Data Mining and Knowledge Discovery (2020) 34:980–1021
https://doi.org/10.1007/s10618-020-00686-9

Counting frequent patterns in large labeled graphs: a hypergraph-based approach

Jinghan Meng[1] · **Napath Pitaksirianan**[1] · **Yi-Cheng Tu**[1]

Received: 22 February 2019 / Accepted: 15 April 2020 / Published online: 5 May 2020

Abstract

In recent years, the popularity of graph databases has grown rapidly. This paper focuses on single-graph as an effective model to represent information and its related graph mining techniques. In frequent pattern mining in a single-graph setting, there are two main problems: support measure and search scheme. In this paper, we propose a novel framework for designing support measures that brings together existing minimum-image-based and overlap-graph-based support measures. Our framework is built on the concept of occurrence/instance hypergraphs. Based on such, we are able to design a series of new support measures: minimum instance (MI) measure, and minimum vertex cover (MVC) measure, that combine the advantages of existing measures. More importantly, we show that the existing minimum-image-based support measure is an upper bound of the MI measure, which is also linear-time computable and results in counts that are close to number of instances of a pattern. We show that not only most major existing support measures and new measures proposed in this paper can be mapped into the new framework, but also they occupy different locations of the frequency spectrum. By taking advantage of the new framework, we discover that MVC can be approximated to a constant factor (in terms of number of pattern nodes) in polynomial time. In contrast to common belief, we demonstrate that the state-of-the-art overlap-graph-based maximum independent set (MIS) measure also has constant approximation algorithms. We further show that using standard linear programming and semidefinite programming techniques, polynomial-time relaxations for both MVC and MIS measures can be developed and their counts stand between MVC and MIS. In addition, we point out that MVC, MIS, and their relaxations are bounded within constant factor. In summary, all major support measures are unified in the new hypergraph-based framework which helps reveal their bounding relations and hardness properties.

Keywords Data mining · Graph mining · Support measure · Hypergraph

Responsible editor: M.J. Zaki

Extended author information available on the last page of the article

1 Introduction

Graphs have become increasingly important in modeling complicated structures, such as chemical compounds, bimolecular structures, social networks, aviation maps, and the Web. Recent years have witnessed intensive studies on mining graph databases for interesting patterns. Such endeavors often involve calculating the frequency of the identified patterns (i.e., subgraphs). As shown in many problems, frequent patterns are believed to reveal essential features of the system modeled. A clear definition of any frequent pattern mining problem depends on a *support measure* as a notion of the frequency of the patterns of interest.[1] In a transaction-based frequent pattern mining setup, the development of a support measure is straightforward as we only need to count individual graphs (in a graph database) that contain the query pattern. The problem is more interesting and challenging in a single-graph setup, in which the frequent patterns are to be found in only one graph that often consists of a large number of vertices and edges.

The design of a support measure is non-trivial in the single-graph environment as the measure has to fulfill several requirements. For example, an obvious definition of support of a pattern is the number of its occurrences in the input graph (see more details in Sect. 2). However, this definition possesses a feature in that the support may increase when extending a pattern with more edges/vertices. It is not hard to see such feature is undesirable: when a query pattern grows, the search becomes more selective thus the support should decrease. First introduced by Vanetik et al. (2002), *anti-monotonicity* is well accepted by the graph mining community as an essential rule for support measure design. Vanetik et al. (2002) also proposed an anti-monotonic support measure called the *maximum independent set* (MIS) support. The MIS is built on an important concept named *overlap graph*, which is a graph that consists of the instances of the query pattern in the original graph (database) as vertices and the overlap of such instances as edges. The main problem of MIS is the lack of efficient algorithms—it is proved to be NP-hard.

Another design of support measure named the **mininum-image-based** (MNI) support (see Bringmann and Nijssen 2008 for details) is based on the technique of *vertex images*. Being another anti-monotonic support, MNI requires only linear time to compute. The MNI support, however, has serious drawbacks due to its overestimation of independent occurrences by ignoring the topological structure of the query pattern and partial overlap of occurrences. This lowers its value in real applications. The overlap-graph-based support (represented by MIS) and MNI support, as well as their variants, represent the two major bodies of work in defining support measures in frequent graph mining.

Motivation While both MIS and MNI are anti-monotonic, they stand on opposite sides of the spectra of overestimation and efficiency. Therefore, the main objective of this study is to set up a new framework that unifies existing two categories of support measures, and such a framework serves the following purposes: (1) we can better understand existing support measures (with improved hardness and bounding

[1] For that, we use the words *frequency* and *support* interchangeably in this paper. We also use the word *support* and the phrase *support measure* in the same way.

Table 1 Table of notations

MIS	Maximum independent set measure
$MIES$	Maximum independent edge set measure
MNI	Minimum-image-based measure
MI	Minimum instance measure
MVC	Minimum vertex cover measure
$RMVC$	Polynomial-time relaxation of minimum vertex cover measure
$RMIES$	Polynomial-time relaxation of maximum independent edge set measure
σ_*	Value of measure $*$

theorems); (2) we can build new support measures that combine the best of the two worlds: they are fast (with linear/polynomial time), avoiding the high cost of computing MIS support measure, and avoid over-estimation, without over counting patterns as MNI; and more general, (3) we could develop a good number of choices spreading out the domains of support counts and computational efficiency to allow users choose the proper measure. For example, users can choose a measure, depending on the dataset characteristics (e.g., sparsity, label diversity of data graph) and computational resources (e.g., the time she is willing to wait). Hence there are urgent needs to fill such a gap among existing measures MIS and MNI (Table 1).

In this paper, we first introduce the concept of **occurrence/instance hypergraph**, which is a graph built on the occurrences or instances of the pattern. We show that there is a natural mapping of MNI in the hypergraph setting. As to the MIS, we show it is equivalent (in both value and computational complexity) to a support measure defined from the occurrence/instance hypergraph, the **maximum independent edge set support (MIES)**. We explain that overlap-graph-based MIS is equivalent to MIES in the hypergraph framework. We also discuss the differences between the new hypergraph framework and overlap hypergraph introduced by Wang and Ramon (2012).

Based on the hypergraph concept, we define new support measures: the **minimum instance (MI)** measure in Sect. 3.3, and the **minimum vertex cover (MVC)** measure in Sect. 3.4. For the MI support measure, we show that the existing MNI support is its upper bound, or in other words, MI is closer to the MIS support of a pattern than the MNI. Same as MNI, the MI support is also linear-time computable. The MVC support returns frequency that is even closer to MIS. Although computing MVC measure is NP-hard, MVC enjoys a k-competitive approximate algorithm. We disccuss the relationship between MVC and a overlap-graph-based measure named MCP proposed by Calders et al. (2008) in Sect. 3.4.1. Furthermore, we provide **polynomial-time MVC (RMVC)** and **polynomial-time MIES (RMIES)** relaxations for MVC and MIES respectively. This makes MVC and MIS/MIES more efficient while still providing meaningful frequency values. Bounding theorems that describe the differences among all support measures included in the hypergraph-based framework are also presented. It shown in Bringmann and Nijssen (2008) that

$$\sigma_{\mathbf{MIS}} \leq \sigma_{\mathbf{MNI}}.$$

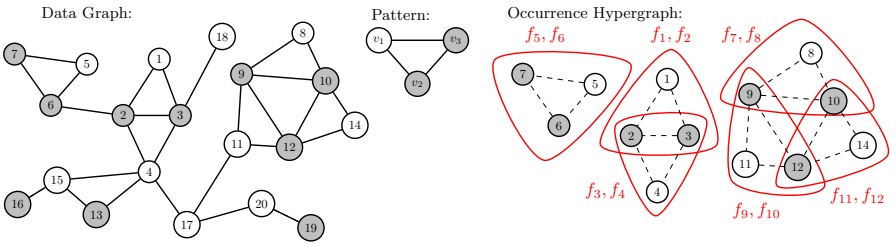


Fig. 1 An example showing support measures of one pattern and a data graph as well as the occurrence hypergraph. In this case, we have $\sigma_{MIS} = \sigma_{MIES} = 3 \leq \sigma_{MVC} = 4 \leq \sigma_{MI} = 5 \leq \sigma_{MNI} = 6$

Table 2 Complexity of measures

Measure	$MIS, MIES, MVC$	$RMIES, RMVC$	MI, MNI
Complexity	NPH const-approx	Polynomial	Linear

In the hypergraph framework, the bounds of all aforementioned support measures are as follows:

$$\sigma_{\mathbf{MIS}} = \sigma_{\mathbf{MIES}} \leq \sigma_{\mathbf{RMIES}} = \sigma_{\mathbf{RMVC}} \leq \sigma_{\mathbf{MVC}} \leq \sigma_{\mathbf{MI}} \leq \sigma_{\mathbf{MNI}}.$$

As an example, Fig. 1 illustrates the hypergraph framework and displays counts of support measures of a one-edge pattern in a small data graph.

The computational complexity of all measures is summarized in the following Table 2.

When modeling pattern occurrences/instances as hypergraphs, an essential finding about pattern occurrences/instances is revealed: the hypergraphs of interest belong to a special group called uniform hypergraphs instead of general hypergraphs. From this finding we show that if a pattern has k nodes, overlap graphs of pattern occurrences (instances) are actually in a subcategory of so called $(k+1)$-claw-free graph, and the MIS support can be approximated within a constant factor. In k-uniform hypergraphs, the ratio between MIES and MVC is within constant k. Although the standard linear programming (LP) and semidefinite programming (SDP) relaxation of MIS are studied in overlap graphs (Calders et al. 2008), to our best knowledge we are the first to study them in uniform hypergraphs. We discover that if the LP relaxation of MIES and SDP relaxation of MIS can be used to derive new polynomial-time measures in hypergraph framework, then the measure derived from the SDP relaxation will be strictly stronger than the one derived from the LP relaxation and the integrality gaps of LP and SDP are $k - 1 + \frac{1}{k}$ and $\frac{k+1}{2}$ respectively.

In addition, we analyze the concepts of overlap such as harmful overlap (HO) in Fiedler and Borgelt (2007) and present potential applications of them in designing and improving support measures in hypergraph framework. We shall give a theorem proving that only same-label node subsets are necessary for effectively counting minimum number of node subset images. From this theorem we develop a new linear-time sup-

port measure named **harmful overlap minimum instance based support (HO-MI)** and show that the value of HO-MI falls between MVC and MI.

A preliminary version of this paper was published in Meng and Tu (2017). We significantly extended that version by showing a series of new findings. These include: (1) new constant approximation theorems, such as the NP-hard MIES support has constant approximation algorithms; overlap graphs fall into a category of so called $(k+1)$-claw-free graph, together with the fact that MIS is equivalent to MIES, the MIS support can be approximated within a constant factor; (2) furthermore we present the improved ratio between measures, for example the ratio between MIES and MVC is within constant k; we also compare polynomial time measures derived from standard linear programming (LP) and semidefinite programming (SDP) in the hypergraph framework and show that the integral gap of LP is $k-1+\frac{1}{k}$ for k-uniform hypergraphs and the SDP relaxation is strictly stronger than the LP relaxation, its integral gap is at most $\frac{k+1}{2}$; (3) we analyze the concepts of overlap and present potential applications of them in designing and improving support measures in the hypergraph framework; we propose a new variant of MI which is also linear-time computable; in addition, we give a theorem proving that only same-label node subsets are necessary for effectively counting minimum number of node subset images, which is the key for designing minimum-image-based support measures.

The rest of this paper is organized as follows: in Sect. 2, we formally define the problem and sketch the necessary background for the problem; in Sect. 3, we present a framework, introduce our new support measures and study their feature, and show that this framework unifies all support measures mentioned in this paper; in Sect. 4, we discuss its potential in defining and studying a wide range of support measures, and new hardness and approximation theorems among all support measures studied in this paper; in Sect. 5 we present and review the experimental evaluations of the major support measures discussed in this paper; in Sect. 6, we present a brief review of related work; and we conclude our paper in Sect. 7.

2 Preliminaries

In this section, we introduce basic notations to describe the problem and the necessary background.

2.1 Labeled graphs

In this paper, we only consider the case of a labeled graph, which is simply referred to as *graph* hereafter. In all figures of this paper, the shade of a vertex represents its label.

Definition 1 A (undirected) **labeled graph**

$$G = (V_G, E_G, \lambda_G)$$

consists of a set of vertices V_G, a set of edges $E_G \subseteq V_G \times V_G := \{(u, v) \mid u, v \in V_G,\ u \neq v\}$ and a labeling function $\lambda_G : V_G \to \Sigma$ that maps each vertex of the graph to an element of the alphabet Σ. We use $\mathcal{G}$ to denote the class of all graphs.

Definition 2 A graph $S = (V_S, E_S, \lambda_S)$ is a **subgraph** of $G = (V_G, E_G, \lambda_G)$ if V_S is a subset of V_G and E_S is a subset of E_G and for all $v \in V_S,\ \lambda_S(v) = \lambda_G(v)$.

Definition 3 A **pattern** $P = (V_P, E_P, \lambda_P)$ is a labeled graph we use as a query against another graph.

Definition 4 Let P be a graph pattern, and p a subgraph of P, denoted by $p \subseteq P$. We call p a **subpattern** of P, and likewise, we call P a **superpattern** of p.

2.2 Graph isomorphism

Given the problem of finding pattern P in a large dataset graph G, we need techniques for determining whether P is structural identical to G or a subgraph of G, and consequently decide if pattern P appears in dataset graph G.

Definition 5 A graph G_1 is **isomorphic** to G_2 if and only if there exists is a bijection (one-to-one mapping) between the vertex sets of G_1 and G_2

$$f : V_{G_1} \to V_{G_2}$$

that preserves vertex labels and

$$(v_1, v_2) \in E_{G_1} \text{ if and only if } (f(v_1), f(v_2)) \in E_{G_2}.$$

Generally speaking, an isomorphism is an edge-preserving bijection between the vertex sets of two graphs, say G_1 and G_2. In this case, one can take G_1 as a copy of G_2, or vise versa.

Definition 6 An **automorphism** of graph G is an isomorphism from G onto itself.

Definition 7 A graph G_1 is **subgraph isomorphic** to G_2 if and only if G_1 is isomorphic to a subgraph of G_2.

In order for us to know how many times a pattern appears in a data graph, we need to define the concept of an occurrence and an instance of the pattern in the data graph.

In this article when there is no confusion we write graph $G = (V_G, E_G, \lambda_G)$ as $G = (V_G, E_G)$ for simplicity.

Definition 8 Given a pattern $P = (V_P, E_P)$ and a graph $G = (V_G, E_G)$, an **occurrence** is an isomorphism f from pattern P to a subgraph of G. That is to say f is also a subgraph isomorphism from P to G.

Definition 9 Given a pattern $P = (V_P, E_P)$ and a graph $G = (V_G, E_G)$, a subgraph S of G is an **instance** of pattern P in G when there exists an isomorphism between P and S.

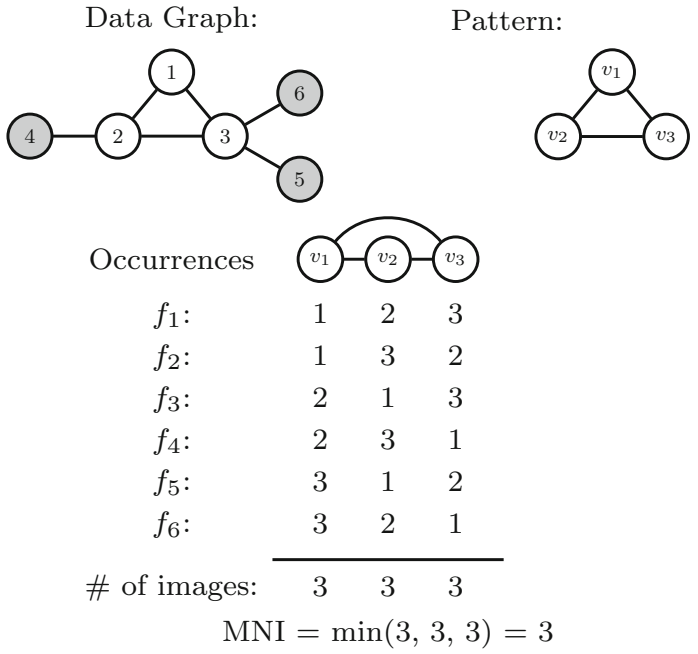


Fig. 2 An example showing a triangle-shaped pattern with 6 occurrences and 1 instance in a data graph. In this case, MNI overestimates the count of pattern—we have an MIS measure of 1 but the MNI measure equals 3

Note that occurrence and instance are two different concepts. An occurrence is an isomorphism between pattern P and a subgraph of dataset graph G, while an instance is a subgraph of G that is isomorphic to pattern P. There can be multiple occurrences mapping pattern P to one instance. For example, in Fig. 2 the triangle-shaped pattern has 6 occurrences f_1, f_2, f_3, f_4, f_5, f_6 in the data graph, while it has only one instance which is the subgraph induced by vertices 1, 2 and 3. Occurrence and instance are key components in the support measure framework we propose.

2.3 Overlap concepts and support measure

The purpose of defining support measure is to count the appearances of a pattern P in a data graph G. The definition of support measure is given below:

Definition 10 A **support measure** of pattern P in data graph G is a function $\sigma : \mathcal{G} \times \mathcal{G} \to \mathbb{R}^+$, which maps (P, G) to a non-negative number $\sigma(P, G)$.

One natural way of defining a pattern support measure is to use its occurrence count, however this measure does not satisfy the *anti-monotonic* property, which states that the support of a pattern must not exceed that of its subpatterns (Vanetik et al. 2002; Kuramochi and Karypis 2005). A more intuitive support measure is the count of instances of the pattern in a dataset graph. This measure, however, is not anti-monotonic either (Vanetik et al. 2002; Kuramochi and Karypis 2005).

Anti-monotonicity is a basic requirement for support measure because most existing frequent pattern mining algorithms depend on it to safely prune a branch of infrequent patterns in the search space for efficiency. Formally, we have

Definition 11 A support measure σ of pattern P in G is **anti-monotonic** if for any pattern p and its superpattern P, we have $\sigma(p, G) \geq \sigma(P, G)$.

To address the above challenge, Vanetik et al. (2002) proposed the first non-trivial anti-monotonic support measure named *maximum independent set based* (MIS) support. The MIS support is developed on top of the so-called *overlap graph* derived from the data graph. We describe the main ideas of this method as follows. First we should explain the concepts of overlap.

Definition 12 (*Vertex overlap*) A vertex overlap of occurrences f_1 and f_2 of pattern $P = (V_P, E_P)$ in data graph $G = (V, E)$ exists if vertex sets $f_1(V_P)$ and $f_2(V_P)$ intersect, that is, $f_1(V_P) \cap f_2(V_P) \neq \emptyset$ where $f_i(V_P) = \{f_i(v) : v \in V_P\}$, i = 1, 2. A vertex overlap of instances $S_1 = (V_{S_1}, E_{S_1})$ and $S_2 = (V_{S_2}, E_{S_2})$ of pattern P exists if vertex sets of S_1 and S_2 intersect, that is, $V_{S_1} \cap V_{S_2} \neq \emptyset$.

Definition 13 Given a pattern $P = (V_P, E_P)$ and a dataset graph $G = (V, E)$, an occurrence (instance) **overlap graph** is a graph O such that each vertex of O represents an occurrence (instance) of P in G, and two vertices u and v are adjacent if the two occurrences (instances) overlap (in sense of one type of overlap defined above).

In this article, we mainly study how occurrences overlap and we only consider overlap in vertex.

Definition 14 An **independent (vertex) set** of graph $G = (V, E)$ is a subset of V, such that no two of which are adjacent.

Definition 15 Given a pattern $P = (V_P, E_P)$ and a data graph $G = (V, E)$, the **maximum independent set based support** is defined as the cardinality of maximum independent vertex set of occurrence or instance overlap graph O:

$$\sigma_{MIS}(P, G) = \max_I |I|,$$

where I is an independent set of O. We use symbol $|\cdot|$ to denote the number of elements in the set.

For example, in Fig. 3 the triangle pattern has 4 occurrences f_1, f_2, f_3, f_4 in data graph. Because they pair-wise overlap, every pair of vertices is connected by an edge in overlap graph. The maximum size of independent vertex set is 1, hence MIS = 1.

The main drawback of the MIS support is computing efficiency—it is shown by Karp (1972) that maximum independent set problem is NP-hard in number of graph vertices. Because MIS proposed in Vanetik et al. (2002) is based on overlap graph, vertices represent instances of pattern in data graph. Thus computing MIS as a support measure is also NP-hard.[2]

[2] In this paper, following conventions of this field, computing time of support measures does not include that for constructing the framework (e.g., overlap graph in the MIS case).

Bringmann and Nijssen (2008) proposed a support measure called *minimum image based support* (MNI). It is based on a technique different from the overlap graph. The main concept here is *image*, which is an existence of a vertex in the pattern (called *node* hereafter) in the data graph. For example, in Fig. 3, node v_1 in the pattern has 2 distinct images because there are occurrences map v_1 to vertices 1, 4 in data graph (e.g., $f_1(v_1) = 1$, $f_2(v_1) = 1$, $f_3(v_1) = 4$ and $f_4(v_1) = 4$). The formal definition of MNI is given below.

Definition 16 Given a pattern $P = (V_P, E_P)$, a data graph $G = (V, E)$, if P has m occurrences $\{f_1, f_2, \ldots, f_m\}$ in G, the **minimum image based (MNI) support** of P in G is defined as

$$\sigma_{MNI}(P, G) = \min_{v \in V_P} |\{f_i(v) : i = 1, 2, \ldots, m\}|.$$

In other words, for each node v in pattern P, MNI support identifies the count c of its unique images, here $c = |\{f_i(v) : i = 1, 2, \ldots, l\}|$. Then MNI support measure of P in G is the minimum count c among all nodes in pattern P. For example, in Fig. 3, the MNI support measure of the triangle pattern is 2.

The anti-monotonicity of MNI is guaranteed by taking the node in P that is mapped to the least number of unique nodes in G.

A clear advantage of MNI support over the NP-hard MIS support is computation time. The reason is that it only requires a set of images for every node in a pattern, and finding the minimum number of distinct images for each set can be done in $O(n)$ where n is the number of occurrences of a pattern. However, MNI support has an obvious disadvantage, that is over-estimation. Let us take a look at the example in Fig. 2 the MIS support of the triangle-shaped pattern is 1 while MNI support is 3, because the minimum number of images of each node is 3. It does not agree with our intuition that the 6 occurrences f_1, f_2, f_3, f_4, f_5, f_6 of the pattern overlap and there is only one instance, which is the subgraph induced by vertices 1, 2 and 3. In other words, MIS counts only independent pattern occurrences, while MNI allows certain degree of overlap exists in occurrences it counts.

The MIS and MNI supports represent the two main flavors of work in the design of support measure for frequent subgraph mining. Both are anti-monotonic yet they stand on far ends of computing efficiency and overestimation of pattern frequency. While the MIS returns the smallest count, there is no efficient algorithm to compute it (Calders et al. 2008). The MNI requires only linear time to compute but can return an arbitrarily large count for a pattern (Bringmann and Nijssen 2008). Both MIS and MNI have variants other than the basic forms mentioned in this section. We will give more details of the variants in Sect. 6. Here we only emphasize that those variants do not significantly change the features of MIS and MNI.

Intuitively, the MNI support returns counts that are closer to the number of occurrences of a pattern. However, it is more natural to define support measure of a pattern according to the number of instances (note that MIS calculates the number of independent instances). Recall the case in Fig. 2: the number of instance is 1, however its MNI support measure is 3, and this does not follow common sense. It is known, however, that the count of instances as a support measure is not anti-monotonic, in this paper

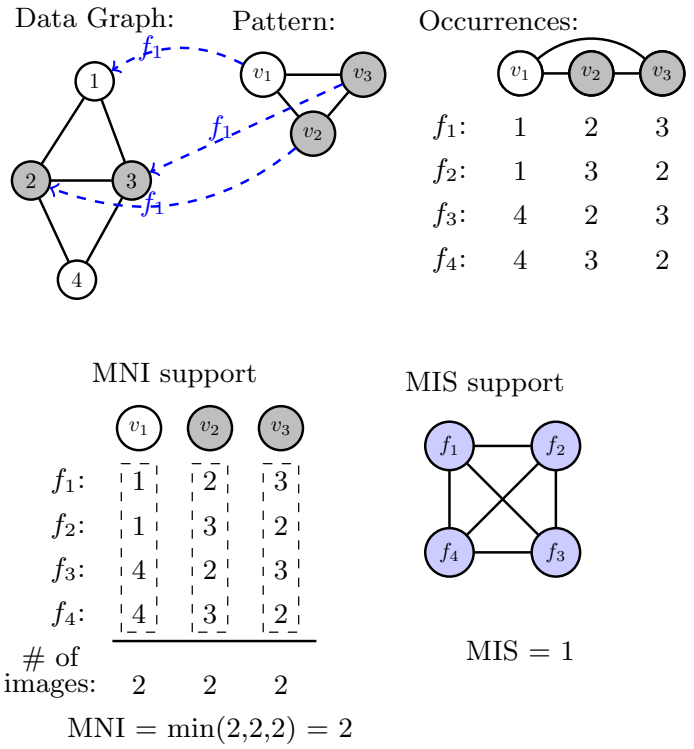


Fig. 3 An example showing the key components (i.e., images and overlap graph) in the calculation of the MNI and MIS support measures

we will present anti-monotonic support measures that achieve counts that are closer to the number of independent pattern instances, as is the MIS support measure.

3 New framework

In this section, we shall introduce a new framework for modeling pattern occurrences. Intuitively, we want to define frequency of pattern occurrences or instances while they overlap in data graph in various ways. Hence we propose a framework that models how occurrences or instances overlap on vertices. In classic graph theory, a graph can be represented by a collection of edges, where each edge joining two vertices. A hypergraph, as a generalization of a graph, in which an edge can join any number of vertices, and it can be represented as a collection of vertex sets. Since occurrences of a pattern usually contains two or more vertices, e.g. in Fig. 3, each occurrence has 3 vertices, and if we study vertex overlap between occurrences, we should be able to model a collection of occurrences as a hypergraph.

In what follows, we introduce a new concept named **occurrence/instance hypergraph** from which not only the existing support measures can be interpreted, but also new support measures can be constructed. Such a concept simplifies the problem of

finding support measures with desired features. Note that this technique is different from the overlap graph used in MIS and the images of occurrences used in MNI. Instead of instances (subgraphs) and occurrences (isomorphisms), we represent a node (i.e., vertex in pattern) image as a vertex and an occurrence/instance as an edge.

Definition 17 A **hypergraph** $H = (V, E)$ consists of a set $V = \{v_1, v_2, \ldots, v_n\}$ of n vertices and a set $E = \{e_1, e_2, \ldots, e_m\}$ of m edges, where each edge is a non-empty subset of V. A **simple hypergraph** H is a hypergraph in which no edge is subset of another edge, that is, if $e_i \subseteq e_j$ then $i = j$. A **k-uniform hypergraph** is a hypergraph such that all its edges have size k.

Definition 18 If pattern $P = (V_P, E_P)$ has m occurrences $\{f_i : i = 1, \ldots, m\}$, the **occurrence hypergraph** of P in G is defined as $H^O = (V, E)$ where $V = f_1(V_P) \cup f_2(V_P) \cup \cdots \cup f_m(V_P)$, and $E = \{e_i : i = 1, \ldots, m\}$, each $e_i = f_i(V_P)$. In other words, hypergraph vertex set V is the collection of all pattern node images, and each edge e_i is a collection of pattern node images mapped by occurrence f_i. We also give each e_i a label f_i to distinguish them from each other.

Definition 19 If pattern $P = (V_P, E_P)$ has m instances $\{S_i = (V_{S_i}, E_{S_i}) : i = 1, \ldots, m\}$ in data graph G, the **instance hypergraph** of P in G is defined as $H^I = (V, E)$ where $V = V_{S_1} \cup V_{S_2} \cup \cdots \cup V_{S_m}$ and $E = \{e_i : i = 1, \ldots, m\}$, each $e_i = V_{S_i}$. We also give each e_i a label S_i to distinguish them from each other.

Note that in Definitions 18 and 19, edges as collections of pattern node images are multisets.

Figure 4 gives a visualization of occurrences of a pattern in a data graph and how occurrences overlap with each other, the upper right part shows pattern occurrences and instances, while the lower right part shows another way of visualize occurrence/instance hypergraph.

Let us use Fig. 4 to show how the hypergraphs are constructed: the occurrence hypergraph $H^O = (V, E)$ has vertex set $V = \{1, 2, 3, 4, 5, 6, 8, 9, 10\}$ and edge set $E = \{e_1, e_2, e_3, e_4\} = \{\{1, 2, 3\}, \{4, 5, 6\}, \{4, 6, 8\}, \{8, 9, 10\}\}$. Note that in occurrence (instance) hypergraph, each edge represents one occurrence (instance) and it is not just a set of vertices. In Fig. 4 the instance hypergraph has 4 edges representing 4 instances, and it is similar to occurrence hypergraph. However, in many other situations occurrence and instance hypergraphs look very different. For example in Fig. 2, occurrence hypergraph of the triangle-shaped pattern has 6 edges because there are 6 occurrences. Although all the edges have the same vertex set $\{1, 2, 3\}$, they are considered as 6 different edges because they represent different occurrences. On the other hand, instance hypergraph of pattern in Fig. 2 has only one edge since there is one instance. This difference between the occurrence hypergraph and instance hypergraph is caused by automorphisms. When a pattern has non-identity automorphisms, multiple occurrences project the pattern to the same instance. If a pattern admits no non-identity automorphism, its occurrence and instance hypergraphs will be quite similar.

For the following discussions, we want to emphasize that, since all edges in occurrence (instance) hypergraph are related to the same pattern, they contain the same

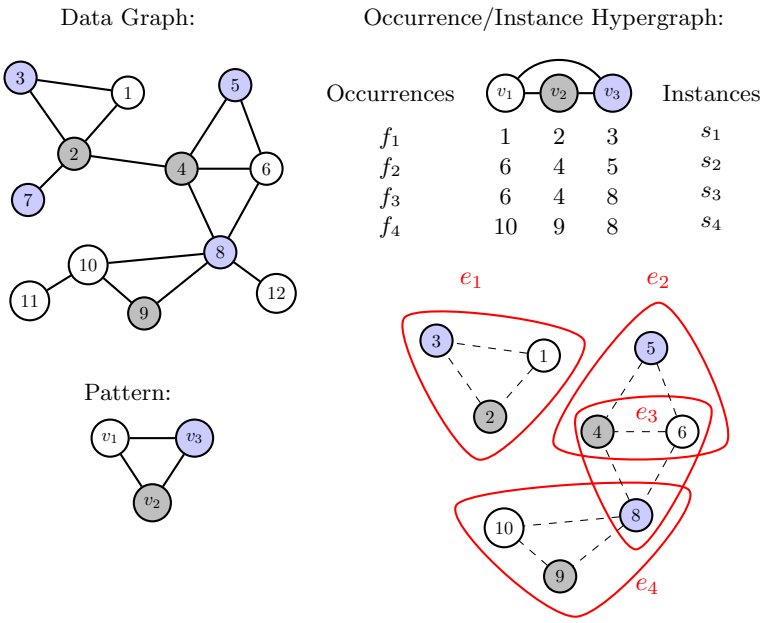


Fig. 4 The occurrence and Instance hypergraph of a triangular pattern within a 12-node data graph

number of vertices which means that occurrence (instance) hypergraphs are *uniform* hypergraphs.

As shown in Fig. 4, pattern occurrences that are represented by hypergraph edges overlap in various degrees and positions. While in occurrence (instance) overlap graphs, each occurrence (instance) is converted to a vertex, if two occurrences (instances) overlap, an edge is generated between them. As a result, how occurrences (instances) overlap is not fully taken into consideration. For example, in Fig. 4, e_3 and e_2 overlap at two vertices but e_3 and e_4 overlap at one vertex. We argue that a hypergraph framework keeps more such information and offers more insights and flexibility for further investigation, as compared to overlap graph based support measure such as MIS presented in Vanetik et al. (2002). In short, the hypergraph is a suitable topological representation of pattern occurrences (instances) for investigating support measures. More details will follow, and let us first discuss how MNI and MIS are embedded in this new framework.

3.1 MIS in hypergraph framework

We now show that, MIS, which is defined based on overlap graphs, can also be mapped to the hypergraph framework. MIS is defined the size of maximum independent set of overlap graph. In other words, MIS counts the maximum number of independent occurrences. In occurrence hypergraph, occurrences are edges, hence intuitively MIS should be equivalent to the size of maximum independent edge set in occurrence hypergraph.

For that, we shall introduce a new measure in hypergraph setting and show it is equivalent to MIS.

Definition 20 Given a pattern P in data graph G and its occurrence (instance) hypergraph $H = (V, E)$, the **maximum independent edge set (MIES) support** measure is defined as

$$\sigma_{MIES}(H) = \max_{E'} |E'|,$$

where E' is an independent edge set of H.

With above formulations, we can show the MIS support measure is equivalent in size to MIES.

Theorem 1 *Given pattern P in data graph G, and its occurrence (instance) hypergraph $H = (V, E)$, we have*

$$\sigma_{MIES}(P, G) = \sigma_{MIS}(P, G).$$

Proof Every edge in occurrence (instance) hypergraph corresponds to a vertex in overlap graph, hence an independent hypergraph edges set corresponds to an independent vertex set in overlap graph. This mapping implies the size of independent edge set in hypergraph is the same as that of independent vertex set in overlap graph. □

We take Fig. 4 as an example, in which the MIES support in overlap graph is 3. Taking a close look, for example, $\{e_1, e_2, e_4\}$ forms a maximum independent set. The MIS in instance hypergraph is also 3.

Theorem 2 *The MIES measure is anti-monotonic.*

In graph theory, there is a concept named conflict graph which is quite similar to overlap graph. In a k-uniform hypergraph $H = (V, E)$, the *conflict graph* is the graph where every edge in E is represented by a vertex. Two vertices are adjacent if and only if the edges in E these vertices correspond to intersect each other.

The overlap graph approach is also similar in nature to how dual hypergraph is built. The definition of dual hypergraph is given as follows:

Definition 21 The **dual hypergraph** $H^* = (E, X)$ of $H = (V, E)$ is a hypergraph whose vertices and edges are interchanged, so that the vertices are given by $E = \{e_1, e_2, \ldots, e_m\}$ and the edges are given by $X = \{X_1, X_2, \ldots, X_n\}$ where $X_j = \{e_i : v_j \in e_i\}$, $j = 1, 2, \ldots, n$, that is, X_j is the collection of all edges in H which contain vertex v_j.

In other words, the dual H^* swaps the vertices and edges of H. For example, in Fig. 5, dual hypergraph has vertices e_1, e_2, e_3 and edges 1, 2, 3, 4. Hence overlap graph and dual hypergraph are similar in the sense that all edges in H are vertices in both dual H^* and overlap graph. If two edges e_i, e_j in H overlap at vertex v then e_i, e_j are contained in edge X_v in dual H^*, while (e_i, e_j) forms an edge in overlap graph.

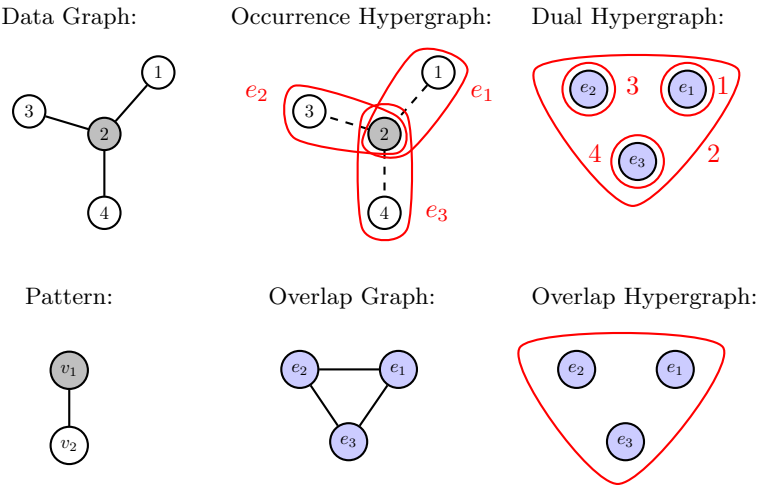


Fig. 5 The instance hypergraph and its dual for a small pattern in a data graph

Actually, each edge in dual H^* is equivalent to a clique in the overlap graph. If H^* is a simple hypergraph, then it is very similar to the overlap hypergraph introduced by Wang and Ramon (2012). The clique in overlap graph is related to intersecting family in occurrence hypergraph (as hypergraph in general), the details are as follows. In a hypergraph $H = (V, E)$, a set F of hyperedges is called an *intersecting family* if every two hyperedges in F have a non-empty intersection. The concepts of clique and intersecting family are important, because they connect the LP and SDP relaxations with maximum independent edge set problem, more details will be presented in Sect. 4.3.

3.2 MNI in hypergraph framework

The technique MNI uses is minimum image, and it uses the minimum number of images of each pattern node as support count. In the hypergraph framework, each occurrence is viewed an edge, and the edge is also an image of the pattern in data graph. A natural question to ask is that does it make sense to count single node images instead of pattern images? MIS gives an answer of counting non-overlapping pattern images, while the count of MNI gives is an obvious upper bound of MIS because the number of single nodes images is alway greater than the number of non-overlapping pattern node images (Theorem 2 in Bringmann and Nijssen 2008). For example, in Fig. 4, nodes v_1, v_2, and v_3 each has 3 distinct images, namely {1, 6, 10}, {2, 4, 9}, and {3, 5, 8}, hence MNI count is 3, which agrees with MIS count. In other words, the difference of handling occurrences between MNI and MIS is single node images and whole pattern node set images. MNI is efficient but with some drawbacks. One comes from a major character of graph pattern, that is multiple isomorphisms can map a pattern into the same subgraph of a data graph, this structural character makes graph pattern counting difficult. We have shown in Fig. 2, that MNI cannot

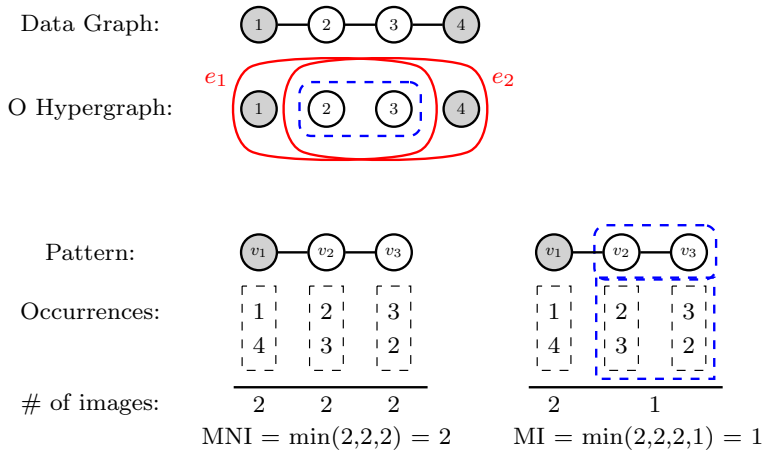


Fig. 6 An example showing the difference between the calculation of the MNI and MI support measures

handle automorphism very well by simply counting single node images. We shall present a new support measure that inherits the efficiency of MNI while improves its intuitiveness by considering automorphism.

3.3 Minimum instance support measure

As described above, the MNI support measure is insensitive to the structure of subgraph patterns. To address this problem of the MNI support, we take the structure of the given pattern into consideration and define a new support measure. Let us explain the main idea by using the example shown in Fig. 6.

In the pattern we have three nodes v_1, v_2, and v_3, each has two images $\{1, 4\}$, $\{2, 3\}$, and $\{3, 2\}$, hence the MNI support of measure of this pattern is 2. However, it misses the fact that the two occurrences overlap on vertices 2 and 3. Apparently the two nodes v_2 and v_3 are symmetric in a subpattern, meaning there is automorphism on the subpattern that maps one to the other. Hence v_2, v_3 can be considered as a set $\{v_2, v_3\}$, which has one image $\{2, 3\}$ as set. This observation leads to the idea of defining a new support measure which takes advantage of patterns' topological structure and reduces overestimation of MNI.

Before defining the new support measure, let us first introduce supportive concepts.

Definition 22 Given a pattern $P = (V_P, E_P)$, a data graph $G = (V, E)$, if P has m occurrences $\{f_1, f_2, \ldots, f_m\}$ in G, a **coarse-grained node subset** W is defined as a subset of V_P that satisfies certain property. The **coarse-grained node subset image count** is defined as

$$c(W) = |\{f_i(W) : i = 1, 2, \ldots, m\}|.$$

Note that in Definition 22 the certain property depends on what support measure is under consideration. For example the property can be that all nodes in the subset have

the same label (as for a measure named HO-MI in Sect. 4.4). In Fig. 6, if a coarse-grained node subset W is $\{v_2, v_3\}$, then its coarse-grained node subset image count $c(W) = |\{\{2, 3\}, \{3, 2\}\}| = 1$. For node subset $M = \{v_2\}$, $c(M) = |\{\{2\}, \{3\}\}| = 2$.

Inspired by our observation, the pattern nodes that are symmetric to each other should be included in the node subsets, hence we shall introduce the definition of *orbit* from classic graph theory as follows.

Suppose that two vertices u and v in graph G are to be considered *related* if there is at least one automorphism f of G such that $f(u) = v$. This is clearly an *equivalence* relation on the vertices of graph G.

Definition 23 The equivalence classes of the vertices of a graph G under the action of the automorphisms are called (vertex) **orbits**.

Note that every node v of a pattern P lies in some orbit (whose size may be 1). Now we are ready to define a new support measure of pattern P using the definition of coarse-grained node subset image count.

Definition 24 Given a pattern $P = (V_P, E_P)$, a data graph $G = (V, E)$, we let T denote the multiset of an orbit in a subgraph of pattern P, and $\mathcal{T} = \{T\}$ denote the collection of all such orbits. The **minimum instance based support (MI)** of P in G is defined as

$$\sigma_{MI}(P, G) = \min_{T \in \mathcal{T}} c(T).$$

Note that $\mathcal{T}$ contains orbits of any subgraph of pattern P. As for the example in Fig. 6, the pattern has coarse-grained node subsets $\{v_1\}$, $\{v_2\}$, $\{v_3\}$ and $\{v_1, v_2\}$, hence $\sigma_{MI}(P, G) = 1$. Now let us study the main properties of the MI support.

Theorem 3 *The MI support measure is anti-monotonic.*

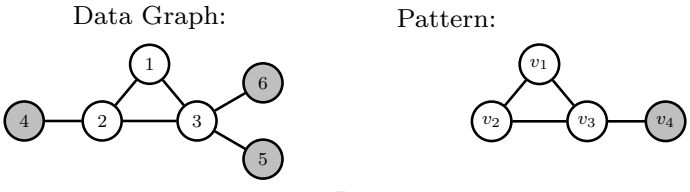
Proof Given pattern $p = (V_p, E_p)$ and its superpattern $P = (V_P, E_P)$ in data graph G, we assume that p has m occurrences $\{f_1, f_2, \ldots, f_m\}$ in G and P has l occurrences $\{f'_1, f'_2, \ldots, f'_l\}$ in G.

First, we have $\sigma_{MI}(p, G) = \min_{T \in \mathcal{T}} c(T)$ and $\sigma_{MI}(P, G) = \min_{T \in \mathcal{T}'} c'(T)$. It is obvious that $\mathcal{T} \subseteq \mathcal{T}'$ by definition. In the next step, we shall prove that for each $T \in \mathcal{T}$ its image count $c(T)$ under the mappings $\{f_1, f_2, \ldots, f_m\}$ is greater than or equal to its image count $c'(T)$ under the mappings $\{f'_1, f'_2, \ldots, f'_l\}$. This is true because any f'_i is an extension of some f_i which implies $f'_i(T) = f_i(T)$, $\forall\ T \in \mathcal{T}$. Therefore $\min_{T \in \mathcal{T}} c(T) \geq \min_{T \in \mathcal{T}} c'(T) \geq \min_{T \in \mathcal{T}'} c'(T)$.

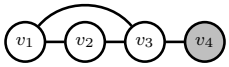
Hence we have $\sigma_{MI}(p, G) \geq \sigma_{MI}(P, G)$. □

Figure 7 shows the anti-monotonicity of MI support measure via an illustrative example. The triangle-shaped pattern induced by nodes v_1, v_2, v_3 has six occurrences $f_1, f_2, f_3, f_4, f_5, f_6$, its MI support is 1. When this pattern extended by connecting node v_4 with node v_3, the new pattern has six occurrences but the MI support of this superpattern is 1 which is not greater than the triangle-shaped pattern.

Theorem 4 *The MI support measure is linear-time computable.*



Occurrences



f_1:	1	2	3	5
g_1:	1	2	3	6
f_2:	1	3	2	4
f_3:	2	1	3	5
g_2:	2	1	3	6
f_4:	2	3	1	
f_5:	3	1	2	4
f_6:	3	2	1	

Fig. 7 An example showing occurrences of a pattern (triangle with v_1, v_2, v_3 nodes) while being extended to a superpattern (by adding one extra node v_4) within the same 6-node data graph

Proof Given $\sigma_{MI}(p, G) = \min_{T \in \mathcal{T}} c(T)$, and there are a fixed number of T for pattern P, it is obvious that calculating $c(T)$ costs $O(n)$ time where n is the number of occurrences. Hence, σ_{MI} is linear-time computable. □

Theorem 5 *Given a pattern P and data graph G, we have*

$$\sigma_{MI}(P, G) \leq \sigma_{MNI}(P, G).$$

Proof Let $\mathcal{W} = \{\{v\} : v \in V_P\}$ we can rewrite MNI support measure as $\sigma_{MNI}(P, G) = \min_{W \in \mathcal{W}} c(W)$.

Since $\mathcal{W} \subseteq \mathcal{T}$, we have $\sigma_{MI}(P, G) = \min_{T \in \mathcal{T}} c(T) \leq \min_{W \in \mathcal{W}} c(W) = \sigma_{MNI}(P, G)$. □

In practice, there will be many cases in which MI measure is strictly smaller than the MNI measure. As in Fig. 6, when considering additional coarse-grained node subsets, minimum count among all of them will decrease. In such a way, we can obtain support count MI that is closer to the number of instances compared with MNI.

In summary, we show that MI support is anti-monotonic, can be computed in linear time, and returns frequency that is bounded by MNI. In the hypergraph setting, MNI and its variant with parameter k reduce the pattern to subsets containing one or k pattern nodes. By revisiting the concept of coarse-grained node subset defined in Sect. 3.3, we see how $\sigma_{MNI}(P, G)$ can be interpreted in terms of such concepts.

In a pattern $P = (V_P, E_P)$, if we let $\mathcal{W} = \{\{v\} : v \in V_P\}$, we can rewrite MNI support measure as

$$\sigma_{MNI}(P, G) = \min_{W \in \mathcal{W}} c(W).$$

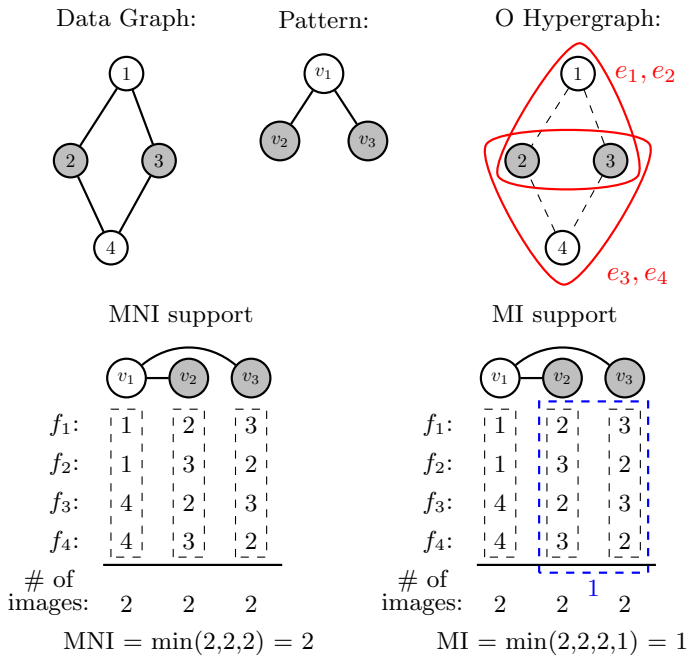


Fig. 8 An example showing the calculation of the MNI and MI support measures in the hypergraph framework

Similarly, we let $\mathcal{W}_k = \{V' : \text{connected } V' \subseteq V_P, |V'| = k\}$, then $\sigma_{MNI}(P, G, k)$ can be interpreted as

$$\sigma_{MNI}(P, G, k) = \min_{V' \in \mathcal{W}_k} c(V').$$

The above definitions show connections among $\sigma_{MNI}(P, G)$, $\sigma_{MNI}(P, G, k)$, and the new support measure $\sigma_{MI}(P, G)$. Figure 8 displays how MNI and MI fit in the hypergraph framework. Note that MI is not simply a transformation from graph pattern to a hypergraph version. For example, in Fig. 8, vertices v_3 and v_2 are not connected by an edge, but they are in an orbit. In conclusion, hypergraph edges (vertex sets) are used to capture desired and essential features of pattern graph for the purpose of designing support measures. From this point of view, hypergraph is indeed a suitable and flexible framework for support measures.

3.4 Minimum vertex cover support measure

The purpose of developing MI is to achieve a reasonable count by avoiding overestimation by MNI. However, MI cannot handle the type of overlap shown in Fig. 9. Although the number of independent instances is only 2 (e.g., $\{1, 5\}$ and $\{4, 8\}$ are independent), we still get MI = MNI = 4. Moreover, there are merely three possible

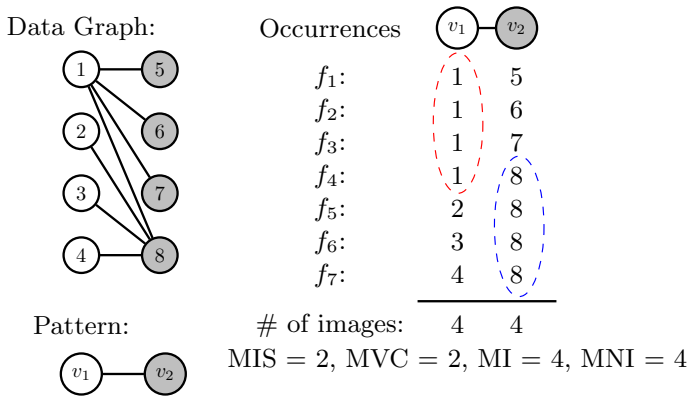


Fig. 9 An example with a 2-node pattern within a 8-node data graph shows that the MNI measure can over-estimate the count of patterns by ignoring partial overlap

coarse-grained node subsets $\{v_1\}$, $\{v_2\}$, $\{v_1, v_2\}$, their images counts are 4, 4, and 7. Hence any variant of MI will not help either.

It seems that for some data graphs (e.g., Fig. 9) the partial overlaps among pattern nodes matter, hence dividing node set in subsets and using their individual minimum image count is not plausible in this case. Thus we treat all pattern nodes as one set, that is, we do not break edges in occurrence (instance) hypergraph. Hence every node image in each occurrence (instance) can be chosen to represent this occurrence (instance). We seek a small number of node images that together represent all occurrences (instances). Intuitively, we want to find an ultimate version of minimum image count, which uses representative node image instead of minimum count of single node images and obtains counts closer to MIS.

Now we introduce a support measure that is even smaller than MI but requires more time to compute. The central idea is related to the well-known vertex cover problem.

Definition 25 A **vertex cover** of hypergraph $H = (V, E)$ is a subset of V that intersects with every edge of H. A **minimum vertex cover** is a vertex cover with the smallest cardinality.

Under the occurrence/instance hypergraph framework, we can transform the minimum vertex cover to a support measure that gives reasonable count of occurrences/instances.

Definition 26 Given pattern P in data graph G, and its occurrence (instance) hypergraph $H = (V, E)$. The **minimum vertex cover based (MVC) support** of P in G is defined as

$$\sigma_{MVC}(P, G) = \min_{C} |C|,$$

where C is a vertex cover of H.

In other words, MVC is defined as the cardinality of a smallest vertex cover set in the occurrence (instance) hypergraph of P in G. For example, in Fig. 9, edges in the occurrence hypergraph are $\{\{1, 5\}, \{1, 6\}, \{1, 7\}, \{1, 8\}, \{2, 8\}, \{3, 8\}, \{4, 8\}\}$, and the vertex set $\{1, 8\}$ is a minimum vertex cover, hence $\sigma_{MVC} = 2$.

The properties of MVC are discussed below.

Theorem 6 *The MVC support is anti-monotonic.*

Proof We shall show that for a pattern p and its superpattern P in graph G, we have $\sigma_{MVC}(p, G) \geq \sigma_{MVC}(P, G)$.

Let $\{f_1, f_2, \ldots, f_m\}$ and $\{f'_1, f'_2, \ldots, f'_{m'}\}$ denote the set of all occurrences of patterns p and P respectively. Let H_p and H_P be occurrence hypergraphs of p and P respectively. Assume that C is a minimum vertex cover of H_p, it intersects with every edge $f_i(V_p)$. Because any occurrence f' of pattern P in G must be an extension of an occurrence f_i of pattern p in G, we obtain that $f_i(V_p) \subseteq f'(V_P)$. If C intersects with $f_i(V_p)$, it must intersect with $f'(V_P)$. Hence a minimum vertex cover C of H_p contains a vertex cover of H_P. Therefore the cardinality of C is greater or equal to that of minimum vertex cover of H_P, that is, $\sigma_{MVC}(p, G) \geq \sigma_{MVC}(P, G)$. □

Let us refer to Fig. 7 for an illustrative example of the anti-monotonicity of σ_{MVC}: when the pattern $\{v_1, v_2, v_3\}$ is extended to include $\{v_4\}$, the MVC support is still 1. For example, vertex set $\{1\}$ is a minimum vertex cover, and it still intersects with each extended hypergraph edges hence it is a vertex cover of superpattern's occurrence hypergraph.

Theorem 7 *Given a pattern P and data graph G, we have*

$$\sigma_{MVC}(P, G) \leq \sigma_{MI}(P, G).$$

Proof Assume that pattern $P = (V_P, E_P)$ has m occurrences $\{f_1, f_2, \ldots, f_m\}$. Since $\sigma_{MI}(P, G) = \min_{T \in \mathcal{T}} c(T)$, there must be one coarse-grained node subset that achieves this minimum count σ_{MI}. We denote this node subset as T, and its images as $\{f_i(T), i = 1, 2, \ldots, m\}$. Because $f_i(T) \subseteq f_i(V_P)$, a minimum vertex cover C of $\{f_i(T) : i = 1, 2, \ldots, m\}$ is also a vertex cover of $\{f_i(V_P) : i = 1, 2, \ldots, m\}$. Therefore we get $\sigma_{MVC}(P, G) \leq |C|$. In the next step, we will show that $|C| \leq |\{f_i(T) : i = 1, 2, \ldots, m\}|$.

If T contains only one vertex, then $|C| = |\{f_i(T) : i = 1, 2, \ldots, m\}|$. If T contains two or more vertices, we can assume that $C = \{u_1, u_2, \ldots u_l\}$, where $u_i \in f_i(T)$, $i = 1, 2, \ldots, l$. Obviously we have $|C| = |\{f_i(T) : i = 1, 2, \ldots, l\}| \leq |\{f_i(T) : i = 1, 2, \ldots, m\}| = c(T) = \sigma_{MI}(P, G)$. Hence $\sigma_{MVC}(P, G) \leq \sigma_{MI}(P, G)$. □

Now we see that MVC is anti-monotonic, and is bounded by MI. In Sects. 4.2 and 4.3 , we shall further show that the MVC measure is actually close to the MIS. As to the computing efficiency, MVC is unfortunately NP-hard—this is easy to prove as it essentially involves solving the minimum vertex cover problem in the occurrence hypergraph. Luckily, in a k-uniform hypergraph, MVC is k-approximable, by choosing a maximal set of independent edges, and picking all vertices in them. Details can be found in Sect. 4.3.

In summary, MVC returns smaller counts but requires more time to compute as compared to MI. Its advantage is that it allows less overlaps between occurrences / instances and obtains the minimum count in the minimum image approach.

3.4.1 Relationship between MVC and MCP

We have shown that MIES in hypergraph framework is equivalent to overlap-graph-based MIS in terms of support value. MVC is actually similar to overlap-graph-based minimum clique partition support (MCP) (Calders et al. 2008). Given a pattern P in data graph G and its occurrence hypergraph H, the size of minimum clique partition of the overlap graph is at most that of minimum vertex cover of the occurrence hypergraph H. The reason is that for each vertex in a vertex cover of H, all edges containing this vertex in the H constitute a clique in the overlap graph, hence a vertex cover of occurrence hypergraph can be converted to a clique cover of overlap graph. On the other hand, in the overlap graph, there are cliques that correspond to multiple vertices in vertex cover of H, as clique in overlap graph corresponds to a intersecting family in occurrence hypergraph. Hence the count of MVC is at least that of MCP in overlap graph.

4 Further study of support measures in hypergraph framework

We shall utilize results in k-uniform hypergraphs to study the relationships between MIS, MIES, MVC and support measures derived from them.

Let us start with MVC by assuming that hypergraph $H = (V, E)$ consists of a set $V = \{v_1, v_2, \ldots, v_n\}$ of n vertices and a set $E = \{e_1, e_2, \ldots, e_m\}$ of m edges. We have a variable $x(v)$ for each vertex $v \in V$ indicating whether v is chosen in the vertex cover or not. The constraints state that in each hypergraph edge e at least one vertex should be chosen and the object is to minimize that number of such chosen vertices. Now we can write:

$$\begin{aligned} \min \quad & \sum_{v \in V} x(v) & \\ \text{subject to} \quad & \sum_{v \in e_i} x(v) \geq 1 & \forall i \\ & x(v) \in \{0, 1\} & \forall v. \end{aligned} \tag{1}$$

By definition the dual hypergraph H^* of H is a hypergraph whose vertices and edges are interchanged, so that the vertices are given by $\{e_i : i = 1, 2, \ldots, m\}$ and the edges are $X = \{X_1, X_2, \ldots, X_n\}$ where X_j is the collection of all edges in H which contain vertex v_j. Let variable $y(e)$ indicate whether e is in the independent set or not. The constraints state that in each edge X only one vertex be chosen and the object is to maximize that number of independent vertices. Therefore the dual of minimum vertex cover problem in H is maximum independent vertex set problem in H^*, which can be formulated as:

$$\begin{aligned} \max \quad & \sum_{e \in E} y(e) & \\ \text{subject to} \quad & \sum_{e \in X_i} y(e) \leq 1 & \forall i \\ & y(e) \in \{0, 1\} & \forall e. \end{aligned} \tag{2}$$

4.1 Polynomial time relaxations

The relaxation technique transforms an NP-hard optimization problem into a related problem that is solvable in polynomial time.

In addition, the solution obtained from relaxation gives information about the solution to the original problem. For example, a solution to a linear programming gives an upper (lower) bound on the optimal solution to the original maximization (minimization) problem.

Now we have presented the integer programming transformation of the problems. Based on that, we are ready to relax the integrability conditions of these two problems to obtain linear programming relaxations and formally define LP relaxations, the relaxed versions of the MVC and MIES measures. It is easy to verify that the optimal solutions exist, and we shall also show that they are both anti-monotonic.

Definition 27 Given a pattern P in a data graph G, and its occurrence (instance) hypergraph $H = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$, the **polynomial-time MVC (RMVC)** support measure of pattern P in graph G is defined as

$$\begin{aligned} \sigma_{RMVC}(P, G) = \min \quad & \sum_{v \in V} x(v) & \\ \text{subject to} \quad & \sum_{v \in e_i} x(v) \geq 1 & \forall i \\ & 0 \leq x(v) \leq 1 & \forall v. \end{aligned} \tag{3}$$

Likewise, we relax the integrability conditions of maximum independent edge set problem to obtain a linear programming formulation and another polynomial-time support.

Definition 28 Given a pattern P in a data graph G, and its occurrence (instance) hypergraph $H = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$, dual hypergraph $H^* = (E, X)$, $X = \{X_1, X_2, \ldots, X_n\}$, the **polynomial-time MIES (RMIES)** support measure of pattern P in graph G is defined as

$$\begin{aligned} \sigma_{RMIES}(P, G) = \max \quad & \sum_{e \in E} y(e) & \\ \text{subject to} \quad & \sum_{e \in X_i} y(e) \leq 1 & \forall i \end{aligned}$$

$$0 \leq y(e) \leq 1 \qquad \forall e. \tag{4}$$

Theorem 8 *The polynomial-time LP relaxation of MVC support measure is anti-monotonic.*

Proof We shall show that $\sigma_{RMVC}(p, G) \geq \sigma_{RMVC}(P, G)$ for any pattern p and its superpattern P in data graph G. Let us assume that the occurrence hypergraphs of p and P in G are $H_p = (V, E)$ and $H_P = (V', E')$ respectively.

Our approach is that: we use a solution $x^* = \sigma_{RMVC}(p, G)$ to the LP (3) to construct another function x^{**} such that $x^* \geq x^{**} \geq \sigma_{RMVC}(P, G)$, in this way we can prove that $\sigma_{RMVC}(p, G) \geq \sigma_{RMVC}(P, G)$.

Let $\sigma_{RMVC}(p, G) = \Sigma_{v \in V} x^*(v)$ be a solution to the LP (3) associated with H_p, where $\sum_{v \in e} x^*(v) \geq 1$ for any $e \in E$ and $0 \leq x^*(v) \leq 1$ for any $v \in V$. From that we construct a function $x^{**} = \Sigma_{v \in V'} x^{**}(v)$ on V' such that

$$x^{**}(v) = \begin{cases} x^*(v), & \text{if } v \in V' \cap V. \\ 0, & \text{otherwise } v \in V' - V. \end{cases}$$

Note that, for every $e' \in E'$ there is some $e \in E$ such that $e \subseteq e'$, hence we have

$$\begin{aligned} \Sigma_{v \in e'} x^{**}(v) &= \Sigma_{v \in e' - e} x^{**}(v) + \Sigma_{v \in e} x^{**}(v) \\ &= \Sigma_{v \in e' - e} x^{**}(v) + \Sigma_{v \in e} x^*(v) \\ &\geq 0 + \Sigma_{v \in e} x^*(v) \geq 1. \end{aligned}$$

Therefore, x^{**} satisfies constraints of LP (3) associated with H_P, that is, $\sum_{v \in e'} x^{**}(v) \geq 1$ for any $e' \in E'$ and $0 \leq x^{**}(v) \leq 1$ for any $v \in V'$. Consequently, we obtain that $x^{**} \geq \min \sum_{v \in V'} x(v) = \sigma_{RMVC}(P, G)$. On the other hand, we have

$$\begin{aligned} \Sigma_{v \in V'} x^{**}(v) &= \Sigma_{v \in V' - V}\ x^{**}(v) + \Sigma_{v \in V' \cap V}\ x^{**}(v) \\ &= 0 + \Sigma_{v \in V' \cap V} x^*(v) \\ &\leq \Sigma_{v \in V} x^*(v). \end{aligned}$$

Finally, we have $\Sigma_{v \in V} x^*(v) \geq \Sigma_{v \in V'} x^{**}(v) \geq \sigma_{RMVC}(P, G)$, which implies $\sigma_{RMVC}(p, G) \geq \sigma_{RMVC}(P, G)$. □

Theorem 9 *The polynomial-time LP relaxation of MIES support measure is anti-monotonic.*

The proof is similar to that of Theorem 8. We omit the details here.

To conclude, we have shown that standard linear programming relaxations can derive anti-monotonic support measures in hypergraph framework. In the state-of-the-art overlap graph framework, Calders et al. (2008) proposed the Lovász θ function (see e.g., Lovász 1979) to derive a new support measure, which is computable in time polynomial in the order of the overlap graph vertices (occurrences) using semidefinite programming (SDP). We will discuss the relationship of support measures derived from LP and SDP techniques in Sect. 4.3.

4.2 Bounding theorems

To explore the relationship among all the support measures within the new framework, we investigate uniform hypergraph maximum independent edge set problem and minimum vertex cover problem so as to obtain the following theorems. We first study the difference between the MIES and MVC measures.

Theorem 10 *Given a pattern P, data graph G, and occurrence (instance) hypergraph $H = (V, E)$, we have*

$$\sigma_{MIES}(P, G) \leq \sigma_{MVC}(P, G).$$

Proof Assume that I is a maximum independent edge set and C a minimum vertex cover in H. For every edge $e \in I$ there is a corresponding vertex $v \in C$ such that $v \in e$. Furthermore, for any $e, e' \in I$, we have $e \cap e' = \emptyset$, hence their corresponding vertices in C are different. Therefore, we get $|I| \leq |C|$ and then we have $\sigma_{MIES}(P, G) \leq \sigma_{MVC}(P, G)$. □

The above theorem shows that MVC is larger than MIES (that equals MIS according to Theorem 1).

Based on well-established results in linear programming (see e.g., Pach and Agarwal 2011), we obtain the following relationship between σ_{MIS}, σ_{MVC}, and support measures created from relaxations of the corresponding linear programming problems.

Theorem 11 *Given a pattern P, data graph G, and occurrence (instance) hypergraph H, we have*

$$\sigma_{MIES}(P, G) \leq \sigma_{RMIES}(P, G) = \sigma_{RMVC}(P, G) \leq \sigma_{MVC}(P, G).$$

Proof If we are given a linear program $\min\{c^T x : x \in \mathbb{R}^n, Ax \geq b, x \geq 0\}$, called the primal, its dual $\max\{y^T b : y \in \mathbb{R}^m, A^T y \leq c, y \geq 0\}$, where matrix $A \in \mathbb{R}^{m \times n}$, vector $b \in \mathbb{R}^m, c \in \mathbb{R}^n$. The theorem of weak duality tells us that if x^* and y^* are primal and dual feasible solutions respectively, then $c^T x^* \geq b^T y^*$. The strong duality theorem tell us that if there exist feasible primal and dual solutions, then they have the same objective value (see e.g., Pach and Agarwal 2011). Since we formulate MVC and MIES measures as solutions of primal and dual linear programs, the first and last inequality are given by the definitions of corresponding linear programming problems. The equality follows from the duality theorems of linear programming problems (see e.g., Pach and Agarwal 2011). □

A key idea in Theorem 11 is that we can switch between considering the relaxations of the maximum independent edge set and minimum vertex cover problems of a hypergraph. In dual linear programming relaxation problems, a feasible primal solution has a value greater than or equal to that of any feasible dual solution. Furthermore strong duality theorem says that if the primal program has an optimal solution, so does the dual and they have the same objective value.

In practice, if each hypergraph vertex is contained in relatively few edges we have a stronger bound between the original and relaxed versions of MVC. Explorations along this direction constitute a very interesting topic for future research.

The comparison between σ_{MVC}, σ_{MI} and σ_{MNI} were examined in Theorems 5 and 7 . Putting all together, we have

$$\sigma_{MIS} = \sigma_{MIES} \leq \sigma_{RMIES} = \sigma_{RMVC} \leq \sigma_{MVC} \leq \sigma_{MI} \leq \sigma_{MNI} \tag{5}$$

The above formula shows a series of measures that can be built in the same framework and occupy different locations of the frequency spectrum.

Nevertheless, the results in Theorem 11 show that, by relaxing the original problem, we further reduce the gap between MVC and MIES/MIS. Of course, we must emphasize that the results shown here are obtained in the relaxed problem settings. In the next section we will explore the close relationship between MVC and MIES using vertex cover and independent edge set theorems in uniform hypergraphs.

4.3 Approximation and hardness theorems

So far we understand the MIS/MIES measures have the highest level of intuitiveness as it counts independent occurrences (instances) and returns the smallest count among all measures we have discussed. Due to their being NP-hard, it is meaningful to study approximate algorithms. In this section, we present our new discovery of hardness theorems of MIS/MIES support measures and approximation theorems of relaxation measures and relationship of MIES, MVC and the SDP, LP relaxations in the hypergraph framework. Here is a summary of our findings (assume we study the support measures of a pattern with k nodes).

(1) First of all, we show that MIES is NP-hard with constant (in terms of k) approximation algorithms (Theorem 13);
(2) The second finding is that overlap graphs fall into a category of so called $(k+1)$-claw-free graph (Theorem 14), together with the fact that MIS is equivalent to MIES, the MIS support can be approximated within a constant factor (Theorem 15);
(3) The ratio between MVC and MIES is within constant k (Theorem 16);
(4) Not only standard linear programming (LP), but also semidefinite programming (SDP) relaxation of MIES can help explore new polynomial time measures. We compare them in the hypergraph framework and show that: (i) the integral gap of LP is $k - 1 + \frac{1}{k}$ for k-uniform hypergraphs (Theorem 17); and (ii) the SDP relaxation is strictly stronger than the LP relaxation, its integral gap is at most $\frac{k+1}{2}$ (Theorem 18);
(5) We also analyze current overlap concepts and present a new linear-time support measure, point out different means of utilizing overlap concepts for improving and designing meaningful support measures (Sect. 4.4).

4.3.1 Approximation of MIES/MIS and MVC

We start our discussions by the existence of approximate algorithms for MVC.

Theorem 12 *Given a pattern P, data graph G, the MVC support measure can be approximated within constant factor.*

Proof The MVC support measure treats each occurrence as a k-node set, hence we can utilize results from MVC approximation theorems in k-uniform hypergraphs. We give a natural greedy algorithm. We first construct a maximal matching by greedily adding edges.Then we let a vertex cover contain all vertices in each edge in this matching. This vertex cover is a set of vertices that covers all the edges and its size is at most k times of the size of the minimum vertex cover. □

Simple algorithms exist that provide k-approximations for MVC, however, despite considerable efforts, state-of-the-art techniques in Holmerin (2002) show that in k-uniform hypergraphs ($k \geq 3$) it is NP-hard to approximate MVC within factor $k^{1-\epsilon}$ for any $\epsilon > 0$.

In the k-uniform hypergraph setting, we can also develop constant factor approximation bound for MIES support measure.

Theorem 13 *Given a pattern P, data graph G, the MIES support measure can be approximated within constant factor.*

Proof We have shown that MIES support treats each occurrence as a vertex set of size k, and it is related to maximum independent edge set (also called maximum matching) problem in k-uniform hypergraphs. The greedy algorithm of picking a maximal set of disjoint hypergraph edges and including all the vertices in those hypergraph edges gives a factor k approximation. Therefore, we conclude that constant factor approximations exist for MIES support measure. □

Recent years, one of the best known approximation ratios for the maximum matching problem in k-uniform hypergraph (or k-set packing) problem is $\frac{k}{2} + \epsilon$ for any fixed $\epsilon > 0$, given by Hurkens and Schrijver (1989). It is further improved to $\frac{k+1}{3} + \epsilon$ for any fixed $\epsilon > 0$ by Cygan (2013).

Previous work (e.g. Wang and Ramon 2012; Kuramochi and Karypis 2005) believed that MIS cannot be approximated even within a factor of $n^{1-o(1)}$ in polynomial time unless $P = NP$, where n is the number of vertices in the overlap graph. Because MIS is equivalent to MIES, which is constant-approximable, it would be interesting to study what causes this contradiction.

A key observation is that the occurrence hypergraph belongs to a special category of hypergraphs instead of general hypergraph. The same is also true for overlap graphs. In an occurrence hypergraph, each edge represents an occurrence and has a fixed size k. For an edge e in a k-uniform hypergraph, there can be at most k mutually-disjoint edges intersect with e, that is when each overlaps with e at a distinct vertex. For example, in Fig. 10, a three-node triangle-shaped pattern has 4 occurrences e_1, e_2, e_3, e_4. From the occurrence hypergraph in Fig. 10, we can tell that it is impossible to have a fifth

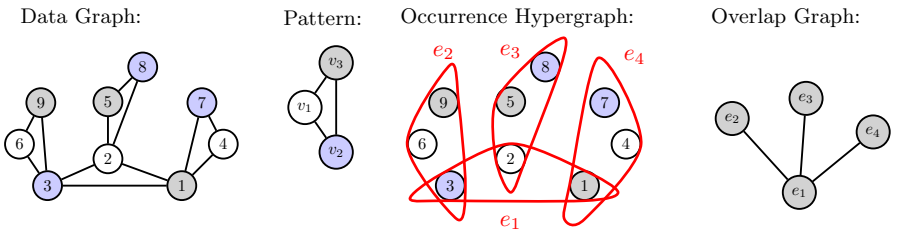


Fig. 10 An example with a 3-node pattern within a 9-node data graph shows that the k-uniform MIES problem can be mapped to MIS on a (k + 1)-claw-free graph. In this case, we have k = 3

occurrence intersecting e_1 and being independent from e_2, e_3, e_4. As a result, overlap graphs are not general graphs either. In overlap graph, an edge is a vertex and all other edges intersecting it become its neighbors (adjacent vertices). Hence although a vertex in overlap graph can have any number of neighbors, the maximum number of mutually independent (non-adjacent) neighbors is k. It turns out that overlap graph is in the category of the so-called "claw-free" graph.

A claw is a complete bipartite graph. A claw-free graph is a graph that does not have a claw as an induced subgraph. In other words, a graph is claw-free if no vertex has three pairwise non-adjacent neighbors. A k-claw graph consists of a center node that is adjacent to k mutually independent vertices, and it is also known as the complete bipartite graph, denoted as $K_{1,k}$. It is a generalization of claw—in this sense, a claw is equivalent to a 3-claw.

In what follows, we give a generalization of claw-free graph.

Definition 29 A graph is k-claw-free if and only if it does not contain the complete bipartite graph $K_{1,k+1}$ as an induced subgraph, where $K_{n,m}$ is the complete bipartite graph on n and m vertices.

Theorem 14 *Let P be a pattern with k nodes, and G a data graph. The overlap graph of occurrences or instances of P in G is in the category of $(k+1)$-claw-free graph.*

Proof From the above discussion we know that for a pattern of k nodes, an occurrence (denoted as vertex) in its overlap graph can intersect with at most k mutually-independent occurrences (vertices), which means its overlap graph does not have $K_{1,k+1}$ as an induced subgraph. That is to say, the overlap graph of a pattern of k nodes is not a general graph, it is in the category of so-called $(k+1)$-claw-free graph. □

Figure 10 shows the overlap graph of a pattern of three nodes is $(k+1)$-claw-free, where k is 3.

Now we can conclude that an overlap graph is a $(k+1)$-claw-free graph. On the other hand, $(k+1)$-claw-free graphs is more general than the overlap graph of k-node pattern. We have an example in what follows.

In graph theory, a cycle graph be a graph that consists of a single cycle (in other words, all its vertices are connected in a closed chain). We use C_n to denote the cycle graph with n vertices. A wheel graph W_{n+1} is formed by joining a single vertex to all

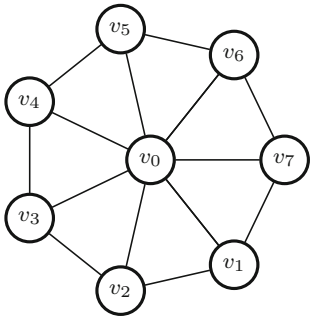


Fig. 11 The wheel graph W_8 here is 4-claw-free but it does not correspond to the overlap graph of any 3-node pattern in a data graph

vertices of a cycle of length n (note that some authors use W_n to denote this type of wheel graph). We shall consider the wheel graph W_{2k+2}, which is obtained by adding a new vertex and joining it to all vertices of the an odd cycle C_{2k+1} of length $2k+1$. As shown in Fig. 11, a cycle is formed by chaining vertices $v_1, v_2, v_3, v_4, v_5, v_6, v_7$. After adding v_0 to it, and connect it with all other vertices, we obtain a wheel graph W_8.

A W_{2k+2} wheel graph is $(k+1)$-claw-free because the size of any independent set is at most k which makes it impossible to find a center vertex joining $k+1$ mutually independent vertices.

A W_{2k+2} wheel graph however cannot be an overlap graph of any pattern in a data graph. The largest complete subgraph in the overlap graph is a triangle, which includes two adjacent vertices on the cycle and the center vertex. Disjoint triangles should not include the same vertices, otherwise extra edges should be added into the overlap graph. Therefore such W_{2k+2} wheel graph cannot be overlap graph of any k-node pattern in a data graph. Hence, overlap graph is a subclass of $(k+1)$-claw-free graph, and we should study overlap graph within the area of $(k+1)$-claw-free graph, not general graph.

After showing that the overlap graph as a special class of graph, one can see why MIS measure should not be as hard as a general maximum independent set problems.

Theorem 15 *Given a pattern P, data graph G, the MIS support measure can be approximated within constant factor.*

Proof Because MIS and MIES are found to be equivalent by Theorem 1, and MIES can be approximated within constant factor k (if the pattern P has k nodes) by Theorem 13, we can conclude that MIS also has constant approximation ratio. □

In the k-uniform hypergraphs, we are able to find not only constant approximation ratios but also bounds between different support measures.

Theorem 16 *Given a pattern P containing k nodes, data graph G, we have*

$$\sigma_{MIES}(P, G) \leq \sigma_{MVC}(P, G) \leq k \cdot \sigma_{MIES}(P, G).$$

Proof The first part of the formula can be easily derived from well-established results of Duality Theorem.

Consider a maximal independent edge set I of H. Let X be the set of vertices contained in the edges of H and τ be the cardinality of maximum independent edge set. Because every edge has size k, the size of set X is at most $k \cdot \tau$. (Otherwise, those edges are not independent).

Because X is the set of all vertices in this hypergraph, X intersects with every edge which means X is a vertex cover. Therefore, the cardinality of minimum vertex cover is less than that of X, beside we know that σ_{MVC} is the cardinality of minimum vertex cover, τ is assumed to be the maximum independent set size which is equal to σ_{MIES}, the cardinality of X is at most $k \cdot \tau$, hence $\sigma_{MVC} \leq k \cdot \sigma_{MIES}$. □

The above theorem shows that, although MVC is larger than MIES, the gap between MVC and MIES/MIS is within a constant factor k. This is also an interesting and encouraging finding that basically shows the MVC defined in the hypergraph framework is actually close to MIS.

4.3.2 Maximum ratios of relaxations of MIES

Because MIES and MVC are NP-hard to compute, the relaxation techniques are of practical importance. In the past years, based on overlap graph, the standard LP and SDP techniques were used to derive polynomial-time support measures. In relaxation approaches, a key concept is the integrality gap which is the maximum ratio between the solution quality of the integer program and of its relaxation. The integrality gap enforces a limit on the approximation power of the relaxation, and different linear programming formulations for the same problem may have different integrality gaps.

In the uniform hypergraph framework, we are able to find many exciting results and insights. In the following, we present new relations among all support measures in uniform hypergraph settings.

Theorem 17 *For any k-node pattern, the maximum ratio between support measure derived from LP and MIES is* $k - 1 + \frac{1}{k}$.

Proof Because MIES support treats the occurrence hypergraphs as k-uniform hypergraphs, the proof follows from the result that the integrality gap of LP is $k - 1 + \frac{1}{k}$ for k-uniform hypergraphs (Füredi et al. 1993). An algorithmic proof in Chan and Lau (2010) shows that this bound is tight. □

In overlap graph framework, Calders et al. (2008) proposed the Lovász θ function of overlap graph to derive a polynomial-time relaxation support measure. If one can show that support measures derived from such SDP are anti-monotonic, then theorems in the k-uniform hypergraph field tell the relationships of support measures constructed from LP and SDP. We obtain maximal ratios and interesting interpretations of relations of such measures from uniform hypergraph theory. The study in Chan and Lau (2010) viewed the maximum independent edge set problem as the independent set problem in a $(k + 1)$-claw-free graph and made the connection between the linear and semidefinite programs in k-uniform hypergraphs which bounded the integrality gap of a semidefinite programming relaxation (the Lovász θ function) for the k-uniform hypergraph matching problem.

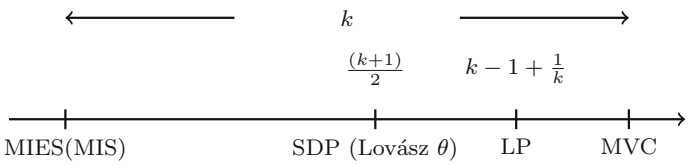


Fig. 12 Ratios between MIS/MIES and its relaxations and MVC

The following theorem shows that the SDP, LP relaxations and maximum independent edge set problem are connected via concepts of clique in overlap graph and intersecting family in k-uniform hypergraphs.

Theorem 18 *For any k-node pattern, the maximum ratio between support measures derived from the Lovász θ function as SDP relaxation and MIES is* $\frac{k+1}{2}$.

Proof For any pattern with k nodes, its occurrence hypergraph is a k-uniform hypergraph. It is proven in Chan and Lau (2010) that the Lovász θ function $\theta(G)$ is equivalent to θ-LP, which is a stronger relaxation than the clique LP which is equivalent to intersecting-family LP. Furthermore, the integrality gap of the intersecting-familty LP is at most $\frac{k+1}{2}$. Hence the proof follows from Theorem 1.5 in Chan and Lau (2010) stating that there is a polynomial size semidefinite program for hypergraph matching (maximum independent edge set) problem, with integrality gap at most $\frac{k+1}{2}$ for k-uniform hypergraphs. □

In conclusion, we believe that SDP and LP are viable techniques for getting polynomial-time support measures in the hypergraph framework. In addition, we discover interesting bounds and interpretation in this setting. Figure 12 shows our main findings: SDP is strictly stronger than LP as a relaxation method, and both of them can lead to support measures that have nice integrality gaps between MIES and MVC.

4.4 Overlap concepts in hypergraph framework

We believe by adopting the hypergraph settings, we can utilize resourceful classic hypergraph theorems to further investigate and gain more thoughtful insights for connections among support measures, or even define more support measures.

A variant of vertex overlap, called harmful overlap, was introduced in Fiedler and Borgelt (2007). We present a new concept of **structural overlap** that can be compared with harmful overlap in studying MIS-flavored support measures. Additionally, we show how structural overlap can be used in the study of support measures.

In this section, we call the concept of vertex overlap in Definition 12 **simple overlap** to distinguish it from the two new overlap concepts.

Definition 30 (Fiedler and Borgelt 2007) A **harmful overlap (HO)** of occurrences f_1 and f_2 of pattern P exists, if $\exists\ v \in V_P$, such that $f_1(v), f_2(v) \in f_1(V_P) \cap f_2(V_P)$.

Definition 31 A **structural overlap (SO)** of occurrences f_1 and f_2 of pattern P exists if $\exists\ v, w \in V_P$, satisfying that v and w are contained in an orbit in a subgraph of pattern P, and $f_1(v) = f_2(w) \in f_1(V_P) \cap f_2(V_P)$.

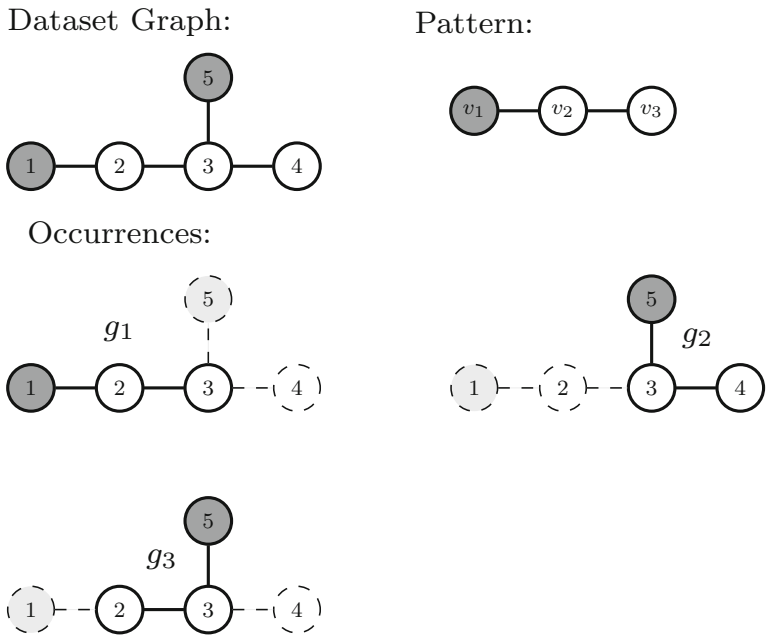


Fig. 13 An example showing the structural overlap is not the same as harmful overlap

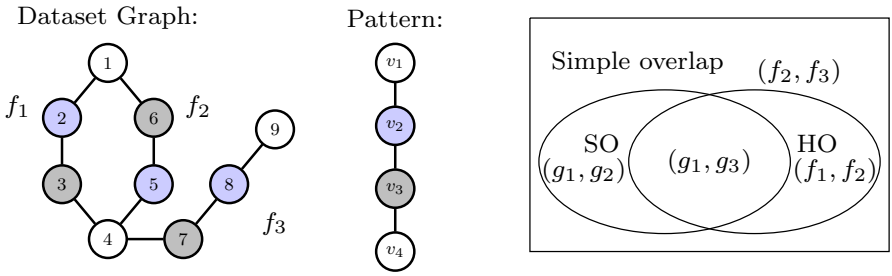


Fig. 14 With the example of a 4-node pattern within a 9-node data graph, we draw Venn diagram to illustrate the relationship among structural overlap, harmful overlap, and simple overlap

The concept of structural overlap is originated from MI support measure which considers overlap on orbits. Take Fig. 13 as an example, when calculating MI, because node v_2 and v_3 are in an orbit, and the orbit has two images {2, 3} and {3, 4}, we get MI = 2. However, these two images have vertex 3 in common. Now we have occurrences g_1 and g_2 overlap in structural overlap sense.

In addition, we use Figs.13 and 14 to show that structural overlap and harmful overlap are different concepts. For example, although structural overlap of g_1 and g_2 exists, harmful overlap of them does not exist. The reason is that $g_1(V_P) \cap g_2(V_P) = \{3\}$, but 3 is an image of two different nodes v_2, v_3 where $g_1(v_3) = 3$ and $g_2(v_2) = 3$. On the other hand, a harmful overlap of f_1 and f_2 exists but no structural overlap of them exists. We state that both harmful overlap and structural overlap implies simple

overlap, and there are cases when simple overlap exists but neither harmful overlap nor structural overlap exists (e.g., f_2 and f_3 in Fig. 14). Harmful overlap and structural overlap can exist at the same time (e.g., g_1 and g_3 in Fig. 13).

Fiedler and Borgelt (2007) explain that some type of overlap of two occurrences should not be considered harmful. According to the definition of harmful overlap, for a pattern $P = (V_P, E_P)$, if a simple overlap of its two occurrences f_1 and f_2 exist and there is at least one node's images are in both of node images $f_1(V_P)$ and $f_2(V_P)$, a harmful overlap of f_1 and f_2 exists. A similar argument applies to our structural overlap concept. When a simple overlap exists, in addition if two nodes are in an orbit in a subgraph of P and their images are in both images $f_1(V_P)$ and $f_2(V_P)$, a structural overlap of f_1 and f_2 exists. That is to say structural overlap addresses more on topological structure of the pattern which is at the core of graph isomorphism problem.

The common ground of harmful overlap and structural overlap is that both are weaker concepts compared to simple overlap. Hence, like harmful overlap, the concept of structural overlap can also be used in various ways to explore frontiers of support measure theory.

One possible direction is that, instead of simple overlap, one can use structural overlap to decide whether two occurrences (instances) overlap, and then proceed to construct overlap graph. The resulted overlap graph is sparser (i.e., with fewer edges) than the one generated from simple overlap. Consequently, one can use MIS, MCP, and other overlap graph based measures to obtain count of pattern occurrences (instances).

The other direction is motivated by the close connection of overlap concepts to MI support measure, various overlap concepts can be potentially used to explore variants of MI support measures. We analyze harmful overlap to explore new support measures. By definition, harmful overlap requires the existence a node $v \in V_P$, such that $f_1(v)$, $f_2(v) \in f_1(V_P) \cap f_2(V_P)$. Hence it is stricter than simple overlap because the latter only requires $f_1(V_P) \cap f_2(V_P) \neq \emptyset$. MI is derived from symmetric character of pattern nodes; however harmful overlap does not utilize topological structure of graph. One way of detecting harmful overlap is to check the images of nodes with the same labels. It is also a generalization of MI support. Hence we can define a variant of MI as follows:

Definition 32 Given a pattern $P = (V_P, E_P)$, a data graph $G = (V, E)$, let T be a subset of V_P such that all nodes in T have the same label, the collection of all such T is denoted as $\mathcal{T} = \{T\}$. The **harmful overlap minimum instance based support (HO-MI)** of P in G is defined as

$$\sigma_{HO-MI}(P, G) = \min_{T \in \mathcal{T}} c(T).$$

Let us revisit the example in Fig. 14: the pattern has same-labeled node subsets $\{v_1\}$, $\{v_2\}$, $\{v_3\}$, $\{v_4\}$ and $\{v_1, v_4\}$, hence $\sigma_{HO-MI}(P, G) = 1$.

The intuition behind the design of this new support is that occurrences are likely to overlap at the images of same-labeled nodes. Even if we group nodes with different labels together as a node subset, the image count of this node subset is not smaller than that of any subsets.

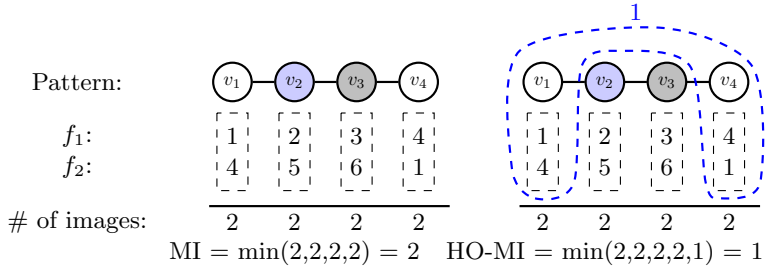


Fig. 15 HO-MI is a new variant of MI derived from harmful overlap. Here we show an example of the calculation of MI and HO-MI by using the same pattern and data graph mentioned in Fig. 14

Theorem 19 *Given a pattern P and a data graph G, assume that P has m occurrences in G, for a pattern node set $T \subseteq V_P$ containing nodes with the same label, if one add any node v_0 with a different label, its image count $c(T)$ under the mappings $\{f_1, f_2, \ldots, f_m\}$ is lesser than or equal to its image count $c(T \cup v_0)$ under the mappings $\{f_1, f_2, \ldots, f_m\}$.*

Proof Assume that $c(T) = l$, and $f_1(T), \ldots, f_l(T)$ are l mutually distinct images. If we can show that $f_1(T \cup v_0), \ldots, f_l(T \cup v_0)$ are also mutually distinct images, then we obtain that $c(T)$ is lesser or equal to the image count $c(T \cup v_0)$.

For any two images among $f_1(T \cup v_0), \ldots, f_l(T \cup v_0)$, say $f_1(T \cup v_0)$ and $f_2(T \cup v_0)$, if $f_1(T \cup v_0)$ is identical to $f_2(T \cup v_0)$ then $f_1(v_0) \in f_2(T)$ and $f_2(v_0) \in f_1(T)$. This is a contradiction to the fact that the label of v_0 is different from that of nodes in T. □

For example in Fig. 15, nodes v_1 and v_2 have different labels, their image sets do not intersect. From Theorem 19 we know that when designing any variant of MI, we can only use count of images of nodes with the same label, therefore it makes the calculation of support measure efficient. Theorem 19 tells us that investigation of pattern structures and features are necessary for developing efficient support measures, randomly group pattern nodes together and count their images under occurrences is not the optimal solution.

This new support measure has similar main properties as the MI support.

Theorem 20 *The HO-MI support measure is anti-monotonic.*

Proof Given pattern $p = (V_p, E_p)$ and its superpattern $P = (V_P, E_P)$ in data graph G, we assume that p has m occurrences $\{f_1, f_2, \ldots, f_m\}$ in G and P has l occurrences $\{f'_1, f'_2, \ldots, f'_l\}$ in G.

Assume that $\sigma_{MI}(p, G) = \min_{T \in \mathcal{T}} c(T)$ and $\sigma_{MI}(P, G) = \min_{T \in \mathcal{T}'} c'(T)$. Because $V_p \subseteq V_P$, we have $\mathcal{T} \subseteq \mathcal{T}'$. Therefore $\min_{T \in \mathcal{T}} c'(T) \geq \min_{T \in \mathcal{T}'} c'(T)$.

Since any f'_i is an extension of some f_i which implies $f'_i(T) = f_i(T), \forall\, T \in \mathcal{T}$. Therefore $\min_{T \in \mathcal{T}} c(T) \geq \min_{T \in \mathcal{T}} c'(T) \geq \min_{T \in \mathcal{T}'} c'(T)$.

Hence we have $\sigma_{HO-MI}(p, G) \geq \sigma_{HO-MI}(P, G)$. □

Theorem 21 *The HO-MI support measure is linear-time computable.*

Proof By definition $\sigma_{HO-MI}(P, G) = \min_{T \in \mathcal{T}} c(T)$, T is subset of V_P, hence the number of T for pattern P is a function of the number of pattern nodes. It implies that calculating $c(T)$ costs $O(n)$ time where n is the number of occurrences. Therefore, σ_{MI} is linear-time computable. □

Theorem 22 *Given a pattern P and data graph G, we have*

$$\sigma_{HO-MI}(P, G) \leq \sigma_{MNI}(P, G).$$

Proof The MNI support can be written as $\sigma_{MNI}(P, G) = \min_{W \in \mathcal{W}} c(W)$, where $\mathcal{W} = \{\{v\} : v \in V_P\}$. Because $\mathcal{W} \subseteq \mathcal{T}$, we have $\sigma_{HO-MI}(P, G) = \min_{T \in \mathcal{T}} c(T) \leq \min_{W \in \mathcal{W}} c(W) = \sigma_{MNI}(P, G)$. □

Because HO-MI is a variant of MI, it is not surprising that HO-MI is an upper bound of MVC.

Theorem 23 *Given a pattern P and data graph G, we have*

$$\sigma_{MVC}(P, G) \leq \sigma_{HO-MI}(P, G).$$

Proof Assume that pattern $P = (V_P, E_P)$ has m occurrences $\{f_1, f_2, \ldots, f_m\}$. Since $\sigma_{HO-MI}(P, G) = \min_{T \in \mathcal{T}} c(T)$, there must be one same-label node subset that achieves this minimum count σ_{HO-MI}. It follows from Theorem 7 that for any node subset as T, and its images as $\{f_i(T), i = 1, 2, \ldots, m\}$, and a minimum vertex cover C of $\{f_i(T) : i = 1, 2, \ldots, m\}$, we have $|C| \leq |\{f_i(T) : i = 1, 2, \ldots, m\}|$.

Hence $\sigma_{MVC}(P, G) \leq \sigma_{HO-MI}(P, G)$. □

From the fact that nodes in the same orbit are of the same label, we conclude that HO-MI counts images of a boarder range of node subsets hence its value should be smaller than that of MI.

Theorem 24 *Given a pattern P and data graph G, we have*

$$\sigma_{HO-MI}(P, G) \leq \sigma_{MI}(P, G).$$

Proof Because nodes in one orbit have the same label, it follows from their definition that $\sigma_{HO-MI}(P, G) \leq \sigma_{MI}(P, G)$. □

In conclusion, structural overlap differs from harmful overlap by considering overlap at images of symmetric pattern nodes. To sum up, we believe that the concepts of overlap should reflect how occurrences share the same vertices in different ways, and we shall explore potential applications of them in designing and improving support measures in the future.

5 Experiments

As most of the findings in this paper are supported by rigorous proof, our experimental evaluations focus on the actual performance of MI, MVC, and polynomial-time MVC

Table 3 Graph datasets used in our experiments

Dataset	Total Edges	Total Vertices	Description
Crime	1380	1476	Interaction
Euroad	1174	1417	Infrastructure
FAA	2615	1226	Routes Database
Figeys	6452	2239	Protein network
PowerGrid	4941	6594	Infrastructure
Protein	1870	2277	Protein network
Stelzl	6207	1706	Protein network
Vidal	6726	3133	Protein network

in comparison to existing support measures MIS and MNI. Specifically, we study the actual measure counts returned and the computational time, in attempt to get insights on the gaps between neighboring support measures shown in Eq. (5). For convenience, we denote polynomial-time MVC as RMVC. Recall that MIS gives the most reasonable measure but its computation is a NP-hard problem. MNI could return a value that is significantly higher than MIS but it can be solved in linear time.

We invested much efforts into the implementation of a comprehensive framework for studying different support measures in the hypergraph setup. As input, the framework takes a pattern, a data graph, and generates a list of its occurrences in the data graph in the form of DFScode, which represents the DFS lexicographic order of the pattern (Yan and Han 2002). We obtain occurrences by using the DistGraph approach introduced in Talukder and Zaki (2016). We implemented the following support measures: MNI, MI, MVC, RMVC, and MIS. In particular, we implement RMVC and MVC by using the Cplex package, which is a commercial software for optimization problems (IBM 2011). We use Nauty library (McKay and Piperno 2014) for finding isomorphism in the MI implementation. All of our code can be downloaded via a Github folder (Pitaksirianan 2019). We run all of our experiments in a workstation running Linux (Ubuntu 18.04 LTS) with an Intel i9-7920X 12-core CPU and 94GB of DDR4 2666-MHz memory.

Data Graphs We use eight different datasets for our experiments (Table 3). All datasets are collected from real-world applications and acquired from the well-known KONECT (Kunegis 2018) website. Note that the graph sizes are on the smaller side by today's standards. While our new support measures can process much bigger datasets, the long-running time of the NP-hard MIS means a comparative study is impossible under large datasets.

Patterns To generate patterns, we set a frequency cutoff according to MNI (being the largest support measure) for the data graph. We report five measures (i.e., MI, RMVC, MVC, MNI, and MIS) for EVERY pattern in the dataset with an MNI value higher than that cutoff. The cutoff is set to a relatively small number to ensure we study a large number of patterns therefore the workload is not biased towards particular patterns.

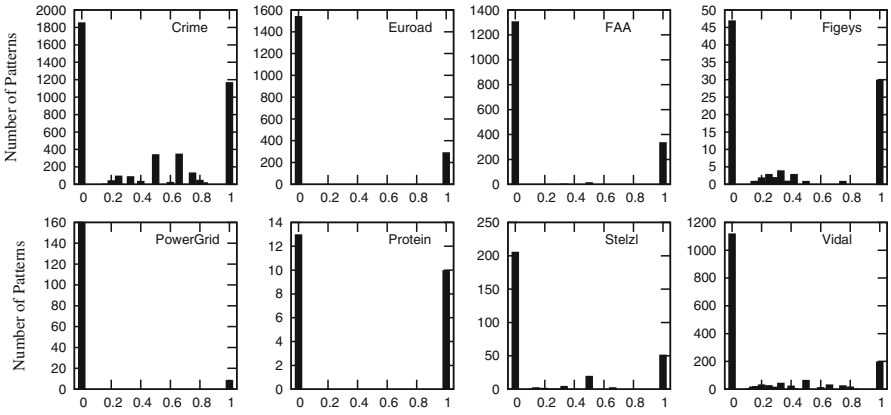


Fig. 16 Distribution of the relative values of RMVC to MIS (lower bound) and MVC (upper bound) for all patterns visited in different datasets. Here the value 0 on the x axis means RMVC returns the same value as MIS, and 1 means RMVC is the same as MVC

5.1 Experimental results

For comparisons, we present counts returned in relation to those returned by other measures as follows:

$$R = \frac{\sigma_X - \sigma_A}{\sigma_B - \sigma_A} \tag{6}$$

where X is the value of a support measure to be studied, A is a measure whose counts serve as the lower bound, and B the higher bound. Thus, an R value close to 1 (0) means that measure returned by X is close to that of B (A). We report the results for all frequent patterns we encountered in each dataset. Following Eq. (5), we study the new support measures RMVC, MVC, and MI, each with its two immediate neighbors in Eq. (5) as lower and upper bounds.

In theory, we have MIS $\leq$ RMVC $\leq$ MVC, our results support these bounding theorems. We first compare RMVC with MIS and MVC. We show the distribution of the R values among all considered patterns in relation to MVC (upper bound) and MIS (lower bound) in Fig. 16. In particular, the distribution is plotted as a histogram with 0.05 as the bucket width. The exciting news is that, for most of the patterns, RMVC returns the same value as MIS (i.e., $R = 0$). There are very few patterns for which RMVC is the same as MVC (except for the 'Protein' and 'Figeys' datasets), and even fewer patterns go between MIS and MVC.

According to Fig. 17, MVC returns a significantly smaller value than MI in most of the cases. In all eight datasets, there is big percentage of patterns that show the same value for MVC and RMVC. On the other hand, there are cases where we found MVC to be the same as MI, but they are less popular—only in three datasets FAA (20%), PowerGrid (30%), and Vidal (2%). For all datasets, there are a big percentage of cases in which the value of MVC goes between RMVC and MI. The experimental results support the fact that RMVC $\leq$ MVC $\leq$ MI.

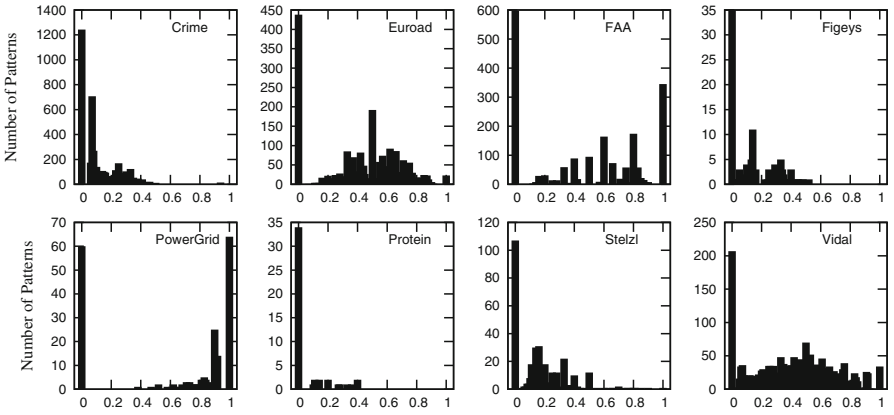


Fig. 17 Distribution of the relative values of MVC to RMVC (lower bound) and MI (upper bound) for all patterns visited in different datasets

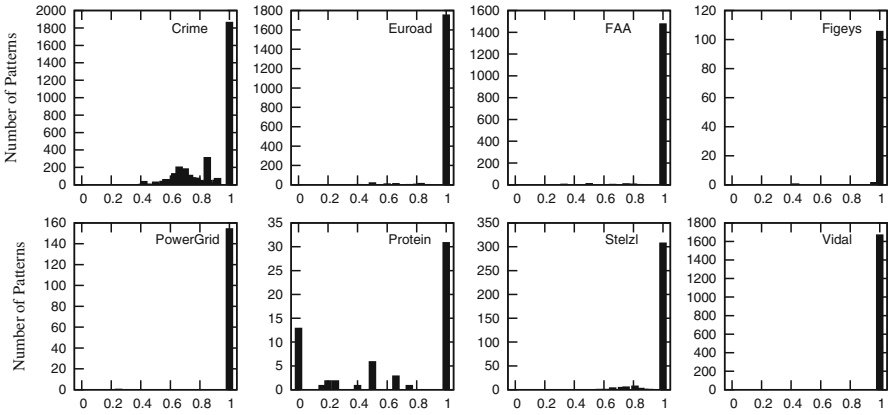


Fig. 18 Distribution of the relative values of MI to MVC (lower bound) and MNI (upper bound) for all patterns visited in different datasets

By looking at Fig. 18, however, we found that, in most of the cases, MI is equal to MNI (i.e., $R = 1$). The only exception seems to be the *Crime* and *Protein* datasets, there are 40% of the patterns in *Protein* and 57% of the patterns in *Crime* with a MI value smaller than MNI. Further investigation of the patterns involved show the reason behind such a phenomenon: MI takes advantage of a pattern's topological structure to reduce overestimation. However, for most of the datasets, the patterns do not show a symmetric form therefore the returned counts are the same as MNI.

We must emphasize that our bounding theorems are supported by the experiments—we never encountered cases that violate such theorems. As a special note, in the above three figures, we did not report that cases where $\sigma_X = \sigma_A = \sigma_B$ (i.e., quantitiy R is undefined). However, such cases are rare (less than 5% in all plots) therefore their absence does not change the big picture.

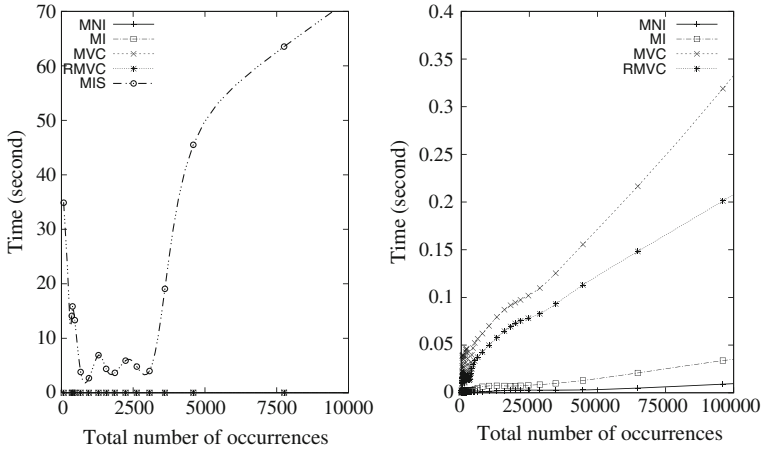


Fig. 19 Time to compute different support measures

Computational efficiency We report the computational time of generating the support measures in Fig. 19. We report time for all patterns with up to 100,000 occurrences each. We do not record the time for generating the pattern occurrences, as that is the same for all measures. This is also the convention followed by all published work in this topic. Note that we report MIS running time (left subfigure of Fig. 19) for only up to 10,000 occurrences because the time is nonpolynomial. As a result, it is many orders of magnitude slower than that for other support measures. According to Fig. 19 (right subfigure), MNI and MI are truly efficient given the linear complexity. On the other hand, it takes much more time to compute MVC and RMVC. The interesting observation is: although MVC is an NP-hard problem, CPLEX seems to be reasonably fast in computing it. It is known that CPLEX uses the Simplex algorithm to solve linear programming problems, although the problem is worst-case exponential, Simplex can often deliver polynomial-time solutions for average cases (Spielman and Teng 2004). By comparing RMVC and MVC, as expected, the time for computing RMVC is much shorter than that for MVC. The difference between the linear-time MI and MNI is insignificant.

In summary, our experiments show that MVC can reduce the overestimation seen in MNI and MI with a reasonable computational overhead. RVMC stands out as a clear winner—it returns counts very close to that of MIS yet only requires polynomial time to compute. The improvement of MI over MNI is less exciting, as it heavily depends on the topological features of the patterns to show its advantage.

6 Related work

The frequent subgraph mining (FSM) problem is to find subgraphs in a data graph, and then enumerate all subgraphs with support (or frequency) above some minimum support threshold. FSM can be divided into two categories: finding frequent patterns

in transactional data graph (a graph database comprising multiple small graphs) and a single large data graph.

6.1 Mining algorithms

In the past years, fruitful graph mining algorithms have been published in the graph-transaction setting: a few representative publications include Borgelt and Berthold (2002), Yan and Han (2002), Yan and Han (2003), Inokuchi et al. (2003), Hong et al. (2003), Huan et al. (2003) and Kuramochi and Karypis (2004a). Although FSM in a single large graph setting has been studied (e.g., Kuramochi and Karypis 2005, 2004b; Elseidy et al. 2014), it receives less attention. The reason is that it is more challenging in both stages of finding pattern occurrence in large data graph and computing support.

6.2 Support measures

Related to the problem of support counting in a single graph setting, currently there are two major approaches. The first one is well-established overlap graph based support measure introduced in Vanetik et al. (2002) and its formal definitions were given in Vanetik et al. (2006) together with proofs for the sufficient and necessary conditions required for overlap graph based measure to be anti-monotonic. Several variations and extensions of overlap graph based measure were also proposed and analyzed, including exact and approximate MIS measures presented by Kuramochi and Karypis (2005), and overlap graph based MCP by Calders et al. (2008). In Calders et al. (2008), the authors also proposed the Lovász measure (SDP) by using the Lovász θ function that is bounded between MIS and MCP in overlap graph. There is another measure named *Schrijver* graph measure presented in Wang et al. (2013) that is very similar in nature to Lovász measure.

A relaxation of overlap graph based MIS is given by Wang and Ramon (2012). The concept of hypergraph is used in Wang and Ramon (2012) to define a variant of overlap graph (Vanetik et al. 2002) by replacing cliques by hypergraph edges and deleting non-dominating hypergraph edges. Hence it is still overlap-graph-based method in which vertices denote pattern occurrences or instances, and edges represent overlaps.

In Kuramochi and Karypis (2005), a greedy algorithm named GMIS is used as upper bound MIS. GMIS picks a vertex of the minimum degree, deletes that vertex together with all its adjacent vertices from the graph, and repeats this process until the graph becomes empty. In particular, for a graph G with a maximum degree Δ and an average degree d, the size $|I|$ of the MIS satisfies the following

$$|I| \leq \min\left(\frac{\Delta+2}{3}|GMIS(G)|, \frac{d+2}{2}|GMIS(G)|\right)$$

where $|GMIS(G)|$ is the size of the approximate MIS given by the GMIS algorithm.

In Calders et al. (2008), for the purpose of studying support measure in graph mining, subgraph isomorphism is extended to homomorphisms, isomorphisms and

homeomorphisms on both labeled and unlabeled, directed and undirected graphs, for both vertex and edge overlap.

Three ideal properties of a frequency measure have been concluded in Wang and Ramon (2012): (1) *Anti-monotonic* (we have investigated this property in this article); (2) *Normalized:* if for every pattern which has only independent images in a database graph, its support in that database graph equals the number of images. Independent images mean that they do not overlap according to some notion of overlap, such as sharing a vertex or an edge. (3) *Statistical soundness*: the function should give a measure of the number of independent observations of a phenomenon (the pattern). With this paper focusing on (1), (2) and (3) are obviously interesting topics for future research within the hypergraph framework. Earlier, Calders et al. (2008) has shown that overlap-graph-based support measures MIS, MCP and Schrijver are normalized.

7 Conclusions and future work

In this paper, we propose a new framework for studying support measures in frequent subgraph mining. This framework transforms pattern and data graph into hypergraphs containing occurrences and instances of the pattern as well as information of the original graph, in contrast to existing overlap graph techniques that only contain the former. Under the new hypergraph setting, encouraging results are achieved including the newly-defined linear-time MI and its variants that returns counts closer to number of independent pattern instances, the MVC measure that is very close to the MIS, and the MIES measure that is an equivalent version of MIS under the hypergraph framework.

Moreover, as for well accepted belief that MIS is NP-hard and cannot be approximated within a constant factor unless P = NP, we show that if the a pattern has k nodes, overlap graphs of pattern occurrences (instances) are actually in a subcategory of so called $(k+1)$-claw-free graph, and the MIS support can be approximated within a constant factor. The ratio between MIES and MVC is also within constant k. From the k-uniform hypergraph theory, it is clear that the SDP relaxation is strictly stronger than the LP relaxation of MIES. The integrality gap of LP relaxation of MIES is $k-1+\frac{1}{k}$, while that of SDP relaxation of MIES is at most $\frac{k+1}{2}$.

For future research, in the hypergraph-based framework, there are abundant opportunities for interesting theoretical and experimental research. In particular, explorations in the following directions are worth immediate attention. (1) Further investigation of support measures in hypergraph settings is promising since in this paper we only utilize the k-uniform property of the occurrence (instance) hypergraph to analyze the hardness and bounds of support measures, however the features such as pattern vertex labels have not been considered so far; (2) new overlap concepts can be investigated, as we have briefly mentioned in Sect. 4.4; (3) more support measures can be designed to fill the gap between MVC and MI via the subedge (subset) approach. For example, it would be useful to have a support measure with super-linear time complexity and counts smaller than MI; We can also explore the design of variations of MI that utilize a multitude of topological properties of pattern to find coarse-grained node subsets; (4)

It is also possible to include other desirable features in the design of support measure. One important example is called *additiveness*, meaning the computing can be done in a parallel manner therefore it brings great value to the implementation of the theoretical results; (5) More *user control* can be introduced into the framework in defining and selecting support measures for different applications.

Acknowledgements This work is supported by a grant (IIS-1253980) from the National Science Foundation (NSF) of U.S.A. Jinghan Meng was partially supported by an award (R01GM086707) from US National Institutes of Health (NIH).

References

Borgelt C, Berthold MR (2002) Mining molecular fragments: finding relevant substructures of molecules. In: Proceedings of the 2002 IEEE international conference on data mining, pp 51–58. https://doi.org/10.1109/ICDM.2002.1183885

Bringmann B, Nijssen S (2008) What is frequent in a single graph? In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 858–863

Calders T, Ramon J, Van yck D (2008) Anti-monotonic overlap-graph support measures. In: 2008 eighth IEEE international conference on data mining. IEEE, pp 73–82

Chan YH, Lau LC (2010) On linear and semidefinite programming relaxations for hypergraph matching. In: Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, pp 1500–1511

Cygan M (2013) Improved approximation for 3-dimensional matching via bounded pathwidth local search. In: 2013 IEEE 54th annual symposium on foundations of computer science (FOCS). IEEE, pp 509–518

Elseidy M, Abdelhamid E, Skiadopoulos S, Kalnis P (2014) Grami: frequent subgraph and pattern mining in a single large graph. Proc VLDB Endow 7(7):517–528

Fiedler M, Borgelt C (2007) Support computation for mining frequent subgraphs in a single graph. In: MLG, Citeseer

Füredi Z, Kahn J, Seymour PD (1993) On the fractional matching polytope of a hypergraph. Combinatorica 13(2):167–180

Holmerin J (2002) Improved inapproximability results for vertex cover on k-uniform hypergraphs. In: Proceedings of the 29th international colloquium on automata, languages and programming. Springer, London, ICALP '02, pp 1005–1016. http://dl.acm.org/citation.cfm?id=646255.756764

Hong M, Zhou H, Wang W, Shi B (2003) An efficient algorithm of frequent connected subgraph extraction. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 40–51

Huan J, Wang W, Prins J (2003) Efficient mining of frequent subgraphs in the presence of isomorphism. In: Third IEEE international conference on data mining, 2003. ICDM 2003. IEEE, pp 549–552

Hurkens CAJ, Schrijver A (1989) On the size of systems of sets every t of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. SIAM J Discrete Math 2(1):68–72. https://doi.org/10.1137/0402008

IBM (2011) IBM ILOG CPLEX optimization studio CPLEX user's manual

Inokuchi A, Washio T, Motoda H (2003) Complete mining of frequent patterns from graphs: mining graph data. Mach Learn 50(3):321–354

Karp RM (1972) Reducibility among combinatorial problems. In: Miller R (ed) Complexity of computer computations. Springer, New York, pp 85–103

Kunegis J (2018) Konect. http://konect.uni-koblenz.de/

Kuramochi M, Karypis G (2004a) An efficient algorithm for discovering frequent subgraphs. IEEE Trans Knowl Data Eng 16(9):1038–1051

Kuramochi M, Karypis G (2005) Finding frequent patterns in a large sparse graph. Data Min Knowl Discov 11(3):243–271

Kuramochi M, Karypis G (2004b) Grew-a scalable frequent subgraph discovery algorithm. In: Fourth IEEE international conference on data mining, 2004, ICDM'04. IEEE, pp 439–442

Lovász L (1979) On the shannon capacity of a graph. IEEE Trans Inf Theory 25(1):1–7

McKay BD, Piperno A (2014) Practical graph isomorphism, II. J Symb Comput 60:94–112. https://doi.org/10.1016/j.jsc.2013.09.003
Meng J, Tu Yc (2017) Flexible and feasible support measures for mining frequent patterns in large labeled graphs. In: Proceedings of the 2017 ACM international conference on management of data. ACM, New York, SIGMOD '17, pp 391–402. https://doi.org/10.1145/3035918.3035936
Pach J, Agarwal PK (2011) Combinatorial geometry, vol 37. Wiley, New York
Pitaksirianan N (2019) Graphmining. https://github.com/napath-pitaksirianan/GraphMining
Spielman DA, Teng SH (2004) Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. J ACM 51(3):385–463. https://doi.org/10.1145/990308.990310
Talukder N, Zaki MJ (2016) A distributed approach for graph mining in massive networks. Data Min Knowl Discov 30(5):1024–1052
Vanetik N, Shimony SE, Gudes E (2006) Support measures for graph data. Data Min Knowl Discov 13(2):243–260
Vanetik N, Gudes E, Shimony SE (2002) Computing frequent graph patterns from semistructured data. In: Proceedings of the 2002 IEEE international conference on data mining. IEEE Computer Society, Washington, ICDM '02, pp 458–465
Wang Y, Ramon J, Fannes T (2013) An efficiently computable subgraph pattern support measure: counting independent observations. Data Min Knowl Discov 27(3):444–477
Wang Y, Ramon J (2012) An efficiently computable support measure for frequent subgraph pattern mining. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 362–377
Yan X, Han J (2002) gSpan: graph-based substructure pattern mining. In: Proceedings of the 2002 IEEE international conference on data mining (ICDM 2002), 9–12 December 2002, Maebashi City, Japan, pp 721–724. https://doi.org/10.1109/ICDM.2002.1184038
Yan X, Han J (2003) Closegraph: mining closed frequent graph patterns. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 286–295

Affiliations

Jinghan Meng[1] · Napath Pitaksirianan[1] · Yi-Cheng Tu[1]

✉ Yi-Cheng Tu
tuy@mail.usf.edu

Jinghan Meng
jmeng@mail.usf.edu

Napath Pitaksirianan
napath@mail.usf.edu

[1] University of South Florida, 4202 E Fowler Ave, Tampa, FL 33620, USA