Flexible and Feasible Support Measures for Mining Frequent Patterns in Large Labeled Graphs

Jinghan Meng
Department of Computer Science & Engineering
University of South Florida
jmeng@mail.usf.edu

Yi-Cheng Tu [*]
Department of Computer Science & Engineering
University of South Florida
tuy@mail.usf.edu

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 constructing 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 that, we present two new support measures: minimum instance (MI) measure and minimum vertex cover (MVC) measure, that combine the advantages of existing measures. In particular, 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. Although the MVC measure is NP-hard, which means it is as hard as the existing overlap-graph-based measure, it can be approximated to a constant factor in polynomial time. We also provide polynomial-time relaxations for both measures and bounding theorems for all presented support measures in the hypergraph setting. We further show that the hypergraph-based framework can unify all support measures studied in this paper. This framework is also flexible in that more variants of support measures can be defined and profiled in it.

Keywords

Data mining; graph mining; support measures; hypergraph

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 Section 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.* [17], *anti-monotonicity* is well accepted by the graph mining community as an essential rule for support measure design. Vanetik *et al.* [17] also proposed an anti-monotonic support measure called the *maximum independent set based support* (MIS). 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. Its extensions (e.g., minimum clique partition (MCP) measure developed by Calders *et al.* [3]) also suffer from the same problem.

Another support measure named the **mininum-image-based support** (MNI) [2] 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 lack of *intuitiveness*. Specifically, by ignoring the topological structure of the query pattern, MNI could arbitrarily overestimate the frequency of a pattern, and this lowers its value in real appli-

[*]Y. Tu is also affiliated with Interdisciplinary Data Sciences Consortium (IDSC), University of South Florida

[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.

SIGMOD'17, May 14-19, 2017, Raleigh, NC, USA
© 2017 ACM. ISBN 978-1-4503-4197-4/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3035918.3035936

cations. 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. While both are anti-monotonic, they stand on opposite sides of the spectra of intuitiveness and efficiency. Therefore, the main objective of this study is to develop 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 intuitive, without over counting patterns as in MNI-based measures.

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. Based on the hypergraph concept, we define two new support measures: the **minimum instance (MI)** measure and the **minimum vertex cover (MVC)** measure. For the MI support measure, we show that the existing MNI support is an upper bound for it, or in other words, it 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, which means it is as hard as the overlap-graph-based MIS measure, MVC enjoys a k-competitive approximate algorithm. This is in sharp contrast to the proved fact that the MIS measure cannot be approximated to a constant factor in polynomial time unless P = NP. Furthermore, we provide polynomial-time computable relaxations of both MVC and MIS measures. This makes MVC and MIS more efficient while still providing meaningful frequency values.

We further demonstrate that our hypergraph-based method serves as a unified framework that encapsulates not only MI and MVC, but also the existing support measures including MIS and MNI. Specifically, we first 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 instance hypergraph, the **maximum independent edge set support (MIES)**. Bounding theorems that describe the differences among all support measures included in the hypergraph-based framework are also presented.

To summarize the main findings of this paper, let us denote the values of MIS, MIES, polynomial relaxation of MIES and MVC, MI, MNI as σ_{MIS}, σ_{MIES}, ν_{MIES}, ν_{MVC}, σ_{MVC}, σ_{MI}, σ_{MNI}, respectively. The bound and computational complexity of all aforementioned support measures are as follows.

$$\sigma_{\mathbf{MIS}} = \sigma_{\mathbf{MIES}} \leq \nu_{\mathbf{MIES}} = \nu_{\mathbf{MVC}} \leq \sigma_{\mathbf{MVC}} \leq \sigma_{\mathbf{MI}} \leq \sigma_{\mathbf{MNI}}$$

Measure	$\sigma_{MIS}(\sigma_{MIES})$	σ_{MVC}	ν_{MIES}
Complexity	NPH	NPH k-approx	Poly
Measure	ν_{MVC}	σ_{MI}	σ_{MNI}
Complexity	Poly	Linear	Linear

Furthermore, we showcase the potential of the new framework as a platform for defining and profiling a wide ranges of support measures. Figure 1 is a sketch of the hypergraph framework and displays counts of support measures of a one-edge pattern in a small data graph.

In a nutshell, the occurrences of a pattern in a data graph are viewed as edges in hypergraph framework. From the perspective of MIS and MVC support measures, vertices in

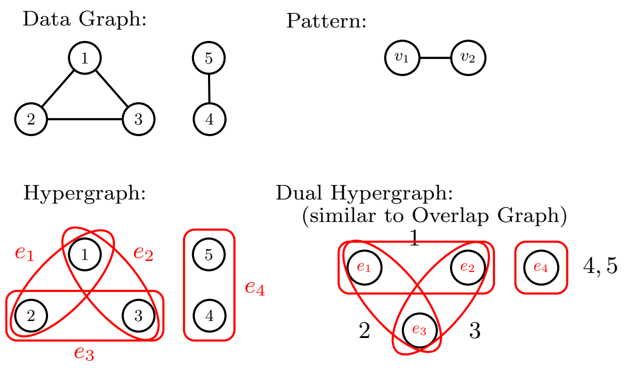


Figure 1: Example showing support measures of the pattern and the hypergraph framework. In this case, we have $\sigma_{MIS} = 2 \leq \sigma_{MVC} = 3 \leq \sigma_{MI} = 4 \leq \sigma_{MNI} = 5$

edges are treated as one set, and the cardinality of maximum independent edge set and minimum vertex cover set are used as support measure respectively. MNI and MI support measures break the edges into subedges (subsets) and then adopt subedges' minimum distinct images count as support. In this way, MNI and MI support measures reduce computation complexity to linear time (in number of occurrences).

The rest of this paper is organized as follows: In Section 2, we formally define the problem and sketch the necessary background for the problem; In Section 3, we introduce our new support measures and study their features; In Section 4, we present a framework that unifies all support measures mentioned in this paper and discuss its potential in defining and studying a wide range of support measures; In Section 5, we present a brief review of related work; and we conclude our paper in Section 6.

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.

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 dataset 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.

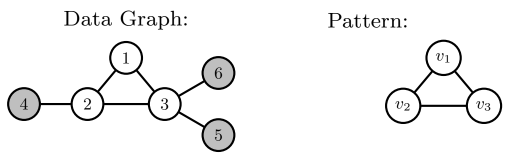
Note that occurrence and instance are two different concepts. An occurrence is an isomorphism between pattern P and a subgraph of dataset graph G, while an instance is a subgraph of G that is isomorphic to pattern P. There can be multiple occurrences mapping pattern P to one instance. For example, in Figure 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.

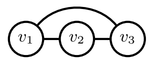
2.3 Overlap Concepts and Support Measure

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 $\sigma : G \times G \to \mathbb{R}^+$, 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





Occurrences	v_1	v_2	v_3
f_1:	1	2	3
f_2:	1	3	2
f_3:	2	1	3
f_4:	2	3	1
f_5:	3	1	2
f_6:	3	2	1
# of images:	3	3	3

MNI = 3

Figure 2: Example showing the triangle-shaped pattern has 6 occurrences and 1 instance in data graph. We have an MIS measure of 1 but the MNI measure equals 3 – it overestimates the count of pattern

[17, 15]. 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 [17, 15].

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.* [17] 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. **Edge overlap**: An edge overlap of occurrences f_1 and f_2 of pattern $P = (V_P, E_P)$ in data graph $G = (V, E)$ exists if edge sets of $f_1(E_P)$ and $f_2(E_P)$ intersect, that is, $f_1(E_P) \cap f_2(E_P) \neq \emptyset$ where $f_i(E_P) = \{(f_i(u), f_i(v)) : (u, v) \in E_P\}$, $i = 1, 2$. An edge 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 edge sets of S_1 and S_2 intersect, that is, $E_{S_1} \cap E_{S_2} \neq \emptyset$.

Definition 14. 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 15. An **independent (vertex) set** of graph $G = (V, E)$ is a subset of V, such that no two of which are adjacent.

Definition 16. 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.

The main drawback of the MIS support is computing efficiency - it is shown [12] that maximum independent set problem is NP-hard in number of graph vertices. Because MIS [17] 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]

Bringmann and Nijssen [7] 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 Figure 2, node v_1 in the pattern has 3 distinct images because there are occurrence map v_1 to vertices 1, 2, 3 in data graph (e.g., $f_1(v_1) = 1$, $f_3(v_1) = 2$ and $f_5(v_1) = 3$).

Definition 17. Given a pattern $P = (V_P, E_P)$, a data graph $G = (V, E)$, if P has m occurrences $\{f_1, f_2, \cdots, f_m\}$ in G, the **minimum image based (MNI) support** of P in G is defined as

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

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

MNI can be configured to allow certain level of tolerance in images [7]. Given a parameter k, a support measure can be defined based on determining where each connected subgraph containing k nodes of the pattern can be matched with each other.

Definition 18. Given a pattern $P = (V_P, E_P)$, a data graph $G = (V, E)$, and a positive integer parameter k, if P has m occurrences $\{f_1, f_2, \cdots, f_m\}$ in G, the **minimum k-image based support** is defined as

$$\sigma_{MNI}(P, G, k) = \min_{V'} |\{f_i(V') : i = 1, 2, \cdots, m\}|,$$

where V' is connected subset of V_P and $|V'| = k$.

[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).

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. The proof of anti-monotonicity of $\sigma_{MNI}(P, G, k)$ is similar.

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 lack of intuitiveness. Let us revisit the example in Figure 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.

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 [3]. The MNI requires linear time to compute but can return an arbitrarily large count for a pattern [2]. Both MIS and MNI have variants other than the basic forms mentioned in this section. We will introduce some of the variants in Section 5. 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 Figure 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 we present two anti-monotonic support measures that achieve counts that are closer to the number of pattern instances and MIS support measure.

3. NEW SUPPORT MEASURES

In this section, we first introduce a new concept named **occurrence/instance hypergraph** from which our new support measures are 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 19. A **hypergraph** $H = (V, E)$ consists of a set $V = \{v_1, v_2, \cdots, v_n\}$ of n vertices and a set $E = \{e_1, e_2, \cdots, e_m\}$ 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 $i = j$.

For discussions related to the features of relevant support measures, we also introduce the concept of dual hypergraph.

Definition 20. 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 $E = \{e_1, e_2, \cdots, e_m\}$ and the edges are given by $X = \{X_1, X_2, \cdots, X_n\}$ where $X_j = \{e_i : v_j \in e_i\}, j = 1, 2, \cdots, n$, that is, X_j is the collection of all edges in H which contain vertex v_j.

As a key technique, we show how occurrences and instances of a pattern are integrated into a hypergraph and support measures are defined within the hypergraph.

Definition 21. If pattern $P = (V_P, E_P)$ has m occurrences $\{f_i : i = 1, \cdots, m\}$, 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 $E = \{e_i : i = 1, \cdots, m\}$, 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 22. If pattern $P = (V_P, E_P)$ has m instances $\{S_i = (V_{S_i}, E_{S_i}) : i = 1, \cdots, m\}$ 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 $E = \{e_i : i = 1, \cdots, m\}$, each $e_i = V_{S_i}$. We also give each e_i a label S_i to distinguish them from each other.

Let us use Figure 3 to show how the hypergraphs are constructed: the occurrence hypergraph $H^O = (V, E)$ has vertex set $V = \{1, 2, 3,\ 4, 5, 6,\ 8, 9, 10, 11,\ 13, 15, 16, 17\}$ and edge set $E = \{e_1, e_2, e_3, e_4, e_5, e_6\} = \{\{1, 2, 3\}, \{4, 5, 6\}, \{4, 6, 8\}, \{8, 9, 10\}, \{11, 13, 17\}, \{11, 15, 16\}\}$. Note that in occurrence (instance) hypergraph, each edge represents one occurrence (instance). In Figure 3 the instance hypergraph also has the same edge set, hence it looks like the same as occurrence hypergraph. However, in many other situations occurrence and instance hypergraph are different. For example in Figure 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 edge labels are different. On the other hand, instance hypergraph of pattern in Figure 2 has only one edge since there is one instance. The reason why we give such labels for each edge is to save necessary information from data graph for studying various support measures.

The differences between the occurrence hypergraph and instance hypergraph are partly caused by the pattern's topological structure, or more specifically, automorphisms. When a pattern has non-identity automorphisms, multiple occurrences project the pattern to the same subgraph of dataset graph. When pattern admits no automorphism, its occurrence and instance hypergraphs are quite similar.

As shown in Figure 3, 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 Figure 3, e_3 and e_2 overlap on two vertices but e_3 and e_4 overlap on one vertex. We argue that a hypergraph framework keeps more such information and offers more insight and flexibility for further investigation, compared to overlap graph based support measure such as MIS [17].

In summary, the hypergraph is a suitable topological representation of pattern occurrences (instances) for investigating support measures. More details will follow.

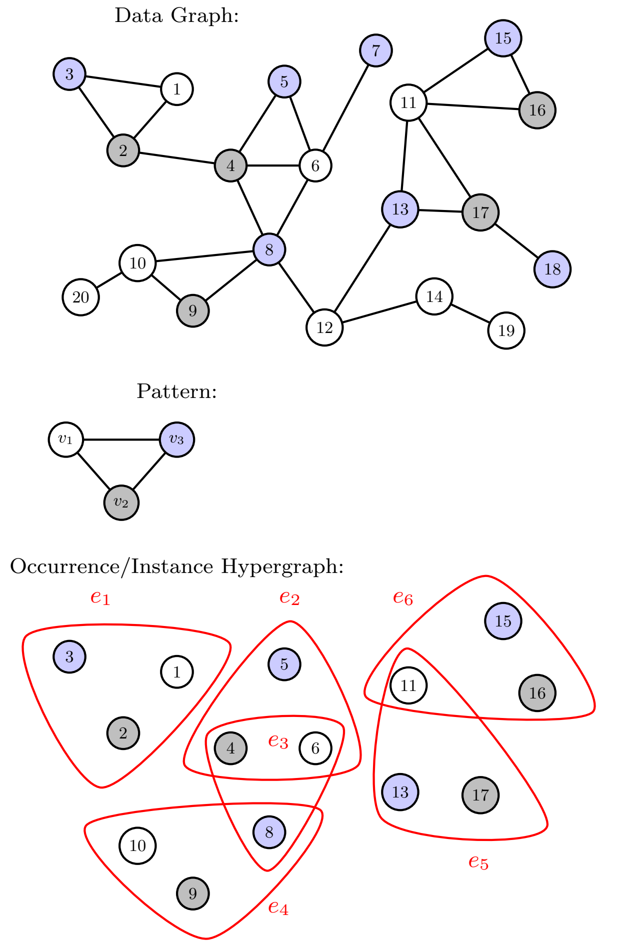


Figure 3: Occurrence/instance hypergraph of a triangular pattern

3.1 Minimum Instance Support Measure

As described above, the MNI support measure is insensitive to structures 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 Figure 4.

We can see three pattern 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.

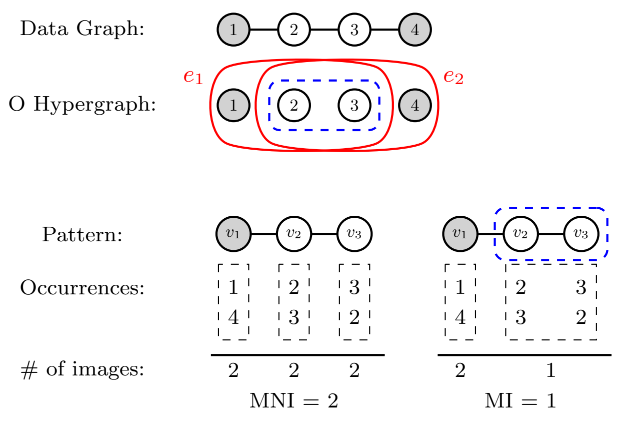
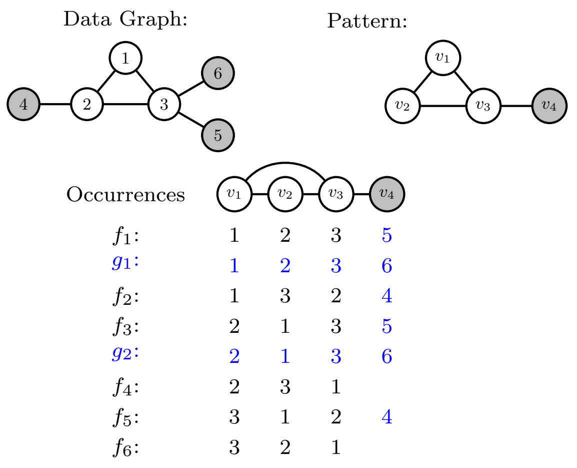


Figure 4: MNI vs MI Support Measure



	v_1	v_2	v_3	v_4
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	

Figure 5: An example showing occurrences of a pattern while being extended to a superpattern

Definition 23. Given a pattern $P = (V_P, E_P)$, a data graph $G = (V, E)$, if P has m occurrences $\{f_1, f_2, \cdots, f_m\}$ in G, we define **coarse-grained node subset** W as a subset of V_P. The **coarse-grained node subset image count** is defined as

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

In Figure 4, if 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 introduce the definition of transitive node subset as follows.

Definition 24. A pair of vertices u and v in graph G is **transitive** if there is at least one automorphism f of G such that $f(u) = v$.

Note that in this definition, u and v can be equal.

Theorem 1. *For vertices u, v and w in graph G, if the pair of u, v and the pair v, w are both transitive, the pair of u, w is transitive.*

Proof. Since the pair of u, v and the pair v, w are both transitive in G, there are two automorphisms f and g of G such that $f(u) = v$ and $g(v) = w$. Hence the composition of f and g, denoted as $f \circ g$, is also an automorphism of G and $f \circ g(u) = w$, which states that the pair u, w is also transitive in G. $\square$

Definition 25. The **transitive node subset** T in pattern $P = (V_P, E_P)$ is a subset of V_P such that any pair of vertices in T is transitive in P.

The transitive node subset is a special case of coarse-grained node subset, we shall show that MNI and MI measures can be connected by using these concepts.

Now we are ready to define new support measure of pattern P using the definition of coarse-grained node subset image count.

Definition 26. Given a pattern $P = (V_P, E_P)$, a data graph $G = (V, E)$, let T be a transitive node subset in a subgraph of pattern P, the collection of all such T is denoted as $\mathcal{T} = \{T\}$. The **minimum instance based support (MI)** of P in G is defined as

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

As for the example in Figure 4, 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 2. *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 $\{f_1, f_2, \cdots, f_m\}$ in G and P has l occurrences $\{f'_1, f'_2, \cdots, f'_l\}$ 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 $\{f_1, f_2, \cdots, f_m\}$ is greater or equal to its image count $c'(T)$ under the mappings $\{f'_1, f'_2, \cdots, f'_l\}$. 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 $\sigma_{MI}(p, G) \geq \sigma_{MI}(P, G)$. $\square$

Figure 5 shows the anti-monotonicity of MI support measure via an illustrative example.

Theorem 3. *The MI support measure is linear-time computable.*

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. $\square$

Theorem 4. *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 $\sigma_{MI}(P, G) = \min_{T \in \mathcal{T}} c(T) \leq \min_{W \in \mathcal{W}} c(W) = \sigma_{MNI}(P, G)$. $\square$

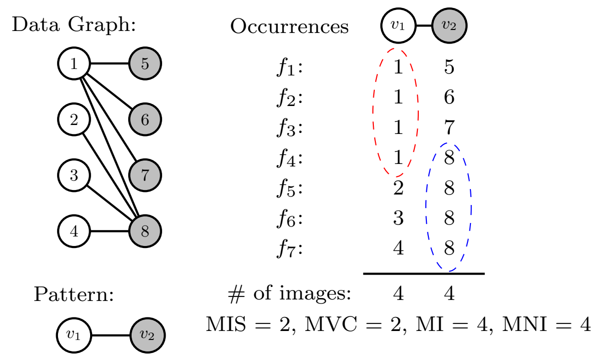


Figure 6: MNI measure can over-estimate count of patterns as it ignores partial overlap

In practice, there will be many cases in which MI measure is strictly smaller than the MNI measure. As in Figure 4, 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.

3.2 Minimum Vertex Cover Support Measure

The purpose of developing MI is to achieve reasonable count by avoiding overestimation by MNI. However, MI cannot handle the type of overlap shown in Figure 6. 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 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., Figure 6) 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. We opt to 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 27. 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 28. 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 Figure 6, 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 5. *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 $\{f_1,f_2,\cdots,f_m\}$ and $\{f'_1,f'_2,\cdots,f'_{m'}\}$ 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 is also 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, $\sigma_{MVC}(p,G)\geq\sigma_{MVC}(P,G)$. $\square$

Let us refer to Figure 5 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 supperpattern's occurrence hypergraph.

THEOREM 6. *Given a pattern P and data graph G, we have*

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

PROOF. 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 $\{f_i(T), i=1,2,\cdots,m\}$. It is obvious that a minimum vertex cover of $\{f_i(T): i=1,2,\cdots,m\}$ is also a vertex cover of occurrence hypergraph because $f_i(T)\subseteq f_i(V_P)$. Hence $\sigma_{MI}=|\{f_i(T): i=1,2,\cdots,m\}|\geq$ cardinality of minimum vertex cover of $\{f_i(T), i=1,2,\cdots,m\}\geq\sigma_{MVC}(P,G)$. $\square$

Now we see that MVC is anti-monotonic, and is bounded by MI. In Section 4.4, 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, the best approximate algorithms achieve a factor $k-o(1)$ approximation under polynomial time [8]. In summary, MVC returns smaller counts but requires more time to compute as compared to MI.

4. THE HYPERGRAPH FRAMEWORK

A very interesting result of our work is that existing categories of support measures (i.e., MNI and MIS), although

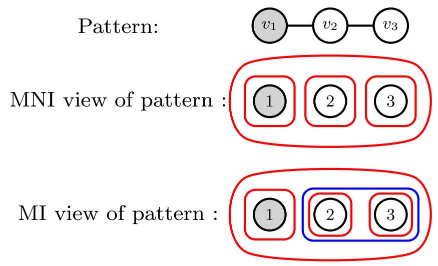


Figure 7: The MNI and MI's view of pattern in hypergraph framework

constructed from different techniques, can also be incorporated into the new hypergraph settings. Therefore, we have a unified framework that encapsulates all major support measures mentioned in this paper.

4.1 MNI in Hypergraph Framework

We first show that the MNI support can be easily related to the occurrence hypergraph and the new MI support measure. 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 Section 3.1, we see how $\sigma_{MNI}(P, G)$ and its parameterized version $\sigma_{MNI}(P, G, k)$ 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

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

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 7 displays how MNI and MI fit in hypergraph framework. Note that MI is not simply a transformation from graph pattern to a hypergraph version. For example, in Figure 7, if we disconnect v_3 with v_2 and then connect it with v_1, the pair v_2 and v_3 is still in a transitive node subset while it is no longer connected by one edge in pattern graph. Hypergraph edges are used to capture desired and essential features of pattern graph. From this point of view, hypergraph is indeed a suitable and flexible framework for support measures.

4.2 MIS in Hypergraph Framework

We now show that, MIS, which is defined based on overlap graphs, can also be mapped to the hypergraph framework. For that, we shall introduce a new measure in hypergraph setting and show it is equivalent to MIS.

Definition 29. 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.

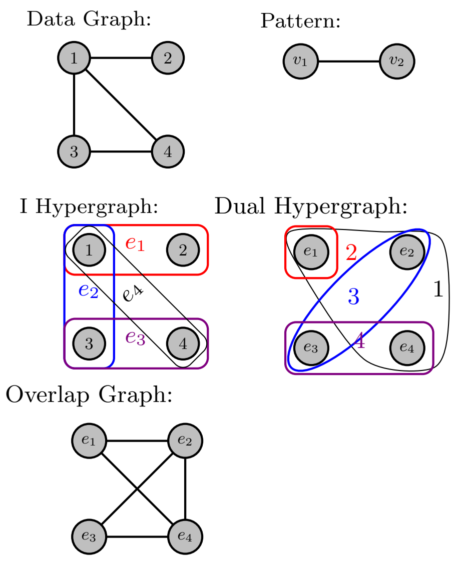


Figure 8: The instance hypergraph and its dual for a small pattern in a data graph

The overlap graph approach is similar in nature to how dual hypergraph is built. Edges in instance hypergraph represent instances of a pattern, therefore the MIS support is equal to the maximum cardinality of independent edge set of the instance hypergraph. For example, according to the definition of dual hypergraph, all edges in H are vertices in dual H^*, and they are also vertices in overlap graph. If two edges e_i, e_j in H overlap on 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. Actually, each edge in dual H^* is equivalent to a clique in the overlap graph. If H^* is a simple hypergraph, then it is the same as the overlap hypergraph introduced in [19]. The example in Figure 8 shows how similar dual hypergraph and overlap graph are to each other.

Within the hypergraph framework, we shall show that the MIS is equivalent to MIES, and the latter is also anti-monotonic. To achieve such analysis, an integer programming formulation can be developed. Such formulation also serves as relaxation for reducing computing costs of expensive measures such as MVC and MIES (see Section 4.3).

Let us start with MVC by assuming that hypergraph $H = (V, E)$ consists of a set $V = \{v_1, v_2, \cdots, v_n\}$ of n vertices and a set $E = \{e_1, e_2, \cdots, e_m\}$ 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 in it should be chosen and the object is to minimize that number of vertices intersecting all edges. Now we can write:

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

By definition the dual hypergraph H^* of H is a hypergraph whose vertices and edges are interchanged, so that the vertices are given by $\{e_i : i = 1, 2, \cdots, m\}$ and the edges are X

$= \{X_1, X_2, \cdots, X_n\}$ 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:

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

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

THEOREM 7. *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. The problem of finding maximum independent edge set in occurrence hypergraph H is equivalent to finding maximum independent vertex set in dual hypergraph H^* with vertices corresponding to edges in H and vice versa. Although the dual hypergraph and overlap graph can be different in their forms, we can show that their sizes of maximum independent vertex set are the same. We use the linear programming techniques to show this equivalence.

In dual hypergraph H^*, MIES is equal to the solution of maximum optimization problem (Eq. (2)), while overlap graph based MIS is equal to the solution of problem: max $\sum_{e\in E} y(e)$ subject to $y(e_j) + y(e_k) \le 1$, if e_j and e_k overlap, $y(e) \in \{0, 1\}$, $\forall e$. Figure 8 shows an illustrative example of this.

We shall show that the constraints of the two maximum optimization problems are equivalent, therefore their solution values shall be the same. Furthermore, because any edge (two vertices) in overlap graph is contained in a dual hypergraph edge, we only need to show that for X_i with size larger that one, the equalities max $\sum_{e\in X_i} y(e)$ subject to $y(e) \in \{0, 1\}$, $\forall e$ is equivalent to $y(e_j) + y(e_k) \le 1$ for any $e_j, e_k \in X_i, 1 \le j \ne k \le n$, where $y(e_j), y(e_k)$ are restricted to $\{0, 1\}$.

It is obvious that $\sum_{e\in X_i} y(e) \le 1$ implies $y(e_j)+y(e_k) \le 1$ for any $e_j, e_k \in X_i, 1 \le j \ne k \le n$ because every $y(e), e \in X_i$ is non-negative. Hence, we need to prove that $y(e_j)+y(e_k) \le 1$ for any $e_j, e_k \in X_i, 1 \le j \ne k \le n$ implies $\sum_{e\in X_i} y(e) \le 1$.

Assume that $y(e_j) + y(e_k) \le 1$ for any $e_j, e_k \in X_i, 1 \le j \ne k \le n$ and $y(e) \in \{0, 1\}, \forall e \in X_i$. If there exits an $e \in X_i - \{e_j, e_k\}$, then there must be two more edges $\{e, e_j\}$ and $\{e, e_k\}$ in overlap graph because all e, e_i, e_k overlap at vertex i. Therefore we should have $y(e) + y(e_j) \le 1, y(e) + y(e_k) \le 1, y(e_j) + y(e_k) \le 1$ and $y(e), y(e_j), y(e_k) \in \{0, 1\}$, which implies $y(e_j) + y(e_k) + y(e) \le 1$. Similarly we can add all other $e \in X_i$ into this equality so that we have $\sum_{e\in X_i} y(e) \le 1$. In the end, we obtain $\sigma_{MIES}(P, G) = \sigma_{MIS}(P, G)$. □

We take Figure 8 as an example, in which the MIS support in overlap graph is 2. Taking a close look, for example, $\{e_1, e_3\}$ forms a maximum independent set. The MIES in instance hypergraph is also 2.

THEOREM 8. *The MIES measure is anti-monotonic.*

PROOF. Since MIES is equivalent to anti-monotonic MIS, it is also anti-monotonic. □

4.3 Polynomial Time Relaxation

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, the solution for a linear programming gives a upper (lower) bound on the optimal solution to the original maximization (minimization) problem.

In Section 4.2, we have presented the integer programming transformation of the problems. Based on that, we are ready to relax the integrability conditions of minimum vertex cover problem to obtain linear programming problem and formally define the relaxed versions of the MVC and MIES measures. We shall also show that they are both anti-monotonic.

Definition 30. Given a pattern P in a data graph G, and its occurrence (instance) hypergraph $H = (V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ and $E = \{e_1, e_2, \cdots, e_m\}$, the **polynomial-time MVC** support measure of pattern P in graph G is defined as

$$\begin{aligned} \nu_{MVC}(P, G) = \min \quad & \sum_{v\in V} x(v) && (3)\\ \text{subject to} \quad & \sum_{v\in e_i} x(v) \ge 1, \quad \forall i\\ & 0 \le x(v) \le 1, \quad \forall v. \end{aligned}$$

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

Definition 31. Given a pattern P in a data graph G, and its occurrence (instance) hypergraph $H = (V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ and $E = \{e_1, e_2, \cdots, e_m\}$, dual hypergraph $H^* = (E, X)$, $X = \{X_1, X_2, \cdots, X_n\}$, the **polynomial-time MIES** support measure of pattern P in graph G is defined as

$$\begin{aligned} \nu_{MIES}(P, G) = \max \quad & \sum_{e\in E} y(e) && (4)\\ \text{subject to} \quad & \sum_{e\in X_i} y(e) \le 1, \quad \forall i\\ & 0 \le y(e) \le 1, \quad \forall e. \end{aligned}$$

THEOREM 9. *The polynomial-time MVC support measure is anti-monotonic.*

PROOF. We shall show that $\nu_{MVC}(p, G) \ge \nu_{MVC}(P, G)$ for any pattern p and its superpattern P in dataset 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 $x^* = \nu_{MVC}(p, G)$ to the LP (3) to construct another function x^{**} such that $x^* \ge x^{**} \ge \nu_{MVC}(P, G)$, in this way we can prove that $\nu_{MVC}(p, G) \ge \nu_{MVC}(P, G)$.

Let $\nu_{MVC}(p, G) = \Sigma_{v\in V} x^*(v)$ be a solution to the LP (3) associated with H_p, where $\sum_{v\in e} x^*(v) \ge 1$ for any $e \in E$ and $0 \le x^*(v) \le 1$ 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 $e \in E$ as its subset, 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 $x^{**} \geq \min \sum_{v\in V'} x(v) = \mu_{MVC}(P,G)$. Since $V \subseteq V'$, it is obvious that

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

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

THEOREM 10. *The polynomial-time MIES support measure is anti-monotonic.*

PROOF. The proof is similar to that of Theorem 9. We omit the details here. □

4.4 Bounding Theorems

To explore the relationship among all the support measures within the new framework, we derive the following theorems from the classic results in the hypergraph field. 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 number of vertices which means that occurrence (instance) hypergraphs are uniform hypergraphs. We first study the difference between the MIES and MVC measures.

THEOREM 11. *Given a pattern P containing k nodes, 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. 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 measure is larger than the MIES (that equals MIS according to Theorem 7).

Based on well-established results in linear programing [16], we obtain the following relationship between σ_{MIS}, σ_{MVC}, and support measures created from relaxation on constraints in their corresponding linear programming problems.

THEOREM 12. *Given a pattern P, data graph G, and occurrence (instance) hypergraph H, we have*

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

PROOF. The first and last inequality are directly given by the definitions of corresponding linear programming problems. The equality follows from the duality theorems of linear programming [16]. □

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.

Nevertheless, the results in Theorem 12 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. Despite the close relationship between vertex cover and independent edge set in graphs, without the relaxation, it is not possible to find a vertex cover under polynomial time and then derive the complementary maximum independent set.

The comparison between σ_{MVC}, σ_{MI} and σ_{MNI} were examined in Theorems 4 and 6. Putting all together, we have

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

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

4.5 Other Extensions Within the Framework

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

A variant of the simple overlap, called harmful overlap, was introduced in [6]. 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.

Definition 32. [6] 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 33. 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 a transitive node subset in a subgraph of pattern P, and $f_1(v) = f_2(w) \in f_1(V_P) \cap f_2(V_P)$.

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

Note that structural overlap is different from harmful overlap. In Figure 9, a structural overlap of occurrences g_1 and g_2 exists because, for transitive pair v_2, v_3, we have $g_1(v_3) = 3$, $g_2(v_2) = 3 \in \{3\} = g_1(V_P) \cap g_2(V_P)$. The concept of structural overlap is originated from MI support measure which considers overlap on transitive node subsets. When calculating MI, we let node v_2 and v_3 form a transitive node subset, it has two images $\{2, 3\}$ and $\{3, 4\}$, and 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 Figures 9 and 10 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 does not exist. The reason is that $g_1(V_P)\cap g_2(V_P) = \{3\}$, but 3 is 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,

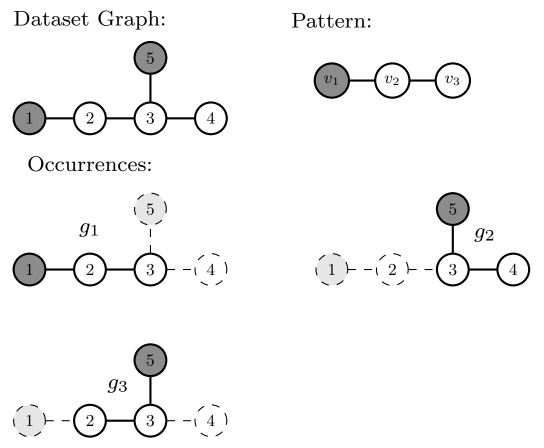


Figure 9: Structural overlap $\neq$ harmful Overlap

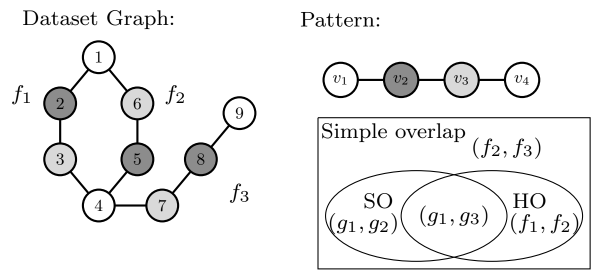


Figure 10: Example and Venn diagram for illustrating the relationship among structural overlap, harmful overlap, and simple overlap

a harmful overlap of f_1 and f_2 exists but structural overlap does not. 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 Figure 10). Harmful overlap and structural overlap can exist at the same time (e.g., g_1 and g_3 in Figure 9).

Fiedler and Borgelt [6] 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 one pair of nodes is transitive in a subgraph of P and their images are in both images $f_1(V_P)$ and $f_2(V_P)$, then 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 help define frontier to explore in support measure theory. For example, 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 (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). In the hypergraph setting, because of its close connection to MI support measure, structural overlap can be potentially utilized to explore variants of MI support measures.

5. RELATED WORK

The frequent subgraph mining (FSM) problem is to find subgraphs in a data graph, and them 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. In the past years, fruitful results have been published in the graph-transaction setting: a few representative publications include Borgelt and Berthold [1], Yan and Han [21, 22], Inokuchi *et al.* [11], Hong *et al.* [9], Huan *et al.* [10], Kuramochi and Karypis [13]. Although FSM in a single large graph setting has been studied (e.g., Kuramochi and Karypis [14, 15], Elseidy *et al.* [5]), 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.

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, which was first introduced in Vanetik [17] and its formal definitions were given in Vanetik *et al.* [18] 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 [15], and overlap graph based MCP by Calders *et al.* [4]. In [4], the authors also propose the Lovasz measure by using the theta function that is proven to be bounded between MIS and MCP in overlap graph. This is very similar in nature to another measure named *Schrijver* graph measure [20].

A relaxation of overlap graph based MIS is given by Wang *et al.* [19]. Note that the concept of hypergraph is also used in [19] to define a variant of overlap graph [17] by replacing cliques by hypergraph edges and deleting non-dominating hypergraph edges. In our method, we do not build overlap graph, our hypergraph vertices are node images of query pattern, and edges represent occurrences and instances.

6. 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 latter. By doing this, state-of-art hypergraph theorems can provide theoretical explanations to interpret the relationship between occurrences (mapped as edges in hypergraph). Under the new hypergraph setting, encouraging results are achieved including the linear-time MI measure that returns counts closer to pattern instance, 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. Furthermore, the MVC measure can be approximated by polynomial time algorithm within a constant factor while MIS measure does not have this privilege.

With 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) New overlap concepts can be investigated, as we have briefly mentioned in Section 4.5; (2) More support measures can be designed that fill the gap between MVC and MI. For example, it would be useful to have a support measure with super-linear time complexity but is smaller than the counts of MI; We can also explore the design of variations of MI that utilize a multitude of topological properties of pattern to find the transitive vertex set; (4) Inclusion of 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; and (5) More *user control* can be introduced into the framework in defining and selecting support measures for different applications.

Acknowledgment

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

7. REFERENCES

[1] C. Borgelt and M. R. Berthold. Mining molecular fragments: Finding relevant substructures of molecules. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 51–58. IEEE, 2002.

[2] B. Bringmann and S. Nijssen. What is frequent in a single graph? In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 858–863. Springer, 2008.

[3] T. Calders, J. Ramon, and D. Van Dyck. Anti-monotonic overlap-graph support measures. In *2008 Eighth IEEE International Conference on Data Mining*, pages 73–82. IEEE, 2008.

[4] T. Calders, J. Ramon, and D. Van Dyck. Anti-monotonic overlap-graph support measures. In *2008 Eighth IEEE International Conference on Data Mining*, pages 73–82. IEEE, 2008.

[5] M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis. Grami: Frequent subgraph and pattern mining in a single large graph. *Proceedings of the VLDB Endowment*, 7(7):517–528, 2014.

[6] M. Fiedler and C. Borgelt. Support computation for mining frequent subgraphs in a single graph. In *MLG*. Citeseer, 2007.

[7] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422. ACM, 2013.

[8] E. Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM Journal on Computing*, 31(5):1608–1623, 2002.

[9] M. Hong, H. Zhou, W. Wang, and B. Shi. An efficient algorithm of frequent connected subgraph extraction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 40–51. Springer, 2003.

[10] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 549–552. IEEE, 2003.

[11] A. Inokuchi, T. Washio, and H. Motoda. Complete mining of frequent patterns from graphs: Mining graph data. *Machine Learning*, 50(3):321–354, 2003.

[12] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[13] M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1038–1051, 2004.

[14] M. Kuramochi and G. Karypis. Grew-a scalable frequent subgraph discovery algorithm. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 439–442. IEEE, 2004.

[15] M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph. *Data mining and knowledge discovery*, 11(3):243–271, 2005.

[16] J. Pach and P. K. Agarwal. *Combinatorial geometry*, volume 37. John Wiley & Sons, 2011.

[17] N. Vanetik, E. Gudes, and S. E. Shimony. Computing frequent graph patterns from semistructured data. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 458–465. IEEE, 2002.

[18] N. Vanetik, S. E. Shimony, and E. Gudes. Support measures for graph data. *Data Mining and Knowledge Discovery*, 13(2):243–260, 2006.

[19] Y. Wang and J. Ramon. An efficiently computable support measure for frequent subgraph pattern mining. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 362–377. Springer, 2012.

[20] Y. Wang, J. Ramon, and T. Fannes. An efficiently computable subgraph pattern support measure: counting independent observations. *Data Mining and Knowledge Discovery*, 27(3):444–477, 2013.

[21] X. Yan and J. Han. 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*, pages 721–724, 2002.

[22] X. Yan and J. Han. Closegraph: mining closed frequent graph patterns. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 286–295. ACM, 2003.