Available online at www.sciencedirect.com

SciVerse ScienceDirect

Fuzzy Sets and Systems 221 (2013) 24–47

FUZZY sets and systems

www.elsevier.com/locate/fss

Graph ambiguity

Lorenzo Livi*, Antonello Rizzi

Department of Information Engineering, Electronics, and Telecommunications, SAPIENZA University of Rome, Via Eudossiana 18, 00184 Rome, Italy

Received 3 May 2011; received in revised form 29 December 2012; accepted 2 January 2013
Available online 9 January 2013

Abstract

In this paper, we propose a rigorous way to define the concept of ambiguity in the domain of graphs. In past studies, the classical definition of ambiguity has been derived starting from fuzzy set and fuzzy information theories. Our aim is to show that also in the domain of the graphs it is possible to derive a formulation able to capture the same semantic and mathematical concept. To strengthen the theoretical results, we discuss the application of the graph ambiguity concept to the graph classification setting, conceiving a new kind of inexact graph matching procedure. The results prove that the graph ambiguity concept is a characterizing and discriminative property of graphs.

Keywords: Information measures; Fuzzy sets; Fuzzy clustering; Graph matching; Graph representation; Graph mining

1. Introduction

Data mining and knowledge discovery methods [1,31] play a central role in many modern scientific contexts, especially considering the availability of large datasets. The employment of *fuzzy sets* [81] as the basic mathematical objects to represent complex systems has received a great deal of attention from a broad range of scientific disciplines, because they are naturally able to model the *uncertainty* of the data and the *soft* relations between the elements of the modeled system. Graphs are powerful representation instruments, able to model both relational and semantical information in data. They are ubiquitous in every field of science and engineering [25,29,63,65]. Consequently, the field of *structural pattern recognition* is growing fast, and is aimed to the establishment of standard pattern recognition techniques on different structured domains, such as the ones of graphs, strings, and automata [28,51].

The *ambiguity*, or uncertainty, of a fuzzy set is a well-known concept and its mathematical formulation is well described using a concept similar to the one of *information* from information theory [68,69], that is, the *fuzzy information* [24,44]. The first objective of this paper is to show that it is possible to measure the ambiguity of a complex combinatorial object, such as a graph. The intuition is the following: the ambiguity in the definition of a graph is a matter of observation, or better, of representation. The *intrinsic*, or *characteristic*, ambiguity is the evaluation of the graph considering the best possible representation; we show that the ambiguity of the graph is measurable using the concept of *fuzzy entropy*

* Corresponding author. Tel.: +39 0644585745.
E-mail addresses: livi@diet.uniroma1.it, lorenz.livi@gmail.com (L. Livi), antonello.rizzi@uniroma1.it (A. Rizzi)
URLs: http://sites.google.com/site/lorenzlivi/, http://infocom.uniroma1.it/~rizzi/index.html.

0165-0114/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.fss.2013.01.001

[42,45,52]. Our approach is based on the fact that the graph can be explicitly represented in a vector space. Consequently, we do not need to define a new fuzzy entropy function in the domain of graphs; instead, our objective is to be able to import the whole set of already defined fuzzy entropy functions to this new representation of the graph. This is a novel problem considering the domain of graphs, and in our analysis we will refer to it with the name *graph ambiguity*. This result has both theoretical and practical relevance, as we show in this paper. Indeed, on the basis of this new theoretical concept, we will develop a set of new procedures able to deal with the graph matching problem [51].

This paper is organized as follows. Some basic preliminary definitions are given in Section 1.1. A brief introduction to the concepts of fuzzy entropy and graph partitioning is given in Sections 1.2 and 1.3, respectively. The graph ambiguity concept is exposed mainly in Section 2, where the representation of the graph in a suited embedding space is described (Section 2.2) and the graph ambiguity evaluation is carried out (Section 2.3). In Section 2.4, we show some properties related to the developed embedding procedure. The graph ambiguity problem is described in Section 3, that is, the computational problem aimed to the identification of the best possible representation for the graph. In Section 3.1 we analyze some of the derived properties of the measure, while its computation is treated in Section 3.2. In Section 4, we define the *Fuzzy Information-Theoretic Graph Kernel* (Fuzzy IT Graph Kernel), that is, a new similarity measure between graphs. This new way of conceiving graph matching is based on the theoretical results provided by the proposed graph ambiguity concept. Moreover, we benchmarked this new graph similarity measure against other state-of-the-art approaches, obtaining comparable results. Finally, in Section 5 we draw our conclusions, together with a description of different future applications concerning the proposed graph ambiguity.

1.1. Preliminary definitions

Firstly, we will briefly introduce the contexts of fuzzy sets and graphs, considering only the most basic definitions needed for our paper. A more complete and in depth treatment of the subject of fuzzy sets is given by Pedrycz and Gomide [58], while for graph theory we refer to Bollobás [9] and Diestel [23]. Other less general definitions will be provided only when necessary. For simplicity, we will restrict the main observations to the class of (possibly weighted) simple graphs, if not stated otherwise.

A generic (fuzzy) set is denoted in calligraphic style, such as $\mathcal{X}$. A fuzzy set $\mathcal{X}$ [81], defined on the universe of discourse $\mathcal{U}$, is a set of pairs $(x, \mu(x))$, where $x \in \mathcal{U}$ represents an element and $\mu(x)$, sometimes denoted as $\mu_{\mathcal{X}}(x)$, stands for its *membership value*, or *membership degree*, to the fuzzy set $\mathcal{X}$. The *membership function* describing the fuzzy set $\mathcal{X}$ is a function of the form $\mu_{\mathcal{X}} : \mathcal{U} \to [0, 1]$, where if $\mu_{\mathcal{X}}(x) = 1$ ($\mu_{\mathcal{X}}(x) = 0$) holds, we say that the element $x \in \mathcal{U}$ belongs totally (does not totally belong) to the fuzzy set $\mathcal{X}$. A *crisp* set $\mathcal{X}$ is a set where each element of $\mathcal{U}$ belongs, or does not belong, totally to $\mathcal{X}$. The set denoted with $[1/2]$ is a fuzzy set with all membership values equal to $1/2$.

Real column vectors are denoted with $\underline{\mathbf{x}}$ and an $n \times m$ matrix with $\mathbf{M}^{n\times m}$ or simply $\mathbf{M}$ where the dimension is easily inferable from the context. A mapping function $f : \mathcal{X} \to \mathcal{Y}$ will also be denoted with $f(\cdot)$. $\mathbb{R}_0^+$ stands for the set of positive real values plus the zero.

Definition 1.1 (*Graph*). A graph is a pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the finite set of vertices (also referred to as nodes) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the finite set of edges.

The items of $\mathcal{E}$ can be denoted with $e_{ij} = (v_i, v_j)$, meaning an edge from vertex v_i to vertex v_j. If $\mathcal{E}$ is a symmetric relation, $(v, u) \in \mathcal{E} \Leftrightarrow (u, v) \in \mathcal{E}, \forall v, u \in \mathcal{V}$, then the graph G is called *undirected*, conversely, it is referred to as *directed* graph. G is called a *weighted* graph if for each edge e_{ij} there is a real (usually nonnegative) number w_{ij} called the weight. An unweighted graph can be seen as a weighted one with $w_{ij} = 1, \forall e_{ij} \in \mathcal{E}$. The notations $\mathcal{V}(G)$ and $\mathcal{E}(G)$ will refer, respectively, to the sets of vertices and edges of the graph G.

Definition 1.2 (*Adjacency matrix*). Given a graph $G = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}| = n$, the adjacency matrix of G is denoted with $\mathbf{A}^{n\times n}$ and is defined as

$$A_{ij} = \begin{cases} 1 & \text{if } e_{ij} \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Similarly, if the graph G is weighted then $A_{ij} = w_{ij}$ is a *weighted adjacency matrix*. For undirected graphs this matrix is symmetric. The *degree* of a vertex v_i is denoted with $\deg(v_i)$ and is equal to the number of incident edges, that is: $\deg(v_i) = \sum_j A_{ij}$.

Definition 1.3 (*Walks*). A *walk* of length k in a graph G is a sequence of vertices $(v_1, \ldots, v_{k+1})$ with $v_i \in \mathcal{V}$ and $e_{i,i+1} = (v_i, v_{i+1}) \in \mathcal{E},\ i = 1 \to k$.

A *path* in a graph G is a walk with no repeated vertices. If two vertices are not connected by a path their distance is assumed to be ∞. A *cycle* of length k in a graph G is a path of length k with $(v_{k+1}, v_1) \in \mathcal{E}$. A graph G is called connected if there is a walk between any two vertices.

Definition 1.4 (*Diameter*). The *diameter* Δ of a graph $G = (\mathcal{V}, \mathcal{E})$ is the greatest shortest path length between any two vertices of the graph.

Definition 1.5 (*Subgraph*). Let $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $G_2 = (\mathcal{V}_2, \mathcal{E}_2)$ be two graphs. G_1 is a subgraph of G_2, written as $G_1 \subseteq G_2$, if $\mathcal{V}_1 \subseteq \mathcal{V}_2$ and $\mathcal{E}_1 \subseteq \mathcal{E}_2 \cap (\mathcal{V}_1 \times \mathcal{V}_1)$ hold.

Definition 1.6 (*Induced subgraph*). Given a graph $G = (\mathcal{V}, \mathcal{E})$ and a subset $\mathcal{C} \subseteq \mathcal{V}$, the induced subgraph of the vertices in $\mathcal{C}$ is denoted with $G[\mathcal{C}] = (\mathcal{C}, \mathcal{E}')$, where $\mathcal{E}' = \mathcal{E} \cap (\mathcal{C} \times \mathcal{C})$.

1.2. Fuzzy information theory

Starting from fuzzy set theory developed by Zadeh [81], the principal aim of the fuzzy information research branch is to extend and generalize the concept of uncertainty of a generic object to different contexts, where ambiguity, or imprecision, of definitions and measurements are key elements to be considered. The basic building block of the classical information theory can be identified certainly in the concept of a *entropy* measure of a probability distribution. Considering fuzzy sets, the concept of fuzzy entropy was developed [42,45,52]. The original Shannon entropy [68] measures the average information provided by a probability distribution. A fuzzy entropy measure is conceived to estimate the degree of ambiguity of an object, defined in terms of fuzzy sets. However, both concepts define a *kind* of uncertainty [44].

Zadeh [80] defines the concept of entropy of a *fuzzy event*, conceived as a weighted sum of the Shannon entropy, using the membership values as weights. The first formulation that is not based on the classic probabilistic interpretation can be found in Luca and Termini [52]; in their work there is also the definition of the base axioms that a general fuzzy entropy measure should satisfy.

Definition 1.7 (*Fuzzy entropy*). Given a crisp set $\mathcal{X}$, $|\mathcal{X}| = k$, we define its *fuzzy power set* with $\mathcal{F}(\mathcal{X})$, that is, the crisp set of all fuzzy sets of $\mathcal{X}$. In general, $\mathcal{F}(\mathcal{X})$ has continuum-many elements (fuzzy sets). The function $E : \mathcal{F}(\mathcal{X}) \to [0, \infty)$ is a fuzzy entropy function if the following axioms are satisfied:

1. (Sharpness) $\forall \mathcal{Y} \in \mathcal{P}(\mathcal{X}),\ E(\mathcal{Y}) = 0$, where $\mathcal{P}(\mathcal{X})$ is the power set of $\mathcal{X}$,
2. (Maximality) $E([\frac{1}{2}]) = \max_{\mathcal{A} \in \mathcal{F}(\mathcal{X})} E(\mathcal{A})$,
3. (Resolution) $\forall \mathcal{Y}, \mathcal{Z} \in \mathcal{F}(\mathcal{X})$, if $\mathcal{Y}$ is *less fuzzy* than $\mathcal{Z}$ we have $E(\mathcal{Y}) \leq E(\mathcal{Z})$,
4. (Symmetry) $E(\mathcal{Y}) = E(\mathcal{Y}^c)$, where $\mathcal{Y}^c$ is the fuzzy complement set of $\mathcal{Y}$.

For the third condition of Definition 1.7, less fuzzy means that if $\mu_{\mathcal{Z}}(x) \leq 1/2 \Rightarrow \mu_{\mathcal{Y}}(x) \leq \mu_{\mathcal{Z}}(x)$, and if $\mu_{\mathcal{Z}}(x) \geq 1/2 \Rightarrow \mu_{\mathcal{Y}}(x) \geq \mu_{\mathcal{Z}}(x)$ holds $\forall x \in \mathcal{X}$. These axioms say that a fuzzy entropy is a function of the membership values characterizing a given fuzzy set. If a fuzzy set contains objects with uncertain membership values (i.e., around 1/2), then the definition of the fuzzy set is not clear – i.e., the elements belong to the fuzzy set with uncertainty – and hence its fuzzy entropy should be high. Conversely, if the membership values are mostly crisp, that is, only a small number of objects has uncertain memberships, the fuzzy set should have a small fuzzy entropy value.

Other concepts from information theory have been extended and generalized in the domain of fuzzy sets, such as the *fuzzy directed divergence* and *fuzzy symmetric divergence* [7,44].

1.2.1. Fuzzy entropy measures

Many fuzzy entropy functions have been developed and analyzed in the literature, such as the ones proposed by Bhandari and Pal [7], Fan and Ma [26], Gang [27], Kaufmann [42], Kosko [45,46], Luca and Termini [52], Pal and Pal [56], Shang and Jiang [67], Yager [78], You and Wen [79], as well as Hooda [35]. In a recent survey of Al-Sharhan et al. [3], a taxonomy is presented for the most important formulations. Roughly speaking, it is possible to identify three main categories of fuzzy entropy formulations: *probabilistic*, *non-probabilistic*, and *hybrid*.

Considering the non-probabilistic fuzzy entropy functions, we can cite the formulation of Kosko [45] that analyzes the geometry of a fuzzy set considering the relationship between its fuzzy complement, defining a new vectorial interpretation for discrete fuzzy sets. Given a crisp set $\mathcal{X}$, $|\mathcal{X}| = k$, the subset $\mathcal{C} \subseteq \mathcal{X}$ can be represented as a binary vector such as $[1, 1, \ldots, 0]^T$, where the i-th element is equal to one if the i-th element of $\mathcal{X}$ belongs to $\mathcal{C}$, being equal to zero otherwise. In the case of a (discrete) fuzzy set $\mathcal{A} \in \mathcal{F}(\mathcal{X})$, its vector representation consists in representing the membership values of the elements of $\mathcal{X}$ as a k-dimensional vector $\underline{\mathbf{f}} \in [0, 1]^k$, where $f_i = \mu_{\mathcal{A}}(x_i)$; we will make use of this handy notation in this paper.

Kosko computes the fuzzy entropy of $\mathcal{A}$ as

$$E_{Ko}(\underline{\mathbf{f}}) = \frac{c(\underline{\mathbf{f}} \top \underline{\mathbf{f}}^c)}{c(\underline{\mathbf{f}} \perp \underline{\mathbf{f}}^c)} \in [0, 1], \tag{2}$$

where $f_i^c = 1 - f_i$, and the function $c(\underline{\mathbf{f}}) = \sum_i f_i$ is a measure of the cardinality of the fuzzy set. Symbols $\perp$ and $\top$ represent the standard t-conorm and t-norm operators [4,81]. When $\underline{\mathbf{f}}$ is unidimensional, Eq. (2) reduces to

$$E_{Ko}(f) = \begin{cases} \dfrac{f}{1-f} & \text{if } f \leq \dfrac{1}{2}, \\ \dfrac{1-f}{f} & \text{otherwise.} \end{cases} \tag{3}$$

Other three formulations that consider the relationship between the fuzzy set $\mathcal{A}$ and its fuzzy complement $\mathcal{A}^c$ are the one of Kaufmann [42], Yager [78], and Shang and Jiang [67]. Let $\mathcal{A} \in \mathcal{F}(\mathcal{X})$ be a fuzzy set and let nearest crisp set $\mathcal{A}_{\text{near}}$ have a membership function defined as

$$\mu_{\mathcal{A}_{\text{near}}}(x) = \begin{cases} 1 & \text{if } \mu_{\mathcal{A}}(x) \geq 1/2, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The fuzzy entropy of Kaufmann [42] measures the distance of a fuzzy set between its nearest crisp set, and it is defined as

$$E_K(\mathcal{A}) = \frac{1}{n^{1/p}} \sum d_p(\mathcal{A}, \mathcal{A}_{\text{near}}). \tag{5}$$

The formulation of Yager [78] is

$$E_Y(\mathcal{A}) = 1 - \frac{d_p(\mathcal{A}, \mathcal{A}^c)}{n^{1/p}}, \tag{6}$$

where in both Eqs. (5) and (6) $d_p(\cdot, \cdot)$ is a *Minkowski* distance of order p applied to the membership values of the fuzzy sets, that is,

$$d_p(\mathcal{A}, \mathcal{B}) = \left(\sum_{x \in \mathcal{X}} |\mu_{\mathcal{A}}(x) - \mu_{\mathcal{B}}(x)|^p \right)^{1/p}. \tag{7}$$

Another interesting formulation is the one of Shang and Jiang [67]. Given a fuzzy set $\mathcal{A}$ and its fuzzy complement $\mathcal{A}^c$, the fuzzy entropy of $\mathcal{A}$ is defined considering the memberships of the intersection and union of those two fuzzy sets,

$$E_{SJ} = \frac{1}{n} \sum_{i=1}^{n} \frac{\mu_{\mathcal{A} \top \mathcal{A}^c}(x_i)}{\mu_{\mathcal{A} \perp \mathcal{A}^c}(x_i)}. \tag{8}$$

An example of generalization of the classical Shannon entropy comes from Pal and Pal [56] and is referred to as *hybrid entropy*. Considering the scenario of a digital communication over a noisy channel and a binary probability distribution over the alphabet {0, 1}, the hybrid entropy of the fuzzy set $\mathcal{A}$, semantically interpreted as "symbol close to 1", is defined as

$$E_H(\mathcal{A}) = -p(0)\,\log(1 - F_0) - p(1)\,\log(F_1), \tag{9}$$

where $p(0)$ and $p(1)$ are the probabilities of the occurrence of the symbols 0 and 1, respectively. The average likeliness of interpreting the i-th symbol as 0 and 1, respectively, is defined as

$$F_0 = \frac{1}{k}\sum_{i=1}^{k}(1 - \mu(x_i))\,\exp(\mu(x_i))$$

and

$$F_1 = \frac{1}{k}\sum_{i=1}^{k}(\mu(x_i))\,\exp(1 - \mu(x_i)). \tag{10}$$

An example of pure probabilistic formulation can be found in Bhandari and Pal [7], based on Rényi's entropy [61]. It is a parametric fuzzy entropy function that depends on $\alpha > 0$, with $\alpha \neq 1$, defined as

$$E_\alpha(\mathcal{A}) = \frac{1}{1-\alpha}\sum_{i=1}^{n}\log[\mu^\alpha(x_i) + (1 - \mu(x_i))^\alpha]. \tag{11}$$

1.3. Graph partitioning

In this section we will introduce the problem known as *graph partitioning* that seeks to find a partition of the set of vertices of a graph that optimizes a suited performance measure. Our interest is not focused on issues related to graph partitioning, but only in the definition of the problem. In this paper, when we talk about clustering of the vertex set or issues related to graph clustering, we are referring to the problem of partitioning the vertex set of a given graph. For examples on graph partitioning see Brandes et al. [14–16], Jianbin et al. [37].

Definition 1.8 (*Hard k-clustering*). Given a graph $G = (\mathcal{V}, \mathcal{E})$, a hard k-clustering problem over G consists in finding a partition $\mathcal{K} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$, with $1 \leq k \leq |\mathcal{V}|$ disjoint non-empty sets $\mathcal{C}_i$ of vertices, such that $\bigcup_{i=1}^{k}\mathcal{C}_i = \mathcal{V}$ holds. k is referred to as the *order* of the clustering.

Obviously k is a natural number. Each cluster $\mathcal{C}_i, i = 1 \to k$, is a set of vertices, $\mathcal{C}_i \subseteq \mathcal{V}$, which induces a proper subgraph $G[\mathcal{C}_i]$. Given a $1 \leq k \leq |\mathcal{V}(G)|$, we can ask how many clusterings of order k does the graph G have.

Definition 1.9 (*Set of k-clusterings*). Given a graph $G = (\mathcal{V}, \mathcal{E})$ and a number $1 \leq k \leq |\mathcal{V}|$, we denote with $\Gamma^k(G)$ the set of all possible hard k-clusterings of the graph G.

Where it is not ambiguous, we will use also the notation Γ^k. Generalizing for k, we obtain the set $\Gamma^*(G)$ of all possible hard k-clusterings of the graph G.

Definition 1.10 (*Set of *-clusterings*). Given a graph $G = (\mathcal{V}, \mathcal{E})$, $\Gamma^*(G) = \{\mathcal{K}_1, \ldots, \mathcal{K}_{|\Gamma^*|}\}$ is the set of all possible hard k-clusterings of the graph G, for $k = 1 \to |\mathcal{V}|$.

We can calculate the size of the set $\Gamma^*(G)$ of graph $G = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}| = n$ vertices, using the *Stirling numbers of the second kind* [70]. The sum of these numbers is expressed with the *Bell numbers* [64],

$$|\Gamma^*(G)| = \sum_{k=0}^{n} S(n,k) = \sum_{k=0}^{n}\left\{{n \atop k}\right\} = \sum_{k=0}^{n}\frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n. \tag{12}$$

For example, if we have a graph with $|\mathcal{V}(G)| = 3$ vertices, there are five different partitions, $|\Gamma^*(G)|=0+1+3+1=5$.

Considering Eq. (12), it is easy to understand that the number of partitions of a graph grows exponentially with the number of vertices, that is,

$$|\Gamma^*(G)|_n = \mathcal{O}(\exp(n)). \tag{13}$$

Now we introduce the important concept of *fuzzy partitioning* of the vertices set of a graph. This concept plays an important role in the definition of the graph ambiguity.

Definition 1.11 (*Soft k-clustering*). Given a graph $G = (\mathcal{V}, \mathcal{E})$, a soft k-clustering of $\mathcal{V}(G)$ is defined as a collection $\mathcal{S} = \{\mathcal{F}_1, \ldots, \mathcal{F}_k\}$ of $1 \leq k \leq |\mathcal{V}|$ fuzzy sets $\mathcal{F}_i$, such that each $v \in \mathcal{V}(G)$ belongs with a certain membership degree to every fuzzy set of $\mathcal{S}$.

Considering our needs, it is important to treat also a particular case, where the graph is *empty*, $G = (\emptyset, \mathcal{E})$, that is, when there are no vertices to partition. This kind of graph is also referred to as *zero-order* graph.

Definition 1.12 (*Empty graph clustering*). Given an empty graph $G=(\emptyset, \mathcal{E})$, the only possible clustering is the empty clustering $\Gamma^*(G) = \mathcal{K}_\emptyset = \emptyset$.

2. Graph ambiguity

In this section, we introduce all the concepts related to the graph ambiguity. We show that partitioning a graph can be interpreted formally as an algebraic representation. To this end, given a hard k-clustering $\mathcal{K}$, we analyze the derived grouping of the vertices, firstly inducing another type of clustering. This second interpretation of the graph is elaborated as an explicit embedding of the graph in a suitable vector space. What follows is almost directly applicable also to directed graphs, but for the sake of simplicity, we will restrict our analysis to the class of undirected ones.

2.1. Induced soft clustering

Given a hard k-clustering $\mathcal{K} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$, our objective is to obtain what we call the relative *induced* soft clustering $\mathcal{S}$, of the same order k. To make this possible, we model each cluster $\mathcal{C}_i$ with a specific fuzzy set $\mathcal{F}_i$, defining a collection of fuzzy sets $\mathcal{S} = \{\mathcal{F}_1, \ldots, \mathcal{F}_k\}$. Each vertex $v \in \mathcal{C}_i$ becomes then an element of a fuzzy set $\mathcal{F}_i$ considering the pair $(v, \mu_{\mathcal{F}_i}(v))$, $i = 1 \to k$. For this purpose, the membership function $\mu_{\mathcal{F}_i}(\cdot)$ is defined as the product of two distinct functions of the same argument, that is,

$$\mu_{\mathcal{F}_i}(v) = \alpha_i(v) \cdot \tau_i(v). \tag{14}$$

The first factor is defined as

$$\alpha_i(v) = \frac{\deg_{\mathcal{C}_i}(v)}{\sum_{j=1}^{k} \deg_{\mathcal{C}_j}(v)}, \tag{15}$$

where the degree evaluation $\deg_{\mathcal{C}_i}(v)$ can take into account also the weights of the edges. $\deg_{\mathcal{C}_i}(v)$ is the degree of the vertex v in the specific cluster $\mathcal{C}_i$, that is, the number of intra-cluster edges. $\tau_i(v)$ is a radial basis function (RBF) nonlinear modifier defined as follows:

$$\tau_i(v) = \exp\left(-\frac{d(v, \delta(\mathcal{C}_i))}{t^2}\right), \quad t \geq 1. \tag{16}$$

The function $\delta(\mathcal{C}_i)$ retrieves the *representatives* of the set $\mathcal{C}_i$, that is, a subset of vertices, say $\mathcal{M} \subseteq \mathcal{C}_i$, that are maximally *central* in the induced subgraph $G[\mathcal{C}_i]$. The function $d(v, \delta(\mathcal{C}_i))$ computes the minimum distance (i.e., the shortest path) from v to the closest vertex in $\mathcal{M}$. Details on the computation of the representatives are given in Section 2.1.3. The more peripheral a vertex is, the higher should be the dispersion applied to its membership value. It is quite easy to understand that the membership function shown in Eq. (14) assumes values in [0, 1].

This process of inducing a soft clustering from a hard k-clustering can be modeled by a many-to-one mapping function from Γ^* to the set of all possible soft clusterings Ψ^*, generalizing for the set of all graphs $\mathcal{G}$.

Definition 2.1 (*Induced soft clustering*). The soft clustering $\mathcal{S}$ is an induced soft clustering, and it is denoted with $\mathcal{K} \succ \mathcal{S}$. The mapping function generating those induced soft clusterings is a many-to-one mapping function defined as follows:

$$\psi : \Gamma^*(\mathcal{G}) \to \Psi^*(\mathcal{G}). \tag{17}$$

Given a graph $G = (\mathcal{V}, \mathcal{E})$, the mapping function shown in Eq. (17) is not injective, because two distinct hard k-clusterings $\mathcal{K}_i, \mathcal{K}_j \in \Gamma^*(G)$ *could* be mapped into the same soft clustering $\mathcal{S}$. We will return to this issue later in Section 2.4.2. This map is neither surjective, because not every possible soft clustering of $\Psi^*(\mathcal{G})$ can be associated with a hard k-clustering of $\Gamma^*(\mathcal{G})$. This claim is supported by the fact that there are infinitely many possible configurations of the collection of membership values that any vertex can assume.

As in the hard case, we need to extend the definition of soft clustering to the case of an empty graph.

Definition 2.2 (*Empty induced soft clustering*). Given an empty graph $G=(\emptyset, \mathcal{E})$, the unique hard clustering is $\Gamma^*(G) = \mathcal{K}_\emptyset = \emptyset$. Consequently the only possible soft clustering will be the empty one $\mathcal{S} = \emptyset$.

2.1.1. Membership function

The membership function $\mu_{\mathcal{F}_i}(\cdot)$ shown in Eq. (14) is a convex nonlinear function of two variables: the degree $\deg_{C_i}(v)$ of the vertex v in the i-th cluster and of the minimum distance $d(v, \delta(\mathcal{C}_i))$ to the representatives of $\mathcal{C}_i$. More formally,

Definition 2.3 (*Membership function*). The membership function, defined in Eq. (14), which evaluates the membership value of each vertex to each fuzzy set $\mathcal{F}_i$, is a nonlinear convex function of the form

$$\mu : \hat{\mathcal{D}} \times \Sigma \to [0, 1], \tag{18}$$

where $\hat{\mathcal{D}} = [0, 1]$ is the set of normalized degrees and Σ is the set of minimum distances from each cluster representative.

If the graph is unweighted, then $\Sigma = \mathbb{N}_0 \cup \{\infty\}$, otherwise $\Sigma = \mathbb{R}_0^+ \cup \{\infty\}$. There are two factors that influence the membership of a vertex: the *degree concentration* $\alpha_i(v)$ and the *centrality*, or *nearness*, $\tau_i(v)$. The first one is an estimation of the connectivity of the vertex in a specific cluster $\mathcal{C}_i$, while the second one is an estimation of the centrality of the vertex in the cluster, that is, how far is from the set $\mathcal{M}$ of representatives of this cluster.

The nonlinear modifier $\tau_i(v)$ decays exponentially with the distance, which evaluates to one when the vertex $v \in \mathcal{M}$ and goes to zero when there is no path from v to any of the vertices in $\mathcal{M}$, that is, when the distance is ∞. The parameter t controls the influence of this nonlinear modifier over the whole membership value. If $t \to \infty$, actually we are not considering the centrality factor, because $\lim_{t\to\infty} \exp(-x/t^2) = 1$. In Fig. 1 is shown the shape of the membership function defined in Eq. (14) in a finite range of possible distances.

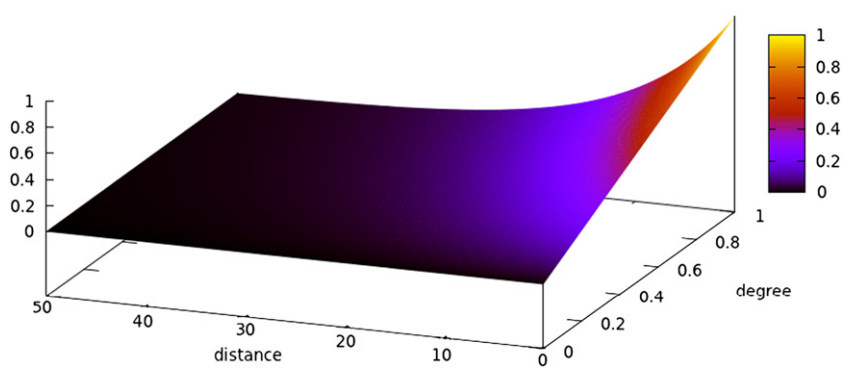

Fig. 1. Shape of the membership function defined in Eq. (14).

2.1.2. Bounding of the membership function

If the graph $G = (\mathcal{V}, \mathcal{E})$ is (strongly) connected, and considering the notation of Eq. (14), we have the following bounds:

$$0 \leq d(v, \delta(\mathcal{C}_i)) \leq \Delta, \quad \forall v \in \mathcal{V},\ i = 1 \to k, \tag{19}$$

where Δ is the diameter of the graph. If there is no path P between v and $\delta(\mathcal{C}_i)$, $d(\cdot, \cdot)$ evaluates to ∞. Formally,

$$\begin{cases} d(v, \delta(\mathcal{C}_i)) \leq \Delta & \text{if } \exists P, \\ d(v, \delta(\mathcal{C}_i)) = \infty & \text{otherwise.} \end{cases} \tag{20}$$

From Eq. (20) it is possible to state that $\forall v \in \mathcal{V}$, and for $i = 1 \to k$, holds

$$\begin{cases} \exp\left(-\dfrac{\Delta}{t^2}\right) \leq \tau_i(v) \leq 1 & \text{if } \exists P, \\ \tau_i(v) = 0 & \text{otherwise.} \end{cases} \tag{21}$$

Consequently, the membership value of each $v \in \mathcal{V}$ to the i-th fuzzy set $\mathcal{F}_i$ is bounded by

$$\begin{cases} \alpha_i(v) \cdot \exp\left(-\dfrac{\Delta}{t^2}\right) \leq \mu_{\mathcal{F}_i}(v) \leq \alpha_i(v) & \text{if } \exists P, \\ \mu_{\mathcal{F}_i}(v) = 0 & \text{otherwise.} \end{cases} \tag{22}$$

If in a cluster there is no connectivity, obviously there cannot be any path, thus $\mu_{\mathcal{F}_i}(v) = 0$. The bound shown in Eq. (22) is a function of a characteristic property of the graph, the diameter Δ. One should investigate further in this direction, searching for a bridge with the *spectral graph theory* [19,20]. Indeed, the diameter of a graph is known to be related to the *eigenvalues* of the *Laplacian* matrix associated with the graph [19].

2.1.3. Cluster representatives

The representatives identification function $\delta(\cdot)$ computes a set of vertices $\mathcal{M} \subseteq \mathcal{C}_i$, $\mathcal{M} \neq \emptyset$, that are central considering the induced subgraph $G[\mathcal{C}_i]$. Formally, we have defined the set of representatives of a cluster $\mathcal{C}_i$ as the one whose elements minimize a given nonnegative finite cost function $f(\cdot)$

$$\delta(\mathcal{C}_i) = \left\{ v : \arg\min_{v \in \mathcal{C}_i} f(v) \right\}. \tag{23}$$

Note that $\delta(\mathcal{C}_i)$ always exists (is non-empty) also for singleton clusters. Intuitively, a vertex is central (i.e., important) in a graph if it is highly connected to the other vertices. To this end, we consider the set of elements that minimizes the sum of distances, known as MinSOD,

$$\delta(\mathcal{C}_i) = \left\{ v : \arg\min_{v \in \mathcal{C}_i} \left(\sum_{u \in \mathcal{C}_i} d(v, u) \right) \right\}. \tag{24}$$

Note that equivalently one could minimize the sum of any (positive) power of the distances. By convention, the summation is taken only for finite-value distances. For large graphs the exact determination of $\delta(\mathcal{C}_i)$ could be prohibitive. The determination of the centrality, or importance, of a vertex in a network is a well-known problem in socio-mathematics and network dynamic fields [33,75], especially for *web information retrieval* systems. For example we can cite *PageRank* of Brin and Page [17] and any of its variations [76], *HITS* of Kleinberg [43], *Betweenness Centrality* [13], and *Closeness Centrality* [73]. We are not interested here in a full treatment of this problem in this paper. We will just assume to be able to determine the set of representatives $\delta(\mathcal{C}_i)$ exactly as defined in Eq. (24).

The asymptotic complexity for the computation of Eq. (24) is given by

$$\mathcal{O}(|\mathcal{V}(G[\mathcal{C}_i])|^2 \cdot (|\mathcal{E}(G[\mathcal{C}_i])| + |\mathcal{V}(G[\mathcal{C}_i])| \log(|\mathcal{V}(G[\mathcal{C}_i])|))), \tag{25}$$

where the second factor is the best known running time for Dijkstra's algorithm [21], used for computing the $|\mathcal{V}(G[\mathcal{C}_i])|^2$ shortest paths in the cluster $\mathcal{C}_i$.

2.2. Graph embedding into the unit hypercube

The graph *embedding* can be described as a function that maps the vertices of the graph to a suitable vector space. In our case this space is a k-dimensional *unit hypercube* defined as $\mathcal{I}^k = [0, 1]^k$, and it will be denoted simply with $\mathcal{I}$ where the dimension k can be implicitly inferred from the context or it is not strictly required for the understanding. This multidimensional bounded (crisp) set contains all the possible realizations of the membership values of the vertices to the given induced soft clustering $\mathcal{S}$. To deal with our embedding, and consequently with the measurement of the graph ambiguity, we do not need any metric distance, neither any norm nor inner product defined on $\mathcal{I}$. Later, we will analyze the properties of this embedding, and then we will equip this space with this kind of functions.

Now we show how to map each vertex of the graph into a point in this space and, consequently, how to calculate the graph ambiguity. Given a graph $G = (\mathcal{V}, \mathcal{E})$ and a hard k-clustering $\mathcal{K} \in \Gamma^*$, we obtain a vector representation of each vertex v in the unit hypercube $\mathcal{I}^k$ that we will denote with $\underline{\mathbf{f}}_v \in \mathcal{I}^k,\ \forall v \in \mathcal{V}(G)$. Formally,

Definition 2.4 (*Vertex mapping function*). A mapping for vertices is a function of the form

$$\varphi^{\mathcal{K}} : \mathcal{V}(\mathcal{G}) \to \mathcal{I} \tag{26}$$

that maps each vertex v with a point in $\mathcal{I}^k$, denoted with a vector $\underline{\mathbf{f}}_v$ of membership values of v given $\mathcal{K} \succ \mathcal{S}$. Each i-th component is defined as $f_i^v = \mu_{\mathcal{F}_i}(v)$, hence

$$\varphi^{\mathcal{K}}(v) = \underline{\mathbf{f}}_v = [\mu_{\mathcal{F}_1}(v), \mu_{\mathcal{F}_2}(v), \ldots, \mu_{\mathcal{F}_k}(v)]^T. \tag{27}$$

We use this notation $\varphi^{\mathcal{K}}(\cdot)$ to remark that this representation is done considering the particular clustering $\mathcal{K} \succ \mathcal{S}$ (Section 2.1). By definition of the membership function, $\varphi^{\mathcal{K}}(\cdot)$ is a nonlinear mapping function. Considering an isolated vertex with $\deg(v) = 0$, we have that the mapping function will map this vertex to the origin of the hypercube, that is, in $\underline{\mathbf{0}}$. Intuitively, these vertices have no ambiguity. We will see later how to justify this claim.

Before we can generalize this mapping to a set of vertices, we need to introduce some mathematical concepts. The unit hypercube is a multidimensional bounded set, and set operations like, for example, union $\cup$ and intersection $\cap$ between fuzzy sets of the collection $\mathcal{S}$ are extended with the respective triangular operators t-conorm $\perp$ and t-norm $\top$ [4,30,81]. The t-conorm is the dual operator of the t-norm, and it is a function of the form $\perp : [0, 1] \times [0, 1] \to [0, 1]$. There are many well-known formulations in the literature [4,30]; some of these are shown in Table 1.

After we have performed the embedding of each vertex $v_i \in \mathcal{V}(G)$, we obtain a collection of vector representations $\{\underline{\mathbf{f}}_{v_i}\}_{i=1}^{|\mathcal{V}(G)|}$ in the unit hypercube $\mathcal{I}^k$. Our aim is to calculate the ambiguity of the graph G, considering this specific representation framework $\mathcal{K} \succ \mathcal{S}$. The graph ambiguity measure will be expressed in terms of the membership values of the vertices to each fuzzy set $\mathcal{F}_i,\ i = 1 \to k$. We can describe each fuzzy set $\mathcal{F}_i$ as an n-dimensional vector

$$\underline{\mathbf{f}}_{\mathcal{F}_i} = [\mu_{\mathcal{F}_i}(v_1), \mu_{\mathcal{F}_i}(v_2), \ldots, \mu_{\mathcal{F}_i}(v_n)]^T, \tag{28}$$

containing the respective membership values of the vertices. Remember that each fuzzy set $\mathcal{F}_i$ models only one hard cluster $\mathcal{C}_i$, and by definition $\bigcup_{i=1}^{k} \mathcal{C}_i = \mathcal{V}(G)$ holds. Consequently, the vector representation of the graph as a whole will be expressed as the union of each fuzzy set $\mathcal{F}_i$, that is, as the t-conorm of each $\mathcal{F}_i,\ i = 1 \to k$.

Table 1
Some t-conorm formulations.

Name	Formula
Maximum t-conorm	$\perp_{\max}(a, b) = \max\{a, b\}$
Probabilistic sum	$\perp_{\text{sum}}(a, b) = a + b - a \cdot b$
Bounded sum	$\perp_{\text{Luk}}(a, b) = \min\{a + b, 1\}$
Einstein sum	$\perp_{\text{H}_2}(a, b) = \dfrac{a + b}{1 + ab}$

Definition 2.5 (*Graph embedding*). An embedding for a graph into the unit hypercube is a function of the form

$$\phi^{\mathcal{K}} : \mathcal{G} \to \mathcal{I}. \tag{29}$$

Given a graph $G = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, a hard k-clustering $\mathcal{K} \in \Gamma^*$ and the relative induced soft clustering $\mathcal{S}$, the embedding of G is defined as

$$\phi^{\mathcal{K}}(G) = \perp_{i=1}^{k} \underline{\mathbf{f}}_{\mathcal{F}_i} = \underline{\mathbf{f}}_G \in \mathcal{I}^n. \tag{30}$$

In Eq. (30) the operator $\perp$ is assumed to be generalized to vectors in $[0, 1]^n$, and will act as a componentwise t-conorm. Note that $\perp$ is associative, so the order in Eq. (30) will not matter. In this way, the entire graph is eventually represented as a single fuzzy set, and hence its membership values can be described with an n-dimensional vector $\underline{\mathbf{f}}_G$, belonging to the unit hypercube $\mathcal{I}^n$.

We need to define how to handle the case of an empty graph (see Definitions 1.12 and 2.2). By convention, the mapping function shown in Eq. (29) will map any empty graph to the scalar zero.

2.2.1. Matrix representation

It is convenient to represent the induced soft clustering $\mathcal{S}$ as a matrix $\mathbf{M}^{n\times k}$, where n is the number of vertices in the graph and k is the order of the chosen hard k-clustering. We have

$$\mathbf{M}^{n\times k} = \begin{bmatrix} \mu_{\mathcal{F}_1}(v_1) & \cdots & \mu_{\mathcal{F}_k}(v_1) \\ \vdots & \ldots & \vdots \\ \mu_{\mathcal{F}_1}(v_n) & \cdots & \mu_{\mathcal{F}_k}(v_n) \end{bmatrix}. \tag{31}$$

The i-th column represents the membership values of the i-th fuzzy set $\mathcal{F}_i$, that is, $\underline{\mathbf{f}}_{\mathcal{F}_i}$. This notation is not strictly necessary but it is a convenient and formal description of the induced soft clustering that could be also used for further investigations, as for example considering *PCA* analysis [38] aimed at reducing the dimensionality of the embedding space $\mathcal{I}$, or any multidimensional scaling techniques [10]. However, in this paper we will not focus on this direction.

2.3. Graph ambiguity

The fuzzy entropy measures (see Section 1.2) are built to be able to measure the ambiguity of a fuzzy set, both on discrete or continuous domains. Our principal objective is to apply those measures also to graphs. In this section we will describe how to measure the ambiguity of the graph, considering a specific representation induced by the choice of the hard k-clustering $\mathcal{K}$.

Definition 2.6 (*Graph ambiguity*). The graph ambiguity is a nonnegative function of the form

$$A : \mathcal{I} \to [0, \infty). \tag{32}$$

Given a graph $G = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}| = n$ and a hard k-clustering $\mathcal{K} \in \Gamma^*$, the ambiguity of the graph is computed as a composite function

$$A(\phi^{\mathcal{K}}(G)) = (g \circ E)(\underline{\mathbf{f}}_G) = g(E(\underline{\mathbf{f}}_G)), \tag{33}$$

where $g : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is a nonnegative monotone nondecreasing function such that $g(0) = 0$ holds. The function $E(\cdot)$ is any fuzzy entropy function that satisfies the requirements of Definition 1.7.

In Table 2 are shown five types of graph ambiguity measures that satisfy the requirements of Definition 2.6. In Fig. 2 are shown some shapes of these functions considering two-dimensional vectors, that is, representations induced with two clusters. Note that these plots are based on Kosko's formulation of the fuzzy entropy (see Eq. (2)). Different formulations of fuzzy entropy will produce different shapes, but preserving the constraints of a general fuzzy entropy function listed in Definition 1.7, such as the maximization and minimization points.

Table 2
Types of graph ambiguity.

Type	Formula	Parameters
Linear	$vE(\underline{\mathbf{f}}_G) + \gamma$	$v \geq 1, \gamma \geq 0$
Logarithmic	$\log(1 + E(\underline{\mathbf{f}}_G))$	
Polynomial	$E(\underline{\mathbf{f}}_G)^p$	$p > 0$
Exponential	$\exp(E(\underline{\mathbf{f}}_G)) - 1$	
Sigmoidal	$\frac{2}{1 + \exp\left(\frac{-E(\underline{\mathbf{f}}_G)}{\sigma}\right)} - 1$	$\sigma > 0$

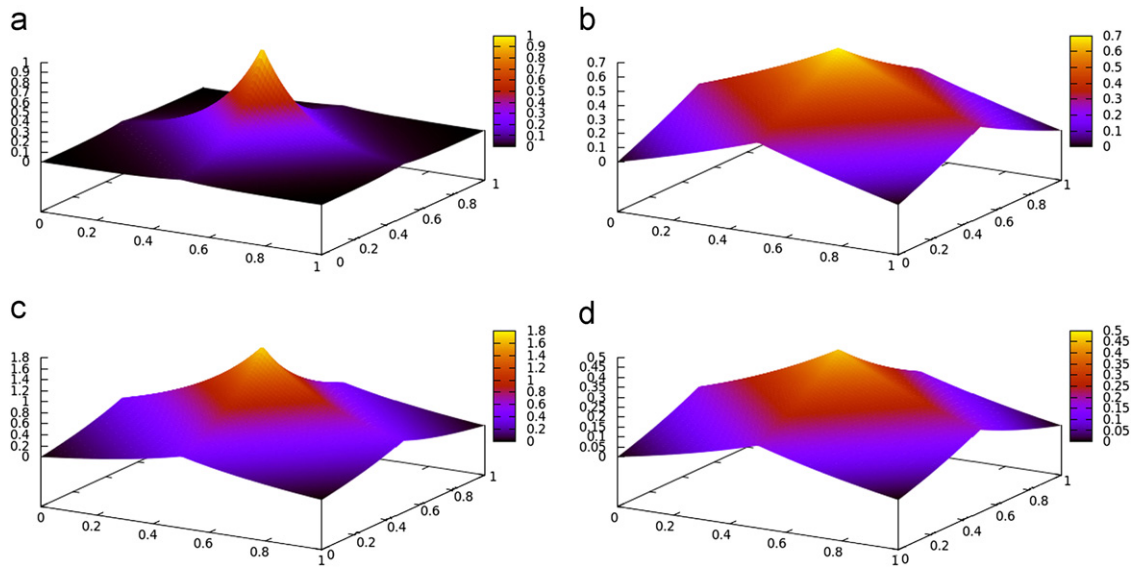

Fig. 2. Some types of graph ambiguity: (a) Polynomial. (b) Logarithmic. (c) Exponential. (d) Sigmoidal.

It is easy to understand that the measure of ambiguity for a graph is conditioned to the specific clustering $\mathcal{K}$ adopted for the representation. Different clusterings *could* produce different ambiguity values. We will return to this topic later.

A representation of the graph is a vector $\underline{\mathbf{f}}_G \in [0, 1]^n$. For instance, consider the representation $[0.5, 0.5, 0.5]^T$. This implies that the three vertices are equally assigned to each fuzzy set, making very hard to say something about their memberships in the fuzzified graph. By definition of fuzzy entropy, this vector representation of the graph will have the maximum possible ambiguity evaluation. Conversely, a representation of the form $[1, 1, 1]^T$ will denote a clear assignment of the vertices. Note that also $[1, 0, 0]^T$ is a totally unambiguous representation. For this purpose, we will call a representation crisp if each component assumes values in $\{0, 1\}$, and soft if at least one component is in the open interval $(0, 1)$.

The ambiguity could be calculated also for each fuzzy set $\mathcal{F}_i \in \mathcal{S}$. Given the vector representation of the fuzzy set $\mathcal{F}_i$, namely $\underline{\mathbf{f}}_{\mathcal{F}_i}$, the ambiguity given in Eq. (33) can be applied directly as

$$A(\underline{\mathbf{f}}_{\mathcal{F}_i}) = g(E(\underline{\mathbf{f}}_{\mathcal{F}_i})). \tag{34}$$

The ambiguity measure $A(\cdot)$ is a nonnegative measure defined, indirectly, on the set of vertices of the graph. The *subadditive* property follows from the subadditivity of the function $E(\cdot)$ [45] and from the fact that any monotone nondecreasing transformation will preserve this property. Thus we have that

$$A(\underline{\mathbf{f}}_G) \leq \sum_{i=1}^{k} A(\underline{\mathbf{f}}_{\mathcal{F}_i}). \tag{35}$$

In this case, the whole is less than or equal to the sum of its parts. $A(\cdot)$ is closed under multiplication by a nonnegative constant $\alpha \in \mathbb{R}_0^+$ and under product of two ambiguity measures, that is, both $A(\phi^{\mathcal{K}}(G)) = A_1(\phi^{\mathcal{K}}(G)) \cdot \alpha$ and $A(\phi^{\mathcal{K}}(G)) = A_2(\phi^{\mathcal{K}}(G)) \cdot A_3(\phi^{\mathcal{K}}(G))$ are again valid graph ambiguity measures satisfying the requirements listed in Definition 2.6.

For completeness, we extend the graph ambiguity measure for the empty set $\emptyset$.

Definition 2.7 (*Empty set ambiguity*). The ambiguity of a graph G with an empty set of vertices $\mathcal{V} = \emptyset$ is zero, since

$$A(\phi^{\mathcal{K}}(G)) = A(0) = 0. \tag{36}$$

2.4. Embedding properties

In this section we will study some properties of the embedding functions $\varphi^{\mathcal{K}}(\cdot)$ and $\phi^{\mathcal{K}}(\cdot)$ (see Definitions 2.4 and 2.5, respectively). To be able to deal with some of these analyses, we need to equip the embedding space with a metric distance. So, we define the *metric space* as $(\mathcal{I}, m_p)$, where $m_p(\cdot, \cdot)$ is a Minkowski distance of order p.

2.4.1. Commonalities among vertices

An interesting property of our embedding function $\varphi^{\mathcal{K}}(\cdot)$ is that it will map two vertices into close points in the unit hypercube $\mathcal{I}$ if they share *commonalities* in terms of a structural definition in the graph. Now we will prove that the structural definitions, that is, degree concentration and centrality factors, of any two vertices in a graph are reflected in the embedding space considering their mutual distance.

Claim 2.1 (*Mutual distance*). *If two given vertices $v, u \in \mathcal{V}(G)$ have similar structural properties in the graph G, regardless of their direct nearness in G, they will be mapped close in $\mathcal{I}^k$ by the embedding function $\varphi^{\mathcal{K}}(\cdot)$, considering any hard k-clustering $\mathcal{K} \in \Gamma^*(G)$.*

Proof. To prove the claim we just need to exploit the metric properties of $\mathcal{I}^k$. Considering this, we have

$$\begin{aligned} \sqrt[p]{k} \geq m_p(\varphi^{\mathcal{K}}(v), \varphi^{\mathcal{K}}(u)) &= \left(\sum_{i=1}^{k} \left| \mu_{\mathcal{F}_i}(v) - \mu_{\mathcal{F}_i}(u) \right|^p \right)^{1/p} \\ &= \left(\sum_{i=1}^{k} \left| \alpha_i(v) \cdot \exp\left(-\frac{d(v, \delta(\mathcal{C}_i))}{t^2} \right) - \alpha_i(u) \cdot \exp\left(-\frac{d(u, \delta(\mathcal{C}_i))}{t^2} \right) \right|^p \right)^{1/p}. \end{aligned} \tag{37}$$

In the last line of Eq. (37), it is possible to clearly see that if they have similar degree concentration ($\alpha_i(v)$ and $\alpha_i(u)$) and similar distances from the chosen representation framework, regardless of their order, they will produce a small numeric value in the summation, that is, a small distance in the embedding space $\mathcal{I}^k$. □

Claim 2.1 will hold also if we consider two non-connected vertices, whose distance in the graph is by definition ∞. An isolated vertex u will be mapped into the origin $\underline{\mathbf{0}}$ and the distance between any other vector representation of another vertex v will be simply the p-th norm of $v - \|\underline{\mathbf{f}}_v\|_p$.

2.4.2. Equivalence between clusterings

As observed in Section 2.1, given two distinct hard k-clusterings $\mathcal{K}_i, \mathcal{K}_j \in \Gamma^*(G)$, they can share the same induced soft clustering $\mathcal{S}$ and then they will produce the same graph ambiguity measure. Consequently, it is necessary to explore the equivalence relation between the clusterings in $\Gamma^*(G)$.

For the property of symmetry of any fuzzy entropy function (see Definition 1.7), and from the fact that any permutations of the components of the input vector will produce the same entropy value (e.g., $E([a, b]^T) = E([b, a]^T)$, $\forall b, a \in [0, 1]$), we can define the equality condition for the vector representations of the graph, based on the induced soft clustering.

Definition 2.8 (*Induced soft clustering equality*). Given two distinct induced soft clusterings $\mathcal{K}_i \succ \mathcal{S}_i, \mathcal{K}_j \succ \mathcal{S}_j$, we say that these are equal if and only if they induce the same vector representation for the graph. Formally,

$$\mathcal{S}_i = \mathcal{S}_j \Leftrightarrow \phi^{\mathcal{K}_i}(G) = \Pi\phi^{\mathcal{K}_j}(G), \tag{38}$$

for all permutation matrices Π of order n. The equality is also given for complementary values, that is, two vector representations are considered equal also if $\phi^{\mathcal{K}_i}(G)_h = 1 - \phi^{\mathcal{K}_j}(G)_h,\ \forall h = 1 \to n$.

What matters are the values of the representations, not their order. For example $[a, b]^T$ would be equal to $[b, a]^T$, $[1 - a, 1 - b]^T$, and $[1 - b, 1 - a]^T$.

Now we introduce the important definition of equivalence between hard k-clusterings.

Definition 2.9 (*Clustering equivalence*). Given two hard k-clusterings $\mathcal{K}_i, \mathcal{K}_j \in \Gamma^*(G)$ of the graph $G = (\mathcal{V}, \mathcal{E})$, we call these equivalent if and only if they have the same ambiguity value. Formally, the equivalence relation $\equiv_A$ on Γ^* is defined as

$$\mathcal{K}_i \equiv_A \mathcal{K}_j \Leftrightarrow A(\phi^{\mathcal{K}_i}(G)) = A(\phi^{\mathcal{K}_j}(G)). \tag{39}$$

It is straightforward to check that this equivalence relation satisfies reflexivity, symmetry, and transitivity constraints for all possible hard k-clusterings in Γ^*.

Considering this definition of equivalence between clusterings, we observe that if two clusterings induce the same representation for a graph, then they will be equivalent.

Claim 2.2. *Given two hard k-clusterings $\mathcal{K}_i, \mathcal{K}_j \in \Gamma^*(G)$, if they induce the same soft clustering $\mathcal{S}$, then $\mathcal{K}_i$ and $\mathcal{K}_j$ are equivalent. Formally*,

$$\mathcal{K}_i \succ \mathcal{S} = \mathcal{K}_j \succ \mathcal{S} \Rightarrow \mathcal{K}_i \equiv_A \mathcal{K}_j. \tag{40}$$

Proof. The proof is trivial and follows from Definition 2.8. □

Now we introduce the concept of equivalence class of a hard k-clustering for a given graph G.

Definition 2.10 (*Clusterings equivalence class*). Given a graph $G = (\mathcal{V}, \mathcal{E})$ and a clustering $\mathcal{K}_i \in \Gamma^*$, the class of equivalent clusterings $\{\mathcal{K}_i\}$ over Γ^* is defined as

$$\{\mathcal{K}_i\}_{\Gamma^*} = \{\mathcal{K} : \mathcal{K} \equiv_A \mathcal{K}_i,\ \mathcal{K} \in \Gamma^*\}. \tag{41}$$

Sometimes, for notational purposes, the equivalence class $\{\mathcal{K}_i\}_{\Gamma^*}$ can be denoted just with one of its elements, say $\mathcal{K}_i$. Given the equivalence relation $\equiv_A$, it is possible to define the *quotient set* of Γ^*.

Definition 2.11 (*Clusterings quotient set*). Given a graph G and the equivalence relation $\equiv_A$, the set of equivalence classes over Γ^*, considering the relation $\equiv_A$, forms the quotient set of Γ^*, and it is denoted with $\mathcal{Q}(G)$.

The quotient set $\mathcal{Q}(G)$ contains only those hard k-clusterings with different induced representations considering the graph G. The existence of the equivalence relations is very important in order to design optimized algorithms for the graph ambiguity computation, avoiding the evaluation of every possible hard k-clustering in $\Gamma^*(G)$, since some of the induced representations share the same ambiguity.

Now we study the cardinality of the set $\mathcal{Q}(G)$. To rigorously analyze the properties of $\mathcal{Q}(G)$, we introduce a notation to denote hard k-clusterings of $\Gamma^*(G)$. With the following notation, we want to denote what we called the *partition types* of G. Let $G = K_4$ be the complete graph of order four, for instance. We know from Eq. (12) that $|\Gamma^*(G)| = 15$. The possible hard k-clusterings are shown in Table 3.

For $k = 1$ we have one type of partition that we can denote as (4). This notation means that there is a single set of four elements (i.e., the four vertices of the graph). For $k = 2$ we have two types of partition. The first one denotes a partition of type (2)(2), meaning that we group the elements using two sets, containing in turn two elements each.

Table 3
Partition types of the 15 hard k-clusterings for $G = K_4$.

k	Partition type	Hard k-clusterings
1	(4)	$\mathcal{K}_1 = \{\{1, 2, 3, 4\}\}$
2	(2)(2)	$\mathcal{K}_2 = \{\{1, 2\}, \{3, 4\}\}, \mathcal{K}_3 = \{\{1, 3\}, \{2, 4\}\}, \mathcal{K}_4 = \{\{1, 4\}, \{2, 3\}\}$
2	(1)(3)	$\mathcal{K}_5 = \{\{4\}, \{1, 2, 3\}\}, \mathcal{K}_6 = \{\{1\}, \{2, 3, 4\}\}, \mathcal{K}_7 = \{\{2\}, \{1, 3, 4\}\}, \mathcal{K}_8 = \{\{3\}, \{1, 2, 4\}\}$
3	(1)(1)(2)	$\mathcal{K}_9$={{3}, {4}, {1, 2}}, $\mathcal{K}_{10}$={{2}, {4}, {1, 3}}, $\mathcal{K}_{11}$={{2}, {3}, {1, 4}}, $\mathcal{K}_{12}$={{1}, {4}, {2, 3}}, $\mathcal{K}_{13}$={{1}, {3}, {2, 4}}, $\mathcal{K}_{14}$={{1}, {2}, {3, 4}}
4	(1)(1)(1)(1)	$\mathcal{K}_{15} = \{\{1\}, \{2\}, \{3\}, \{4\}\}$

The second one denotes a partition of type (1)(3), meaning that the grouping is again done with two sets, but with one and three elements for each set, respectively. For $k = 3$, we have one type of partition, denoted as (1)(1)(2), meaning that we employ three sets, containing one, again one, and two elements, respectively. Finally, for $k = 4$ we have only one type of partition, denoted as (1)(1)(1)(1).

Using this encoding to denote the types of possible partitions, it is easy to make a connection with the well-known *partition function*, denoted as $p(n)$, of number theory [32,34,60]. Indeed, $p(n)$ counts the number of *distinct* partitions of a positive integer n. That is, it counts the number of distinct ways to write n as a sum of positive integers. The hard k-clustering of $\mathcal{V}(G)$ has practically the same meaning.

Claim 2.3. *Given $G = K_4$, we have $|\mathcal{Q}(G)| = p(4) = 5$.*

Proof. Using Table 3, we need to show that all the hard k-clusterings belonging to, say, the partition type (2)(2) share the same ambiguity. As a consequence, we need to prove also that hard k-clusterings belonging to different partition types (e.g., (2)(2) and (1)(3)) have different ambiguity, eventually yielding a total of $p(4) = 5$ equivalence classes. For instance, it is very easy to understand that $\mathcal{K}_2$ and $\mathcal{K}_3$ are equivalent. Indeed, G being complete, their induced representations will be equal, and consequently they result to be equivalent (see Claim 2.2). The reason for this is that in a complete graph, regardless of the particular hard k-clustering $\mathcal{K}_i \in \Gamma^*$, every vertex is a representative of its cluster (the shortest path length is always one). At the same time, the number of edges between vertices belonging to different clusters is always the same. This is not the case considering $\mathcal{K}_2$ and $\mathcal{K}_5$, for instance. Indeed, in this case the respective degree distributions among the two clusters are not equal (or symmetric), and consequently their representations cannot be the same. The same arguments hold for the other inter- and intra-types of hard k-clustering combinations. □

In the case of complete graphs, the result of Claim 2.3 quite easily generalizes to any positive integer n.

Theorem 2.4. *Let K_n be a complete graph of order n. We have $|\mathcal{Q}(K_n)| = p(n)$.*

In the case of non-complete graphs, it is not assured that for any graph of order, say, $n = 4$ the two hard k-clusterings $\mathcal{K}_2$ and $\mathcal{K}_3$ will be equivalent. In fact, being not complete, some edges will be missing and the degree distributions will certainly not be the same. Moreover, in this case it is not assured that, regardless the clustering, every vertex will be a representative of its own cluster, yielding possibly different representations. Consequently, given a generic graph $G = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}(G)| = n$, we have

$$|\mathcal{Q}(G)| \geq p(n). \tag{42}$$

Eq. (42) fixes a lower bound for what concerns the number of possible equivalence classes of a generic graph of order n. This result is quite important because it helps defining the hardness of the problem from the computational viewpoint.

Assuming that the graph G is weighted, the result of Theorem 2.4 may not be still valid, because degree concentrations can vary even between intra-type partitions, as indeed they are dependent on the particular weighting of the graph. Nonetheless, the bound shown in Eq. (42) is valid also in this case.

2.4.3. Compression factor

Given two vertices $u, v \in \mathcal{V}(G)$, and suppose that they have the same vector representation, that is: $\underline{\mathbf{f}}_v = \underline{\mathbf{f}}_u$. We have shown in Section 2.4.1 that v could be different from u. We can state that the unit hypercube $\mathcal{I}$, equipped, for example, with the Euclidean distance $d_2(\cdot,\cdot)$, is actually a *semi-metric space* $(\mathcal{I}, d_2)$, because the definitiveness constraint of metric distances is not satisfied. Considering this, any number of vertices mapped to the same point will not affect differently the determination of the ambiguity of the graph, that is, only one of those equally mapped vertices is useful for the determination of the graph ambiguity. This claim is easily proved considering the behavior of the t-conorm $\perp$ operator. So, we can define another property that we will call *compression factor* of the vertex mapping function $\varphi(\cdot)$.

Definition 2.12 (*Compression factor*). Given a graph $G = (\mathcal{V}, \mathcal{E})$, for any set of vertices $\mathcal{V}' \in \mathcal{P}(\mathcal{V}(G))$ the compression factor η of the embedding of the vertices in $\mathcal{V}'$ is defined as

$$\eta = 1 - \frac{N-1}{|\mathcal{V}'|-1} \in [0,1], \tag{43}$$

where N stands for the number of *distinguishable* vertices in $\mathcal{V}'$, that is, the ones with different vector representations.

When $\mathcal{V}' = \mathcal{V}(G)$ the compression factor is referred to the whole graph G. The compression factor η can be thought of as a measure to quantify the regularity of a graph. In fact, when we consider a d-regular graph, we will see that η will be exactly 1, because each vertex will be represented by the same point in the embedding space.

3. The graph ambiguity problem

The ambiguity of a graph G is expressed through a measure that depends on the specific representation chosen, that is, it is a relative measure. Moreover, it is intuitive to think that the ambiguity of a complex combinatorial object is a relative concept, conditioned to the adopted representation framework. Now we are able to distinguish two concepts: the measure of ambiguity and the *graph ambiguity problem*. The latter is a computational problem, with the aim of identifying the intrinsic, or characteristic, ambiguity of a graph.

Definition 3.1 (*Graph ambiguity problem*). Given a graph $G = (\mathcal{V}, \mathcal{E})$, the graph ambiguity problem is defined as

$$A^*_G = \min_{\mathcal{K} \in \mathcal{Q}(G)} A(\phi^{\mathcal{K}}(G)), \tag{44}$$

where $\mathcal{Q}(G)$ is the solution space of the optimization problem.

Where it is not ambiguous, we will refer to the intrinsic ambiguity value A^*_G of the graph G simply with A^*. This is a discrete unconstrained optimization problem. The intrinsic ambiguity of the graph is the minimum ambiguity achievable over all possible representations. When we talk about the ambiguity of a graph, without considering any specific representation, we are referring to the best one, that is, A^*, the one that minimizes Eq. (44).

3.1. Analysis

In this section we analyze three properties of graph ambiguity. The first one shows that the measure of ambiguity can be seen also as a measure of *regularity* of the graph. The second one deals with the isomorphism between graphs. Finally, the last one is useful for stating the validity of the measure also for some pathological cases.

3.1.1. d-Regular graph

A d-regular (undirected) graph $G = (\mathcal{V}, \mathcal{E})$ is a graph with $\deg(v) = d,\ \forall v \in \mathcal{V}$, where $0 \le d \le |\mathcal{V}| - 1$; when $d = |\mathcal{V}| - 1$, the graph is said to be *complete*. An automorphism of a graph $G = (\mathcal{V}, \mathcal{E})$ is a permutation $p : \mathcal{V} \to \mathcal{V}$ of its vertices satisfying the edge-preserving property, that is, $(v, u) \in \mathcal{E} \Leftrightarrow (p(v), p(u)) \in \mathcal{E}$. A *distance-transitive* graph is a d-regular graph G in which distances between vertices are preserved under automorphisms, i.e., $d(u, v) = d(p(u), p(v))$ holds $\forall u, v \in \mathcal{V}(G)$. Conversely, *asymmetric* d-regular graphs do not have nontrivial symmetries, that is, they do not

have automorphisms other than the *identity mapping* (the mapping of the graph onto itself). Consequently, for these graphs, distances among vertices are not regular [23].

The degree-regularity is reflected directly by the graph ambiguity, observing that any distance-transitive d-regular graph has no ambiguity.

Theorem 3.1 (*Ambiguity of distance-transitive d-regular graphs*). *The ambiguity of any distance-transitive d-regular graph of any order is zero.*

Proof. Consider a (distance-transitive) d-regular graph $G = (\mathcal{V}, \mathcal{E})$ with $1 \leq d \leq |\mathcal{V}| - 1$. If $|\mathcal{K}| = 1$ then every vertex will share the full concentration degree and centrality. Consequently, every vertex will be a representative of this cluster and thus its membership value will be one. It is straightforward to check that the ambiguity of the graph with this clustering will be zero. Now consider the same graph G, but with a clustering of order $m > 1$. Necessarily it follows that at least an edge will be *cut* and the degree concentration of some vertices (the ones interested by the cut) will be shared among $h \leq m$ different clusters, producing a non-crisp vector representation of the graph $\underline{\mathbf{f}}_G$, where at least one component is in $(0, 1)$. This directly implies a non-zero ambiguity for the graph. □

A direct consequence of Theorem 3.1 is that for any d-regular graph we obtain the maximum concentration factor, that is, $\eta = 1$.

It is worth to discuss the particular case of a complete graph $K_n = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, with *self-loops*. A self-loop is an edge of the form $e_{ii} = (v_i, v_i)$. In this case, analyzing the ambiguity of the graph for $t \to \infty$, the exact maximum of Eq. (33) is achieved when $|\mathcal{K}| = n$, that is, when all the clusters are constituted by only one vertex. Conversely it still holds that $A^*(G) = 0$.

3.1.2. Relation with graph isomorphism

It is possible to establish a relation with the concept of isomorphism between graphs. To this aim, we will restrict to the class of unweighted graphs. Given two graphs G and G', a graph isomorphism is a bijection $f : \mathcal{V}(G) \to \mathcal{V}(G')$ such that $(f(v), f(w)) \in \mathcal{E}(G') \Leftrightarrow (v, w) \in \mathcal{E}(G)$ holds $\forall w, v \in \mathcal{V}(G)$ [23]. If an isomorphism exists, then the two graphs are said to be isomorphic, which is denoted with $G \simeq G'$. We will show that the isomorphism relation will imply the equality of the ambiguity; nonetheless, two graphs with the same ambiguity are not necessarily isomorphic.

Lemma 3.1. *If $G \simeq G'$, then*

1. $\alpha_i(v) = \alpha_i(f(v)), \forall v \in \mathcal{V}(G), \forall i = 1 \to k$,
2. $d(v, u) = d(f(v), f(u)), \forall v, u \in \mathcal{V}(G)$,

where $d(v, u)$ and $d(f(v), f(u))$ are the minimum distances computed in G and G', respectively.

The proof of Lemma 3.1 is given in Appendix A.

Theorem 3.2. *If two graphs are isomorphic then they have the same ambiguity. Formally,*

$$G \simeq G' \Rightarrow A^*_G = A^*_{G'}. \tag{45}$$

Proof. A direct consequence of Lemma 3.1 is that $\forall v \in \mathcal{V}(G)$ holds $\varphi^{\mathcal{K}}(v) = \varphi^{\mathcal{K}'}(f(v))$, because if the degrees and minimum distances are preserved the vector representations of the mapped vertices will be the same, regardless of the chosen $\mathcal{K} \in \Gamma^*(G)$ and the correspondent $\mathcal{K}^f \in \Gamma^*(G')$, derived from the application of the isomorphism $f(\cdot)$ to the elements in the clusters of $\mathcal{K}$. If two graphs have the same representation, they will have the same ambiguity (Claim 2.2), therefore $A^*_G = A^*_{G'}$. □

Theorem 3.2 shows that the ambiguity could be seen as a *graph invariant*, that is, a property of graphs that depends only on the abstract structure. Those properties do not change between graphs of the same class induced by the isomorphism relation [23]. This is an interesting theoretical result, because many other relations could be observed between other characteristic properties of graphs, such as the *clustering coefficient*, *clique number*, *graph spectrum*, *algebraic connectivity*, and *independence number*.

Theorem 3.3. *If two graphs have the same ambiguity, then they are not necessarily isomorphic. Formally,*

$$A^*_G = A^*_{G'} \not\Rightarrow G \simeq G'. \tag{46}$$

Proof. To prove this theorem just note that every d-regular graph of any order has ambiguity equal to zero (see Theorem 3.1), and that not any d-regular graph is necessarily isomorphic to some other d-regular graph. For example K_3 and K_4 cannot be isomorphic because they have different orders. □

3.1.3. Pathological graphs

A zero-order graph is a graph $G = (\emptyset, \mathcal{E})$. We have already shown how to handle this particular case, and it is intuitive to think that a graph of this kind cannot have any ambiguity. A zero-size graph is a graph $G = (\mathcal{V}, \emptyset)$, where each vertex is isolated. Intuitively, also in this case the graph should result unambiguous.

Claim 3.4. *The ambiguity of any zero-order or zero-size graph is always zero.*

Proof. Considering the zero-order case, the proof follows quite obviously from Definitions 2.6 and 2.7, and from straightforward calculations. In the case of a zero-size graph, we note that regardless of the hard k-clustering adopted for the representation, each vertex, being isolated, will be mapped to the origin $\underline{\mathbf{0}}$ of the hypercube $\mathcal{I}$. Consequently, the ambiguity of such a representation will be zero. □

Note that a zero-size graph can be seen as a zero-regular graph, consequently the result of Theorem 3.1 can be applied also in this case.

3.2. Graph ambiguity computation

The mathematical formulation of the graph ambiguity problem is given in Eq. (44) as a discrete optimization problem and states that the characteristic ambiguity of a graph is the one with a minimum measure of ambiguity. Now we need an algorithm to find this minimum.

3.2.1. Computational complexity

Given a (non-pathological) graph $G = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, finding the minimum of Eq. (44) requires evaluating every hard k-clustering in $\mathcal{Q}(G)$. As discussed in Section 2.4.2, the cardinality of the solution space $\mathcal{Q}(G)$ is bounded from below by the evaluation of the partition function $p(n)$. As shown by different authors [32,34,60], this number grows exponentially with n. As a consequence, the graph ambiguity problem should be considered as a problem in **NP** [72], since, given $\hat{\mathcal{K}}$, defined as

$$\hat{\mathcal{K}} = \arg\min_{\mathcal{K} \in \mathcal{Q}(G)} A(\phi^{\mathcal{K}}(G)), \tag{47}$$

the evaluation of $A(\phi^{\hat{\mathcal{K}}}(G))$ can be clearly verified in polynomial time (see Definition 2.6). However, by definition of the class **NP**, we do not reject the possibility to conceive a polynomial time deterministic algorithm to find the minimum of Eq. (44).

3.2.2. Genetic algorithm approach

Given a graph $G = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, and a finite alphabet $\Omega = \{1, 2, \ldots, k\}$, it is possible to represent each hard k-clustering with a vector $\underline{\mathbf{c}} \in \Omega^n$. If the i-th component of this vector is equal to, say, $1 \leq l \leq k$, then this means that the i-th vertex of the graph G is assigned to the cluster $\mathcal{C}_l$. To be able to cover every possible partitioning of G, we set $|\Omega| = n$. A vector constructed in this way is able to encode every possible hard k-clustering of $\Gamma^*(G)$.

A suboptimal approach can be conceived considering the stochastic optimization strategy provided by a genetic algorithm. A genetic algorithm can be defined with a fitness function that is based completely on the graph ambiguity evaluation (see Eq. (33)). Each hard k-clustering $\mathcal{K} \in \Gamma^*(G)$ can be encoded by the vector $\underline{\mathbf{c}}$ that will represent its genetic code. The population of the genetic algorithm $\mathcal{P} = \{\underline{\mathbf{c}}_1, \ldots, \underline{\mathbf{c}}_m\}$ is defined as a set of those vectors. Standard selection, crossover, and mutation operators can be applied directly to these vectors, producing a sequence of evolved populations

(i.e., candidate solutions). Clearly the solution with the best fitness is the one with lowest ambiguity measure. It is worth to remark that this approach does not ensures the convergence to the global minimum A^*, but rather to a suboptimal value.

4. Graph similarity measures by graph ambiguity computation

Given a set $\mathcal{X}$, a *positive definite* (pd) kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric function of two variables that satisfies

$$\sum_{i,j\in\{1,\ldots,n\}} c_i c_j k(x_i, x_j) \geq 0, \tag{48}$$

$\forall n \in \mathbb{N}$, $x_1, \ldots, x_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$. In the fields of pattern recognition and soft computing, such a pd kernel is called *Mercer's kernel* [2] (sometimes called also *valid kernel*), and acts as the core similarity procedure of *kernel machines* [66,71]. Kernel functions of this type can be defined on virtually any input set $\mathcal{X}$, such as graphs and strings, for instance. In the first case, they are known in the literature as *graph kernels* [11,50,55,74]. A recent interesting development in this field is the establishment of the so-called *information-theoretic kernels* [18,54,59], aimed at the definition of pd kernel functions on probability distributions, using concepts derived from information theory.

What we introduce in this section is the definition of *Fuzzy Information-Theoretic Kernel* functions (Fuzzy IT Kernels). These functions are Mercer's kernels founded on fuzzy information-theoretic concepts. The aim is to provide general procedures able to compute the similarity value of two fuzzy sets, whatever they represent. Finally, for the ultimate purpose of this paper, we will employ those Fuzzy IT Kernels in the graphs domain $\mathcal{G}$, yielding graphs similarity functions that we call Fuzzy IT Graph Kernels, founded on the new theoretical concepts provided by graph ambiguity.

4.1. Fuzzy information-theoretic graph kernels

A dissimilarity function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$, defined over a generic set $\mathcal{X}$, is a generalized distance measure that satisfies fewer metric constraints with respect to a proper metric distance [57]. For example, given two discrete probability distributions $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_n)$, a measure of dissimilarity between these distributions can be obtained with the classical *divergence* of Kullback and Leibler [48]. This is not a metric distance, because it is not symmetric in general. Nonetheless, later on a symmetric version was proposed [47]. One of the first attempts to define a *fuzzy divergence* was done by Bhandari and Pal [7]. The proposed fuzzy directed divergence $I(\cdot,\cdot)$ among two fuzzy sets $\mathcal{A}$ and $\mathcal{B}$, defined on the same discrete universe of discourse $\mathcal{U}$, is computed as follows:

$$I(\mathcal{A},\mathcal{B}) = \sum_{i=1}^{n} \left[\mu_{\mathcal{A}}(x_i) \log\left(\frac{\mu_{\mathcal{A}}(x_i)}{\mu_{\mathcal{B}}(x_i)}\right) + (1-\mu_{\mathcal{A}}(x_i)) \log\left(\frac{1-\mu_{\mathcal{A}}(x_i)}{1-\mu_{\mathcal{B}}(x_i)}\right)\right], \tag{49}$$

and its symmetric version is given by

$$J(\mathcal{A},\mathcal{B}) = I(\mathcal{A},\mathcal{B}) + I(\mathcal{B},\mathcal{A}). \tag{50}$$

There are other known generalized formulations of the same concept of fuzzy divergence, such as the fuzzy Rényi divergence of order α proposed by Bajaj and Hooda [5],

$$I_\alpha(\mathcal{A},\mathcal{B}) = \frac{1}{\alpha-1} \sum_{i=1}^{n} \log[\mu_{\mathcal{A}}(x_i)^\alpha \cdot \mu_{\mathcal{B}}(x_i)^{1-\alpha} + (1-\mu_{\mathcal{A}}(x_i))^\alpha \cdot (1-\mu_{\mathcal{B}}(x_i))^{1-\alpha}] \tag{51}$$

that is defined for $\alpha > 0$, $\alpha \neq 1$. Its symmetric version is defined as

$$J_\alpha(\mathcal{A},\mathcal{B}) = I_\alpha(\mathcal{A},\mathcal{B}) + I_\alpha(\mathcal{B},\mathcal{A}). \tag{52}$$

Being able to compute the dissimilarity value of two fuzzy sets, we introduce the definition of the Fuzzy IT Kernel.

Definition 4.1 (*Fuzzy information-theoretic kernel*). Given two fuzzy sets $\mathcal{A}, \mathcal{B} \in \mathcal{F}$, where $\mathcal{F}$ is the crisp set of all fuzzy sets, let $J(\cdot, \cdot)$ be a fuzzy symmetric divergence measure defined on $\mathcal{F}$. We define a Fuzzy Information-Theoretic Kernel function $k : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ as

$$k(\mathcal{A}, \mathcal{B}) = \exp(-\gamma \cdot J(\mathcal{A}, \mathcal{B})^2), \quad \gamma > 0. \tag{53}$$

It is possible to recognize in Eq. (53) the well-known RBF kernel scheme. Such a kernel function is a valid Mercer's kernel, provided that the dissimilarity on which it relies is a metric [66]. It is worth to stress that different *distance-based* families of Fuzzy IT Kernels could be defined, using the rational quadratic and the exponential kernels, for instance. However, in this paper we will restrict the analysis to the RBF kernel. The Fuzzy IT Kernel defined in Eq. (53) possesses valuable properties, as indeed it assumes values in the interval [0, 1]. Moreover, $k(\mathcal{A}, \mathcal{B}) = 1$ holds only if the divergence of the two fuzzy sets is zero, that is, when they are equal. Conversely, it tends to zero as the divergence tends to infinity. When $\gamma = 1/2\sigma^2$ holds, the function shown in Eq. (53) is called a Gaussian RBF, and the σ parameter is known in the literature as the *kernel size*, since its adaptation permits to fit data with different distributions.

The embedding of the graph into the unit hypercube $\mathcal{I}$, described in Section 2.2, can be used to establish a measure of dissimilarity between two represented graphs using the concept of fuzzy directed or symmetric divergence. Given two graphs G_1 and G_2, and let $\underline{\mathbf{f}}_{G_1}$ and $\underline{\mathbf{f}}_{G_2}$ be their respective best representations (i.e., the ones that minimize Eq. (44)), a measure of dissimilarity between those two graphs can be computed as

$$J_\alpha(\underline{\mathbf{f}}_{G_1}, \underline{\mathbf{f}}_{G_2}), \tag{54}$$

as defined in Eq. (52), for instance. The explanation of Eq. (54) is that if two given graphs are different in their structure, this must be reflected also in their representation in the unit hypercube $\mathcal{I}$. Therefore, the measure of dissimilarity can be evaluated computing the divergence of the respective vector representations, since, as discussed in Section 2.2, each vector encodes the membership values of the resulting fuzzy set that describes the graph as a whole.

Substituting Eq. (54) into Eq. (53) we can bridge the fields of fuzzy information theory and graph-based pattern recognition and soft computing, through the definition of the Fuzzy IT Graph Kernel. Indeed, the establishment of a (dis)similarity measure between graphs is one of the basic building blocks of data-driven inference systems able to cope with graphs as input patterns [51]. Moreover, such a function is a pd (graph) kernel, and consequently it can be employed directly also in the so-called *large margin classifiers*, such as the famous SVM system [12].

4.2. Experimental evaluation

In this section, we discuss the experiments performed to assess the performance of the newly introduced Fuzzy IT Graph Kernel in common classification problems defined on graphs. We study different classification problem instances, controlling the difficulty of each setup. We benchmark the Fuzzy IT Graph Kernel function against other two well-known inexact graph matching algorithms, called *weighted Best Matching First* (wBMF) [51], and *Graph Coverage* (GC) [50], respectively. These procedures have been employed in a *simple* classifier based on the k-NN rule. The choice of a simple classifier is motivated by the fact that such a system totally relies on the definition of a dissimilarity function. In this way, we are able to better evaluate the capability of each considered graph matching algorithm. The Fuzzy IT Graph Kernel has been configured with the fuzzy Rényi divergence shown in Eq. (52), setting $\alpha = 2$. The kernel size σ has been fixed to one. In addition, the ambiguity of graphs has been computed using directly the fuzzy entropy function shown in Eq. (2), i.e., applying the linear modifier described in Table 2 with $\nu = 1$ and $\gamma = 0$. Those parameters could have been optimized using a suited cross-validation technique. Anyway, in this paper we will use a fixed setup. All the tests have been performed using four values of k in the k-NN rule, namely 1, 3, 5, and 7.

Each synthetic dataset has been generated using a scheme based on Markov chains, described by Livi et al. [49], which consists in producing a sequence of two-class classification problems. Consequently, we generate graphs from two Markov chains that are entirely described by the respective transition matrices. The hardness of the problem is directly controlled by adjusting the similarity of these two matrices. The more similar they are, more difficult the classification problem becomes, as indeed the sampled graphs will result, with high probability, very similar. For this purpose, we have defined 20 classification problem instances. The generated problems are conceived to be of incremental difficulty, where the last instance is actually impossible to solve due to the total superposition of the classes. Since the adopted

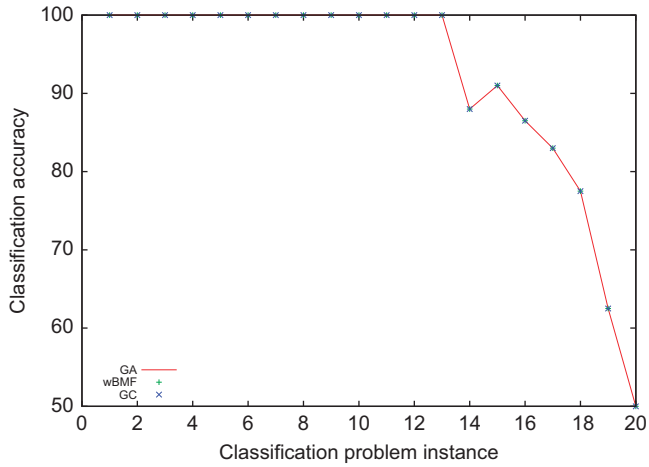

Fig. 3. Classification accuracy with $k = 1$.

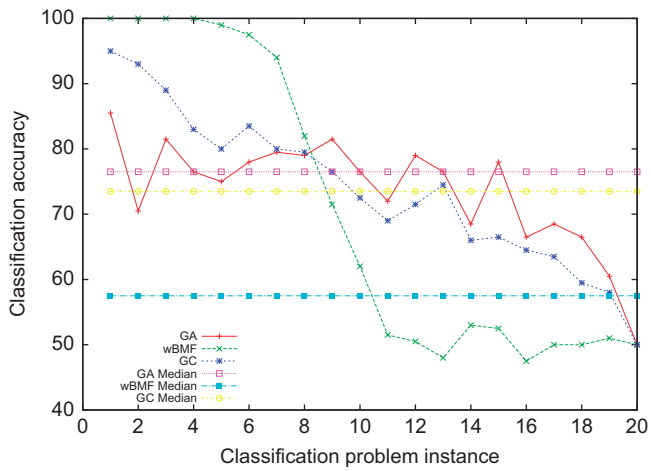

Fig. 4. Classification accuracy with $k = 3$.

implementation of the graph ambiguity depends on the genetic algorithm proposed in Section 3.2.2, we have repeated the test for each problem instance 10 times using different random seeds, reporting the mean values together with the standard deviations. Finally, both training and test set contain 200 graphs of order 50 and variable size between 70 and 150.

Fig. 3 shows the test set classification accuracy (as a percentage) for the three classification systems using $k = 1$. Note that GA in the plot stands for the Graph Ambiguity-based Fuzzy IT Graph Kernel. The outcome is quite interesting because we achieve *exactly* the same results for each system. As expected, being a strictly controlled benchmark test, when the classification becomes really hard, that is, from the instance number 18, we observe a considerable loss of classification accuracy. Nonetheless, even if the employed batch of tests is considered to be very difficult, the three systems show a very robust behavior, with a mean classification accuracy value, achieved over the considered problem instances, greater than 90%. In Figs. 4, 5, and 6 we report the results obtained using a setup for the k-NN classifier with $k = 3$, $k = 5$, and $k = 7$, respectively. It is possible to observe that the wBMF algorithm, regardless of the specific k, shows very good results in the first seven instances. Subsequently, the classification accuracy drops significantly to the level of the random classifier (i.e., 50% of accuracy), while this result is achieved only in the last test by the other two algorithms. Moreover, for $k = 3$ and $k = 5$, its overall median accuracy value is always considerably lower than the other two. On the other hand, GA and GC algorithms show quite similar results, with the GA-based system showing

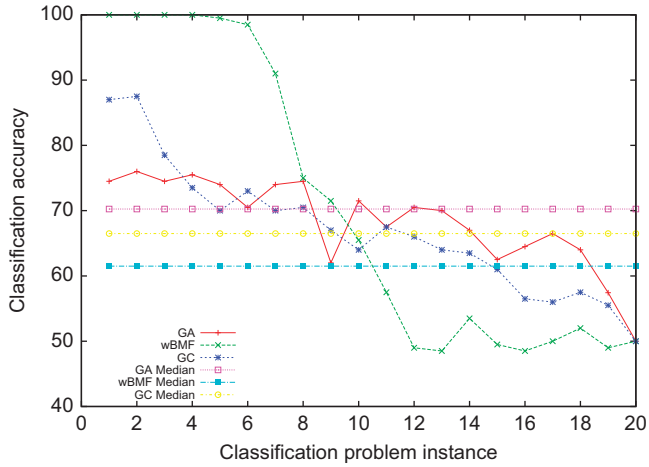

Fig. 5. Classification accuracy with $k = 5$.

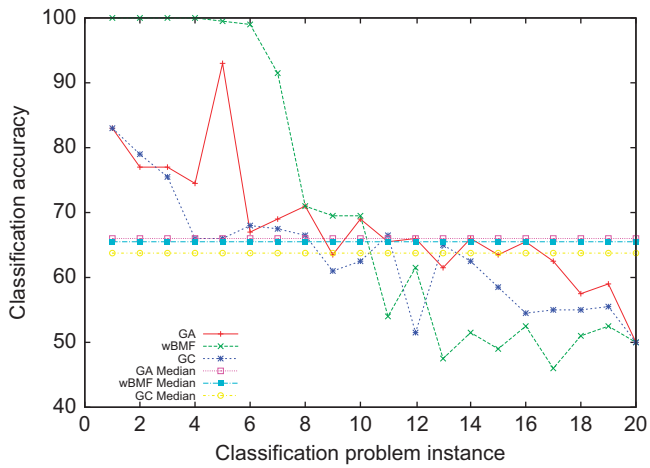

Fig. 6. Classification accuracy with $k = 7$.

always better median accuracy values, denoting a more robust behavior considering the generalization to different sizes of the adopted neighborhood for the classification. Finally, the results adopting the GA-based classification system present always zero standard deviation, confirming the reliability of the results.

The GA-based system employs around 15 min to process a single classification problem instance, while the GC-based system employs around 23 min. The wBMF-based system is significantly faster, since it requires around 1 min to classify a single instance.

5. Conclusions

We have investigated the possibility of the definition of a measure of ambiguity for graphs. The concept of the ambiguity of an object was already explored in the literature using the fuzzy entropy measure, but it has never been formulated on graphs. Our methodology consists in embedding the graph in a suitable vector space $\mathcal{I}$, bringing the problem back to a well-known and understood scenario. We have shown some properties that derive from the proposed embedding functions, such as the fact that the structural definitions between vertices in the graphs are preserved in the embedding space $\mathcal{I}$, considering their mutual distance. We have defined the graph ambiguity problem as a discrete optimization problem, whose aim is to find the best representation, that is, the one that minimizes the measure of

ambiguity of the graph. This ambiguity was referred to as the intrinsic or characteristic ambiguity of the graph. Moreover, by means of the graph ambiguity concept, we have developed the Fuzzy IT Graph Kernel similarity function, joining the fields of fuzzy information theory and graph-based pattern recognition and soft computing. This new approach to graph matching has been tested experimentally on classification problems of controlled difficulty, yielding promising results.

Our aim is to analyze more deeply both theoretical and experimental evaluation of the graph ambiguity concept. There are many areas of interest in which the proposed graph ambiguity measure could be applied. We are particularly interested in the fields of computational intelligence techniques and pattern recognition problems dealing with structured data, where we have to effectively cope with complex problems of data mining-representation and machine learning, such as *graph representation* for clustering and classification [22,36,49,55,62,74] and issues related to *graph partitioning* [14,15,39,40]. In particular, we are highly interested in the evaluation of our new measure of graph ambiguity in real problems concerning inexact graph matching [50,51,77], as discussed and experimented in Section 4. It is worth to stress that graphs are ubiquitous in a considerable number of scientific fields. For instance, we can cite applications to web content-based information retrieval [65], smart grid modeling [25], and complex and biological networks analysis [8,11].

Moreover, module identification in complex graphs could be coped with by solving the optimization problem shown in Eq. (44), and considering as output the optimal hard k-clustering $\hat{\mathcal{K}}$. In this case, possible fields of interest range from bio-chemio informatics [53] to VLSI design [41] and web information retrieval systems [6].

Acknowledgments

The authors of this paper would like to thank the anonymous reviewers for their very helpful comments.

Appendix A

Proof of Lemma 3.1. If the two graphs, G and G', are isomorphic, then by definition they are of the same order and of the same size, and there will exist a bijection $f(\cdot)$ that maps each vertex $v \in \mathcal{V}(G)$ with a vertex $u \in \mathcal{V}(G')$. This means that v has the same topological definition in G as u in G', that is, the same neighborhoods given $f(\cdot)$. Assume for example $\mathcal{V}(G) = \{v_1, v_2, \ldots, v_n\}$, $\mathcal{V}(G') = \{u_1, u_2, \ldots, u_n\}$. Given $\mathcal{K} \in \Gamma^*(G)$ there always exists a correspondent $\mathcal{K}^f \in \Gamma^*(G')$, of the same order k, built considering the mapped vertices by the isomorphism $f(\cdot)$. The proof proceeds considering this correspondence.

Now we check for $\alpha_i(v) = \alpha_i(f(v)), \forall v \in \mathcal{V}(G),\ \forall i = 1 \to k$. By definition of isomorphism $f(\cdot)$, for any two vertices $v, z \in \mathcal{V}(G)$, if they are adjacent in G then $f(v)$ and $f(z)$ must be adjacent in G'. Moreover, if v and z are (are not) in the same cluster in $\mathcal{K}$, then $f(v)$ and $f(z)$ are (are not) in the same cluster of $\mathcal{K}^f$. It follows that the degree concentration of each vertex $v \in \mathcal{V}(G)$ is the same as the one of $f(v) \in \mathcal{V}(G')$, and then $\alpha_i(v) = \alpha_i(f(v)),\ \forall i = 1 \to k$.

Now we check for $d(v, u) = d(f(v), f(u)), \forall v, u \in \mathcal{V}(G)$. The case $(v, u) \in \mathcal{E}(G)$ is trivial. If $(v, u) \notin \mathcal{E}(G)$ we have $d(v, u) = l > 1$. Suppose that the shortest path from v to u is $p = (v = z_0, z_1, z_2, \ldots, z_{l-1}, z_l = u)$. Then, by definition of isomorphism, $\forall (z_i, z_{i+1}) \in p, i = 0 \to l - 1$, must exist an edge $(f(z_i), f(z_{i+1})) \in \mathcal{E}(G')$. This implies that the shortest path sequence of vertices from $f(v)$ to $f(u)$ in G' is $p' = (f(v) = f(z_0), f(z_1), f(z_2), \ldots, f(z_{l-1}), f(z_l) = f(u))$. □

References

[1] C. Aggarwal, H. Wang, Managing and Mining Graph Data, Advances in Database Systems, Springer, 2010.
[2] A. Aizerman, E.M. Braverman, L.I. Rozoner, Theoretical foundations of the potential function method in pattern recognition learning, Autom. Remote Control 25 (1964) 821–837.
[3] S. Al-Sharhan, F. Karray, W. Gueaieb, O. Basir, Fuzzy entropy: a brief survey, in: Proceedings of the 10th IEEE International Conference on Fuzzy Systems, vol. 3, pp. 1135–1139.
[4] C. Alsina, B. Schweizer, M. Frank, Associative Functions: Triangular Norms and Copulas, World Scientific, 2006.
[5] R.K. Bajaj, D. Hooda, On some new generalized measures of fuzzy information, in: Proceedings of World Academy of Science, Engineering and Technology, vol. 62, pp. 642–648.
[6] A. Bernardini, C. Carpineto, M. D'Amico, Full-subtopic retrieval with keyphrase-based search results clustering, in: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT '09, vol. 1, IEEE Computer Society, Washington, DC, USA, 2009, pp. 206–213.

[7] D. Bhandari, N.R. Pal, Some new information measures for fuzzy sets, Inf. Sci. 67 (1993) 209–228.
[8] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D. Hwang, Complex networks: structure and dynamics, Phys. Rep. 424 (2006) 175–308.
[9] B. Bollobás, Modern Graph Theory, Graduate Texts in Mathematics, Springer, 1998.
[10] I. Borg, P. Groenen, Modern Multidimensional Scaling: Theory and Applications, Springer Series in Statistics, Springer, 2005.
[11] K.M. Borgwardt, C.S. Ong, S. Schönauer, S.V.N. Vishwanathan, A.J. Smola, H.P. Kriegel, Protein function prediction via graph kernels, Bioinformatics 21 (2005) 47–56.
[12] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92, ACM, New York, NY, USA, 1992, pp. 144–152.
[13] U. Brandes, A faster algorithm for betweenness centrality, J. Math. Sociol. 25 (2001) 163–177.
[14] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, D. Wagner, On finding graph clusterings with maximum modularity, in: A. Brandstädt, D. Kratsch, H. Müller (Eds.), Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science, vol. 4769, Springer, Berlin, Heidelberg, 2007, pp. 121–132.
[15] U. Brandes, M. Gaertler, D. Wagner, Experiments on graph clustering algorithms, in: Proceedings of 11th European Symposium on Algorithms, Springer-Verlag, 2003, pp. 568–579.
[16] U. Brandes, M. Gaertler, D. Wagner, Engineering graph clustering: models and experimental evaluation, J. Exp. Algorithmics 12 (2008) 1–26.
[17] S. Brin, L. Page, The anatomy of a large-scale hypertextual Web search engine, Comput. Networks ISDN Syst. 30 (1998) 107–117.
[18] A. Carli, M.A.T. Figueiredo, M. Bicego, V. Murino, Generative embeddings based on Rician mixtures: application to kernel-based discriminative classification of magnetic resonance images, in: Proceedings of the First International Conference on Pattern Recognition Applications and Methods, vol. 1, 2012, pp. 113–122.
[19] F. Chung, Spectral Graph Theory, AMS, 1994.
[20] F. Chung, Laplacians and the Cheeger inequality for directed graphs, Ann. Combin. 9 (2005) 1–19.
[21] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd revised edition, MIT Press, 2001.
[22] G. Del Vescovo, A. Rizzi, Automatic classification of graphs by symbolic histograms, in: Proceedings of the 2007 IEEE International Conference on Granular Computing, GRC '07, IEEE Computer Society, 2007, pp. 410–416.
[23] R. Diestel, Graph Theory, Graduate Texts in Mathematics, Springer, 2006.
[24] S. Ding, Z. Shi, S. Xia, F. Jin, Studies on fuzzy information measures, in: Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD '07, vol. 3, IEEE Computer Society, Washington, DC, USA, 2007, pp. 376–380.
[25] F. Dorfler, F. Bullo, Kron Reduction of Graphs with Applications to Electrical Networks, ArXiv e-prints. Eprint: http://arxiv.org/abs/1102.2950, 2011.
[26] J.L. Fan, Y.L. Ma, Some new fuzzy entropy formulas, Fuzzy Sets Syst. 128 (2002) 277–284.
[27] C. Gang, Discussion on new integral entropy and energy of fuzzy sets, in: Proceedings of the Sixth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD'09, vol. 6, IEEE Press, Piscataway, NJ, USA, 2009, pp. 181–185.
[28] X. Gao, B. Xiao, D. Tao, X. Li, A survey of graph edit distance, Pattern Anal. Appl. 13 (2010) 113–129.
[29] R. Guimera, L.A.N. Amaral, Functional cartography of complex metabolic networks, Nature 433 (2005) 895.
[30] P. Hájek, Metamathematics of Fuzzy Logic, Trends in Logic Series, Kluwer, 2001.
[31] D. Hand, H. Mannila, P. Smyth, Principles of Data Mining, Adaptive Computation and Machine Learning, MIT Press, 2001.
[32] G.H. Hardy, S. Ramanujan, Asymptotic formulaæ in combinatory analysis, Proceedings of the London Mathematical Society s2-17 (1918) 75–115.
[33] S. Havlin, Complex Networks, Cambridge University Press, Cambridge, 2010.
[34] J. Hendrik Bruinier, K. Ono, Algebraic formulas for the coefficients of half-integral weight harmonic weak Mass forms, ArXiv e-prints. Eprint: http://arxiv.org/abs/1104.1182, 2011.
[35] D. Hooda, On generalized measures of fuzzy entropy, Math. Slovaca 54 (2004) 315–325.
[36] B.J. Jain, K. Obermayer, Structure spaces, J. Mach. Learn. Res. 10 (2009) 2667–2714.
[37] C. Jianbin, F. Deying, S. Tong, A graph partition-based soft clustering algorithm, in: Proceedings of the Second International Symposium on Intelligent Information Technology Application, vol. 02, IEEE Computer Society, Washington, DC, USA, 2008, pp. 572–577.
[38] I. Jolliffe, Principal Component Analysis, Springer Series in Statistics, Springer, 2002.
[39] R. Kannan, S. Vempala, A. Vetta, On clusterings: good, bad, and spectral, J. ACM (JACM) 51 (2004) 497–515.
[40] G. Karypis, Multi-constraint mesh partitioning for contact/impact computations, in: Proceedings of the 2003 ACM/IEEE Conference on Supercomputing, SC '03, ACM, New York, NY, USA, 2003, pp. 56.
[41] G. Karypis, V. Kumar, Multilevel k-way hypergraph partitioning, in: Proceedings of the 36th Annual ACM/IEEE Design Automation Conference, DAC '99, ACM, New York, NY, USA, 1999, pp. 343–348.
[42] A. Kaufmann, Introduction to the Theory of Fuzzy Subsets, vol. 2, Academic Press, 1975.
[43] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM 46 (1999) 604–632.
[44] G. Klir, Uncertainty and Information: Foundations of Generalized Information Theory, Wiley-Interscience, 2006.
[45] B. Kosko, Fuzzy entropy and conditioning, Inf. Sci. 40 (1986) 165–174.
[46] B. Kosko, Addition as fuzzy mutual entropy, Inf. Sci. 73 (1993) 273–284.
[47] S. Kullback, Information Theory and Statistics, Dover Books on Mathematics, Dover Publications, 1997.
[48] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1951) 79–86.
[49] L. Livi, G. Del Vescovo, A. Rizzi, Graph recognition by seriation and frequent substructures mining, in: Proceedings of the First International Conference on Pattern Recognition Applications and Methods, vol. 1, pp. 186–191.
[50] L. Livi, G. Del Vescovo, A. Rizzi, Inexact graph matching through graph coverage, in: Proceedings of the First International Conference on Pattern Recognition Applications and Methods, vol. 1, pp. 269–272.

[51] L. Livi, A. Rizzi, The graph matching problem, Pattern Anal. Appl. (2012), http://dx.doi.org/10.1007/s10044-012-0284-8.
[52] A.D. Luca, S. Termini, A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory, Inf. Control 20 (1972) 301–312.
[53] H.W. Ma, X.M. Zhao, Y.J. Yuan, A.P. Zeng, Decomposition of metabolic network into functional modules based on the global connectivity structure of reaction graph, Bioinformatics 20 (2004) 1870–1876.
[54] A.F.T. Martins, N.A. Smith, E.P. Xing, P.M.Q. Aguiar, M.A.T. Figueiredo, Nonextensive information theoretic kernels on measures, J. Mach. Learn. Res. 10 (2009) 935–975.
[55] M. Neuhaus, H. Bunke, Bridging the Gap between Graph Edit Distance and Kernel Machines, Series in Machine Perception and Artificial Intelligence, World Scientific, 2007.
[56] N.R. Pal, S.K. Pal, Higher order fuzzy entropy and hybrid entropy of a set, Inf. Sci. 61 (1992) 211–231.
[57] E. Pękalska, R. Duin, The Dissimilarity Representation for Pattern Recognition: Foundations and Applications, Series in Machine Perception and Artificial Intelligence, World Scientific, 2005.
[58] W. Pedrycz, F. Gomide, An Introduction to Fuzzy Sets: Analysis and Design, Complex Adaptive Systems, MIT Press, 1998.
[59] J.C. Príncipe, Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives, Information Science and Statistics, Springer, 2010.
[60] H. Rademacher, On the partition function $p(n)$, Proc. London Math. Soc. s2-43 (1938) 241–254.
[61] A. Rényi, On measures of entropy and information, in: Proceedings of the Fourth Berkeley Symposium on Mathematics, Statistics and Probability, pp. 547–561.
[62] K. Riesen, H. Bunke, Graph Classification and Clustering Based on Vector Space Embedding, Series in Machine Perception and Artificial Intelligence, World Scientific Pub Co. Inc., 2010.
[63] A. Rizzi, G. Del Vescovo, Automatic image classification by a granular computing approach, in: Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing, pp. 33–38.
[64] G.C. Rota, The number of partitions of a set, Am. Math. Mon. 71 (1964) 498–504.
[65] A. Schenker, Graph-Theoretic Techniques for Web Content Mining, Series in Machine Perception and Artificial Intelligence, World Scientific, 2005.
[66] B. Schölkopf, A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, Adaptive Computation and Machine Learning, MIT Press, 2002.
[67] X.G. Shang, W.S. Jiang, A note on fuzzy information measures, Pattern Recognition Lett. 18 (1997) 425–432.
[68] C.E. Shannon, A Mathematical Theory of Communication, CSLI Publications, 1948.
[69] C.E. Shannon, W. Weaver, A Mathematical Theory of Communication, University of Illinois Press, Champaign, IL, USA, 1963.
[70] H. Sharp Jr., Cardinality of finite topologies, J. Combin. Theory 5 (1968) 82–86.
[71] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.
[72] M. Sipser, Introduction to the Theory of Computation, 2nd edition, PWS Pub., Boston, 2006.
[73] K. Stephenson, M. Zelen, Rethinking centrality: methods and examples, Soc. Networks 11 (1989) 1–37.
[74] S.V.N. Vishwanathan, K.M. Borgwardt, R.I. Kondor, N.N. Schraudolph, Graph kernels, J. Mach. Learn. Res. 11 (2010) 1201–1242.
[75] S. Wasserman, K. Faust, Social Network Analysis: Methods and Applications, Cambridge University Press, 1994.
[76] S. White, P. Smyth, Algorithms for estimating relative importance in networks, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, ACM, New York, NY, USA, 2003, pp. 266–275.
[77] B. Xiao, X. Gao, D. Tao, X. Li, HMM-based graph edit distance for image indexing, Int. J. Imaging Syst. Technol. 18 (2008) 209–218.
[78] R.R. Yager, On measures of fuzziness and fuzzy complements, Int. J. General Syst. 5 (1979) 221–229.
[79] C. You, M. Wen, The entropy of fuzzy vectors, Comput. Math. Appl. 56 (2008) 1626–1633.
[80] L. Zadeh, Probability measures of Fuzzy events, J. Math. Anal. Appl. 23 (1968) 421–427.
[81] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.