochastic Neighbor Embedding. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 857–864. MIT Press.

Höppner, F. and Klawonn, F. (2006). Visualising clusters in high-dimensional data sets by intersecting spheres. *Proceedings of the 2006 International Symposium on Evolving Fuzzy Systems, EFS'06*, 2(2):106–111.

Hu, Y. and Shi, L. (2015). Visualizing large graphs. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(2):115–136.

Martin, S., Brown, W. M., Klavans, R., and Boyack, K. W. (2011). OpenOrd: an open-source toolbox for large graph layout. In *Proceedings of SPIE*, number January, pages 7868 – 7868 – 11.

Sato-Ilic, M. and Ilic, P. (2016). Visualization of Fuzzy Clustering Result in Metric Space. *Procedia Computer Science*, 96(September):1666–1675.

Sharko, J. and Grinstein, G. (2009). Visualizing fuzzy clusters using radviz. *Proceedings of the International Conference on Information Visualisation*, pages 307–316.

van der Maaten, L. (2014). Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245.

van der Maaten, L. and Hinton, G. (2008). Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Wang, W. and Zhang, Y. (2007). On fuzzy cluster validity indices. *Fuzzy Sets and Systems*, 158(19):2095–2117.

Zhou, F., Chen, M., Wang, Z., Luo, F., Luo, X., Huang, W., Chen, Y., and Zhao, Y. (2017). A radviz-based visualization for understanding fuzzy clustering results. In *Proceedings of the 10th International Symposium on Visual Information Communication and Interaction*, pages 9–15, New York, USA. ACM.

Chapter 4

Entropy-based features and graph-based data models

Contents

4.1 Introduction

Finding simple representations that allow uncovering and portraying the underlying structure of real-world data is a non-trivial task (Vogt, 2015). Unsupervised methods are used to learn the inner structure from this kind of data, which can consist of unlabeled observations.

However, conventional approaches proposed to accomplish this task rely on methods that make assumptions about data distribution (e.g., center-based clustering (Gan et al., 2007)) or require the reduction of the dimensionality of the input data (e.g., via principal component analysis (Bécavin and Benecke, 2011)).

Graphs are a mathematical concept that allows us to represent and analyze the complex interactions between real-world entities (Newman, 2003; Lambiotte et al., 2008; McCormick et al., 2010; Traud et al., 2012; Schmidt et al., 2017), thereby helping us to extract useful information or knowledge from data. Graph-based methods for detecting structure in data have been previously reported, mainly based on similarity measures between data objects (de Arruda et al., 2012; Zhang and Chen, 2013; Vogt, 2015). Such approaches implement graphs in which each data object acts as a vertex and is connected to every other data object in the data set.

Rather than using every data object to construct a graph-based model, here we intend

to represent data in a compact manner. To this end, we propose to sample a given data set to obtain a subset of data objects based on which the above graph can be constructed. Such sampled data objects are thought of as prototypes that represent neighborhoods of the data set. From an information theory perspective, there exists a relationship between compactly describing a set of data and its entropy (Balakrishnan and Touba, 2007). Consequently, we also propose to incorporate entropy-related measures into the process of identifying the aforementioned prototypes and constructing graph-based models for data sets to uncover their inner structure.

Experiments conducted on both synthetic and real data sets proved the usefulness and effectiveness of the proposed approach. The experimental results on biological data sets shown the potential of the proposed entropy-based graph models to cope with biological applications, such as metagenomic binning and neuronal spike sorting, where it is required to organize data into unknown and meaningful groups.

The rest of this chapter is organized as follows. Section 4.2 recalls some background regarding the concept of entropy and the binary entropy function. Section 4.3 describes the proposed methods. Experimental results on both synthetic and real data sets are illustrated in Section 4.4, whereas Section 4.5 concludes the chapter. In particular, this chapter is based on Ariza-Jiménez et al. (2018), Ariza-Jiménez et al. (2020), and Ceballos et al. (2020).

4.2 Background

4.2.1 Shannon entropy

Shannon introduced the concept of entropy as a measure of information, choice and uncertainty in terms of probability theory (Shannon, 1948). Let $(p_1, p_2, \ldots, p_n)$ be a finite discrete probability distribution, therefore $p_i \geq 0$ for $i = 1, 2, \ldots, n$ and $\sum_{i=1}^{n} p_i = 1$. Then, the (Shannon) entropy of this discrete distribution is defined by

$$\begin{aligned} H &: \mathscr{P} \to [0, \infty) \\ H\left(p_1, p_2, \ldots, p_n\right) &= -\sum_{i=1}^{n} p_i \log_2 p_i, \end{aligned} \tag{4.1}$$

where $\mathscr{P}$ denotes the set of all the probability distributions on finite sets (Klir and Folger, 1987).

Remark 4.1. It is assumed that $p_i \log_2 p_i = 0$, when $p_i = 0$; since $y \log_2 y \to 0$ as $y \to 0$.

Remark 4.2. Entropy, as defined in (4.1), has units *bits*. If the base of the logarithm is changed to e, then the entropy units are called *nats* (see e.g., Cover and Thomas (2006)).

Note that Shannon's entropy measures uncertainty using a combination of outcome unexpectedness weighted by its probability. In particular, this approach creates a balance between unlikely outcomes, with high information content since they are unexpected, which are mitigated by their rare occurrence; and very likely outcomes, with low information content, also mitigated through the product with their probability (Principe, 2010).

Regarding the extreme values of Shannon's entropy, suppose that there exists a single component p_i of the probability distribution $\left(p_1, p_2, \ldots, p_n\right)$ for which $p_i = 1$. Then, there is no uncertainty concerning the outcome of a hypothetical experiment whose results have the probabilities $p_1, p_2, \ldots, p_n$ and thus, the entropy is the minimum possible, i.e., $H\left(p_1, p_2, \ldots, p_n\right) = 0$. On the other hand, when there is maximum uncertainty because all the possible results are equally probable, i.e., $p_i = 1/n$, the entropy is the maximum possible, and equal to $\log_2 n$.

4.2.2 Binary entropy function

Consider now the case of a discrete probability distribution (p, q) such that $q = 1 - p$. Then, the entropy, in this case, is equal to

$$H\left(p, q\right) = -p \log_2 - q \log_2 q. \tag{4.2}$$

The above entropy measure can be seen as a function that takes a single real number as a parameter. This function, known as the *binary entropy function* (MacKay, 2005), is the function defined by

$$\begin{aligned} H_{bin} &: [0, 1] \to [0, 1] \\ H_{bin}\left(p\right) &= -p \log_2 p - \left(1 - p\right) \log_2 \left(1 - p\right). \end{aligned} \tag{4.3}$$

As illustrated in Figure 4.1, H_{bin} reaches a maximum value of 1 when p is 0.5 and a minimum value of 0 when p is either 0 or 1.

Remark 4.3. Entropy is characterized by being a versatile concept, having distinct and intuitive interpretations in multiple applied domains. In physics, where the concept of entropy was first introduced, the missing information on the actual state of a system is related to the entropy of the system (Beck, 2009). In fuzzy set theory, an "entropy" is introduced, using using no probabilistic concepts, as a measure of the degree of fuzziness (De Luca and Termini, 1972). In our case, we will use the binary entropy function as a mechanism to map pairwise similarity measures into entropy-like values in order to characterize a set of data objects.

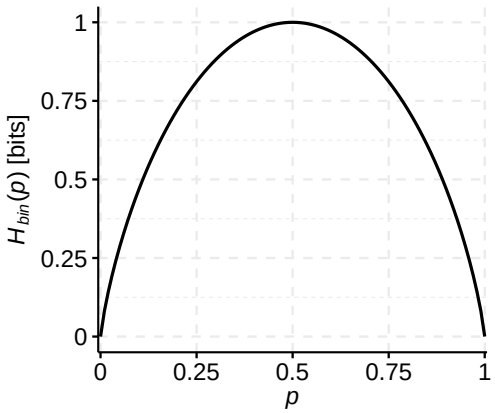


Figure 4.1: The binary entropy function.

4.3 Proposed method

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ be a data set of N t-dimensional data objects, where each data object $\mathbf{x}_i$ consists of a vector of t measurements, features, or attributes, i.e., $\mathbf{x}_i = (x_{i1}, \ldots, x_{it}) \in \mathbb{R}^t$. Our goal is to use pairwise similarities and entropy-based measures to construct a graph that allows us to represent and discover the inner structure of the input data; the latter described as a partition of $\mathbf{X}$ into groups.

The overall process of the method consists of five stages, as outlined below. Input data is first normalized, and then a pairwise similarity measure is computed for every pair of data objects using a Gaussian-based function. Next, an iterative entropy-based procedure is performed to sample the data set. Samples are used together with a fuzzy partition of the data set to built a fuzzy connectivity graph. Finally, this graph, which represents the entire data set, is clustered, and then the resulting clustering is applied to the remaining data objects.

Stage 1: Data normalization

To begin with, $\mathbf{X}$ is normalized to make the input data dimensionless. This is done to make the similarity measuring less sensitive to the differences in the magnitudes of the data objects attributes (Gan et al., 2007). To this end, the value of the j-th feature x_{ij} of each $\mathbf{x}_i$ is re-scaled to the interval [0,1] by applying the following min-max normalization

$$x_{ij}^{*} = \frac{x_{i,j} - \min\left(x_{ij}\right)}{\max\left(x_{ij}\right) - \min\left(x_{ij}\right)}, \tag{4.4}$$

where $\min\left(x_{ij}\right)$ and $\max\left(x_{ij}\right)$ are the minimum and maximum values taken by x_{ij}, with $i = 1, \ldots, N$, and x^*_{ij} represents the re-scaled value of the feature x_{ij}.

Convention 4.1. From now on, the data set $\mathbf{X}$ is assumed to be normalized.

Stage 2: Measuring data similarity

Once data are normalized, the similarity between every pair of data objects $\mathbf{x}_i$ and $\mathbf{x}_j$ is calculated. For this purpose, we used a modified version of the well-known Gaussian neighborhood function:

$$\begin{aligned} S &: \mathbf{X} \times \mathbf{X} \to [0,1] \\ S\left(\mathbf{x}_i, \mathbf{x}_j\right) &= \exp\left(-\alpha\left[D^*\left(\mathbf{x}_i, \mathbf{x}_j\right)\right]^{1/2}\right), \end{aligned} \tag{4.5}$$

where $D^*\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is the normalized Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and α is a scale parameter. If $D\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, then

$$D^*\left(\mathbf{x}_i, \mathbf{x}_j\right) = \frac{D\left(\mathbf{x}_i, \mathbf{x}_j\right)}{t^{1/2}}, \tag{4.6}$$

where t is the dimensionality of the data space (Kantardzic, 2011), and, adapted from elsewhere (Yao et al., 2000; Kantardzic, 2011), α is given by

$$\alpha = -\frac{\ln 0.5}{\sqrt{D^*_{avg}}}, \tag{4.7}$$

where D^*_{avg} is the average (normalized Euclidean) distance among data objects in $\mathbf{X}$. As a result, data points separated by a distance equal to D^*_{avg} are assigned a similarity of 0.5.

Convention 4.2. To shorten notation, the similarity $S\left(\mathbf{x}_i, \mathbf{x}_j\right)$ between data objects $\mathbf{x}_i$ and $\mathbf{x}_j$ will be denoted as S_{ij}.

Stage 3: Iterative entropy-based data sampling

In this stage, the binary entropy function (4.3) is used to map each pairwise similarity S_{ij} into a non-negative value $H_{bin}\left(S_{ij}\right)$. Then, for each data object $\mathbf{x}_i$, an entropy-based feature H_i is calculated as (Yao et al., 2000)

$$H_i = \sum_{j=1}^{N} H_{bin}(S_{ij}). \tag{4.8}$$

To achieve a compact representation of the structure of data, a copy $\mathbf{X}'$ of the original data set $\mathbf{X}$ is sampled by an iterative procedure based on the above entropy-based features. A sampled data object, henceforth referred to as "prototype", is obtained in each iteration. The first prototype $\mathbf{p}_1$ is the data object in $\mathbf{X}'$ whose corresponding entropy-based feature H_i has the least value. This object is removed afterward from $\mathbf{X}'$, as well as all its nearby neighbors $\mathbf{x}_i$ such that $S(\mathbf{x}_i, \mathbf{p}_1) > \beta$. Next, the object having the least value of the entropy-based feature among the remaining objects in $\mathbf{X}'$ is selected as the second prototype $\mathbf{p}_2$. Then this object, together with all its close neighbors, is removed from $\mathbf{X}'$. This iterative procedure is performed until no data object remains in $\mathbf{X}'$. Note that parameter β conditions the number of prototypes found through the iterative procedure. Indeed, there is an inverse relationship between β and the number of samples retrieved from $\mathbf{X}'$. Preliminary tests showed that setting

$$\beta = S_{avg} + \gamma S_{\sigma}, \tag{4.9}$$

where S_{avg} and S_{σ} are respectively the sample mean and standard deviation of all the pairwise similarities S_{ij}, and γ a positive integer constant, allowed to obtain a large and representative number of data samples from $\mathbf{X}'$ (see Section 4.4.1).

Stage 4: Fuzzy connectivity graph construction

Let $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_K$ be K prototypes sampled from $\mathbf{X}$, where $K \ll N$. Suppose that each prototype $\mathbf{p}_l$, where $l = 1, 2, \ldots, K$, is the center of a fuzzy cluster C_l, which is part of a fuzzy partition of $\mathbf{X}$ into K clusters. Then, the grade of membership of object $\mathbf{x}_j \in \mathbf{X}$ in cluster C_l, denoted by u_{lj}, can be computed in a manner similar to the FCM algorithm (Xu and Wunsch, 2010):

$$u_{lj} = \left[\sum_{k=1}^{K} \frac{D\left(\mathbf{x}_j, \mathbf{p}_l\right)^2}{D\left(\mathbf{x}_j, \mathbf{p}_k\right)^2} \right]^{-1}. \tag{4.10}$$

Prototypes are then used to build a fuzzy connectivity graph G (Laskaris and Zafeiriou, 2008), which is an undirected weighted graph able to represent the structure of the input data. To this end, each prototype is considered as the vertex of an edge-less graph G on K nodes, i.e., $G = (V, E)$ where $V = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_K\}$ and $E = \emptyset$. To populate G with weighted edges, for each object $\mathbf{x}_j$, the two larger membership values u_{aj} and u_{bj} are identified, and the edge $\{\mathbf{p}_a, \mathbf{p}_b\}$ is created afterwards. If edge $\{\mathbf{p}_a, \mathbf{p}_b\}$ already existed, its current weight is increased by one; otherwise, an unitary weight is initially assigned on this edge. Edges and their corresponding weights in the fuzzy connectivity graph G represent the extent of overlap between fuzzy clusters sharing a common boundary.

Stage 5: Graph clustering

Graph clustering is performed on the resulting fuzzy connectivity graph G to organize the original data set $\mathbf{X}$ into groups and uncover the underlying natural structure of the data. Clusters on G are found using the so-called Louvain method (Blondel et al., 2008). This is a fast and greedy method for community detection in graphs (also called graph clustering), which can extract the community structure of large graphs based on modularity optimization. The modularity, initially proposed by Newman and Girvan (2004), is a measure of the quality of a particular division of a graph into modules, communities, groups, etc.

Once prototype nodes in G are clustered, data objects which are not part of the graph G receive a cluster membership from prototypes following the mechanism suggested by Laskaris and Zafeiriou (2008). Let $\mathbf{x}_j \in \mathbf{X}$ be a non-prototype data object, if

$$i = \underset{l=1,\ldots,K}{\arg\max}\ u_{lj} \tag{4.11}$$

then $\mathbf{x}_j$ is assigned to the same cluster to which prototype node $\mathbf{p}_i$ was assigned on the graph G. Thus, each non-prototype data object is classified in the same way as the prototype that is the center of the fuzzy cluster, to which the latter has the largest membership. The above assignation mechanism is somewhat inspired on the so-called maximum defuzzification process that is commonly used to obtain a hard clustering from a fuzzy clustering (see e.g., Geng and Hu (2012); Quintero Montoya et al. (2015); Liu et al. (2017))

4.4 Experimental results

In this section, we describe the experiments that have been performed to evaluate the usefulness and effectiveness of the proposed method. Experiments have been performed both on synthetic data sets and data sets with biological origin.

4.4.1 Experiments on synthetic data

Two synthetic data sets were created to test the performance of the proposed method. The first synthetic data set, shown in Fig. 4.2a consists of several clusters of different shapes and densities in a two-dimensional space. The second data set, presented in Fig. 4.3, is a three-dimensional data set, usually known as a "swiss-roll" data set. The number of data objects of each data set is 1756 and 5000, respectively.

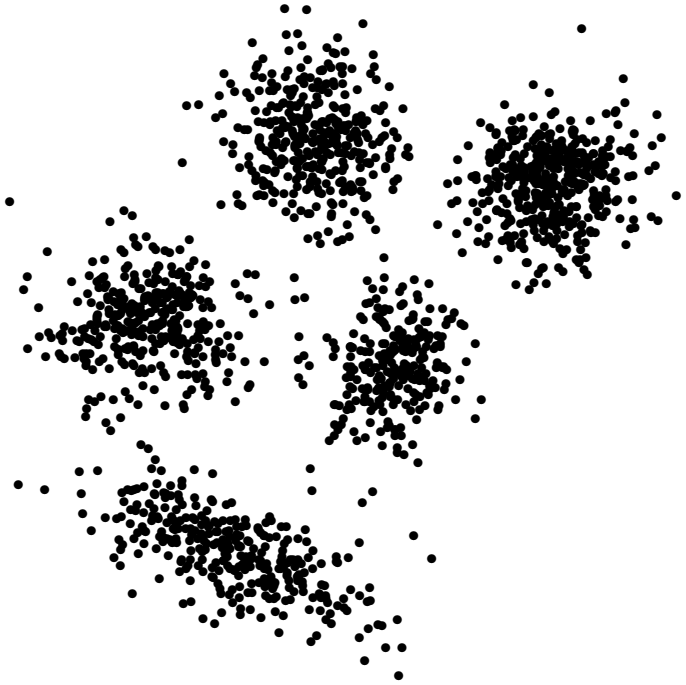
Figure 4.2 illustrates the overall process of the proposed method on the first synthetic data set. Figure 4.2a and 4.2b depict the first synthetic data set and the corresponding entropy-based feature of each data object using colors, respectively. For clarity, the color

of each data object is proportional to the value of its entropy-based feature. A hot color scale, which varies from yellow, through orange and red, to black, is used here. Yellow and black represent "less entropic" and "more entropic" data objects, respectively. Particularly, lower values of the entropy-based feature are assigned to data points near the center of each cluster in comparison with the ones located in the cluster boundaries or regions with a lower density of points.

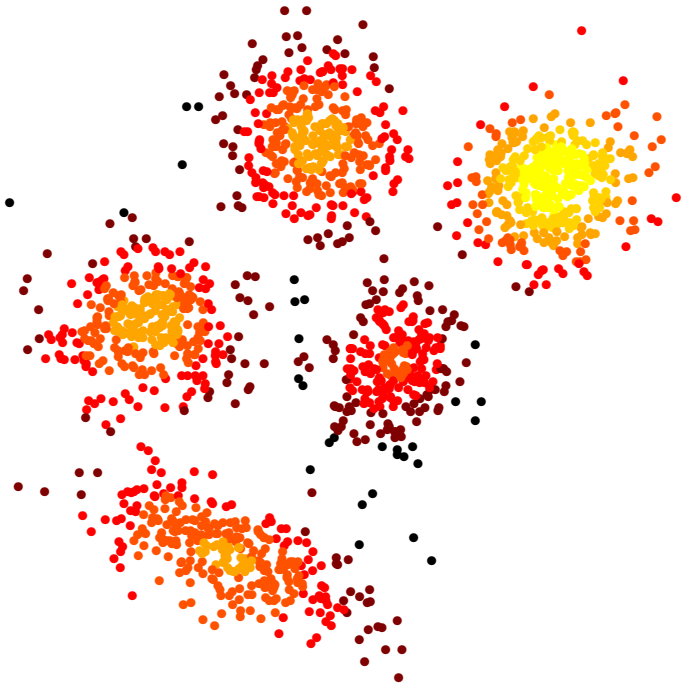
Figure 4.2c shows the prototypes (colored in white) provided by the iterative entropy-based procedure. As expected, prototypes are distributed over the data set and they constitute a down-sampled version of the synthetic data set. The fuzzy connectivity graph generated based on them is depicted in Fig. 4.2d. It can be observed that due to the number and distribution of the prototypes, it is possible to extract a graph-based representation that offers a compact description of each cluster, and it is in accordance with the structure of the data set. In the fuzzy connectivity graph, for clarity, nodes are colored according to the similarity-based entropy measure of the prototypes. As mentioned before, edges and their weights represent the extent of overlap between adjacent fuzzy clusters whose centroids are the prototypes in the ends of each edge. The thickness of the edges represents their weights, and thus thicker edges indicate a higher amount of overlap between two adjacent fuzzy clusters. Notice that edges connecting prototypes in regions with a higher density of points (e.g., in the center of a cluster) are thicker than edges along the boundary of a cluster and those acting as bridges between clusters. Also, edges in the fuzzy connectivity graph are able to reflect how groups are constituted (e.g., see the elongated cluster in the bottom part of Fig. 4.2d) and related to each other (e.g. nearby clusters are bridged by an edge).

Figures 4.2e and 4.2f show, using different colors, the clustering performed by the Louvain algorithm on the fuzzy connectivity graph and the resulting clustering applied to the entire data set based on (4.11), respectively. Fig. 4.2f evidences that the final clustering structure reflects the natural groupings in the synthetic data set.

Figure 4.3 illustrates the overall process on the so-called swiss-roll data set. As before, figures in the first row depict the synthetic data set and the corresponding entropy-based feature of each data object, which is again represented using a hot color scale. In particular, data points in the "origin" of the swiss-roll are the least entropic. Prototypes (depicted in green at the second row) are distributed over the data set, and the generated fuzzy connectivity graph almost represents the spiral structure of the data set, with few edges between the different "layers" of the spiral. Unlike Fig. 4.2d, the fuzzy connectivity graph is not superimposed on the original swiss-roll data set for clarity. Regarding the graph-based resulting clustering, the Louvain algorithm provides clusters that do not leak into different layers of the spiral. These outcomes attested that our method can uncover the underlying structure of the data set.

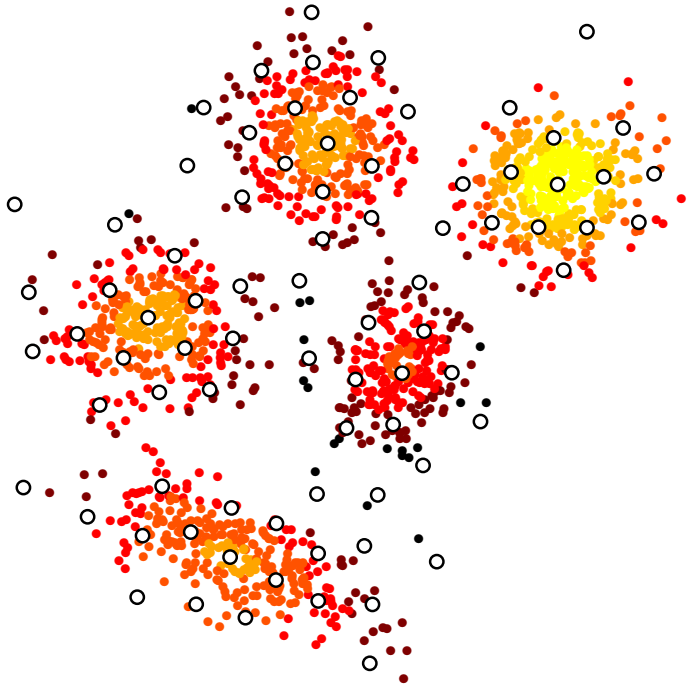


(a) First synthetic data set.

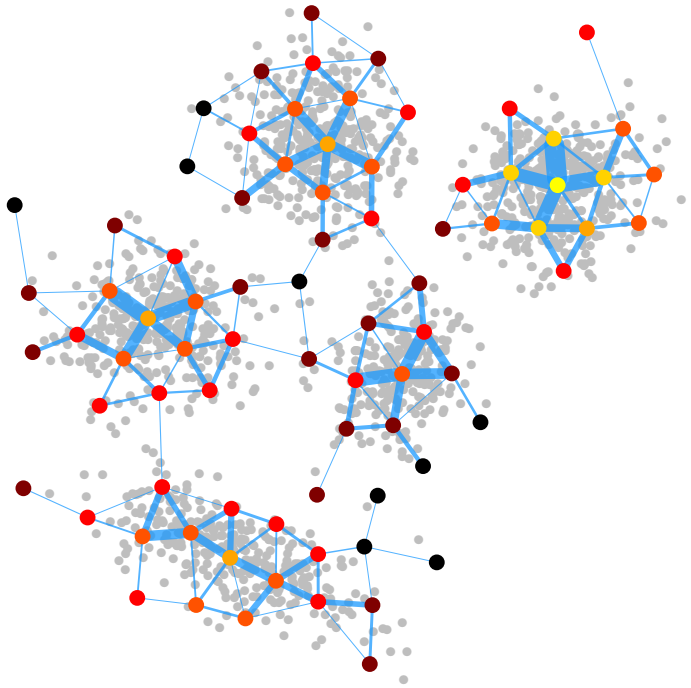


(b) Representation of the entropy-based feature of each data object employing a hot color scale.

Figure 4.2: Experimental results on a synthetic data set that consists of several clusters of different shapes and densities in a two-dimensional space.

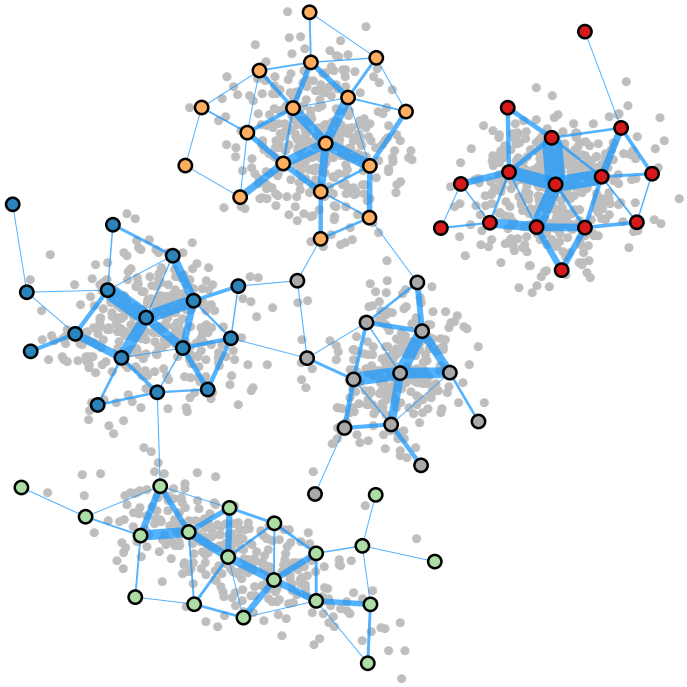


(c) Prototypes (in white) sampled from the data set by the iterative entropy-based procedure.

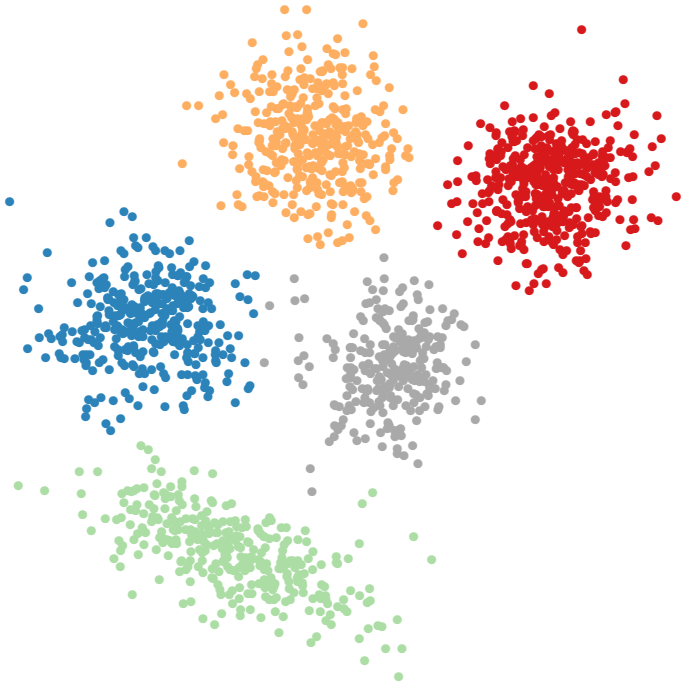


(d) Fuzzy connectivity graph built based on the prototypes.

Figure 4.2: Experimental results on a synthetic data set that consists of several clusters of different shapes and densities in a two-dimensional space (cont.).



(e) Clustering performed by the Louvain method on the fuzzy connectivity graph. Prototypes are colored according to the community to which they are assigned.



(f) Final clustering applied to the entire data set.

Figure 4.2: Experimental results on a synthetic data set that consists of several clusters of different shapes and densities in a two-dimensional space (cont.).

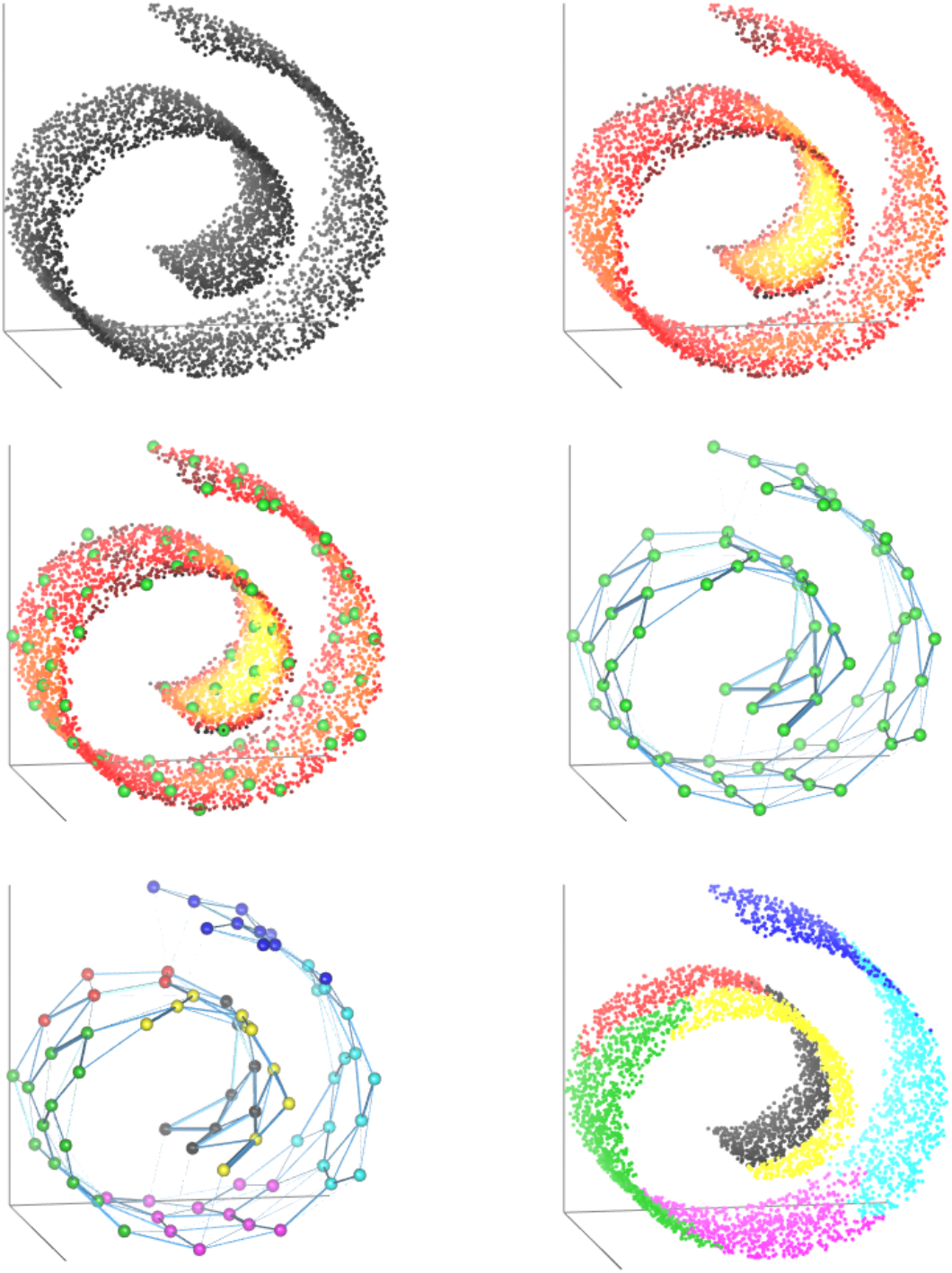


Figure 4.3: Experiment results on a swiss-roll data set.

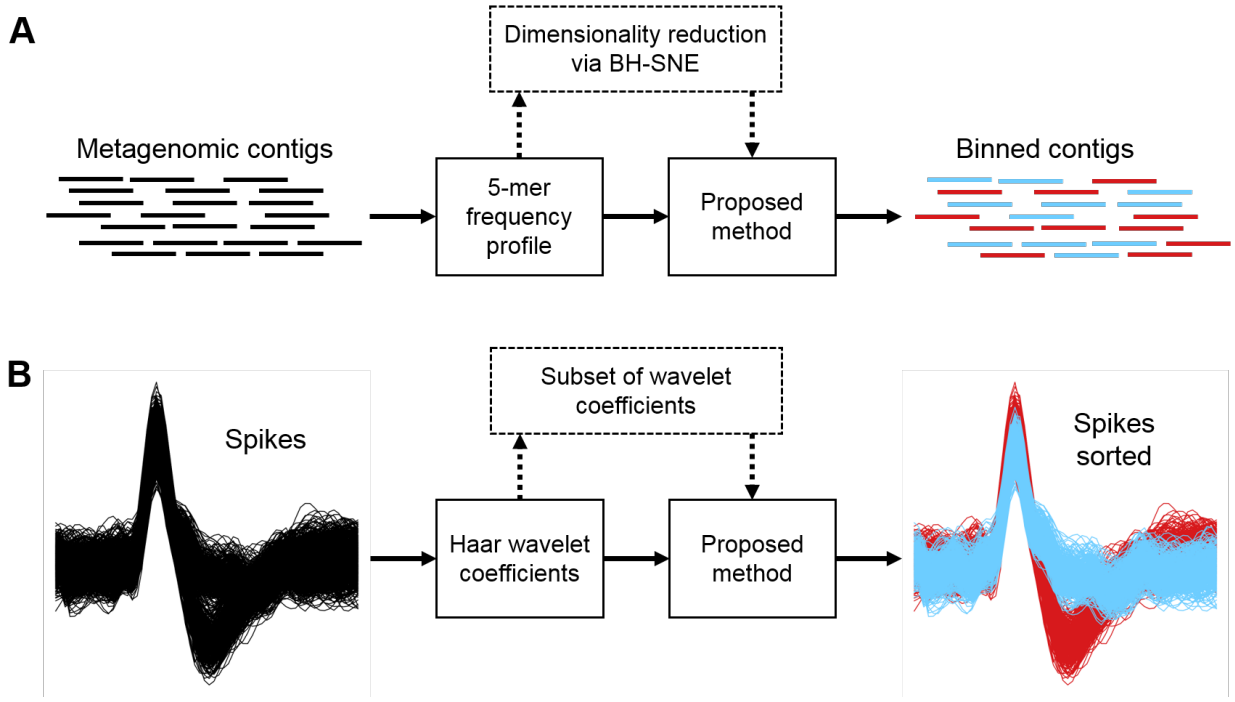


Figure 4.4: Workflow of the experiments conducted on two biological applications of the proposed method. A: Metagenomic binning. B: Spike sorting. Dotted blocks and arrows represent experimental variations.

4.4.2 Experiments on real data

Experiments were also conducted on data sets from two different types of biological applications (Fig. 4.4), both related to problems in which data are organized into unknown yet meaningful groups.

First, we approached metagenomic data sets. Metagenomics is the study of microbial communities through DNA sequencing, sampling directly from their natural environment without prior culturing (Wooley and Ye, 2009). These metagenomic data sets are composed of a mixture of genomic sequence fragments. Our goal is to organize them into groups or "bins" reflecting organismal origin. This process is known as "binning" in metagenomics (Leung et al., 2011). The binning ability of the proposed method was tested on 37 simulated data sets presented in Ceballos et al. (2020) for benchmarking metagenomic binning methods. Metagenomic data sets varied in composition and structure. Each of them comprised fragments from either five or ten genome sequences of distinct microorganisms, and the pairwise genetic relatedness between these genome sequences, measured by the genome similarity metric Average Amino Acid identity (AAI) (Konstantinidis and Tiedje, 2005), is below 55%. Moreover, data sets contained between 6443 and 33,199 fragments, each of them with a length equal to 1000 base pairs.

For a metagenomic data set, let $\mathbf{X}$ be the set of N fragments of t-dimensional features,

and let $\mathbf{x}_i$ denote the i-th fragment. The binning on each data set $\mathbf{X}$ was performed by our method, either directly or indirectly, based on the pentamer frequency profile of each genomic fragment (obtained as is in Ariza-Jiménez et al. (2018)). In the direct approach, the i-th fragment was described by a 512-length frequency profile vector, i.e., $\mathbf{x}_i \in \mathbb{R}^{512}$. Whereas in the indirect approach, prior binning, the above high-dimensional profiles were reduced via the Barnes-Hut t-Stochastic Neighbor Embedding (BH-SNE) algorithm (van der Maaten, 2014) to three dimensions (Ariza-Jiménez et al., 2018), i.e. $\mathbf{x}_i \in \mathbb{R}^3$.

Figure 4.5 shows the binning performance of the proposed method on the ground-truth data sets, benchmarked against a reference method, in terms of two external validity indices, namely, Adjusted Mutual Information (AMI) and the Adjusted Rand Index (ARI). Such indices essentially measure the extent of agreement between the provided binning solution and the available ground-truth binning. All of them give values between zero and one, where one means that the two clustering outcomes match identically (Hubert and Arabie, 1985; Vinh et al., 2010). In the indirect (low-dimensional) approach, the reference method, presented in Ariza-Jiménez et al. (2018), performed unsupervised binning on BH-SNE-based low-dimensional representations of metagenomic data using a combination of Subtractive and Fuzzy c-Means clustering algorithms with the Silhouette index. In the direct (high-dimensional) approach, the reference method was MetaCluster 3.0 (Leung et al., 2011), an unsupervised state-of-art binning method that is compatible with the metagenomic data used in our study.

The overall impression, based on Figure 4.5, is that our method and the reference method perform similarly. Indeed, two-sided sign tests indicated that there is no statistically significant difference between the proposed method and the reference methods in both direct (AMI, $P = 0.998$; ARI, $P = 0.324$) and indirect (AMI, $P = 0.099$; ARI, $p = 0.02$) approaches. In addition, the same results suggest that as data dimensionality increased, the binning performance of our method significantly decreased. We set $\gamma = 2$ for all the experiments and applications described in this section. The only exception was in the indirect binning approach where $\gamma = 1$.

As a second application, the proposed method was used for neuronal spike sorting. That is, given a set of recorded neuronal action potentials, or spikes, our purpose was to sort them according to what hypothetical neuron created them, based on features extracted from each spike waveform (Chaure et al., 2018). Experiments were conducted on simulated data, previously introduced by Pedreira et al. (2012), namely on 80 single-channel extracellular recordings of a varying number of neurons, ranging from 5 to 20. The reference method for this application was a state-of-art spike sorting software known as *Wave_clus* (Chaure et al., 2018). This software was used to detect and isolate spikes automatically from each simulated recording. Among 1063 and 12,618 spikes were obtained per simulated recording.

Given a simulated recording, suppose that $\mathbf{X}$ is the set of N spikes of t-dimensional fea-

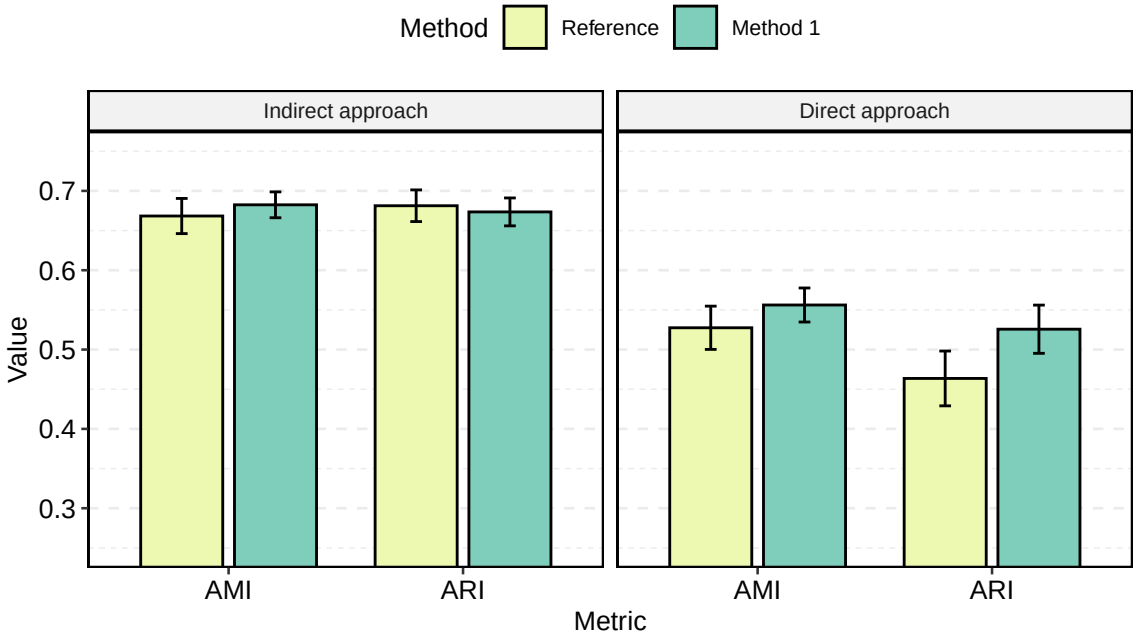


Figure 4.5: Comparison of binning performance between the proposed method and reference methods on 37 ground-truth data sets. External validation measures AMI and ARI are used for evaluating the performance. Mean values and standard error bars are shown. Indirect and directed approaches stand for binning on an embedded three-dimensional space and the original space, respectively.

tures, and let $\mathbf{x}_i$ denote the i-th spike. Spikes were sorted by our method based on two sets of wavelet-based features extracted from each spike waveform. The first set consisted of all the 64 coefficients extracted from the 4-level wavelet decomposition of each spike waveform using the Haar wavelet, while the second one was a variable subset of wavelet coefficients (between 10 and 29 out of 64) that were selected based on an automatic criterion implemented by *Wave_clus*. Therefore, for the full set, $\mathbf{x}_i \in \mathbb{R}^{64}$, and for subset case, $\mathbf{x}_i \in \mathbb{R}^t$, where $10 \leq t \leq 29$.

The performance, in terms of the aforementioned external validation measures, of both spike sorting methods on simulated recordings is shown in Figure 4.6. Results suggests that our method exhibit a lower performance than *Wave_clus*, when the former used all 64 wavelet coefficients as inputs. However, when both methods sorted spikes in the same reduced feature space, our method's performance seems to increase (ARI, $P = 0.033$; AMI, $P = 0.093$). Accordingly, for the proposed method, the results indicate that achieved performance based on selected coefficients was higher than the performance exhibited with 64 wavelet coefficients.

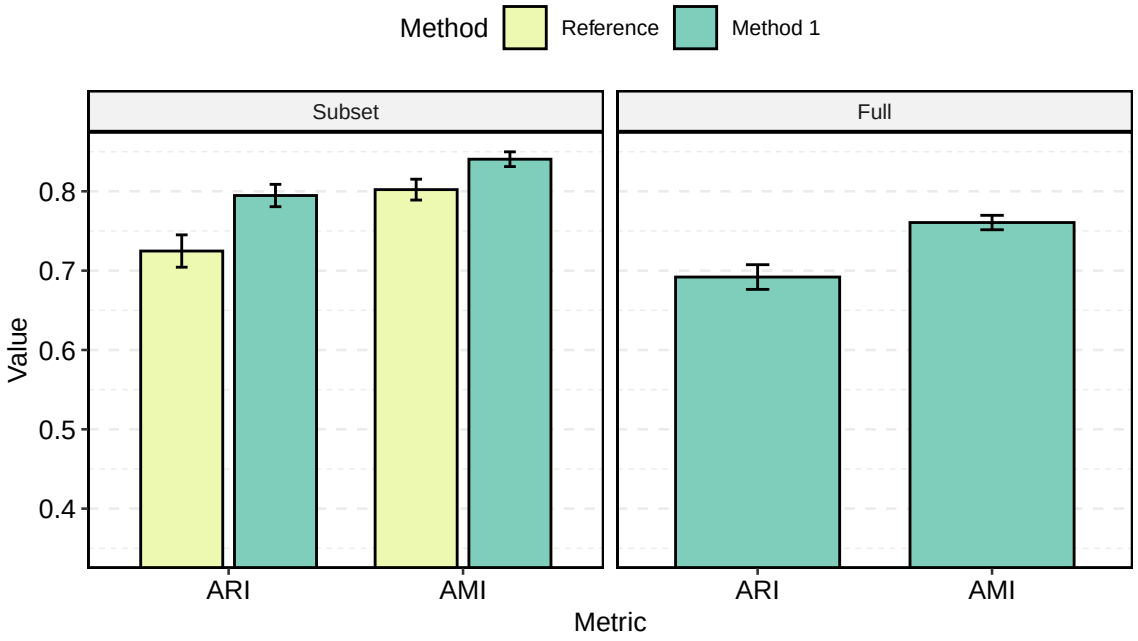


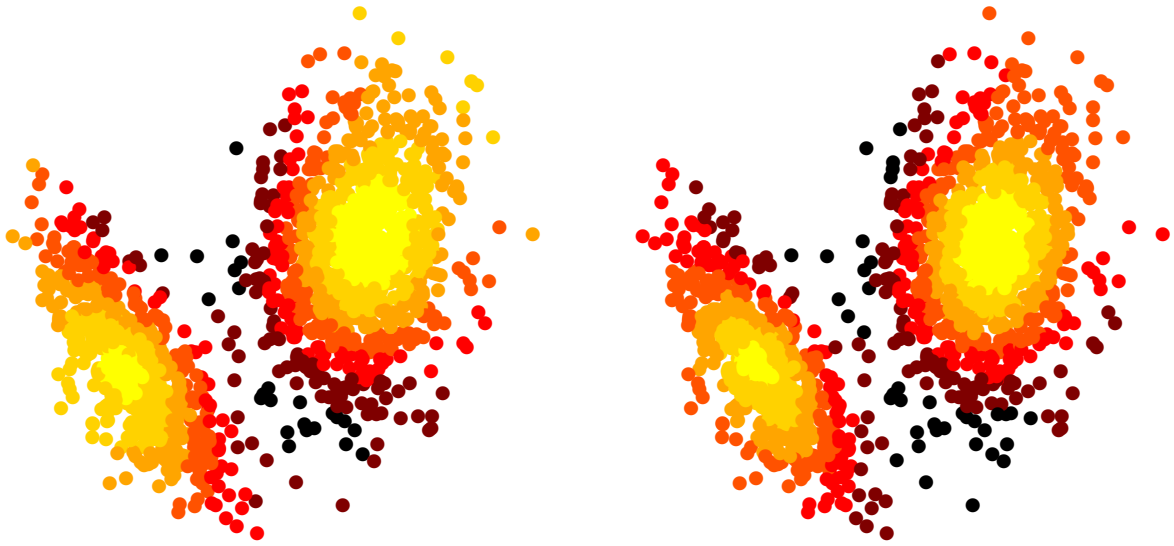
Figure 4.6: Comparison of sorting performance between the proposed method and the reference method on 80 ground-truth data sets. External validation measures AMI and ARI are used for evaluating the performance. Mean values and standard error bars are shown. Full and subset indicate whether results are from sorting spikes based on the full set of wavelet coefficients or a subset of them. There is no reference method for the full case.

4.5 Discussion

An entropy-based graph construction method for representing and finding structure on both synthetic and biological data sets was proposed. Given a data set, our method applies an iterative entropy-based data sampling approach to obtain a subset of representative data objects based on which a fuzzy connectivity graph is built, rather than using all the data objects to obtain a graph-based representation (e.g., see de Arruda et al. (2012); Zhang and Chen (2013); Vogt (2015)). Then, graph clustering is performed to detect structure in data. Experiments evidenced the usefulness and effectiveness of the proposed method on biological applications related to unsupervised learning problems, such as metagenomic binning and neuronal spike sorting, wherein it is expected to organize data into unknown and meaningful groups. Moreover, our proposed method does not require any prior knowledge of the input data, and it displays performance comparable, or higher, than state-of-art methods in the aforementioned biological applications.

The proposed method furthers and integrates the ideas found in Yao et al. (2000) and Laskaris and Zafeiriou (2008). Yao et al. (2000) introduced a method to calculate an entropy-based feature at each data object using together similarity measures and the binary entropy function. This method was initially intended to identify potential cluster centers on data sets (i.e., one per cluster). On the other hand, our method runs in an over-clustering mode (i.e., several per cluster), and to this end, two main modifications were introduced in our method. First, a variation of the usual Gaussian function, i.e. $f(x) = \exp(-\alpha x)$, was used as a similarity measure. Preliminary tests motivate us to propose a function that computes a square root on the pairwise normalized Euclidean distances (4.5) in order to achieve a better characterization of the data objects in a given data set, in terms of the entropy-based feature. Figures 4.7a and 4.7b depict a set of data objects in a two-dimensional space for which their corresponding entropy-based features were obtained, using (4.8), from pairwise similarities based on the usual Gaussian function and the proposed variation, respectively. A visual comparison between both figures evinces that the modified version of the Gaussian function enhances the proper assignation of lower values of the entropy-based feature to the data points located around the center of each group. Second, our method aims to sample the input data set, a process that enables us to obtain a number of representative data objects (the so-called prototypes) for each cluster present in the data set (e.g., see Fig. 4.2c), and it can be interpreted as an over-clustering of the input data set. Specifically, this is achieved by applying an iterative entropy-based strategy that is conditioned by data-driven parameters rather than employing parameters whose value is assigned by a user.

Laskaris and Zafeiriou (2008) suggested the concept of fuzzy connectivity graph, derived from a fuzzy over-clustering, and showed that it captures rich topological information regarding the structure of an unlabeled data set. However, they do not provide an unsu-



(a) Usual Gaussian function. (b) Modified version of the Gaussian function.

Figure 4.7: Entropy-based features of data objects in a two-dimensional space, represented using a hot color scale, as a function of the Gaussian function used as a similarity measure.

pervised strategy for determining a proper number of cluster centers based on which the over-clustering is performed. The number of centers is indeed established *ad hoc*, and they rely on a clustering algorithm, specifically the FCM algorithm, to identify their positions in the data space. Furthermore, they assume that FCM converges satisfactorily to a solution wherein the centers are well distributed throughout the whole data space. In our method, this issue is addressed in an unsupervised way by deriving the fuzzy connectivity graph from the set of prototypes generated through the entropy-based sampling procedure.

In the future, similarity measures potentially less prone to the curse of dimensionality than standard distance measures will be addressed to compute the entropy-based features, especially when it comes to high-dimensional data sets like the metagonomic data sets studied here. This specific consideration has been already taken into account in a recent metagenomic study (Kanj et al., 2018) where a robust similarity measure based on the concept of “shared nearest neighbors” is used.

Bibliography

Ariza-Jiménez, L., Pinel, N., Villa, L. F., and Quintero, O. L. (2020). An Entropy-Based Graph Construction Method for Representing and Clustering Biological Data. In *VIII Latin American Conference on Biomedical Engineering and XLII National Conference on Biomedical Engineering*, pages 315–321, Cham. Springer International Publishing.

Ariza-Jiménez, L., Quintero, O., and Pinel, N. (2018). Unsupervised fuzzy binning of metagenomic sequence fragments on three-dimensional Barnes-Hut t-Stochastic Neighbor Embeddings. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1315–1318.

Balakrishnan, K. J. and Touba, N. A. (2007). Relationship between entropy and test data compression. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(2):386–395.

Bécavin, C. and Benecke, A. (2011). New dimensionality reduction methods for the representation of high dimensional 'omics data. *Expert Review of Molecular Diagnostics*, 11(1):27–34.

Beck, C. (2009). Generalised information and entropy measures in physics. *Contemporary Physics*, 50(4):495–510.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Ceballos, J., Ariza-Jiménez, L., and Pinel, N. (2020). Standardized Approaches for Assessing Metagenomic Contig Binning Performance from Barnes-Hut t-Stochastic Neighbor Embeddings. In *VIII Latin American Conference on Biomedical Engineering and XLII National Conference on Biomedical Engineering*, pages 761–768, Cham. Springer International Publishing.

Chaure, F. J., Rey, H. G., and Quian Quiroga, R. (2018). A novel and fully automatic spike-sorting implementation with variable number of features. *Journal of Neurophysiology*, 120(4):1859–1871.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA.

de Arruda, G. F., Costa, L. D. F., and Rodrigues, F. A. (2012). A complex networks approach for data clustering. *Physica A: Statistical Mechanics and its Applications*, 391(23):6174–6183.

De Luca, A. and Termini, S. (1972). A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. *Information and Control*, 20(4):301–312.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Geng, X. and Hu, G. (2012). Unsupervised feature selection by kernel density estimation in wavelet-based spike sorting. *Biomedical Signal Processing and Control*, 7(2):112–117.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

Kanj, S., Brüls, T., and Gazut, S. (2018). Shared Nearest Neighbor Clustering in a Locality Sensitive Hashing Framework. *Journal of Computational Biology*, 25(2):236–250.

Kantardzic, M. (2011). *Data Mining: Concepts, Models, Methods, and Algorithms*.

Klir, G. J. and Folger, T. A. (1987). *Fuzzy Sets, Uncertainty, and Information*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Konstantinidis, K. T. and Tiedje, J. M. (2005). Towards a Genome-Based Taxonomy for Prokaryotes. *Journal of Bacteriology*, 187(18):6258–6264.

Lambiotte, R., Blondel, V. D., de Kerchove, C., Huens, E., Prieur, C., Smoreda, Z., and Van Dooren, P. (2008). Geographical dispersal of mobile communication networks. *Physica A: Statistical Mechanics and its Applications*, 387(21):5317–5325.

Laskaris, N. A. and Zafeiriou, S. P. (2008). Beyond FCM: Graph-theoretic post-processing algorithms for learning and representing the data structure. *Pattern Recognition*, 41(8):2630–2644.

Leung, H. C., Yiu, S. M., Yang, B., Peng, Y., Wang, Y., Liu, Z., Chen, J., Qin, J., Li, R., and Chin, F. Y. (2011). A robust and accurate binning algorithm for metagenomic sequences with arbitrary species abundance ratio. *Bioinformatics*, 27(11):1489–1495.

Liu, Y., Hou, T., Kang, B., and Liu, F. (2017). Unsupervised Binning of Metagenomic Assembled Contigs Using Improved Fuzzy C-Means Method. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(6):1459–1467.

MacKay, D. J. C. (2005). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA.

McCormick, T. H., Ferrell, R., Karr, A. F., and Ryan, P. B. (2010). Complex Networks as a Unified Framework for Descriptive Analysis and Predictive Modeling in Climate Science. *Science And Technology*, 4(5):497–511.

Newman, M. E. J. (2003). The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 69(2):1–15.

Pedreira, C., Martinez, J., Ison, M. J., and Quian Quiroga, R. (2012). How many neurons can we see with current spike sorting algorithms? *Journal of Neuroscience Methods*, 211(1):58–65.

Principe, J. C. (2010). *Information Theoretic Learning*. Information Science and Statistics. Springer, New York, NY, USA.

Quintero Montoya, O. L., Villa, L. F., Muñoz, S., Arenas, A. C. R., and Bastidas, M. (2015). Information retrieval on documents methodology based on entropy filtering methodologies. *International Journal of Business Intelligence and Data Mining*, 10(3):280.

Schmidt, A. L., Zollo, F., Del Vicario, M., Bessi, A., Scala, A., Caldarelli, G., Stanley, H. E., and Quattrociocchi, W. (2017). Anatomy of news consumption on Facebook. *Proceedings of the National Academy of Sciences*, 114(12):3035–3039.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(July 1928):379–423.

Traud, A. L., Mucha, P. J., and Porter, M. A. (2012). Social structure of Facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180.

van der Maaten, L. (2014). Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245.

Vinh, N. X., Epps, J., and Bailey, J. (2010). Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *Journal of Machine Learning Research*, 11(95):2837–2854.

Vogt, J. E. (2015). Unsupervised Structure Detection in Biomedical Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(4):753–760.

Wooley, J. C. and Ye, Y. (2009). Metagenomics: Facts and artifacts, and computational challenges. *Journal of Computer Science and Technology*, 25(1):71–81.

Xu, R. and Wunsch, D. C. (2010). Clustering algorithms in biomedical research: a review. *IEEE reviews in biomedical engineering*, 3:120–54.

Yao, J., Dash, M., Tan, S. T., and Liu, H. (2000). Entropy-based fuzzy clustering and fuzzy modeling. *Fuzzy Sets and Systems*, 113(3):381–388.

Zhang, H. and Chen, X. (2013). Network-based clustering and embedding for high-dimensional data visualization. *Proceedings - 13th International Conference on Computer-Aided Design and Computer Graphics, CAD/Graphics 2013*, pages 290–297.

Chapter 5

Entropy-based node strength and graph-based data representations

Contents

5.1 Introduction

High dimensionality is a common feature of real-world data; so is the absence of labels, that is, meaningful information about each piece of data. Among these two features, it is the former that represents a notable challenge to methods focused on uncovering the underlying structure of real-world data.

The problems that obstruct the labor of extracting useful information or knowledge from high dimensionality data have been well studied, and they are commonly referred to using the term “the curse of dimensionality”.

Experimental results presented in the previous chapter attested to a reality: The effectiveness of our method significantly increased as data dimensionality decreased. Indeed, this is an expected result since dimensionality reduction methods were applied to improve the data description and avoid the curse of dimensionality, either based on feature selection as in the spike sorting setting or feature extraction as in the metagenomic binning problem.

From a different perspective, as the conventional clustering methods, ours then fail to produce meaningful clusters when data originate from a high-dimensional space. A primary reason behind this behavior could be the use of a distance metric to compute the entropy-based features, like the Euclidian metric that is said to be not suited for estimating

proximity of data objects in high-dimensional spaces (Aggarwal et al., 2001). Therefore, an immediate solution should be focused on employing appropriate proximity measures that are less sensitive to the dimensionality of a given data space.

Traditional distance measures, such as the family of Minkowski distances, can be used as the basis of secondary similarity measures. An early secondary similarity measure is based on the idea that data objects should be considered similar to the extent that they share the same near neighbors (Jarvis and Patrick, 1973), which in turn are determined according to a primary distance measure. Recently, shared nearest neighbor measures have been reported to be more effective than conventional distance measures for high-dimensional data (Houle et al., 2010). Inspired by this, here is presented an approach that involves the use of shared nearest neighbor information to improve the construction of the graph-based models proposed in the previous chapter. The experimental results suggest that this new approach enables our proposed method to have a significantly better performance, in comparison with the method's former implementation and state-of-art methods, in the task of organizing high-dimensional data into unknown and meaningful groups on data sets of biological relevance.

The rest of the chapter is structured as follows. In Section 5.2, we provide some background on k-nearest neighbor graphs and the shared nearest neighbor distance measure, while Section 5.3 describes a variant of the proposed method that incorporates secondary similarity measures. Section 5.4 presents the experiments we performed on both synthetic and real-world data, and Section 5.5 concludes the chapter.

5.2 Background

5.2.1 *k*-nearest neighbor graphs

Consider a set of N t-dimensional data objects $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_j \in \mathbb{R}^t$. Then, let $\mathcal{N}_i^k$ be the set of k nearest neighbors of the data object $\mathbf{x}_i \in \mathbf{X}$, according to a distance metric d.

A k-nearest neighbor (kNN) graph of $\mathbf{X}$, denoted by $G_{kNN} = (\mathbf{X}, E_{kNN})$, is a directed graph in which the elements of $\mathbf{X}$ themselves are the nodes in the graph, and the set of edges is defined by $E_{kNN} = \{(\mathbf{x}_i, \mathbf{x}) : \mathbf{x}_i \in \mathbf{X} \text{ and } \mathbf{x} \in \mathcal{N}_i^k\}$. Further, a mutual k-nearest neighbor (MNN) graph of $\mathbf{X}$ is an undirected graph $G_{MNN} = (\mathbf{X}, E_{MNN})$, in which the elements of $\mathbf{X}$ are connected by a set of edges, given by $E_{MNN} = \{\{\mathbf{x}_i, \mathbf{x}_j\} : \mathbf{x}_i \in \mathcal{N}_j^k \text{ and } \mathbf{x}_j \in \mathcal{N}_i^k\}$ (Von Luxburg, 2007).

Besides having each other in their respective neighborhood sets, a pair of data objects $\mathbf{x}_i$ and $\mathbf{x}_j$ may share some neighbors. Accordingly, it is customary to define a weight function $\omega : E_{MNN} \to \mathbb{R}$ that assigns a positive weight on an edge $\{\mathbf{x}_i, \mathbf{x}_j\} \in E_{MNN}$ to character-

ize the affinity of the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ based on their shared nearest neighbors. In particular, this kind of weighted variant of a MNN graph of $\mathbf{X}$ is called as a shared nearest neighbor (SNN) graph of $\mathbf{X}$ (Jarvis and Patrick, 1973; Boryczko and Kurdziel, 2008; Xu and Su, 2015; Kanj et al., 2018).

5.2.2 Shared nearest neighbor distance measure

Consider a MNN graph $G_{MNN} = (\mathbf{X}, E_{MNN})$ of $\mathbf{X}$, for a given k. A proximity measure between two nodes in the MNN graph can be defined based on the number of neighbors they share (Boryczko and Kurdziel, 2008):

$$\begin{aligned} d_{SNN} &: \mathbf{X} \times \mathbf{X} \to [0, k] \cup \{\infty\} \\ d_{SNN}(\mathbf{x}_i, \mathbf{x}_j) &\equiv \begin{cases} k - |S_{ij}|, & \{\mathbf{x}_i, \mathbf{x}_j\} \in E_{MNN} \\ \infty, & \text{otherwise} \end{cases} \end{aligned} \tag{5.1}$$

where $S_{ij} = \{\mathbf{x} \in \mathbf{X} : \mathbf{x} \in \mathcal{N}_i^k \text{ and } \mathbf{x} \in \mathcal{N}_j^k\}$ is set of shared nearest neighbors between $\mathbf{x}_i$ and $\mathbf{x}_j$. In particular, d_{SNN} is a semimetric on $\mathbf{X}$.

5.3 Proposed method

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ be a data set of N t-dimensional data objects, where $\mathbf{x}_i \in \mathbb{R}^t$. Here our goal is to improve the construction of the graph-based models proposed in the previous chapter. To this end, a secondary distance measure based on the concept of shared neighbors (which in turn are determined based on a primary distance measure) is incorporated in the first stages of the construction process of the aforementioned graph-based models. The rationale behind this decision is to enhance the iterative entropy-based data sampling procedure proposed in Section 4.3, which it is the cornerstone of the subsequent stages. Therefore, key modifications are introduced in the first two stages whereas the latter two stages remain the same. All of the above is achieved through the use of a SNN graph of $\mathbf{X}$.

The first stage begins with the normalization of the input data $\mathbf{X}$ using (4.4) and the subsequent construction of a SNN graph $G_{SNN} = (X, E_{SNN})$, for a given k, based on Euclidean distances. In this SNN graph, data objects with a higher level of proximity, given by the amount of neighbors they share, are connected each other in comparison with disconnected data objects. Then, each edge is first weighted with a value that describe the SNN-based distance between the data objects at its ends. This initial weight function is defined as follows

$$\begin{aligned} w_d &: E_{SNN} \longrightarrow [1, k] \\ &\{\mathbf{x}_i, \mathbf{x}_j\} \mapsto d_{SNN}\left(\mathbf{x}_i, \mathbf{x}_j\right) . \end{aligned} \tag{5.2}$$

Convention 5.1. For simplicity of notation, the edge weight $w_d\left(\{\mathbf{x}_i, \mathbf{x}_j\}\right)$ will be denoted by $w_{d_{ij}}$.

To measure the similarity between each pair of connected data points in the SNN graph, a second weight function, which maps distance measures into similarities, is defined as follows

$$\begin{aligned} w_s &: E_{SNN} \longrightarrow [0,1] \\ &\{\mathbf{x}_i, \mathbf{x}_j\} \mapsto \exp\left(-\hat{\alpha} w_{d_{ij}}\right) , \end{aligned} \tag{5.3}$$

where $\hat{\alpha}$ is a scale parameter. We can compute $\hat{\alpha}$, in a manner similar to (4.7), based on all the individual edge weights $w_{d_{ij}}$ of the SNN graph:

$$\hat{\alpha} = -\frac{\ln 0.5}{\text{median}\left(w_{d_{ij}}\right)} . \tag{5.4}$$

As a results of the above definition, edges with an initial weight equal to the median of all of the individual edge weights $w_{d_{ij}}$ in the SNN graph will then receive a second weight $w_{s_{ij}} = 0.5$.

Convention 5.2. For abbreviation, the edge weight $w_s\left(\{\mathbf{x}_i, \mathbf{x}_j\}\right)$ will be denoted as $w_{s_{ij}}$.

As in the previous chapter, the second stage consist of a two-step procedure: computing an entropy-based feature for each data object and then perform an iterative procedure to sample the data set based on the above features. To compute the entropy-based feature $\hat{H}_i$ of the data object $\mathbf{x}_i$, we employed the SSN graph of $\mathbf{X}$ and the concept of strength of a node (Barrat et al., 2004), which is the weighted analog of node degree. Thus,

$$\hat{H}_i = \sum_{j \in \mathcal{J}} H_{bin}(w_{s_{ij}}) , \tag{5.5}$$

where $\mathcal{J} = \{j : \{\mathbf{x}_i, \mathbf{x}_j\} \in E_{SNN}\}$. In particular, this quantity measures an entropy-based strength of each node in terms of the total weight of their connections to other nodes, that is, the pairwise similarities mapped by the binary entropy function. The subsequent sampling process starts with the identification of the data object with the highest entropy-based strength. Then, this first sample and their neighbors are discarded from the SNN graph. A second sample is selected by looking for the data object with the highest entropy-based strength among the remaining nodes in the SNN graph, and the above discarding procedure is applied again. This iterative procedure is performed until no vertex remains in SNN graph. As before, the set of sampled data objects are the so-called prototypes of the input data set $\mathbf{X}$.

Regarding the last two stages, once a set of prototypes are obtained from the input data

X, a fuzzy connectivity graph is build based on them and the resulting graph is clustered using the Louvain method. Finally, the non-prototypes data objects are organized into groups based on how the prototypes were clustered (see Section 4.3 for details).

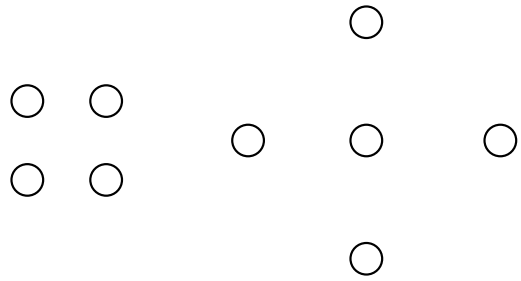
5.4 Experimental results

Like in the previous chapter, experiments were first conducted on synthetic data sets and then the applicability of the proposed method on data sets with biological origin is demonstrated. Additionally, performance comparisons between the algorithm introduced in this chapter and the proposed in Section 4.3 are presented.

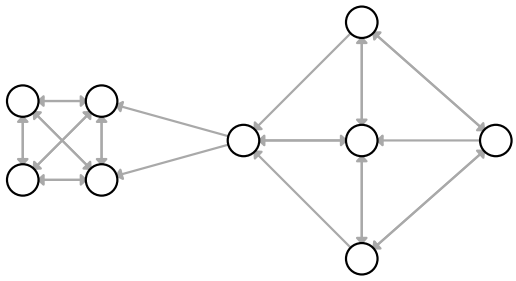
5.4.1 Experiments on synthetic data

A toy data set is first used to exemplify the incorporation of the SNN graph notions onto the process of computing the data objects' entropy-based features. Figure 5.1a shows the nine data points that compose the toy data set, all of them placed in a two-dimensional space. Next, a 3-nearest neighbor (3NN) graph and the corresponding MNN graph are built upon the toy data set, as depicted in Figures 5.1b and 5.1c, respectively. Then, weighted versions of the above MNN graph, i.e. SNN graphs, are generated based on the edge weight functions w_d and w_s. These SNN graphs are shown in Figs. 5.1d and 5.1e. Next to each edge is shown the weight assigned by the functions w_d and w_s. It can be observed that the four data points on the left side are closer to each other and thus they share more neighbors between them, in comparison with the remaining data points on the right side of the toy data set. Consequently, edges connecting data points in the latter group receive distance-based weights with the lowest values (Fig. 5.1d) as well as similarity-based weights with the highest values (Fig. 5.1e).

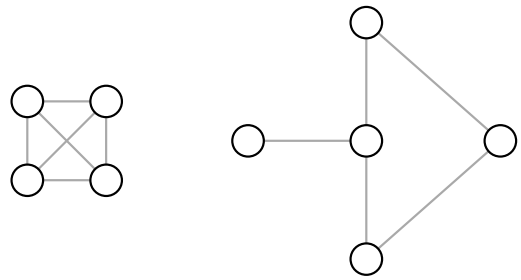
As before, the corresponding entropy-based feature of each data point is described using a hot color scale (varying from yellow, through orange and red, to black) in Fig. 5.1f. This feature is computed for each node, according to (5.5), by adding up the values obtained after mapping, using the binary entropy function, the weights $w_{s_{ij}}$ of the edges attached to the node. For clarity, next to each edge is shown the outcome of the aforementioned mapping. Two factors determine the magnitude of the node's entropy-based feature, namely, the degree of the node and the outcome of the similarity-based weight mapping. Indeed, it is expected that a node becomes "more entropic" as its degree increases, according to (5.5), as well as when its incident edges have mainly weights $w_{s_{ij}}$ equal to 0.5. The latter due to they previously have weights $w_{d_{ij}}$ that equal the median of all of the edge weights $w_{d_{ij}}$ in the SNN graph. This situation is evident in Fig. 5.1f when comparing the only node with degree three at the right group and any node with the same degree at the left side. In particular, note that median$(w_{d_{ij}}) = 1$, in Fig. 5.1d.



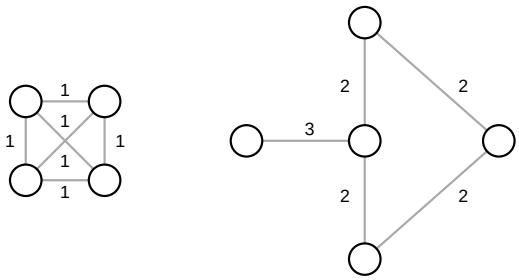
(a) Toy data set.



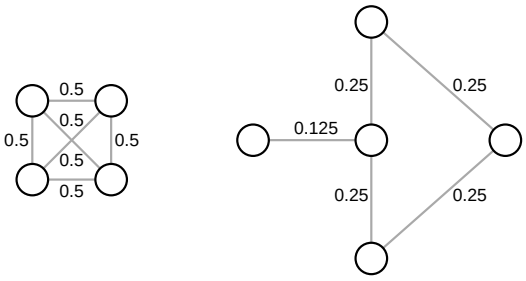
(b) 3NN graph.



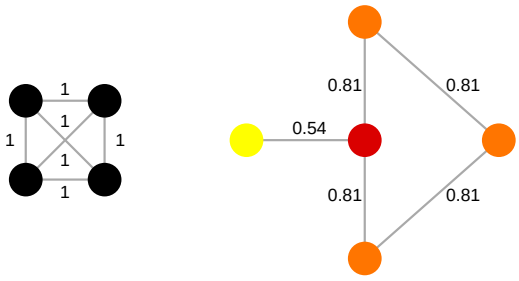
(c) MNN graph ($k = 3$).



(d) SNN graph with weight function w_d.



(e) SNN graph with weight function w_s.



(f) Entropy-based feature of each data point.

Figure 5.1: Entropy-based features of data objects in a two-dimensional space, represented using a hot color scale, as a function of the Gaussian function used as a similarity measure.

A second experiment was performed on a synthetic data set that had been previously presented in Section 4.4.1. The overall process of the method proposed in this chapter on the above synthetic data set is shown in Fig. 5.2. Figure 5.2b depicts the entropy-based feature of each data object employing a hot color scale. These features were obtained following the steps illustrated in Fig. 5.1. In this second experiment, the number of nearest neighbors was set equal to 10% of the total number of data points in the data set (i.e., $k = 176$). Figure 5.2b evidences that, in opposition to the results showed in Fig. 4.2b, "more entropic" data objects, represented in black, can be found near the center of each cluster in comparison with the "less entropic", in yellow, which are located in the cluster boundaries or regions with a lower density of points.

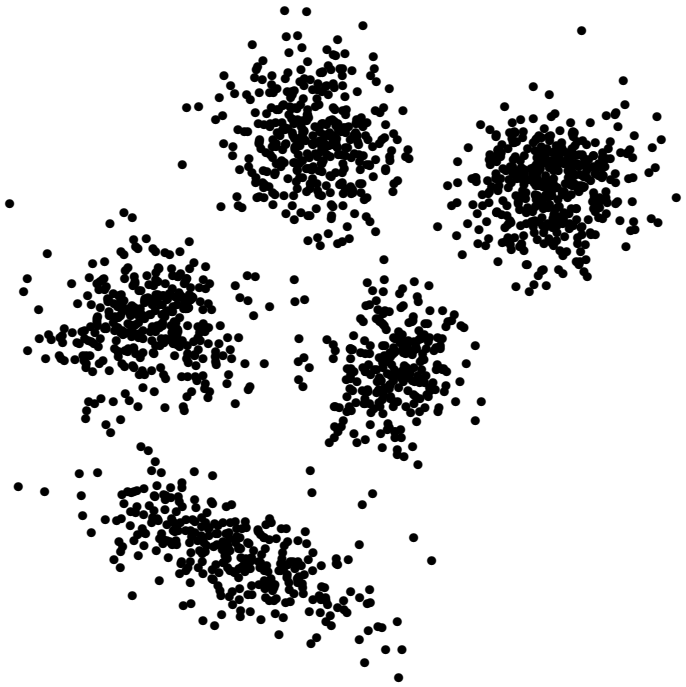
Prototypes found by the iterative sampling procedure are colored in white in Fig. 5.2c, and the fuzzy connectivity graph built based on them is shown in Fig. 5.2d. In particular, it can be observed that this variation of the proposed method provides a graph-based skeleton able to reflect how groups are constituted and related to each other and to describe the data set's density distribution through the thickness of its edges (e.g., ticker edges lying in denser regions).

Convention 5.3. To distinguish the method presented in Section 4.3 from the one we have been evaluating so far, we will refer to these methods as Method 1 and Method 2, respectively.

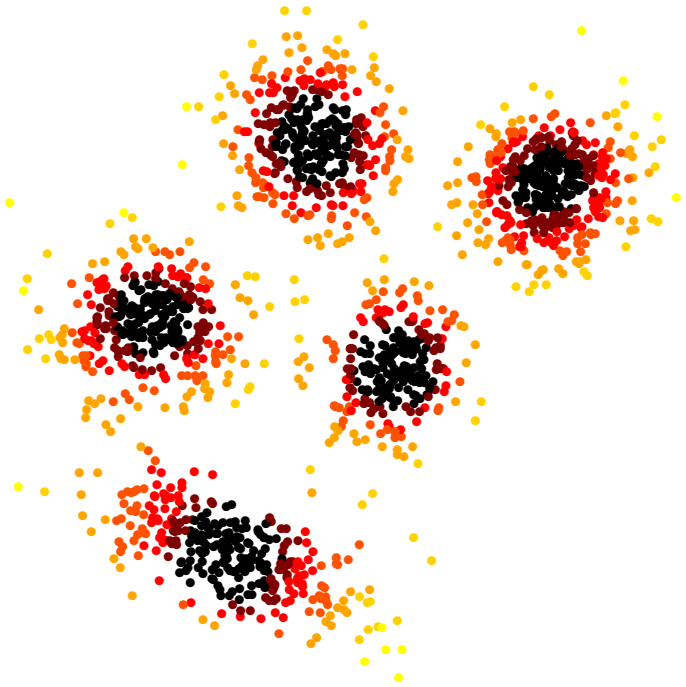
As stated at the beginning of Section 5.3, what motivates the incorporation of a secondary distance measure based on the concept of shared neighbors is the enhancement of the sampling procedure performed by the proposed Method 1 on a given data set. To demonstrate the validity of our decision, we conducted experiments on real-world data sets that are commonly used to test the performance of clustering algorithms. Six data sets were downloaded from UCI Machine Learning Repository[1] and then we executed the sampling procedure on them according to the steps described in Method 1 and Method 2. Specifically, the data sets used for experimentation are *iris*, *thyroid*, *ecoli*, *cancer*, *wine*, and *soybean*.

The results of the above experiments are shown in Fig. 5.3. Since the dimensionality of each data set is greater than three, a BH-SNE-based visualization is employed to depict the results of the experiments. In each row of the figure is displayed the outcome of each method on the same data set. For clarity, the name, size (N), and dimensionality (t) of each data set is shown in the left side of the figure's rows. In addition, the input parameters each method needs are shown on the top side of the figure's rows. These parameters were set in such a way that the number K of samples or prototypes identified by each algorithm almost agree. Prototypes are represented with different glyphs and colors to distinguish them from the remaining data objects in each data set. Fig. 5.3 indicates that, in general, Method 1, which is based on SNN graphs, can provide prototypes (blue triangles in right plots) with

[1] https://archive.ics.uci.edu/ml/index.php

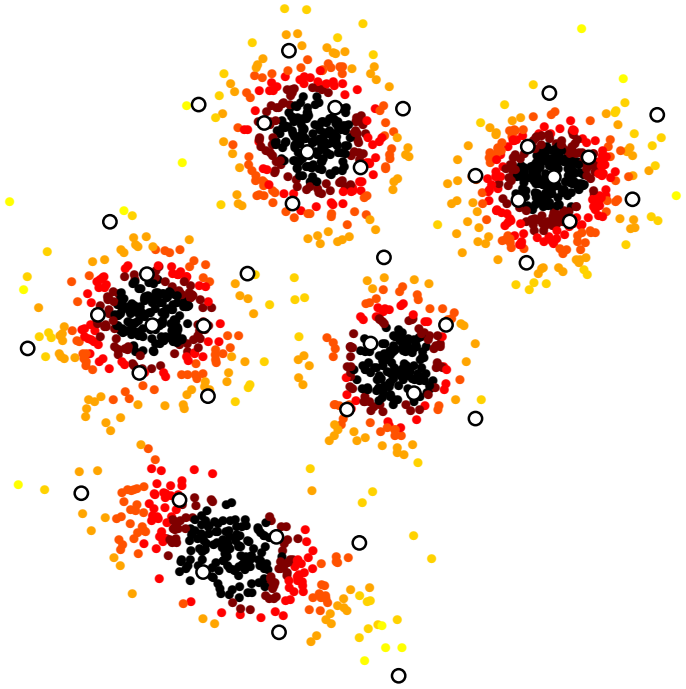


(a) Synthetic data set.

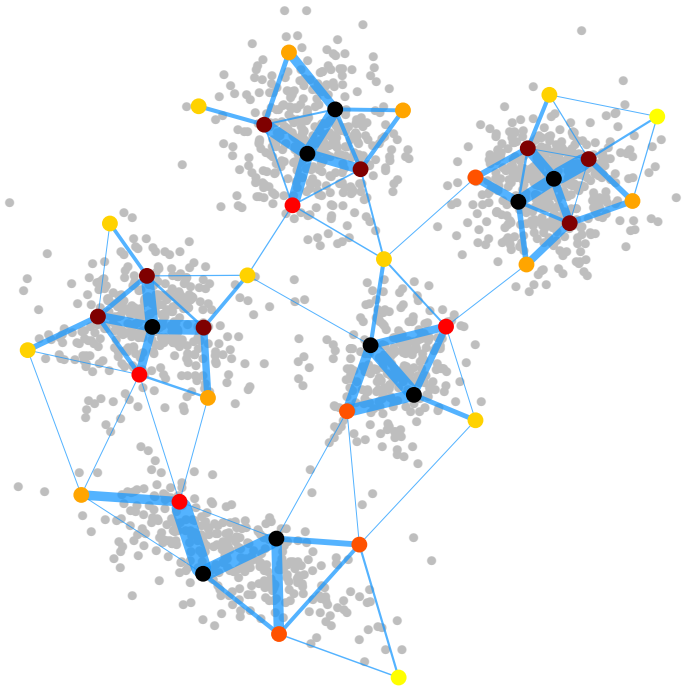


(b) Representation of the entropy-based feature of each data object employing a hot color scale.

Figure 5.2: Experimental results on a synthetic data set that consists of several clusters of different shapes and densities in a two-dimensional space.

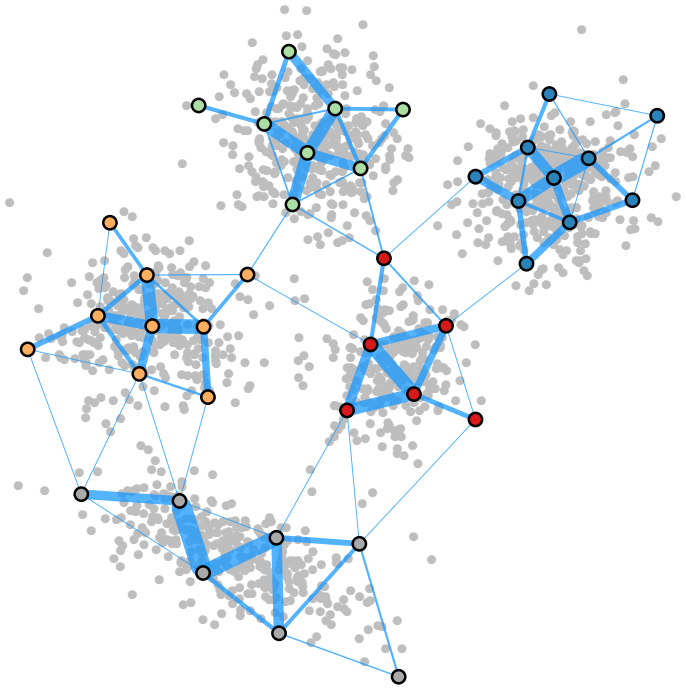


(c) Prototypes (in white) sampled from the data set by the iterative entropy-based procedure.

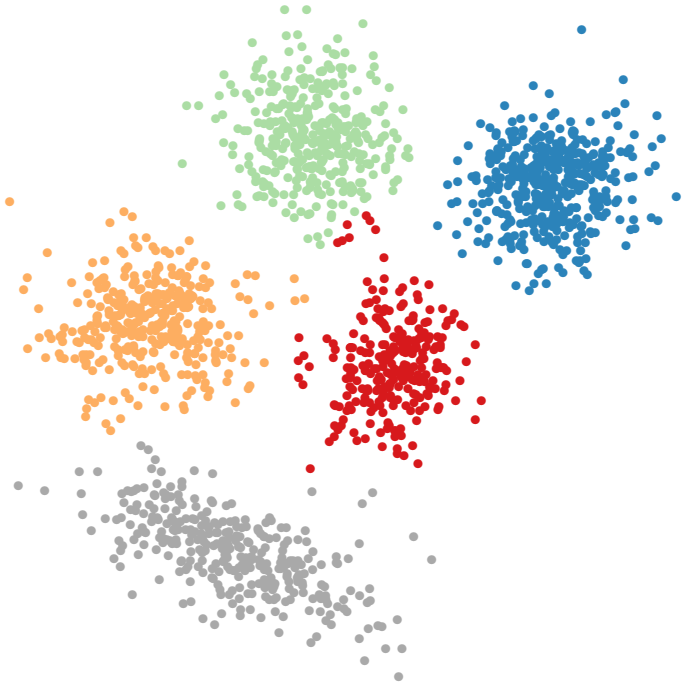


(d) Fuzzy connectivity graph built based on the prototypes.

Figure 5.2: Experimental results on a synthetic data set that consists of several clusters of different shapes and densities in a two-dimensional space (cont.).



(e) Clustering performed by the Louvain method on the fuzzy connectivity graph. Prototypes are colored according to the community to which they are assigned.



(f) Final clustering applied to the entire data set.

Figure 5.2: Experimental results on a synthetic data set that consists of several clusters of different shapes and densities in a two-dimensional space (cont.).

better distribution across the data embeddings in comparison with Method 2 (red squares in left plots). In particular, these results support our intuition about an improved data sampling procedure can be achieved when distance measures derived from the concept of shared neighbors are used.

5.4.2 Experiments on real data

We also extend the performance comparisons between Method 1 and Method 2 to the biological applications described in the preceding chapter, namely, metagenomic binning and spike sorting, in which data is organized into unknown and meaningful groups[2]. The results of both methods were statistically compared using paired sign tests. In-depth details about the above applications and data sets involved in the comparisons presented below are given in Section 4.4.2.

Figure 5.4 shows the performance of Method 2 on binning ground-truth metagenomic data sets, that is, on organizing a mixture of genomic fragments into groups reflecting organismal origin. Two working approaches are considered here: indirect and direct ones. In indirect approach, binning is carried out on a low-dimensional space, whereas in the direct approach, binning is performed on the original high-dimensional data space. Results are benchmarked against a reference method and Method 1, in terms of three different external validity indices. No significant differences ($P > 0.01$) were found in the indirect approach when the performance of Method 2 is compared with the reference method (AMI and ARI, $P = 0.743$) and Method 1 (AMI and ARI, $P = 0.02$). Nevertheless, in the direct approach, Method 2 was significantly better than the reference method (AMI, $P < 0.01$; ARI, $P < 10^{-4}$) and Method 1 (AMI and ARI, $P < 10^{-5}$). For these experiments, the number of nearest neighbors was set equal to 5% of the total number of fragments in each metagenomic data set.

On the other hand, Figure 5.5 shows the performance of Method 2 in a biological application related with the sorting of neuronal spikes, based on wavelet features extracted from each spike waveform. As before, two working scenarios are considered: spikes are sorted whether based on the full set of wavelet features or a subset of them. Results were compared against a reference method and Method 1 using external validity indices. In particular, when it comes to sorting spikes, Method 2 showed a better performance ($P < 10^{-6}$) than Method 1 and the reference method, independently of the considered scenario. For these experiments, we used a number of nearest neighbors equal to 0.5% of the total number of spikes in each data set.

To further assess the performance of the proposed methods, we measured the level of agreement between the detected and ground-truth number of groups in the above biolog-

[2]Additional experiments and an application of Method 2 on the natural language processing of coronavirus-related scholarly articles are presented in Appendix A.

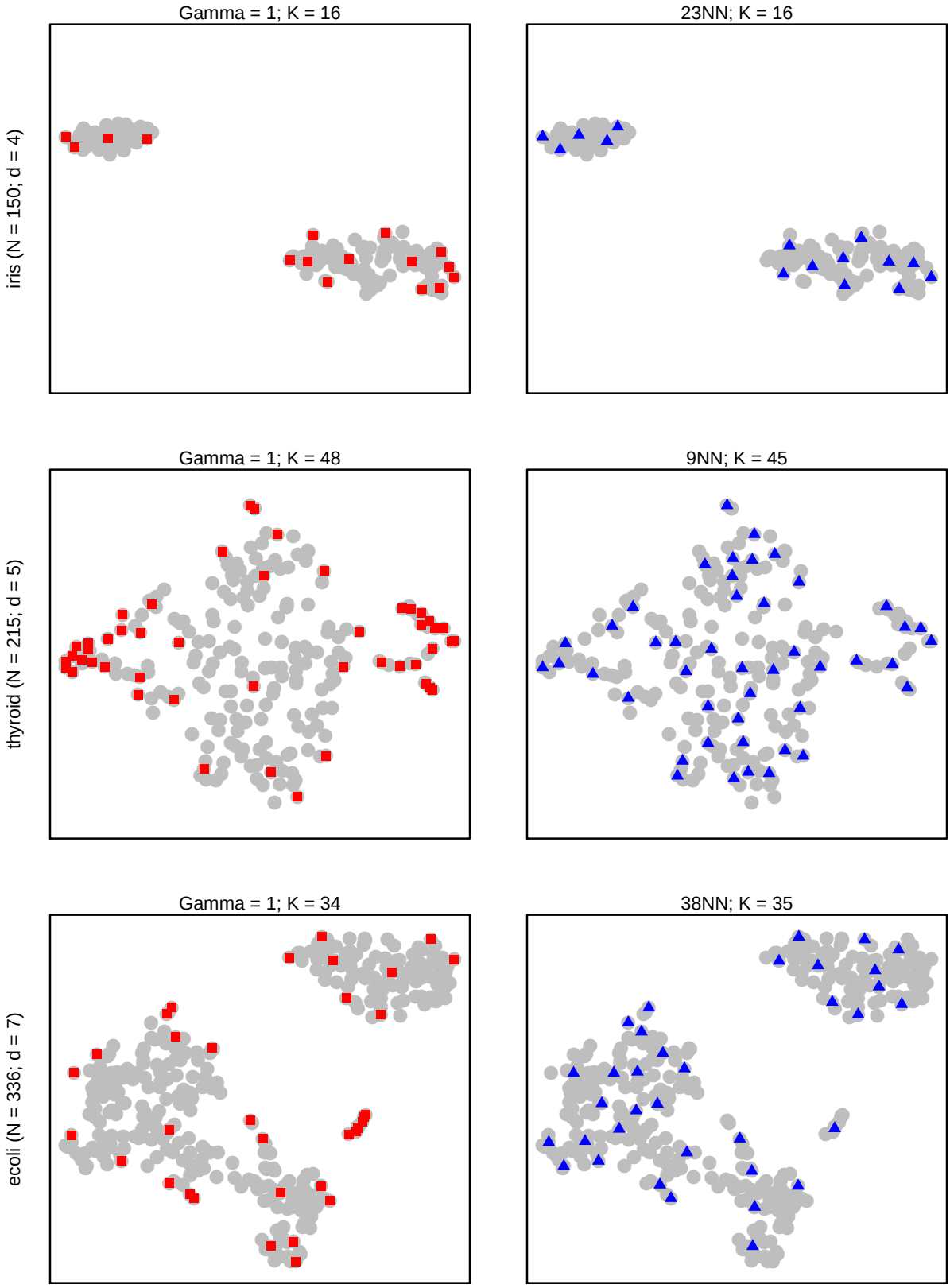


Figure 5.3: Entropy-based data sampling performed on six real-world data sets according to Method 1 and Method 2, left and right plots, respectively.

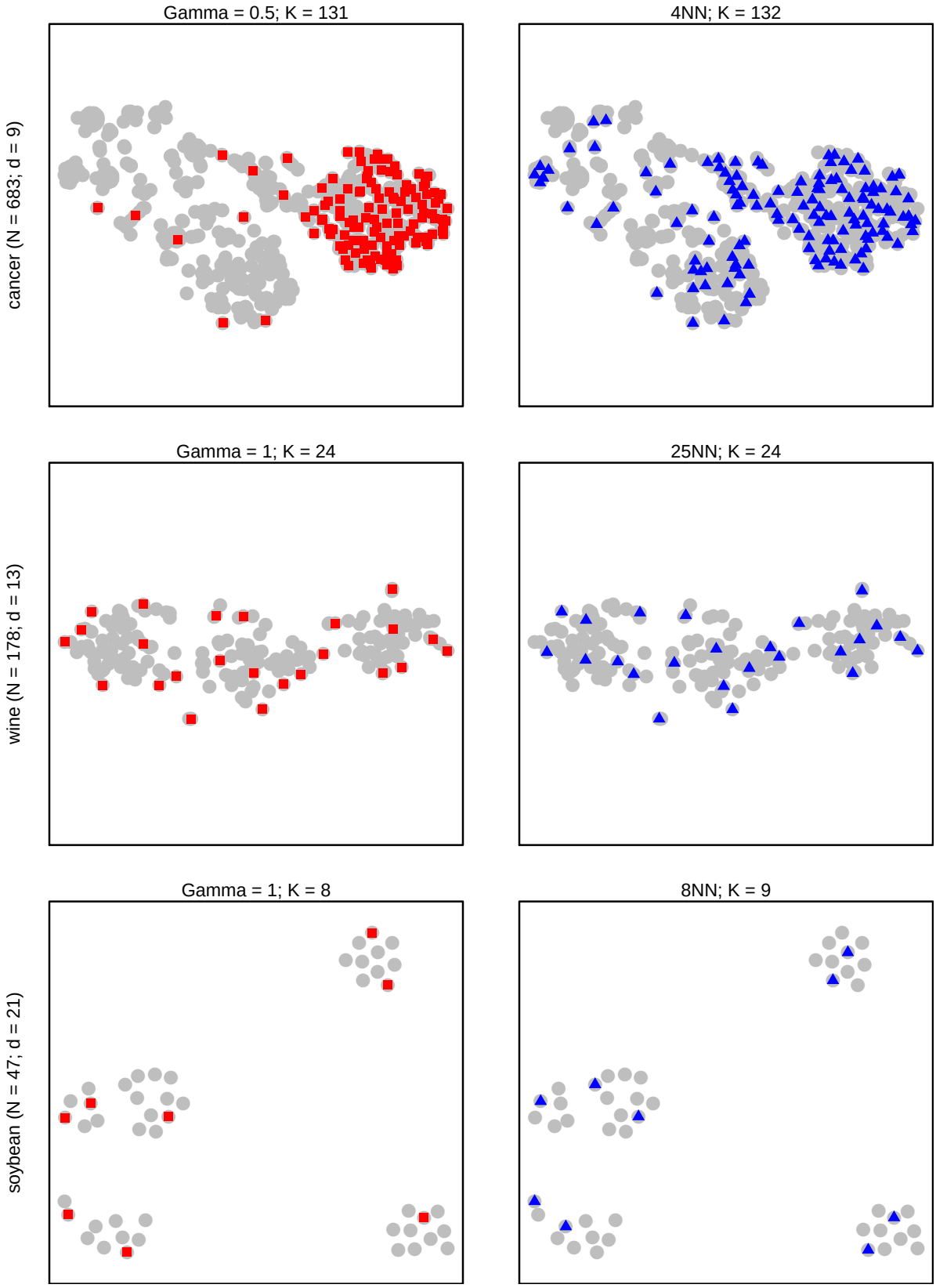


Figure 5.3: Entropy-based data sampling performed on six real-world data sets according to Method 1 and Method 2, left and right plots, respectively.

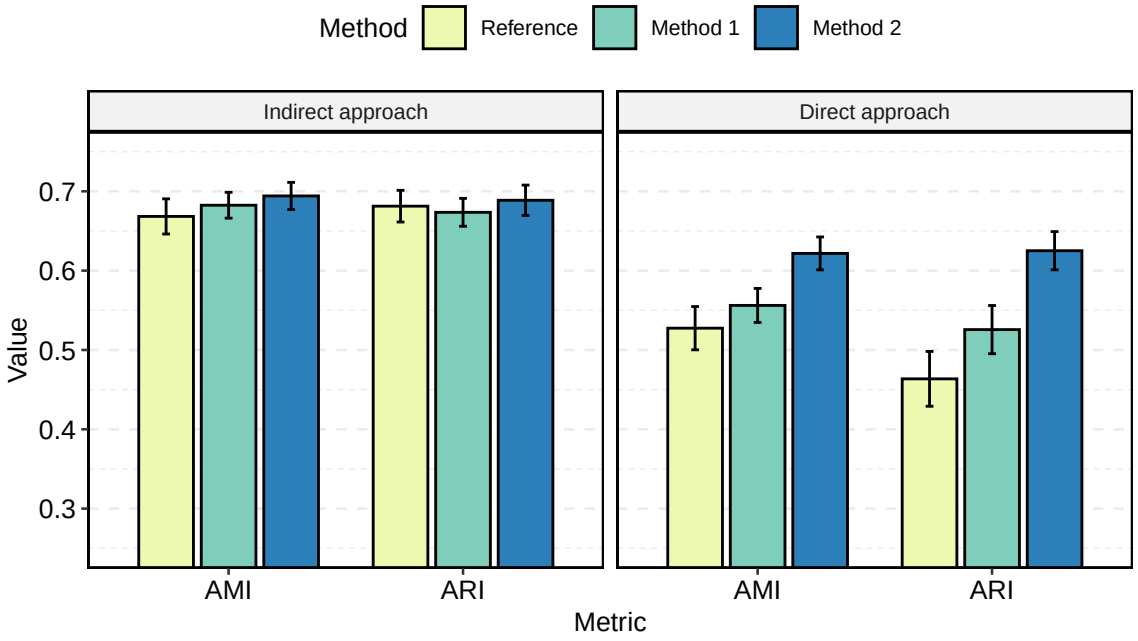


Figure 5.4: Comparison of binning performance between the proposed methods and reference methods on 37 ground-truth data sets. External validation measures AMI and ARI are used for evaluating the performance. Mean values and standard error bars are shown. Indirect and directed approaches stand for binning on an embedded three-dimensional space and the original space, respectively.

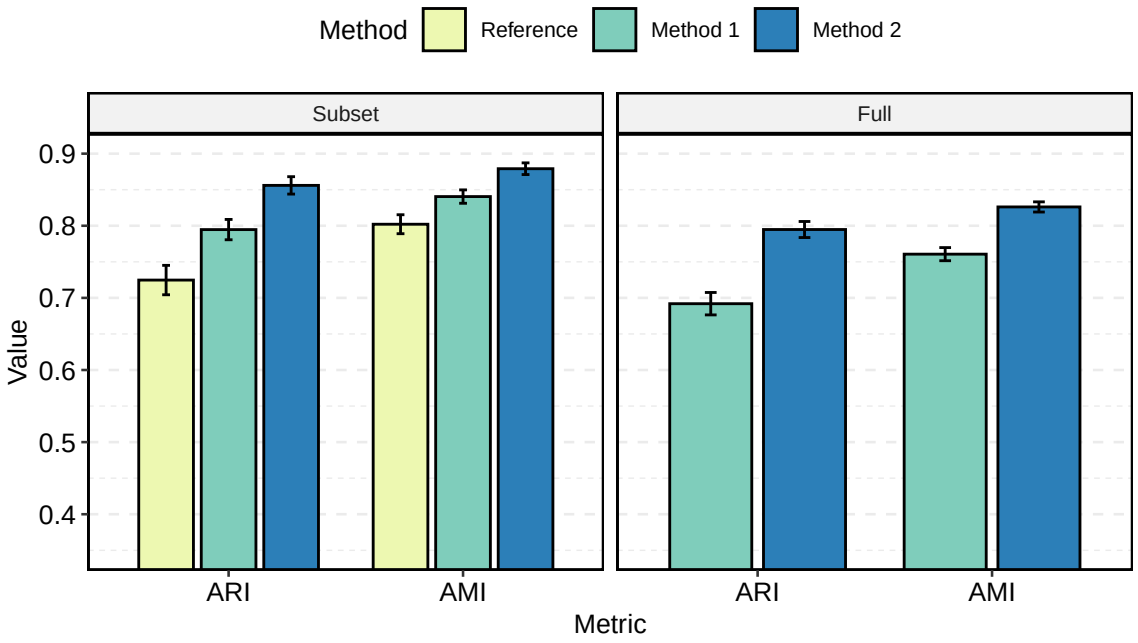


Figure 5.5: Comparison of sorting performance between the proposed methods and the reference method on 80 ground-truth data sets. External validation measures AMI and ARI are used for evaluating the performance. Mean values and standard error bars are shown. Full and subset indicate whether results are from sorting spikes based on the full set of wavelet coefficients or a subset of them. There is no reference method for the full case.

ical applications. Figures 5.6 and 5.7 show the performance of the methods in terms of the absolute relative error in predicting the actual number of genomic populations (in the metagenomic binning scenario) and neurons (in the spike sorting application), respectively. These figures show that Method 2 had a lower amount of misses compared with Method 1 and the reference methods, regardless of the working scenario (i.e., the dimension of the space wherein data is organized in groups) and the biological application.

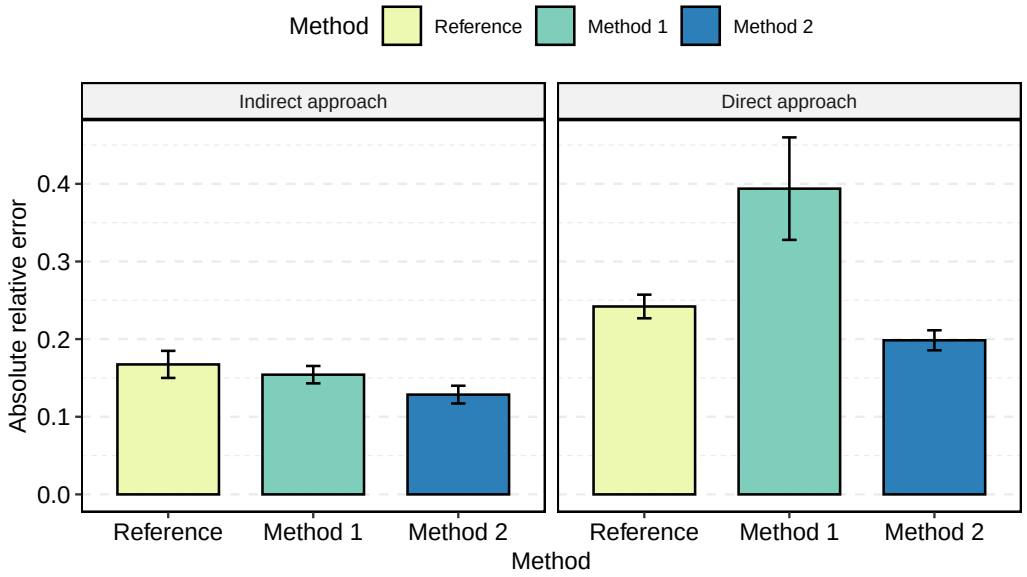


Figure 5.6: Comparison of binning performance between the proposed and reference methods in terms of the absolute relative error in predicting the actual number of genomic populations on 37 ground-truth data sets. Mean values and standard error bars are shown. Indirect and directed approaches stand for binning on an embedded three-dimensional space and the original space, respectively.

5.5 Discussion

A second graph construction method based on entropy features for representing and uncover underlying patterns on both synthetic and biological data was proposed. This second method follows the same stages of the method proposed in the previous chapter, but crucial modifications are introduced to improve the construction of the graph-based representation. These modifications include the use of SNN graphs to initially model the input data, as well as distance and similarity measures based on the concept of shared neighbors, and the concept of node strength, which is the weighted analog of node degree.

Incorporating all of the above elements, in the first stages of the construction process, benefits our work mainly in two ways. First, it enables our method to have a significantly

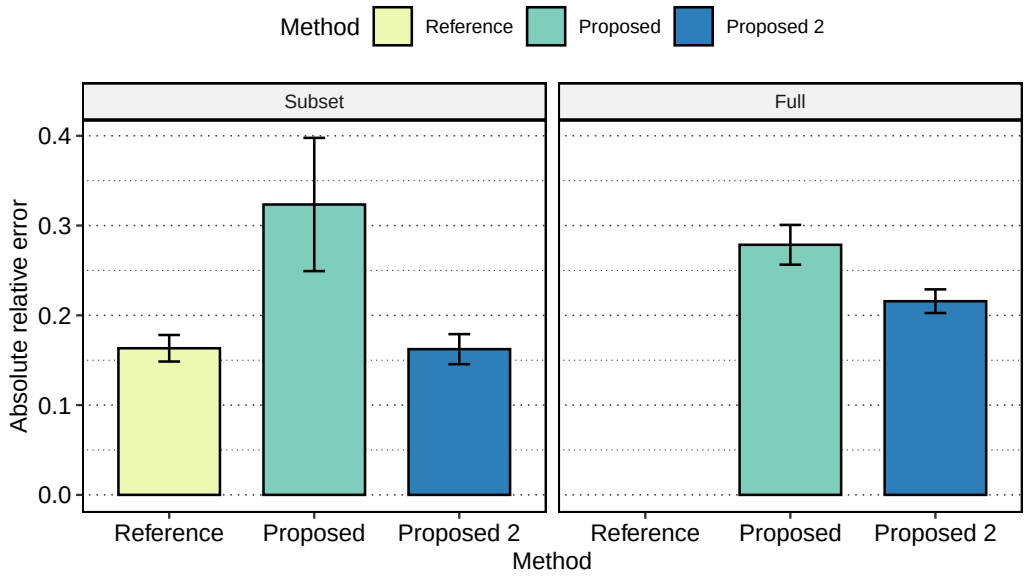


Figure 5.7: Comparison of sorting performance between the proposed and reference methods in terms of the absolute relative error in predicting the actual number of neurons spiking on 80 ground-truth data sets. Mean values and standard error bars are shown. Full and subset indicate whether results are from sorting spikes based on the full set of wavelet coefficients or a subset of them. There is no reference method for the full case.

better performance, in comparison with the method's former implementation and state-of-art methods, in the task of organizing high-dimensional data into unknown and meaningful groups on data sets of biological relevance. Second, it provides the former implementation of our proposed method with a framework based on graph theory concepts wherein distance and similarity measures between data objects are associated with edge weight functions of an SNN graph that models an input data set. Likewise, the entropy-based feature of a data object is now formulated as an entropy-based node strength in the SNN graph. Consequently, this feature is calculated for each data object based on the total weight of its connections to its nearest neighbors.

The development of a proximity measure between any pair of data objects in a given data set and based on the number of neighbors they share is due to Boryczko and Kurdziel (2008). In their work, the SNN distance measure is closely related to the process of obtaining an SNN graph of the input data set. Nevertheless, they limit its use as a decision criterion that allows data objects in the SNN graph, acting as nodes, whether or not to remain connected to reduce the graph size. On the other hand, we extend the role of the SNN distance measure and conceive it as the foundation of several weight functions able to transform the meaning of the edges in an SNN graph. Although diverse attributes have been used to weight the edges of SNN graphs (see e.g., Jarvis and Patrick (1973); Boryczko and Kurdziel (2008); Xu and Su (2015); Kanj et al. (2018)), to the best of our knowledge,

the weighting scheme proposed in our work has not been reported in the literature.

So far, the so-called Louvain algorithm has performed well on clustering the fuzzy connectivity graphs. However, it only considers information regarding edge weights and not the entropy values which were available for prototypes as node attributes. Therefore, the combination of this "metadata" and the information from the fuzzy connectivity graph topology to guide and condition the graph clustering process should be a subject of future research.

Bibliography

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1973:420–434.

Barrat, A., Barthelemy, M., Pastor-Satorras, R., and Vespignani, A. (2004). The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences*, 101(11):3747–3752.

Boryczko, K. and Kurdziel, M. (2008). Approximate clustering of noisy biomedical data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5101 LNCS(PART 1):630–640.

Houle, M. E., Kriegel, H. P., Kröger, P., Schubert, E., and Zimek, A. (2010). Can shared-neighbor distances defeat the curse of dimensionality? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6187 LNCS:482–500.

Jarvis, R. and Patrick, E. (1973). Clustering Using a Similarity Measure Based on Shared Near Neighbors. *IEEE Transactions on Computers*, C-22(11):1025–1034.

Kanj, S., Brüls, T., and Gazut, S. (2018). Shared Nearest Neighbor Clustering in a Locality Sensitive Hashing Framework. *Journal of Computational Biology*, 25(2):236–250.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Xu, C. and Su, Z. (2015). Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, 31(12):1974–1980.

Chapter 6

Further aspects of the graph-based models

Contents

6.1 Introduction

This chapter is devoted to exploring some properties of the graph-based models presented in Chapters 5 and 6. Although the use of this kind of representations is extended (see Section 2.5), it is unusual in the literature to describe the properties of such graph-based models. For instance, Laskaris and Zafeiriou (2008), who introduce the fuzzy connectivity graphs (FCG) based on which we generate graph-based representations, limit themselves to indicate that these are subgraphs of a Delaunay triangulation of the prototypes we sample from an input data set. In particular, we attempt here to address the properties of the developed graph-based models that can be exploited to improve the uncovering of the underlying structure of data.

This chapter is structured as follows. Section 6.2 focused on analyzing the behavior of the entropy-based features associated with each node in the graph-based models and provide additional methods and strategies to complement those presented in this thesis. Section 6.3 looks into the modularity of the graphs that are generated for the biological datasets considered in our experiments. Finally, in Section 6.4, we analyze the time complexity of some of the algorithms developed in this thesis so far.

6.2 On the node entropy-based features

In this section, we will focus on analyzing the behavior of the entropy-based features associated with each node in the graph-based models. Our main purpose is to determine what effects this can have on the functionality of the proposed methods. Based on these findings, we will propose additional methods and strategies to complement those presented in this thesis.

In Chapter 5, a data object $\mathbf{x}_i$ in a data set $\mathbf{X}$ is assigned an entropy-based feature that results from adding up mapped versions of its similarity to all the remaining data objects in the data set (4.8). Since similarities are mapped using the so-called entropy binary function (4.3), which zeroes high values of similarities, data objects with many nearby neighbors, like the ones close to cluster centers or located in high-density regions, obtain an entropy-based feature based on the sum of more near-zero values than the ones located in the cluster boundaries or regions with a lower density of points. Consequently, the former kind of data objects tends to be "more entropic" in comparison with the latter group of data objects, which appear to be "less entropic" (see Fig. 4.2b). Data objects positioned far enough from other data objects can also obtain an entropy-based feature based on the sum of mostly near-zero values, due to the same entropy binary function that also zeroes the low values of similarities. As a result, when the iterative entropy-based procedure is performed to sample the data set, far away data objects could be selected as prototypes and become nodes in the FCG. However, because of their particular location and the mechanism by which the FCG is built, it is unlikely that these data objects can be linked to other prototypes during the process of populating the FCG with weighted edges. Specifically, if a data object $\mathbf{x}_j$ has been selected as a prototype and it is not among the first two nearest neighbors of any data object $\mathbf{x}_i$, with $i \neq j$, then it will remain as an isolated vertex (i.e., a vertex with a degree equal to zero) in the FCG built upon $\mathbf{X}$. Therefore, in such cases, far away data objects can be easily identified from the resulting graph-based representation.

Recall that in Chapter 6 we made use of SNN graphs to compute the entropy-based features for each data object. In such cases, data objects that are not among the set of k nearest neighbors $\mathcal{N}_j^k$ of any other data object $\mathbf{x}_j \in \mathbf{X}$, like data objects away from densely populated regions, will turn into isolated nodes in the SNN graph and their corresponding entropy-based feature will be the minimum possible, i.e. zero, according to (5.5). The identification of such nodes is thus straightforward, and they can be excluded from the subsequent process of data sampling if necessary, thereby preventing them from becoming prototypes in the FCG. On the other hand, the maximum achievable value of the entropy-based feature is equal to k in the SNN graph. This is because the maximum degree of a node in an SNN graph is k and the maximum value of the entropy binary function is equal to one (see Fig. 4.1). Nodes with high entropy values, consequently, can be found in densely populated regions at which nodes tend to share a larger amount of neighbors as they are

closer to each other.

A reasonable and extended assumption is to consider that the natural groupings within data correspond to high-density regions in a feature space separated by low-density regions (Jain, 2010). Based on the assumption and the aforementioned properties of the entropy-based features computed from SNN graphs, it is possible to develop an approach to find natural groupings in data by searching "highly-entropic" data objects. The steps of a strategy that follows this particular approach are summarized below:

1. Build the FCG of a given data set, as indicated in Section 5.3, for a given value of k.

2. Remove from the FCG the prototypes whose corresponding entropy-based feature is below the threshold $T_{\hat{H}}$. The threshold $T_{\hat{H}}$ should close to k but less than it; for instance, $T_{\hat{H}} = \lfloor 0.9k \rfloor$.

3. Find the connected components of the induced subgraph, and assign to each component a unique identifier.

4. Label each node in the induced subgraph with the identifier of the connected component to which it belongs.

5. Run a label propagation algorithm (e.g., Raghavan et al. (2007)) for detecting the community structure in the FCG, based exclusively on the labels assigned in the previous step.

6. Organize, based on (4.11), the non-prototypes data objects into groups according to how prototypes were clustered in the FCG.

To exemplify how this strategy works, we consider the synthetic data set that had been previously presented in Section 5.4.1. This is a typical data set in which clusters correspond to dense regions of data objects in a two-dimensional space surrounded by low-density regions. Figure 6.1a depicts the synthetic data set and the FCG built upon it. As before, we set the parameter k, the number of nearest neighbors, equal to 10% of the total number of data points in the data set; thus $k = 176$. A hot color scale, as usual, is used to represent the entropy-based feature of each prototype in the FCG. Yellow and black represent "less entropic" and "more entropic" data objects, respectively. The distribution of colors evidences that prototypes sampled from high-density regions are "highly-entropic", and thus, as stated above, we can use this information to guide a subsequent process of cluster identification. Figure 6.1b shows how such nodes are detected with the suggested threshold $T_{\hat{H}} = \lfloor 0.9k \rfloor = 158$. Then, a subgraph is induced by deleting the non-detected nodes from the FCG (Fig. 6.1c), and the nodes belonging to each remaining connected component are labeled uniquely (Fig. 6.1d). Rather than use the so-called Louvain algorithm for detecting the community structure of the FCG, we employ a label propagation algorithm because it

is possible to initialize it based on given labeled nodes, which, in this case, are the "highly-entropic" nodes previously detected. Finally, Figs. 6.1e and 6.1f depict the community structure detected by the label propagation algorithm and the cluster structure detected on the synthetic data set, respectively.

Figure 6.1a provides additional information regarding the prototypes in the FCG, namely the order in which they were extracted from the synthetic data set. Numbers are placed near the nodes to indicate the prototype extraction order. "Highly-entropic" nodes, in which we are interested, are extracted first than the other objects. This is because the method proposed in Chapter 5 extracts prototypes from a given data set in decreasing order of their entropy-based feature. To illustrate better this relationship, Fig. 6.2 shows the entropy-based feature of each prototype of the FCG as a function of the order in which it was sampled from the data set.

Figure 6.2 shows that, besides the expected monotonic behavior of the sequence of values, there is a gap that separates the first sixth "highly-entropic" prototypes from the remaining prototypes in the FCG. The existence of this gap motivates us to develop an approach that does not depend on manually setting a threshold to identify the "highly-entropic" nodes from which the clustering structure in data can be discovered.

The alternative approach aims to detect the aforementioned gap in a data-driven manner, similarly to the *eigengap heuristic* proposed for the spectral clustering algorithm (Von Luxburg, 2007), and provide by itself an automatic threshold. Let $\hat{H}_{\mathbf{p}_1}, \hat{H}_{\mathbf{p}_2}, \ldots, \hat{H}_{\mathbf{p}_K}$ be the entropy-based features of the prototypes in the FCG, listed in the order in which the latter were sampled from the data set. Then, we take successive differences between the entropy-based features of the prototypes and divide each of them by a partial cumulative sum. By following this approach, a coefficient g_i is associated with each prototype $\mathbf{p}_i$ as follows:

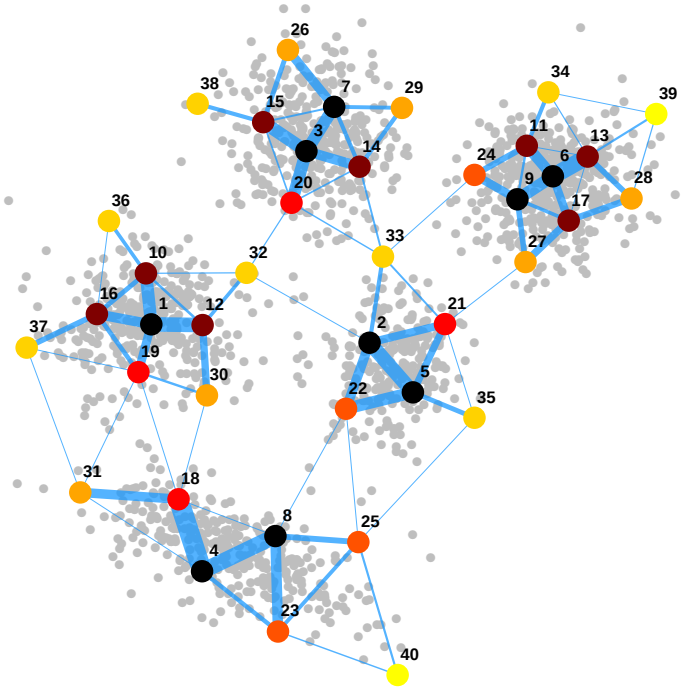
$$g_i = \frac{\hat{H}_{\mathbf{p}_{i-1}} - \hat{H}_{\mathbf{p}_i}}{\sum_{j=1}^{i} \hat{H}_{\mathbf{p}_j}}, \tag{6.1}$$

where $i = 1, 2, \ldots, K$, and K is the total number of sampled prototypes from the data set (as well as the number of nodes in the FCG). For convenience, we take $\hat{H}_{\mathbf{p}_{i-1}} = \hat{H}_{\mathbf{p}_i}$, when $i = 1$. Next, we proceed to identify the prototype $\mathbf{p}_j$ with the largest coefficient g_i, thus:

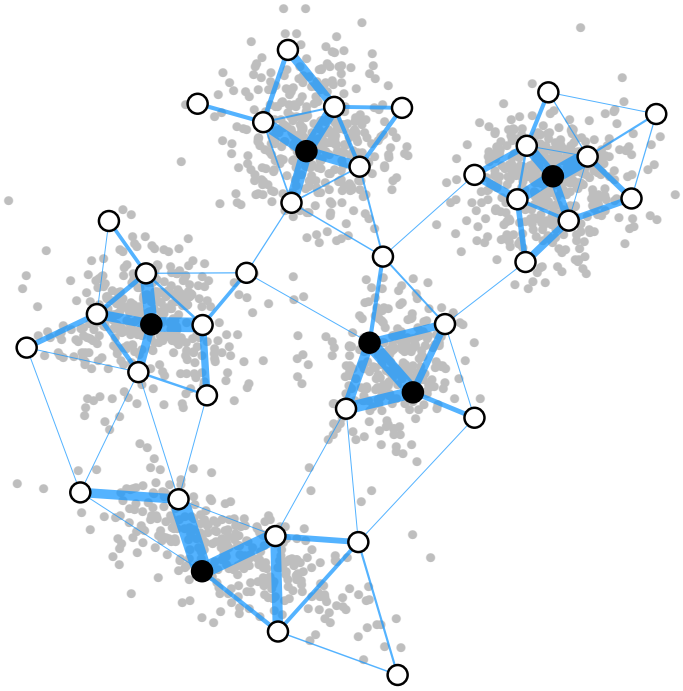
$$j = \underset{i=1,\ldots,K}{\arg\max}\ g_i. \tag{6.2}$$

After that, the entropy-based feature $\hat{H}_{\mathbf{p}_j}$ of the prototype $\mathbf{p}_j$ will be used as a threshold T_g above which the "highly-entropic" nodes of the FCG can be detected.

Figure 6.3 shows the coefficient g_i associated with each prototype in the FCG of Fig. 6.1a against its extraction order. The position where the maximum value of g_i occurs corresponds to the prototype in Fig. 6.1a whose entropy-based feature is immediately below the thresh-

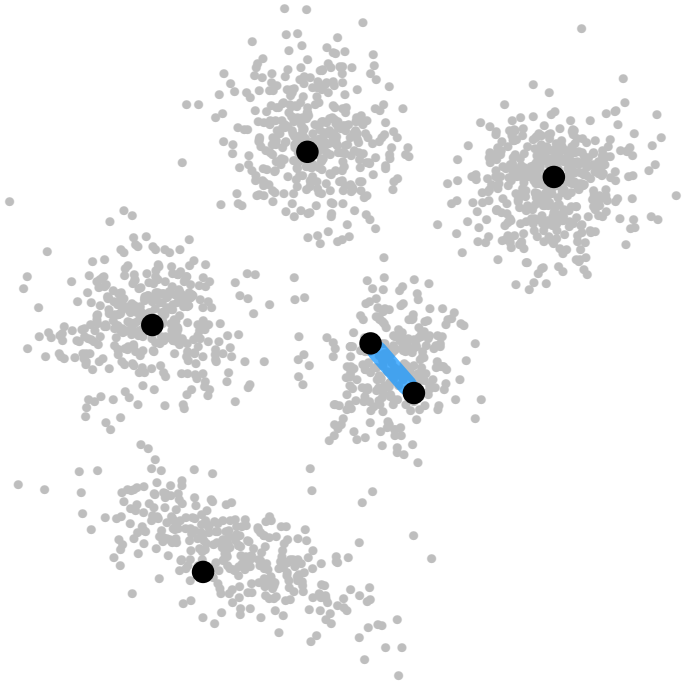


(a) FCG built upon the synthetic data set. The usual hot color scale is employed for clarity purposes. Darker colors correspond to "highly-entropic" vertices. Numbers are placed near the nodes to indicate the prototype extraction order.

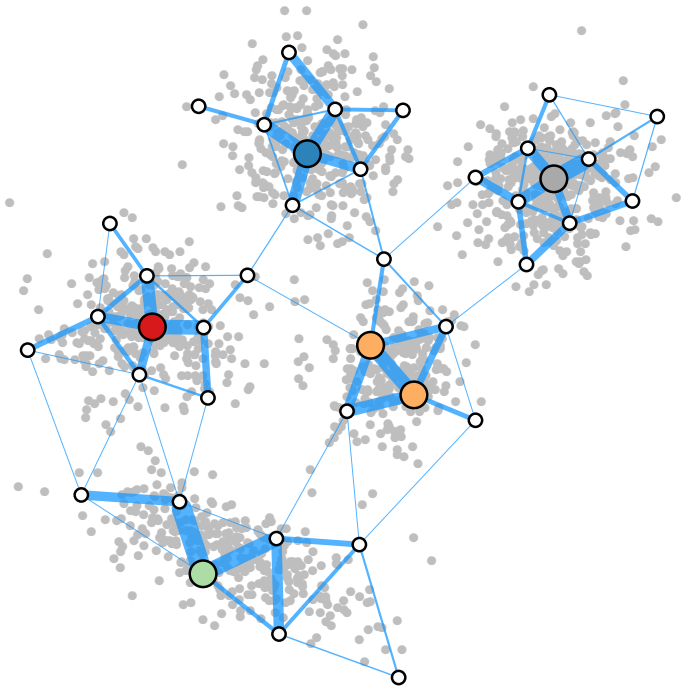


(b) Prototypes colored depending on whether its entropy-based feature is above (black) or below (white) the threshold $T_{\hat{H}}$.

Figure 6.1: Experimental results on a synthetic data set that consists of several clusters of different shapes and densities in a two-dimensional space.

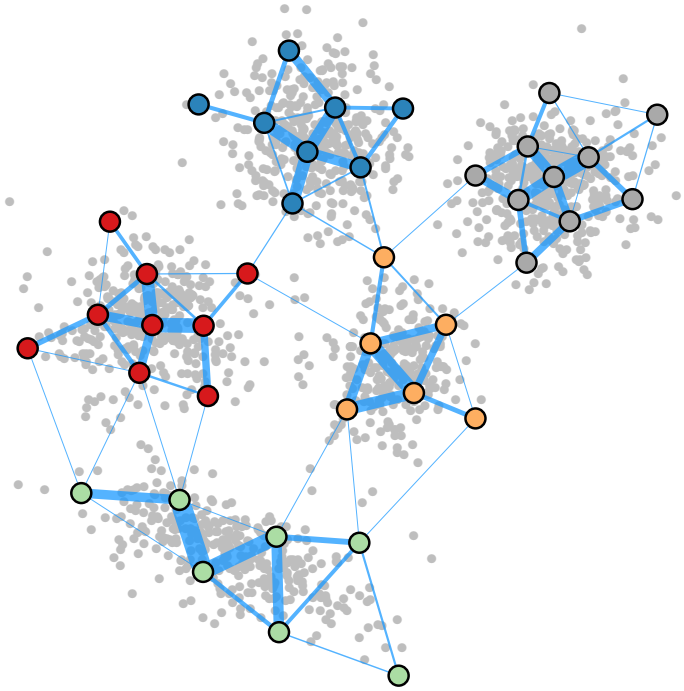


(c) Subgraph induced by removing the nodes whose entropy-based feature is below the threshold $T_{\hat{H}}$ from the FCG.

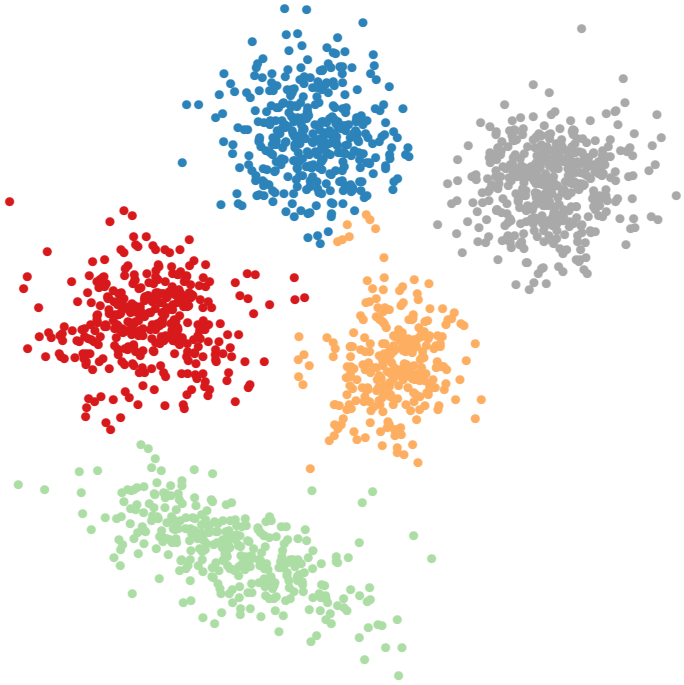


(d) "Highly-entropic" nodes labeled (using colors) depending on the connected component to which they belonged in the induced subgraph.

Figure 6.1: Experimental results on a synthetic data set that consists of several clusters of different shapes and densities in a two-dimensional space (cont.).



(e) Community structure detected by a label propagation algorithm on the FCG. The algorithm is initialized with the labels assigned to the "highly-entropic" nodes. Prototypes are colored according to the community to which they are assigned.



(f) Final clustering applied to the entire data set.

Figure 6.1: Experimental results on a synthetic data set that consists of several clusters of different shapes and densities in a two-dimensional space (cont.).

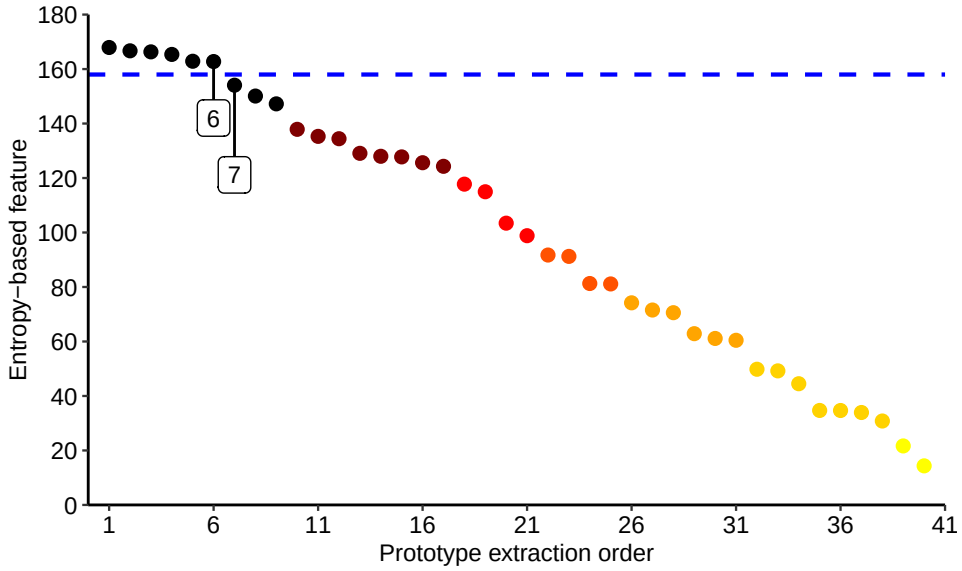


Figure 6.2: Entropy-based feature of the prototypes in the FCG of Fig. 6.1a against the prototype extraction order. The dashed line, which coincides with a gap in the sequence of values, represents the threshold $T_{\hat{H}}$ applied on Fig. 6.1b. The positions below and above the threshold $T_{\hat{H}}$ have been labeled. The usual hot color scale is employed for clarity purposes.

old $T_{\hat{H}}$. Therefore, we have that $T_g = T_{\hat{H}} = 158$ in this particular example.

The above experimental results motivated us to undertake an extensive performance assessment of the novel strategies presented in this section as future research. In addition, we will continue exploring the properties of the graph-based models generated by the methods presented in the previous chapters in order to improve the detection of structure in data. A preliminary result in this work line suggests that the FCG generated by our approach are *small-world networks*, according to a metric introduced by Humphries et al. (2006). Small-world networks, whose name resembles the *small world phenomenon* in which people on Earth are said to be separated by six degrees, are characterized by two main features: dense inter-connectivity within small groups of nodes and the average shortest path is small (Humphries et al., 2006; Watts and Strogatz, 1998). Consequently, we plan to study the interaction between random walk-based algorithms for community detection and our graph-based models, since such algorithms operate based on the assumption that the closeness between nodes in the same community is significantly less than the graph's average shortest path (Coscia et al., 2011).

6.3 Modularity of the graph-based models

In Chapters 4 and 5, we presented methods that model multivariate data using graphs through which the underlying structure in data can be discovered. Our approach relies

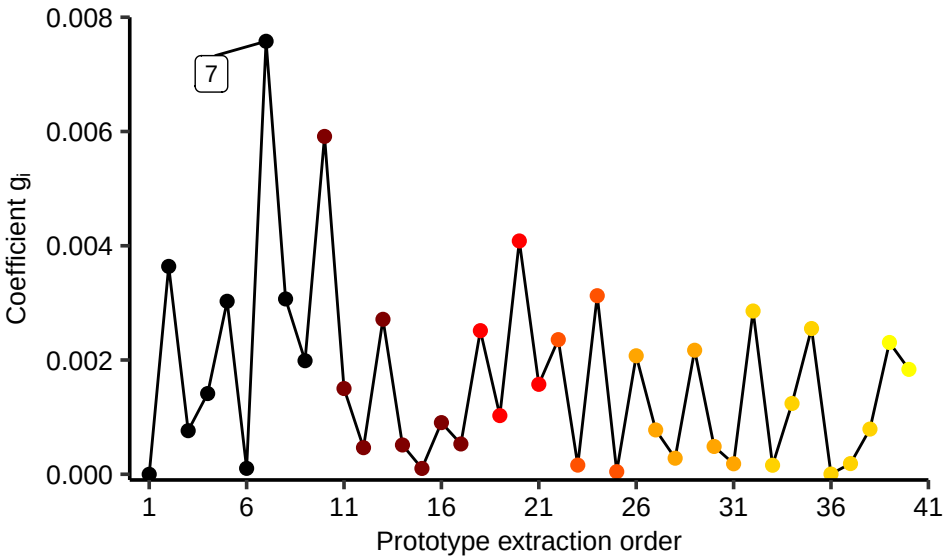


Figure 6.3: Coefficient g_i of each prototype in the FCG of Fig. 6.1a against its extraction order. The position where the maximum value of g_i occurs has been labeled. The usual hot color scale is employed for clarity purposes.

on the underlying assumption that the resulting graph-based models can translate such structures from a multi-dimensional feature space to community structures in the graph-based models. To throw light on this aspect, in this section, we will look into the *modularity* of the graphs that are generated for the biological data sets considered in our experiments. The goal is to provide additional results that support the rationale of our approach and highlight conditions that influence on the effectiveness of the proposed methods.

The modularity, initially proposed by Newman and Girvan (2004), is a measure of the quality of a particular division of a graph into disjoint communities (i.e., groups of connected nodes). It basically compares the actual edge density of each community with the edge density obtained for the same group of nodes for a randomly rewired network (Barabási, Albert-László and Pósfai, 2016). Let $G = (E, V)$ be an undirected and unweighted graph whose nodes are organized into a set $\mathscr{C}$ of M disjoint communities. The modularity $Q_{\mathscr{C}}$ of that specific division of G into communities can be defined as

$$Q_{\mathscr{C}} = \sum_{i=1}^{M} \left[\frac{L_i}{L} - \left(\frac{\deg(\mathscr{C}_i)}{2L} \right)^2 \right], \tag{6.3}$$

where L is the total number of edges in the graph, L_i is the total number of edges joining vertices of the i-th community, and $\deg(\mathscr{C}_i)$ is the sum of the degrees of the vertices of this community. In particular, (6.3) gives an account of the fraction of all edges that lie within communities minus the expected value of the same quantity in a graph wherein vertices have retained their degree but are randomly connected regardless of the communities to

which they belong (Newman, 2004). Modularity takes values between $-1/2$ and 1 (Brandes et al., 2008). Negative values can be obtained when each node is assigned to its own community, whereas a zero value can be obtained when all the nodes are assigned to the same community. Higher modularity values imply a better division of the nodes into communities whose internal connectivity pattern has not emerged by chance. In practice, values greater than 0.3 are associated to graphs with high-quality community structures (Clauset et al., 2004; Newman, 2004).

To examine the extent to which natural group structures in a given data set are organized into network communities by the proposed methods, we compute the "ground-truth modularity" of its corresponding graph-based representation. That is, the modularity of a graph-based model built for a data set when its nodes are organized into communities given by the underlying ground-truth structure of this data set. Figure 6.4 depicts the ground-truth modularity of each biological data set considered in our experiments. In all cases, the modularity values are positive and range from 0.2 to 0.9. Such high modularity values thus indicate that the natural groupings existing in data are being translated into strong community structures in the graph-based models. These results also suggest that the proposed methods tend to provide better representations for the spike-waveform-related data sets than for the metagenomic data sets. Due to how our methods operate, this may be an indication that the wavelet-based features describe better the underlying structure of the spike waveform data in comparison with the pentamer frequency profiles of the genomic fragments.

On the other hand, the scatter plots presented in Fig. 6.4 also evidence the relationship between the modularity values and the performance of the proposed methods on the ground-truth data sets. As before, performance is measured in terms of external validity indices, namely the Adjusted Rand Index (ARI) and the Adjusted Mutual Information (AMI), which have values between zero and one. Particularly, we observe that there is a strong positive correlation between the two quality measures. In other words, our methods' ability to uncover the inner structure of a data set tends to increase when the ground-truth modularity associated with the graph-based model of the same data set increases. This trend could be explained by the fact that our methods discover structure in data based on a community detection process performed on the FCG by the Louvain method, which is a modularity optimization algorithm.

In the future, we plan to perform further experiments to validate our findings using a combination of different measures for characterizing network community structures and additional community detection algorithms.

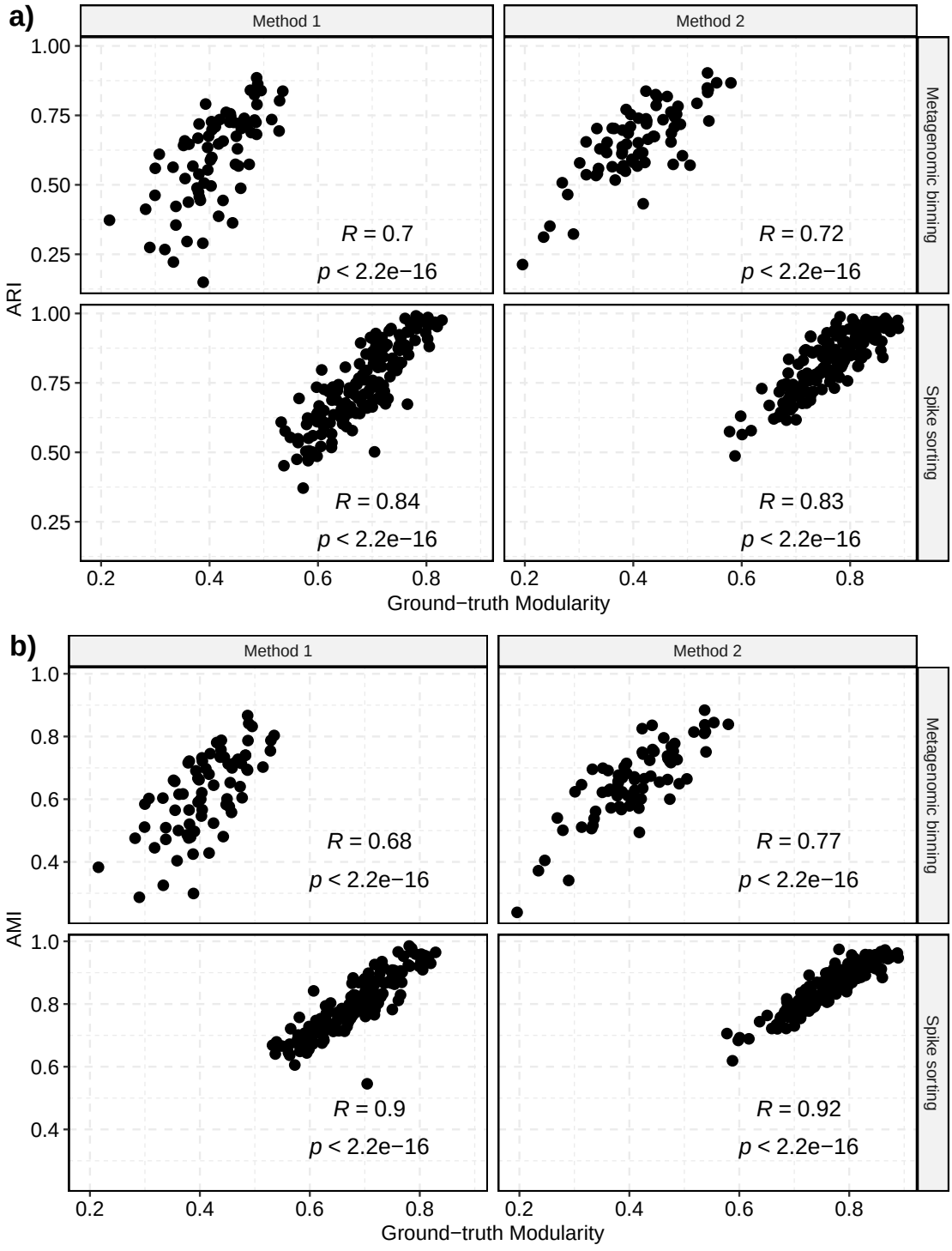


Figure 6.4: Performance of the proposed methods on each of the ground-truth biological data set considered in our experiments against the ground-truth modularity of the graph-based model built on the same data set. External validation measures, a) AMI and b) ARI, are used for evaluating the performance. For convenience, Spearman's rank correlation coefficients and their significance are included.

6.4 Time complexity analysis

In this section, we analyze the time complexity of some algorithms developed in this thesis so far. To this end, we will use the Big-O notation, $O(\cdot)$, to measure the time complexity or running time of these methods for the worst-case scenario.

We start by analyzing the method proposed in Chapter 4. Consider a dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^t$ The time complexity of measuring the similarity between every pair of data objects $\mathbf{x}_i$ and $\mathbf{x}_j$ is $O(N^2)$. The running time needed to compute the entropy-based feature of a data object is $O(N)$. Since we do this step for each of the N data objects, the total complexity amounts to $O(N^2)$ for all of them. In each iteration of the entropy-based sampling process, finding a prototype and removing all its close neighbors has a computational complexity of $O(M^2)$, where M is the current number of data objects and $M \leq N$. Supposing that K iterations are performed, then the total complexity of this process is $O(KM^2)$, where $K \ll N$. The construction of the FCG has a computational complexity of $O(NK^2)$. The running time of the Louvain algorithm is about $O(K \log K)$ (Newman, 2018), where K is the number of nodes in the FCG. Next, the resulting clustering is applied to the remaining data objects in $O(NK)$. Therefore, the overall complexity of Method 1 is $O(N^2)$.

The time complexity of the method proposed in Chapter 5 is described as follows. The complexity of searching the k nearest neighbors of a data object is $O(N)$, so for the N data objects in $\mathbf{X}$ is $O(N^2)$. The construction of the SNN graph, which requires pairwise comparisons of the above sets of nearest neighbors, has a computational complexity of $O(N^2)$. The running time of computing the entropy-based strength of N nodes in the SNN graph is $O(Nk)$, where usually $k \ll N$. The complexity of the remaining steps of this method is the same that for the first proposed method. Therefore, Method 2 has also an overall complexity of $O(N^2)$.

Though each of the above methods run, as a whole, in time proportional to the square of the input size, some of their steps can be separated into many identical sub-processes that can be executed independently. In the first method, the most time-consuming steps, namely computing the similarities between pairs of data objects and the entropy-based feature of each data object, can be fully parallelized. Searching the nearest neighbors per data object can be also done separately in the second method. Alternatively, there are efficient methods, in comparison with the brute force approach, to search a data set for the nearest neighbors to a given query data object. These methods can be either exact, like the still popular kd-tree method (Friedman et al., 1977), or approximate, like the more recent ones based on the locality-sensitive hashing scheme (Wang et al., 2016). In addition, to construct an SNN graph, we only need to test whether two data objects share nearest neighbors when one of them is actually a k-nearest neighbor of the other one.

Bibliography

Barabási, Albert-László and Pósfai, M. (2016). *Network science*. Cambridge University Press, Cambridge.

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2008). On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188.

Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, 70(6):066111.

Coscia, M., Giannotti, F., and Pedreschi, D. (2011). A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, 4(5):512–546.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software*, 3(3):209–226.

Humphries, M. D., Gurney, K., and Prescott, T. J. (2006). The brainstem reticular formation is a small-world, not scale-free, network. *Proceedings of the Royal Society B: Biological Sciences*, 273(1585):503–511.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666.

Laskaris, N. A. and Zafeiriou, S. P. (2008). Beyond FCM: Graph-theoretic post-processing algorithms for learning and representing the data structure. *Pattern Recognition*, 41(8):2630–2644.

Newman, M. (2018). *Networks*, volume 1. Oxford University Press.

Newman, M. E. J. (2004). Detecting community structure in networks. *European Physical Journal B*, 38(2):321–330.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 69(2):1–15.

Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 76(3):1–11.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wang, J., Liu, W., Kumar, S., and Chang, S. F. (2016). Learning to hash for indexing big data - A survey. *Proceedings of the IEEE*, 104(1):34–57.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

Chapter 7

Conclusions and future work

Contents

7.1 Contributions

The goal of this thesis was to develop a framework based on graph theory and entropy measures to learn, model, and represent the underlying structure of multidimensional and unlabeled data, while the information retrieved from the data is maximized. This objective has been achieved with the following contributions:

- We have implemented a characterization of data objects within a given dataset in terms of an entropy-based feature with potential application in the construction of graph-based data representations. We developed two approaches to achieve this task. In the first one, the entropy-based feature at each data object is computed using together similarity measures on the feature space where data objects are represented and the binary entropy function, which in turn is based on Shannon's entropy. In the second approach, we provide the first approach with a framework based on graph theory concepts wherein distance and similarity measures between data objects are associated with edge weight functions of an SNN graph that models an input dataset. Similarly, the entropy-based feature of a data object is formulated as an entropy-based node strength in the SNN graph. Consequently, this feature is calculated for each data object based on the total weight of its connections to its nearest neighbors.

- We have proposed the incorporation of entropy-related measures into the process of constructing graph-based models able to represent and uncover the inner structure of the given data. Our approach is based on the following idea: rather than using every data object to construct the graph-based model, which is the usual approach, the dataset is sampled to obtain a subset of data objects from which the model is

constructed. Such sampled data objects are thought of as prototypes that represent neighborhoods of the dataset, and they are identified based on the aforementioned entropy-based features. Prototypes are then used to build a fuzzy connectivity graph, which is an undirected weighted graph able to represent the structure of the input data. Among the two approaches to obtain the entropy-based features, the one that involves the use of shared nearest neighbor information has shown better results in our experiments. We also analyzed the behavior of the entropy-based features associated with each node in the graph-based models. Our purpose was to determine what effects this can have on the functionality of the proposed methods. Based on our findings, we additionally proposed methods and strategies to complement those presented in this thesis. Moreover, we provided additional results that support the rationale of our approach and highlight conditions that influence on the effectiveness of the proposed methods by looking into the modularity of the graphs that are generated for the biological datasets considered in our experiments.

- We have implemented several unsupervised strategies to organize a given dataset into groups and uncover the underlying natural structure of data. A strategy extensively assessed was based on the so-called Louvain method, which has performed well on clustering the fuzzy connectivity graphs. However, it only considered information regarding edge weights and not the entropy values which were available for prototypes as node attributes. Consequently, we considered additional strategies to combine these attributes and information from the fuzzy connectivity graph topology to guide and condition the graph clustering process. Based on the reasonable assumption that natural groupings within data correspond to high-density regions in a feature space separated by low-density regions, we proposed a preliminary approach to find natural groupings in data by searching "highly-entropic" data objects. Such data objects are found by applying a threshold on the value of the entropy-based features of the nodes in the fuzzy connectivity graphs. Then, the "highly-entropic" data objects can be used as seeds to initialize a label propagation algorithm to detect community structure in the fuzzy connectivity graphs. Additionally, we presented two strategies to set the aforementioned threshold by taking into account the relationship between the entropy-based feature of a data object and its number of nearest neighbors.

- We have developed methods capable of helping or detecting by themselves the internal structure of high-dimensional and unlabeled data such as financial and banking data, metagenomic data, and spike-waveform-related data. Based on benchmarking data, we have demonstrated that the proposed methods are usefulness and effectiveness to cope with biological applications, such as metagenomic binning and neuronal spike sorting, wherein it is required to organize data into unknown and meaningful groups. Our experimental results also suggest that our methods have performance comparable to or higher than some state-of-art methods in the aforementioned biological applications.

7.2 Future work

The following are different directions in which we plan to expand the ideas presented in this thesis:

- We will investigate the use of entropy-related measurements besides Shannon's entropy and the entropy binary function for computing the entropy-based features.
- The single input parameters of the proposed methods, namely γ in Method 1 and k in Method 2, are set experimentally. We will develop strategies to determine them in a data-driven manner.
- We will analyze the relationship between the input parameters of the proposed methods (γ and k) and the number of prototypes that will be sampled from a given dataset and then used as nodes in the fuzzy connectivity graphs.
- We will conduct experiments with additional distance measures (e.g., the cosine distance) that can be used as the basis of the secondary similarity measures (i.e., the shared nearest neighbor information) employed by Method 2. We will also plan to implement and evaluate new secondary similarity measures.
- We will study, or develop, methods that allow us to measure explicitly the quality of the sampling process performed by the proposed methods.
- We will investigate whether the shared nearest neighbor distance measure can be incorporated in the construction of the fuzzy connectivity graphs.
- We will investigate alternative methodologies to obtain a compact graph-based model equivalent to the fuzzy connectivity graph directly from the SNN graph employed in Method 2.
- We will evaluate the performance of the method that combines the thresholding of the node entropy-based features (to identify "highly-entropic" nodes) and the label propagation algorithm. We also plan to evaluate the two proposed thresholding strategies.
- We will investigate new strategies that allow us to exploit the relationship between the entropy-based features and the topology of the graph-based models to improve the performance of the developed methods.
- We will investigate whether the resolution limit of modularity influences the performance of the proposed methods.
- We will study the interaction between random walk-based algorithms for community detection and the proposed graph-based models.

- We will investigate what properties of complex networks, besides the community structure, can be associated with the proposed graph-based models and their implications in the functionality of the proposed methods.
- We will perform further experiments to validate our findings using a combination of different measures for characterizing network community structures and additional community detection algorithms.
- We will study the potential use of the entropy-based node strength as a centrality measure in complex networks.
- We will investigate whether the proposed graph-based models can be adapted to applications related to image segmentation and image compression.
- We will explore additional problems and applications wherein the proposed methods can be applied to uncover the inner structure of high-dimensional and unlabeled data.

Appendices

Appendix A

Experiments on the natural language processing of Coronavirus literature

A.1 Introduction

COVID-19 is the name given by the World Health Organization to the infectious disease caused by a new type of Coronavirus[1]. COVID-19 is currently a major research theme due to the ongoing pandemic that not only puts people's health and lives at risk but also burdens the healthcare systems of countries.

To face this global threat, diverse initiatives have been set in motion by the research community. One of them is the COVID-19 Open Research Dataset (CORD-19) prepared and distributed by worldwide leading research groups[2]. CORD-19 is a free and massive data set consisting of scholarly articles about COVID-19, SARS-CoV-2, and related coronaviruses. This data set is intended to be used by the global research community to generate new insights on the overwhelming amount of coronavirus literature based on natural language processing techniques.

In our case, we intended to provide insight about the data through experiments that allow us to identify the existence of natural groups in which documents could be organized. For this purpose, we use the method presented in Chapter 5 to perform a classification of the documents of the CORD-19 data set into unknown groups under different conditions.

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it

[2] https://azure.microsoft.com/en-us/services/open-datasets/catalog/covid-19-open-research/

A.2 Methods

A sample of 10,000 scientific articles in English language were taken from the CORD-19 data set. The main text of each document was broken down into smaller units called tokens. Here, the tokens are the words of the document and, to reduce the number of words per document, a filtering strategy based on stop words is used (Quintero Montoya et al., 2015). Then, each document $\mathbf{d}_i$ is represented as a vector of t dimensions in vector space model (VSM), i.e., $\mathbf{d}_i = (d_{i1}, d_{i2} \dots, d_{it})$, where each dimension corresponds to a word in a vocabulary of t words that is built based on the non-filtered words of the document collection. Values assigned to a particular d_{ij} indicates whether a word occurs in a document $\mathbf{d}_i$ and how frequently it occurs in the document and the entire document collection. After having the VSM, we proceed to run the Method 2 proposed in Chapter 5.

A.3 Results and discussion

The resulting VSM consists of 10,000 documents, each of them represented as a vector of 4,096 dimensions. Since there is no ground-truth available for the CORD-19 data set, we proceed to explore it by running Method 2 under different working conditions. To this end, we set the input parameter k, the number of nearest neighbors of each document, equal to 0.5, 2.0, 3.5, and 5.0 percent of the total amount of documents in the collection (i.e., $k = \{50, 200, 350, 500\}$). We select these percentages because they are in the range of the percentages successfully used in the experiments described in Chapter 5.

Figure A.1 depicts, using an *alluvial diagram* [3], the resulting clustering structure of each experiment and how documents in the CORD-19 data set are grouped as a function of the input parameter k. In this visualization, the four vertical columns of blocks represent the outcome of each experiment and the blocks represent the different groups identified by our method. In addition, the connections between groups display how documents in each group migrate from one clustering structure to another as the parameter k changes. Figure A.1 evidences that documents tend to form a large group under the different working conditions. Moreover, the usage of smaller values of k fragment the data set into a large number of groups that exhibits a possible hierarchical structure.

In future research, we will continue exploring the CORD-19 data set under more diverse working conditions. For instance, we plan to associate to each resulting group a thematic subject based on the common dimensions of its members. In this way, we could identify the causes behind the existence of the aforementioned large cluster of documents and the observed hierarchical structure.

[3] https://datavizproject.com/data-type/alluvial-diagram/

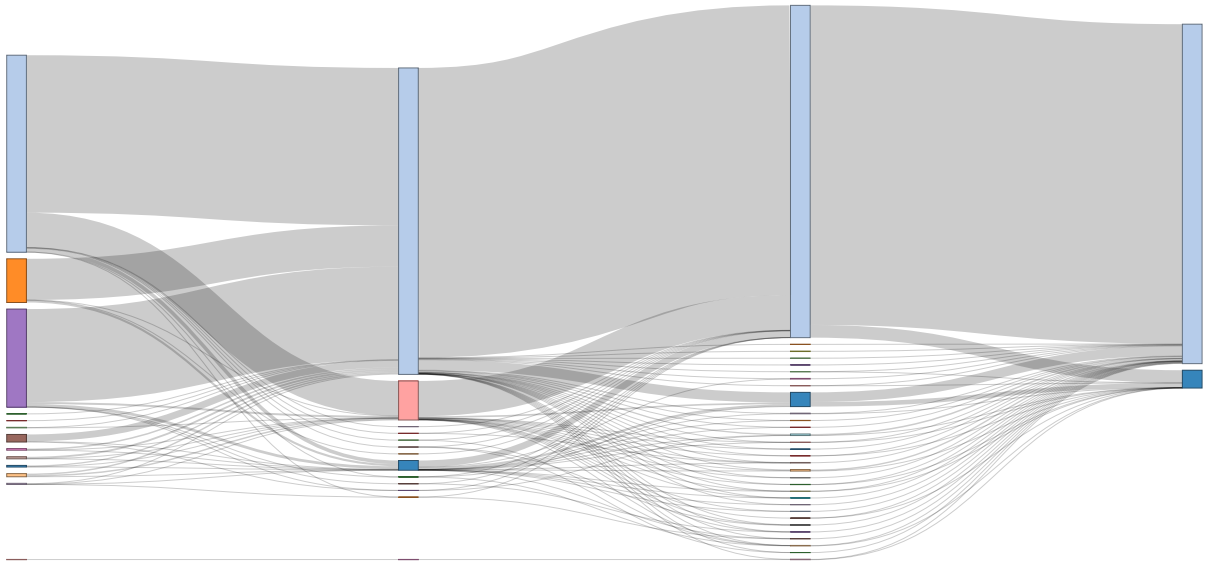


Figure A.1: Alluvial diagram representing the clustering structure identified by Method 2 on the CORD-19 dataset for different values of the input parameter k.

Bibliography

Quintero Montoya, O. L., Villa, L. F., Muñoz, S., Arenas, A. C. R., and Bastidas, M. (2015). Information retrieval on documents methodology based on entropy filtering methodologies. *International Journal of Business Intelligence and Data Mining*, 10(3):280.