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

ABSTRACT

In recent years, graph databases such as Twitter and Facebook social graph and citation maps have grown rapidly, therefore graph mining techniques are becoming more and more important. 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. Within this framework, 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 computable relaxations for both measures. Bounding theorems are given for all presented support measures in the new 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 variations of support measures can be defined and profiled in it.

Categories and Subject Descriptors

H.2.8 [**XXX**]: XXXX—*Data mining*

Keywords

Graph Mining

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 the so-called *monotonicity* 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.* [11], *anti-monotonicity* is well accepted by the graph mining community as an essential rule for support measure design. Vanetik *et al.* [11] 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,

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

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

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 applications. The overlap-graph-based support (represented by MIS) and MNI support, as well as their variations, 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 MVC, 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. Furthermore, we showcase the potential of the new framework as a platform for defining and profiling a wide ranges of support measures.

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**

$$G = \langle V_G, E_G, \lambda_G \rangle$$

consists of a set of vertices V_G, a set of edges $E_G \subseteq V_G \times V_G := \{(u, v) \,|\, u, v \in V_G,\, u \neq v\}$ and a labeling function $\lambda_G : V_G \cup E_G \to \Sigma$ that maps each vertex or edge of the graph to an element of the alphabet Σ.

Definition 2. A graph $G' = \langle V_{G'}, E_{G'}, \lambda_{G'} \rangle$ is a **subgraph** of $G = \langle V_G, E_G, \lambda_G \rangle$ if $V_{G'}$ is a subset of V_G and $E_{G'}$ is a subset of E_G and for all $v \in V_{G'}$, $\lambda_{G'}(v) = \lambda_G(v)$.

Definition 3. A **pattern** $P = \langle V_P, E_P, \lambda_P \rangle$ 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 $p \subset P$. 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 a mapping $f : V(G_1) \to V(G_2)$ such that

- $\forall v \in V_{G_1}$, $f(v) \in V_{G_2}$ and $\forall v \in V_{G_2}$, $f(v) \in V_{G_1}$; and
- $\forall (v_1, v_2, l) \in E_{G_1}$, $(f(v_1), f(v_2), l) \in E_{G_2}$ and $\forall (v_1, v_2, l) \in E_{G_2}$, $(f(v_1), f(v_2), l) \in E_{G_1}$.

The two descriptions state that the isomorphic function preserves both vertex labels and edge labels. The mapping f is called **isomorphism** between G_1 and 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 large data graph, we need to define the concept of an occurrence and an instance of the pattern in the data graph.

Definition 8. Given a pattern $P = \langle V_P, E_P, \lambda_P \rangle$ and a graph $G = \langle V_G, E_G, \lambda_G \rangle$, an **occurrence** is an isomorphism f from pattern P to a subgraph G' of G. That is to say f is also a subgraph isomorphism from P to G.

Definition 9. A subgraph S of G is an **instance** of 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 1 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 dataset 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 [12]. 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 [12].

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 on G is **antimonotonic** 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.* [11] 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 concept of overlap of instances proposed in [11].

Definition 12. 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, $V_{S_1} \cap V_{S_1} \neq \emptyset$.

Definition 13. 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, $E_{S_1} \cap E_{S_1} \neq \emptyset$.

Definition 14. Given a pattern $P = \langle V_P, E_P, \lambda_P \rangle$ and a graph $G = \langle V_G, E_G, \lambda_G \rangle$, an **overlap graph** is a graph O such that each vertex of O represents an instance of P in G, and two vertices u and v are adjacent if the two instances they represent edge overlap (or vertex overlap).

It is also possible to build an overlap graph showing how occurrences overlap as in [5]. In this article, we mainly study how occurrences overlap and we only consider overlap in vertex. Two existing types of occurrence overlap concepts are given as follows.

Definition 15. A **simple overlap (SO)** of occurrences f_1 and f_2 of pattern P exists if $f_1(V_P) \cap f_2(V_P) \neq \emptyset$.

A variant of the simple overlap called harmful overlap, was introduced in [5].

Definition 16. 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 17. An **independent (vertex) set** is a set of vertices in a graph, no two of which are adjacent.

Definition 18. Given a pattern $P = \langle V_P, E_P, \lambda_P \rangle$ and a dataset graph $G = \langle V_G, E_G, \lambda_G \rangle$, the **maximum independent set based support** is defined as the cadiality of maximum independent vertex set of overlap graph O, that is,

$$\sigma_{MIS}(P, G) = \max\{|I| | I \text{ is an independent set of } O\}$$

The main drawback of the MIS support is computing efficiency - it is shown [19] that maximum independent set problem is NP-hard. Because MIS measure is based on overlap graph, where vertices denote instances of pattern in data graph, the total number of instances is proportional to data graph size. Thus computing MIS as a support measure is also NP-hard, that is to say, the time required to solve the problem using any currently known algorithm is exponential to the size of the data graph.[2]

Bringmann and Nijssen [6] 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 1, vertex 1 is an image of any of the nodes v_1, v_2, and v_3 in the pattern.

Definition 19. Assume pattern P has l occurrences in data graph G, and they are denoted as $f_1, f_2, \cdots, f_l$. The **minimum image based (MNI) support** of P in G is defined as

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

In other words, for each node v in pattern P, MNI support measure identifies the count c_v 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 matching patterns. 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 20. For a pattern P, a graph G, and a parameter k, the **minimum k-image based support** is

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

where V is connected subset of V_P and $|V| = k$, and f_i is an occurrence of P in G.

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

[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).

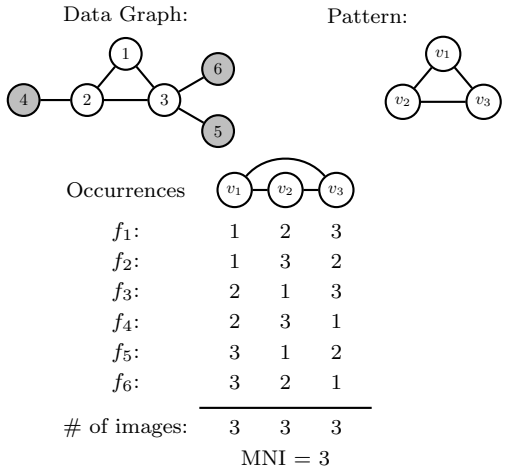

Figure 1: Example showing MNI overestimates the count of patterns. The triangle-shaped pattern has 1 instance but its MNI measure is 3

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 vertex images for every node in a pattern, and finding the minimum number of distinct vertices 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 1: the MNI support of the triangle-shaped pattern is 3, since the minimum number of images of one 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 variations other than the basic forms mentioned in this section. We will introduce some of the variations in Section 5. Here we only emphasize that those variations do not significantly change their features.

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 1: 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.

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.

The following descriptions are based on a data graph G, a pattern P, and the set of l occurrences of pattern P in G denoted as $Occ(P, G) = \{f_i : i = 1, 2, \cdots, l\}$.

Definition 21. A **hypergraph** $H = (V, E)$ consists of a set $V = \{v_1, v_2, \cdots, v_n\}$ of n vertices and a set $E = \{e_1, e_2, \cdots, e_d\}$ of d edges, where each edge is a 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 22. The **dual** H^* of H is a hypergraph whose vertices and edges are interchanged, so that the vertices are given by $\{e_1, e_2, \cdots, e_d\}$ and the edges are given by $X = \{X_1, X_2, \cdots, X_n\}$ where $X_j = \{e_i | v_j \in e_i\}$, 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 measure within the hypergraph framework.

Definition 23. If pattern $P = (V_P, E_P)$ has m instances in data graph G, and the collection of l occurrences is $\{f_i, i = 1, \cdots, l\}$. The **occurrence hypergraph** of P in G is defined as $H^O = (V, E)$ where $V = \cup_{i=1}^{l} f_i(V_P)$ and $E = \{e_i, i = 1, \cdots, l\}$, 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.

Definition 24. If pattern $P = (V_P, E_P)$ has m instances in data graph G, and the collection of instances is $\{S_i = V_{S_i}, E_{S_i}\}, i = 1, \cdots, m$. The **instance hypergraph** of P in G is defined as $H^I = (V, E)$ where $V = \cup_{i=1}^{m} V_{S_i}$ and $E = \{e_i, i = 1, \cdots, m\}$, each $e_i = V_{S_i}$.

Let us use the example shown in Figure 1 to show how the hypergraphs are constructed: the occurrence hypergraph $H^O = (V, E)$ has vertex set $V = \{1, 2, 3\}$ and edge set $E = \{e_1, e_2, e_3, e_4, e_5, e_6\} = \{\{1, 2, 3\}, \{1, 3, 2\}, \{2, 1, 3\}, \{2, 3, 1\}, \{3, 1, 2\}, \{3, 2, 1\}\}$. Similarly, instance hypergraph $H^I = (V, E)$ has vertex set $V = \{1, 2, 3\}$ and edge set $E = \{e\} = \{\{1, 2, 3\}\}$. Note that since the topological structure of pattern P is incorporated into the occurrence and instance hypergraphs, there is an order in vertices contained in each edge. Another example can be found in Figure 2: for both the instance and occurrence hypergraphs, there are 6 edges and 14 vertices as shown in the figure.

The differences between the concepts 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,

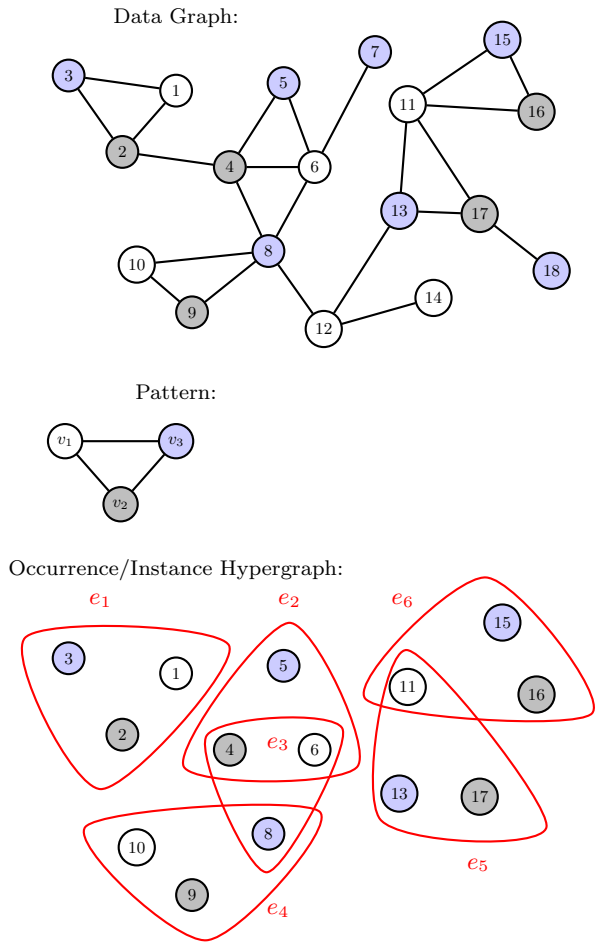

Figure 2: Occurrence/instance hypergraph of a triangular pattern

multiple occurrences project the pattern to the same subgraph of dataset graph. Sometimes, when pattern admits no automorphism, occurrence and instance hypergraphs are quite similar. For example, each instance of the triangle pattern in Figure 2 is associated with one occurrence, hence the number of edges in occurrence hypergraph coincident with that in instance hypergraph.

Judging from the nature of occurrence hypergraphs, as shown in Figure 2, occurrences that are represented by hypergraph edges overlap in various degrees and positions. We argue that a hypergraph framework keeps more such information and offers more insight for further investigation, as compared to overlap graph based support measure such as MIS [11].

Note that the the concept of hypergraph is also used in [18] to define a variant of overlap graph [11]. Given an overlap graph O, if one replaces all cliques in O by hyperedges and deletes non-dominating hyperedges, one can get an overlap hypergraph. In our method, vertices are node images of pattern P, and edges represent occurrences and instances.

In summary, the hypergraph is a suitable topological representation of pattern occurrences (instances) for investigating support measures. We will show that in the remainder of this paper.

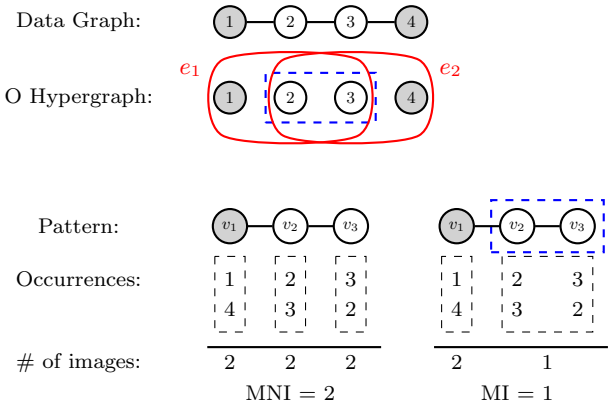

Figure 3: MNI vs MI Support Measure

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 3.

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, apparently the two vertices v_2 and v_3 are symmetric, meaning there is automorphism that maps one to the other. Hence v_2, v_3 can be considered as a group $\{v_2, v_3\}$, which has one image $\{2, 3\}$. This observation leads to the idea of defining a new support measure of a pattern taking advantage of its topological structure and reduce overestimation of MNI.

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

Definition 25. We define **coarse-grained node** W as a subset of V_P. The **coarse-grained node image count** is defined as

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

In Figure 3, if coarse-grained node W is $\{v_2, v_3\}$, then its coarse-grained node image count $c(W) = |\{\{2, 3\}, \{3, 2\}\}| = 1$. However, for $M = \{v_2\}$, since v_2 appeared in two images, we get $c(M) = 2$.

Now we can define new support measure of pattern P using the definition of coarse-grained node image count. Inspired by our observation, the pattern nodes that are symmetric to each other should be affiliated with the same group, hence we definite transitive vertex set as follows.

Definition 26. A pair of vertices u and v in graph G is **transitive** if there is at least one automorphism f of G such that $f(u) = v$.

Definition 27. The **transitive vertex set** T in G is a subset of universal vertex set V such that any pair of vertices in T is transitive.

Definition 28. Given a pattern P, for each pattern node $v \in V_P$, if T is a transitive vertex set in a subgraph of pattern

P, and T contains v, we let v affiliate with T, the collection of such T is $\mathcal{T} = \{T\}$. The **minimum instance based support (MI)** of P in G is defined as

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

As for the example in Figure 3, pattern has coarse-grained nodes $\{v_1\}, \{v_2\}, \{v_3\}$ and $\{v_1, v_2\}$, hence $\sigma_{MI}(P, G) = 1$. Now let us study the main properties of the MI support.

THEOREM 1. *The minimum instance based support measure is anti-monotonic.*

PROOF. The basic idea is: given pattern p and its superpattern P. Any coarse-grained node T considered in $\sigma_{MI}(p, G) = \min_{T \in \mathcal{T}_v} c(T)$ is also considered in $\sigma_{MI}(P, G) = \min_{T \in \mathcal{T}'_v} c(T)$ and the count $c(T)$ does not increase as pattern p extends to superpattern P, we have $\sigma_{MI}(p, G) \geq \sigma_{MI}(P, G)$. □

THEOREM 2. *The minimum instance based support measure is linear-time computable.*

PROOF. Given $\mathcal{T} = \{T\}$, 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, $\sigma_{MI}(p, G)$ is linear-time computable. □

THEOREM 3. *Given a pattern P and data graph G, and occurrence hypergraph $H^O = (V, E)$, we have*

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

PROOF. Given the occurrence hypergraph $H^O = (V, E)$, the differences between $\sigma_{MI}(P, G)$ and $\sigma_{MNI}(P, G)$ are only in coarse-grained nodes. We have $\sigma_{MNI}(P, G) = \min_{W \in \mathcal{W}_v} c(W)$, where $W = \{v\}$, and $\mathcal{W} = \{W\}$. Since $\forall W$, we have $W \in \mathcal{T}$, it is then easy to see $\sigma_{MI}(P, G) = \min_{W \in \mathcal{T}} c(W) \leq \min_{W \in \mathcal{W}_v} c(W) = \sigma_{MNI}(P, G)$. □

In practice, there will be many cases in which MI measure is strictly smaller than the MNI measure. As in Figure 3, when consider additional coarse-grained nodes and minimum count among all of them will decrease. In such 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 support measure is to achieve reasonable count when overlap causes overestimation by MNI. However, MI cannot handle the type of overlap illustrated in Figure 4. Although the number of independent instances is only 2 (e.g., $\{1, 5\}$ and $\{4, 8\}$ are independent), we still get MNI = 4. Moreover, there are merely three possible coarse-grained nodes $\{v_1\}, \{v_2\}, \{v_1, v_2\}$, their images counts are 4, 4, and 4. Hence MI = 4, any variant of MI will not help either.

In this case, we change our interpretation of edges in occurrence/instance hypergraph to set of coarse-grained nodes. It seems for some data graphs (e.g., Figure 4) the connection among pattern nodes matters, and we opt to treat them as a set, hence we now view an edge in an occurrence/instance hypergraph as a set without distinguishing them by images

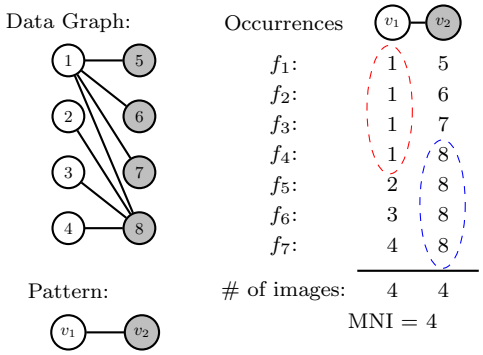

Figure 4: MNI measure can over-estimate count of patterns as it ignores partial overlap.

from different nodes. 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 29. A **vertex cover** of H is a subset of V that intersects with every edge of H. A **minimum vertex cover** is a vertex cover with the smallest cardinality.

In the vertex cover problem, we seek a small number of items that together represent an entire population. Formally, we are given a set of subsets $S = \{s_1, s_2, \cdots, s_m\}$ of the universal set $U = \{1, 2, \cdots, n\}$, and we seek the smallest subset $T \subseteq U$ such that each subset s_i contains at least one element of T.

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 30. Given pattern P in data graph G, and its occurrence hypergraph H^O. Let S_i denote the set of vertex images of each occurrence f_i of pattern p, that is $S_i = \{f_i(v) : f_i \text{ is an occurrence of } P \text{ in } G \text{ and } v \in V_P\}$, the union of them as $U = \cup_{i=1}^{m} S_i$, and a vertex cover $V = \{v_j \subseteq U : S_i \cap V \neq \emptyset, \text{ for any } S_i\}$. The **minimum vertex cover based (MVC) support** of P in G is defined as

$$\sigma_{MVC}(P, G) = \min_{\text{vertex cover } V \subseteq U} |V|.$$

In other words, MVC is defined as the cardinality of a smallest vertex cover set in the occurrence hypergraph of P in G. For example, in Figure 4, 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 4. *The MVC measure is anti-monotonic.*

PROOF. We shall show that if pattern P is a superpattern of pattern p then $\sigma_{MVC}(p, G) \geq \sigma_{MVC}(P, G)$.

Let $\{f_i\}_{i=1}^{l}$ denote the set of all occurrences of pattern P and $S_i = f_i(V_p)$, the union of them as $U = \cup_{i=1}^{l} S_i$, and a minimum vertex cover $V = \{v_j\}_{j=1}^{d}$ with cardinality d. If we can show that a subset W of V is a vertex cover for all occurrences of superpattern P, then obviously $d \geq |W|$. For any occurrence f_i of pattern P in G, there is an occurrence

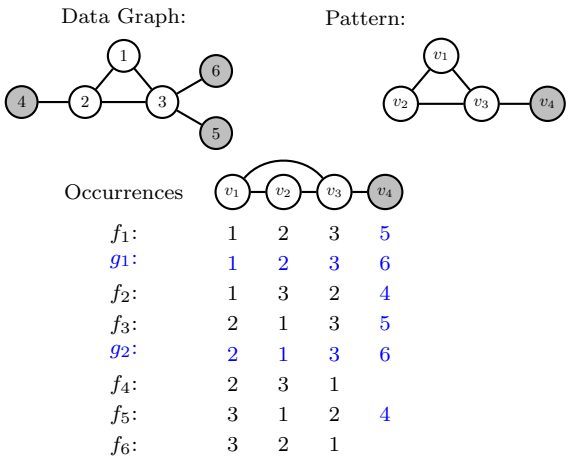

Figure 5: An example showing occurrences of a pattern while being extended to a superpattern

f'_i of pattern p in G, such that $f_i|_p = f'_i$, which implies $f'_i(V_p) \subseteq f_i(V_P)$. If V hits $f_i(V_P)$, it must hit $f'_i(V_P)$. Hence a minimum vertex cover V' for pattern p is still a vertex cover for P. Therefore we have $d \geq \min |W|$, which implies $\sigma_{MVC}(p, G) \geq \sigma_{MVC}(P, G)$. □

Please refer to Figure 5 as an illustrative example of the anti-monotonicity of σ_{MVC}: when the pattern $\{v_1, v_2, v3\}$ is extended to include $\{v_4\}$, the MVC support is still 1.

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

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

PROOF. Since $\sigma_{MI}(P, G) = \min_{T \in \mathcal{T}_v} c(T)$, there must be one coarse-grained node that achieve this minimum count σ_{MI}, denote this coarse-grained node by T, and its images by $\{f_i(T), i = 1, 2, \cdots, l\}$. It is obvious that a minimum vertex cover of $\{f_i(T), i = 1, 2, \cdots, l\}$ is also a vertex cover of all edges in occurrence hypergraph. Hence $\sigma_{MI} = |\{f_i(T), i = 1, 2, \cdots, l\}| \geq$ size of minimum vertex cover of $\{f_i(T), i = 1, 2, \cdots, l\} \geq \sigma_{MVC}(P, G)$. □

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 [20]. 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 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. Intuitively, in the hypergraph setting, MNI support measure itemizes the pattern as individual groups each containing one pattern node. By revisiting the concept of coarse-grained node 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.

If every node v is only affiliated with coarse-grained node $W = \{v\}$, and let $\mathcal{W}$ be a collection of such nodes, we can re-define $\sigma_{MNI}(P, G)$ as

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

Similarly, if every node v is affiliated with coarse-grained node W containing k nodes including v, we can also rewrite $\sigma_{MNI}(P, G, k)$ as

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

The above definitions show connections among $\sigma_{MNI}(P, G)$, $\sigma_{MNI}(P, G, k)$, and the new support measure $\sigma_{MI}(P, G)$. Within the hypergraph setting, edges contain images of coarse-grained nodes, each node has a count of distinct images. Thus, these support measures are all defined as minimum count among coarse grained nodes in the hypergraph.

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

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

where E is an independent edge set of H.

The overlap graph approach is very similar in spirit to how dual hypergraph is built. Edges in instance hypergraph represents 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_P^I are vertices in dual H^*, and if there are only two edges e_i and e_j overlaps at vertex v then there is an edge $X_v = \{e_i, e_j\}$ in dual H^* of H_p^I. 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 [18]. However, instead of instance overlap graphs, we focus on the hypergraph formed by occurrences or instances of pattern, because we believe they contain more information which can be interpreted in a number of different ways in order to find intuitive and computable support measures. As shown in Figure 2, MIES support measure of the triangular pattern is 4, for example $\{e_1, e_2, e_4, e_5\}$ forms an maximum independent edge set. If we construct overlap graph, we should find out the MIS support measure is also 4.

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

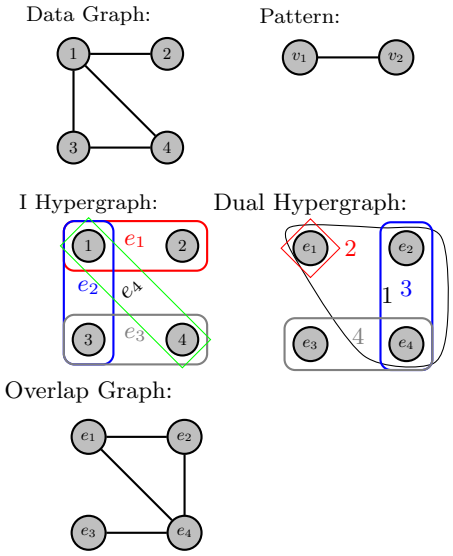

Figure 6: The instance hypergraph and its dual for a small pattern in a data graph

serves as relaxation for reducing computing costs of expensive measures such as MVC and MIES (details in Section 4.3).

Let us start with MIES: 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 for each hyperedge e at least one vertex in it be chosen and the object is to minimize that number of vertices hitting all hyperedges. Now we can write:

$$\min \quad \sum_{v \in V} x(v) \tag{1}$$

$$\sum_{v \in E_i} x(v) \geq 1 \qquad \forall i \tag{2}$$

$$x(v) \in \{0,1\} \qquad \forall v \tag{3}$$

The dual H^* of H is a hypergraph whose vertices and edges are interchanged, so that the vertices are given by $\{e_i\}$ and the edges are given by $\{X_m\}$ where $X_m = \{e_j : v_m \in e_j\}$. Let variable $y(v)$ indicate whether e is in the independent set or not. The constraints state that for each hyperedge 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:

$$\max \quad \sum_{e \in W} y(e) \tag{4}$$

$$\sum_{e \in X_i} y(e) \leq 1 \qquad \forall i \tag{5}$$

$$y(e) \in \{0,1\} \qquad \forall e \tag{6}$$

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

THEOREM 6. *Given a pattern P and data graph G, we have $\sigma_{MIES}(P,G) = \sigma_{MIS}(P,G)$.*

Hypergraph edges	v_1	v_2	v_3
e_1:	1	2	3
e_2:	6	5	4
e_3:	6	4	8
e_4:	10	9	8
e_5:	11	13	17
e_6:	11	16	15
# of images:	4	6	5

MNI = 4

Figure 7: Example shows MNI support measure within hypergraph framework. Data graph and pattern are shown in Figure 2

PROOF. *The problem of finding maximum independent edge set in occurrence hypergraph H^O is equivalent to finding maximum independent vertex set in dual hypergraph H^* with vertices corresponding to edges in H^O 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. (4)), while overlap graph based MIS is equal to the solution of problem: $\max \sum_{e \in W} y(e)$ *subject to* $y(e) + y(e') \leq 1$, $\forall e, e' \in X_i, \forall i$ *and* $y(e) \in \{0,1\}$, $\forall e$. *Figure 6 shows an illustrative example of this.*

We shall show that, if the constraints of the two maximum optimization problems are the same, their solutions shall be equal to each other. Thus we only need to show that the equalities $y(e_1) + y(e_2), + \cdots + y(e_n) \leq 1$ is equivalent to $y(e_i) + y(e_j) \leq 1$ for any $1 \leq i \neq j \leq n$, when every $y(e)$ is restricted to $\{0,1\}$.

It is obvious that $y(e_1) + y(e_2), + \cdots + y(e_n) \leq 1$ implies $y(e_i) + y(e_j) \leq 1$ for any $1 \leq i, j \leq n$ because every $y(e)$ is non-negative. Hence, we need to prove it is also true the other way around.

Assume that $y(e_i) + y(e_j) \leq 1$ for any $1 \leq i \neq j \leq n$ and $y(e)$ is restricted to $\{0,1\}$. Without loss of generality, if there is a e_i such that $y(e_i) = 1$, then for any other e_j we have $y(e_j) = 0, j \neq i$. Hence $y(e_1) + y(e_2), + \cdots + y(e_n) \leq 1$.

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

□

THEOREM 7. *The MIES measure is anti-monotonic.*

PROOF. The proof is obvious since MIES is equivalent to anti-monotonic MIS. □

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 a linear programming problem.

$$\min \quad \sum_{v \in V} x(v) \tag{7}$$

$$\sum_{v \in E_i} x(v) \geq 1 \qquad \forall i \tag{8}$$

$$0 \leq x(v) \leq 1 \qquad \forall v \tag{9}$$

Likewise, we relax the integrability conditions of maximum independent edge set problem to obtain a linear programming problem.

$$\max \quad \sum_{e \in W} y(e) \tag{10}$$

$$\sum_{e \in X_i} y(e) \leq 1 \qquad \forall i \tag{11}$$

$$0 \leq y(e) \leq 1 \qquad \forall e \tag{12}$$

Now we can formally define the relaxed versions of the MVC and MIES measures. We shall also show that they are both anti-monotonic.

Definition 32. The **polynomial-time MVC** support measure of pattern P in graph G is defined as

$$\nu_{MVC} = \min_x \Sigma_v x(v)$$

Definition 33. The **polynomial-time MIES** support measure of pattern P in graph G is defined as

$$\nu_{MIES} = \max_y \Sigma_e y(e)$$

THEOREM 8. *The polynomial-time MVC support measure ν is anti-monotonic.*

PROOF. *We shall show that $\nu(p, G) \geq \nu(P, G)$ for any pattern p and its superpattern P in dataset graph G. Assume that $x* = \Sigma_{v \in K}^{m} x_v^*$ is a solution to the LP (7-9), that is, $\sum_{v \in E_i} x(v) \geq 1$ for any i and $\leq x(v) \leq 1$ for any v. For each edge in the occurrence hypergraph E_i' of P in G, there must be an edge E_i in occurrence hypergraph of p in G such that $E_i \subseteq E_i'$. Therefore $\sum_{v \in E_i} x(v) \geq 1$gives rise to $\sum_{v \in E_i'} x(v) \geq 1$ provided that all x values are non-negative. Finally, because $\nu(P, G)$ is defined as the minimum value of $\sum_{v \in V'} x(v)$ that satisfy all constraints, we reach the conclusion that $\nu(p, G) = x^* \geq \min_x \Sigma_v x(v) = \nu(P, G)$.* □

THEOREM 9. *The linearized maximum independent edge set support measure ν is anti-monotonic.*

PROOF. *The proof is similar to that of Theorem 8. 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 10. *Given a pattern P containing k nodes, data graph G, and occurrence hypergraph $H = (V, E)$, 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, while MVC measure is larger than the MIES (that equals MIS according to Theorem 6), the gap between MVC and MIES/MIS is within a constant factor.

Based on well-established results in linear programing [21], 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 11. *Given a pattern P, data graph G, and occurrence 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 [21]. □

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.

THEOREM 12. *Given a pattern P, data graph G, and occurrence (instance) hypergraph H, if every vertex is contained in at most d edges, then we have*

$$\sigma_{MVC}(P, G) \leq \ln(d + 1)\nu_{MVC}(P, G).$$

PROOF. It is proved in [22] therefore we omit the details here. □

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. 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 3 and 5. Putting all together, we have

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

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

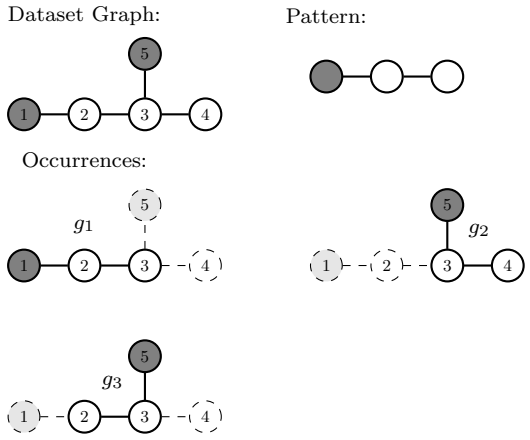

Figure 8: Structural overlap ⇏ Harmful Overlap

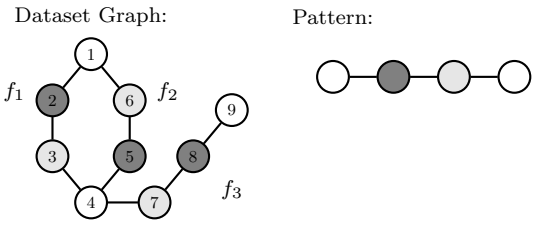

Figure 9: Example for illustrating the relationship among Structural overlap, Harmful Overlap, and Simple Overlap

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.

Here we introduce the concept of **structural overlap** which can be compared with harmful overlap MIS support measure. Then we show how the new structural overlap can be used in the study of support measures.

Definition 34. A **structural overlap** of occurrences f_1 and f_2 of pattern P exists if $\exists\ v, w \in V_P$, satisfying that either v and w are contained in a transitive subgraph of pattern P or $v = w$, such that $f_1(v) = f_2(w) \in f_1(V_P) \cap f_2(V_P)$.

Note that structural overlap is different from harmful overlap, illustration examples are given in Figures 8 and 9. In Figure 8, occurrences g_1 and g_2 are structural overlap but not harmful overlap, and occurrences g_1 and g_3 are both structural and harmful overlap. In Figure 9, the two occurrences f_1 and f_2 are harmful overlap but not structural overlap, f_2 and f_3 are neither structural nor harmful overlap but simple overlap. We can also find out that f_1 and f_3 are overlap in sense of simple, structural and harmful overlap.

Figure 9 explains how harmful and structural overlap are different from simple overlap. Even though the occurrences f_1 and f_2 overlap on the vertices 4, Fiedler and Borgelt [5] argue that they do not have an occurrence of a graph with a single vertex as a common ancestor. In other words, they are not occurrences of the same vertex in the given pattern. Therefore the two occurrences cannot be constructed from the same occurrence of a single vertex, which is then extended in corresponding ways, and the two occurrences have to be built from two different occurrences of a single vertex. Hence the two occurrences are not harmful overlap. A similar logic applies to our structural overlap here. Although they share the same vertex 4, because the two occurrences of vertex 4 have different ancestors that are not topological identical, the two occurrences of vertex 4 serve different role in the given subgraph. As a consequence, there were two occurrences for any ancestors of these occurrences and thus the support has always been 2, that is to say, they are not considered overlap in structural overlap sense either.

An apparent difference between structural overlap and harmful overlap can also be shown in Fugure 9, in which two occurrences f_1 and f_2 are considered harmful overlap but not structural overlap. That is because harmful overlap allows overlap between occurrences of various ancestors without fully consider they topological properties. While structural overlap addresses overlap of occurrences of structural identical vertices, i.e. vertices in a transitive subgraph of the given pattern. In this sense, structural overlap better explains for the topological structure of pattern occurrences.

Moreover, the concept of structural overlap originally came from MI support measure. By observation that MNI does not recognize symmetric nodes in pattern, we group nodes that are symmetric to each other together so that the group images can reflect occurrence overlap.

The concept of structural overlap can be used in various ways to help find 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 overlapping graph. The resulted overlap graph that is sparser than the one generated from simple overlap. Consequently, user can use MIS, MCP, other measures to obtain count of pattern occurrences (instances). In the hypergraph setting, besides its close connection to MI support measure, structural overlap can also be used to find out if edges overlap.

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 [23], Yan and Han [13, 14], Inokuchi *et al.* [24], Hong *et al.* [26], Huan *et al.* [27], Kuramochi and Karypis [25]. Although FSM in a single large graph setting has been studied (e.g., Kuramochi and Karypis [29, 12], Elseidy *et al.* [4]), it receives less attention. The reason for that 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 sup-

port measure, which was first introduced in Vanetik [11] and its formal definitions were given in Vanetik *et al.* [30] together with proofs for the sufficient and necessary conditions required for overlap graph based measure to be anti-monotonic. Several variations and extensions of the MIS measure were also proposed and analyzed. Those include exact and approximate MIS measures presented by Kuramochi and Karypis [12], and overlap graph based MCP by Calders *et al.* [31]. In [31], the authors also propose the Lovasz measure by using the theta function that is proven to be bounded between MIS and MCP. This is very similar in nature to another measure named *Schrijver* graph measure [32]. A relaxation of overlap graph based MIS is given in Wang *et al.* [18].

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 algebraic 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.

7. REFERENCES

[1] M. Berlingerio. Mining graph evolution rules. IN *ECML PKDD*, pages 115–130, 2009.

[2] B. Bringmann and S. Nijssen. What is frequent in a single graph?. In *PAKDD*, pages 858–863, 2008.

[3] Calders, Toon, Jan Ramon, and Dries Van Dyck. Anti-monotonic overlap-graph support measures. IN *ICDM* 2008.

[4] M. Elseidy. Grami: Frequent subgraph and pattern mining in a single large graph. In *VLDB*, pages 517–528, 2014.

[5] M. Fiedler and C. Borgelt. Support computation for mining frequent subgraphs in a single graph. In International Workshop on Mining and Learning with Graphs (MLG), 2007.

[6] L. Galarraga, C. Teflioudi, K. Hose, and F. M.Suchanek. Amie: Association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22th international conference on World Wide Web*, WWW '13, 2013.

[7] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *PKDD*, pages 13–23, 2000.

[8] A. Inokuchi and T. Washio. A Fast Method to Mine Frequent Subsequences from Graph Sequence Data. in *ICDM*, pages 303-312, 2008

[9] G. Jeh and J. Widom. Mining the space of graph properties. In *KDD*, pages 187–196, 2004.

[10] M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. In *TKDE*, pages 1038–1051, 2004.

[11] Vanetik, N., Gudes, E., Shimony, S.E.: Computing frequent graph patterns from semistructured data. In *ICDM* 2002, pp.458-465 (2002)

[12] M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph. In *DMKD*, pages 243–271, 2005.

[13] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *ICDM*, pages 721–724, 2002.

[14] X. Yan and J. Han. Closegraph: mining closed frequent graph patterns. In *KDD*, pages 286–295, 2003.

[15] Moens, Marie-Francine, Juanzi Li, and Tat-Seng Chua, eds. Mining user generated content. CRC Press, 2014.

[16] Dehmer, Matthias, and Frank Emmert-Streib, eds. Quantitative Graph Theory: Mathematical Foundations and Applications. CRC Press, 2014.

[17] Cheng, Hong, Xifeng Yan, and Jiawei Han. Mining graph patterns. Frequent Pattern Mining. Springer International Publishing, 2014. 307-338.

[18] Wang, Yuyi, and Jan Ramon. An efficiently computable support measure for frequent subgraph pattern mining. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2012.

[19] Karp, Richard M. "Reducibility among combinatorial problems." Complexity of computer computations. springer US, 1972. 85-103.

[20] E. Halperin, Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs, SIAM J. Comput. 31 (5) (2002)

[21] Pach, János, and Pankaj K. Agarwal. Combinatorial geometry. Vol. 37. John Wiley & Sons, 2011.

[22] Lovász, László. "On the ratio of optimal integral and fractional covers." Discrete mathematics 13.4 (1975): 383-390.

[23] Borgelt, Christian, and Michael R. Berthold. "Mining molecular fragments: Finding relevant substructures of molecules." Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on.

IEEE, 2002.

[24] Inokuchi, Akihiro, Takashi Washio, and Hiroshi Motoda. "Complete mining of frequent patterns from graphs: Mining graph data." Machine Learning 50.3 (2003): 321-354.

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

[26] Hong, Mingsheng, et al.. An efficient algorithm of frequent connected subgraph extraction. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, 2003.

[27] Huan, Jun, Wei Wang, and Jan Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. *Proc. IEEE International Conference on Data Mining*, 2003.

[28] Khan, Arijit, Xifeng Yan, and Kun-Lung Wu. "Towards proximity pattern mining in large graphs." *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010.

[29] Kuramochi, Michihiro, and George Karypis. Grew-a scalable frequent subgraph discovery algorithm. *Proc. Fourth IEEE International Conference on Data Mining*, 2004.

[30] Vanetik, Natalia, Solomon Eyal Shimony, and Ehud Gudes. Support measures for graph data. *Data Mining and Knowledge Discovery* 13:2 (2006): 243-260.

[31] Calders, Toon, Jan Ramon, and Dries Van Dyck. Anti-monotonic overlap-graph support measures. *2008 Eighth IEEE International Conference on Data Mining* 2008.

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