Chapter 10

Single-Graph Support Measures

Toon Calders, Jan Ramon, and Dries Van Dyck

Contents

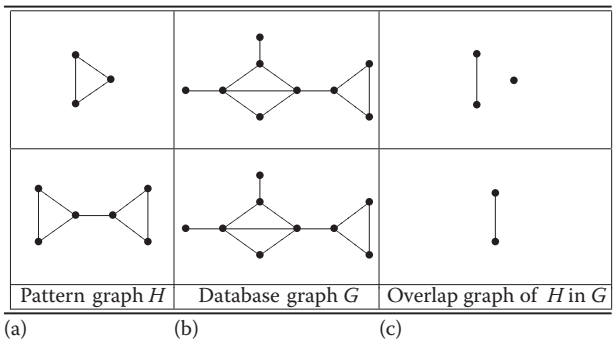
10.1 Introduction

Over the last several years, graph mining has emerged as a new field within contemporary data mining. One of the central tasks is the search for subgraphs, called *patterns*, that occur frequently in either a collection of graphs (e.g., databases of molecules [6], game positions [15], scene descriptions) or in a single large graph

(e.g., the Internet, citation networks [16], social networks [12], protein interaction networks [10]). In the literature, the terms *frequency* and *support* have been used interchangeably to denote the measure to quantify the prevalence of a pattern. In the single-graph setting, however, the notion of frequency is not at all straightforward to define. For example, the obvious definition of taking the number of instances of a pattern as its frequency has the undesirable property that extending the pattern (i.e., making it more restrictive) may actually increase its frequency. Indeed, consider for instance, the unlabeled k-clique K_k as the single graph in which we want to find patterns. There are $\binom{k}{2}$ different embeddings under subgraph isomorphism of the unlabeled path of length 1 in K_k, whereas there are $3\binom{k}{3}$ embeddings of the path of length 2 in K_k. In fact, the number of different embeddings may increase exponentially in the size of the pattern. Hence, as pointed out by Vanetik et al. [17], a good frequency measure must be such that the frequency of a superpattern is always at most as high as that of a subpattern. This property is called the *antimonotonicity*. Also, for reasons of efficiency, antimonotonicity of the frequency measure is highly desirable, as it allows for pruning large parts of the search space in a general-to-specific exploration. The efficiency and correctness of most existing graph pattern miners relies critically on the antimonotonicity of the frequency measure being used.

In this chapter, we give an overview of measures that have been defined in the literature for assessing the frequency of graph patterns in one large graph. We divide the measures into two groups: the ones that are based on the notion of the so-called overlap graph and those that are not. An overlap graph of a pattern graph in a single data graph is itself a graph again that expresses how the different occurrences of the pattern are connected to each other in the data graph. Every node in the overlap graph denotes an occurrence of the pattern and two nodes are connected by an edge if the corresponding occurrences of the pattern graph have an overlap. In Figure 10.1, two examples of the overlap graph of a pattern in another graph have been given.

In Section 10.2, we formally define important notions such as embedding, overlap graphs, and antimonotonic support measures. Then, for reasons of completeness, we start our discussion of graph-support measures with nonoverlap-graph-based mea-



AQ1 **FIGURE 10.1:** Two examples of an overlap graph of H in G.

sures in Section 10.3, although in the rest of the chapter, we will mainly concentrate on the class of overlap-graph-based measures. This important class of graph measures is then introduced in Section 10.4. In this section, we survey important results connecting the antimonotonicity of the support-graph-based measure directly to properties of the overlap graph itself.

In Section 10.5, the results are extended to overlap *hypergraphs* that are able to express the way instances or embeddings overlap in a much more subtle and exact way. The alternative characterization and bounding theorems of the overlap-graph-based support measures are extended to this more fine-grained setting.

Section 10.6 describes an important application of the study of support measures: statistical analysis on graph datasets. Statistical theory often assumes that the objects over which summary statistics are computed are drawn independently. In networked data, however, different occurrences of subgraphs in the single-graph settings are dependent. Therefore, in Section 10.6, we relate the statistical power of a set of observations to its s-measure. In particular, if a pattern has a number of overlapping embeddings in a database graph and every embedding has some properties, one can estimate the distribution of these properties from a sample. We are interested in bounding the variance of such estimates.

Finally, Section 10.7 concludes the chapter.

10.2 Preliminaries

In this section, we introduce important graph-related notions such as graph isomorphisms and embeddings as well as antimonotonicity of graph-support measures. We assume that the reader is familiar with basic graph theoretic notions and with computational complexity. Textbooks in these areas, such as [7] and [14], supply the necessary background.

10.2.1 Graphs

A graph $G = (V, E)$ is a pair in which V is a (nonempty) set of *vertices* or *nodes* and E is either a set of *edges* $E \subseteq \{\{v, w\} \mid v, w \in V, v \neq w\}$ or a set of *arcs* $E \subseteq \{(v, w) \mid v, w \in V, v \neq w\}$. In the latter case, we call the graph *directed*. A *labeled* graph with labels from Σ is a triple $G = (V, E, \lambda)$, with (V, E) a graph, and λ a function $V \to \Sigma$ assigning labels to the vertices. We will use the notation $V(G)$, $E(G)$, and λ_G to refer to the set of vertices, the set of arcs (edges), and the labeling function of a graph G, respectively. Unless explicitly stated otherwise, we will assume to be working over undirected labeled graphs in this chapter. By $\mathcal{G}_\lambda$ and $\mathcal{G}$, we denote, respectively, the set of all labeled graphs and the set of all unlabeled graphs.

A graph $G = (V, E, \lambda)$ is said to be a subgraph of graph $H = (V_H, E_H, \lambda_H)$, denoted $G \subseteq H$, if $V \subseteq V_H$, $E \subseteq E_H$, and $\lambda = \lambda_H|_V$.

For $G \in \mathcal{G}_\lambda$,

$$\overline{G} := (V(G), \{\{v, w\} \mid v, w \in V\} \setminus E(G), \lambda_G)$$

denotes the complement graph of G. By $K_k \in \mathcal{G}$, we denote the *complete graph* on k vertices, that is,

$$K_k := (\{v_1, \ldots, v_k\}, \{\{v_i, v_j\} \mid 1 \leq i \neq j \leq k\}).$$

A subgraph $K \subseteq G$ on k vertices for which all vertices are adjacent to all other vertices is called a k-clique. A *cycle* of length k is a connected subgraph on k vertices each of which is incident with exactly two edges.

An undirected unlabeled *hypergraph* is a pair (V, E) where V is a set of vertices and $E \subseteq 2^V$ is a set of (hyper)edges, each of which is a subset of the set of vertices. We denote the set of all (undirected, unlabeled) hypergraphs with $\mathcal{H}$.

10.2.2 Isomorphisms

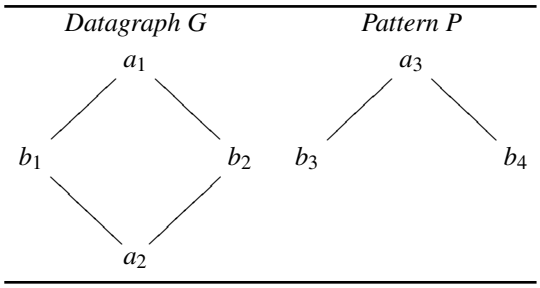
The following concepts introduced in terms of $\mathcal{G}_\lambda$ are also valid for directed and/or unlabeled graphs by adding the direction of the edges and/or dropping the labels of the vertices. For a complete set of definitions of all cases, we refer the interested reader to [4].

A *homomorphism* π from $H = (V_H, E_H, \lambda_H)$ to $G = (V, E, \lambda)$ is a mapping from $V_H \to V$, such that $\forall\{v, w\} \in E_H : \{\pi(v), \pi(w)\} \in E$. We say that H is homomorphic to G.

An *isomorphism* from H to G is a bijective homomorphism π from H to G. In that case, we say that H is isomorphic to G and write $H \cong G$. We use $H \subseteq G$ to denote that $H \cong g$, for some subgraph g of G.

By an *instance* of P in G, we refer to a subgraph g of G such that P and g are isomorphic. Any isomorphism π between P and one of its instances g is called an *embedding*. We denote the set of all instances of a pattern P in the graph G by $\mathrm{Img}(P, G)$, and the set of all embeddings by $\mathrm{Emb}(P, G)$. Notice that the number of instances does not necessarily equal the number of embeddings of P into G, as some embeddings may have the same image.

Example 10.2.1 *Consider the following datagraph G and pattern graph P. The subscripts in the labels have been added for ease of reference only. For instance, the nodes with label a have been annotated* $a_1, a_2, \ldots$



In this example, P has two instances in G (λ denotes the labeling function of G):

$$\left(\{a_1, b_1, b_2\}, \{\{a_1, b_1\}, \{a_1, b_2\}, \{b_1, b_2\}\}, \lambda|_{\{a_1,b_1,b_2\}}\right) \text{ and}$$
$$\left(\{a_3, b_1, b_2\}, \{\{a_3, b_1\}, \{a_3, b_2\}, \{b_1, b_2\}\}, \lambda|_{\{a_3,b_1,b_2\}}\right)$$

but the number of embeddings of P in G is 4*:*

$$\begin{Bmatrix} a_3 \mapsto a_1 \\ b_3 \mapsto b_1 \\ b_4 \mapsto b_2 \end{Bmatrix} \quad \begin{Bmatrix} a_3 \mapsto a_2 \\ b_3 \mapsto b_1 \\ b_4 \mapsto b_2 \end{Bmatrix} \quad \begin{Bmatrix} a_3 \mapsto a_1 \\ b_3 \mapsto b_2 \\ b_4 \mapsto b_1 \end{Bmatrix} \quad \begin{Bmatrix} a_3 \mapsto a_2 \\ b_3 \mapsto b_2 \\ b_4 \mapsto b_1 \end{Bmatrix}$$

For a more complete treatment including homeomorphisms and the extension of the notion of an instance to all morphism types, unlabeled, and directed graphs, we refer the reader to [4].

10.2.3 Support Measures

One of the key elements in a graph mining algorithm is the support measure; that is, a measure expressing the prevalence of a pattern graph in a larger database graph:

Definition 10.2.1 *A* support measure *on $\mathcal{G}$ is a function $f : \mathcal{G} \times \mathcal{G} \to \mathbb{N}$ that maps (P, G) to $f(P, G)$ where P is called the pattern, G is called the database graph, and $f(P, G)$ is called the* support of *P* in *G.*

For efficiency reasons, most graph mining algorithms use a level-wise or depth-first approach to generate frequent patterns, expanding smaller patterns to larger ones. Such an approach requires the support measure being antimonotonic in order to prune efficiently:

Definition 10.2.2 *A support measure f on is* antimonotonic *if for all patterns p and P and database graph G it holds that if $p \subseteq P$, then $f(P, G) \leq f(p, G)$. That is, the support in a graph G does not increase from a subpattern p to a superpattern P.*

In the two next sections, we will see multiple examples of graph-support measures, many of which are antimonotone.

10.3 Nonoverlap-Graph-Based Measures

The first type of single-graph-support measures we consider are those that are not based upon the overlap graph. The advantage of these measures is that they do not require the costly step of building up the overlap graph. All results have been stated in function of isomorphisms but can be extended easily to other morphism types.

10.3.1 Key-Based Support Measures

One approach that is commonly used is to select a fixed key pattern *K* consisting of a number of isolated vertices and to consider as pattern language the space of all

superpatterns of K. This support measure is one of the first ones that was considered in relational learning and is related to the *learning from entailment* setting in the field of inductive logic programming [5]. The K-support of a pattern P in graph G is only defined if K is a subpattern of P and is defined as

$$keycount_E(P, G) = |\{\pi|_K \mid \pi \in \mathrm{Emb}(P, G)\}|$$

where $\pi|_K$ is the restriction of the mapping to K. Clearly, *keycount* is antimonotonic. Indeed, if $p \subseteq P$ and $\pi \in \mathrm{Emb}(P, G)$, then $\pi|_p \in \mathrm{Emb}(p, G)$. Furthermore, if $\pi_1, \pi_2 \in \mathrm{Emb}(P, G)$, and $\pi_1|_p = \pi_2|_p$, then also $\pi_1|_K = \pi_2|_K$. Therefore, $keycount_E(P, G) \leq keycount_E(p, G)$.

The same holds for the image-based version of this support measure:

$$keycount_I(P, G) = |\{\pi(K) \mid \pi \in \mathrm{Emb}(P, G)\}|$$

10.3.2 Minimal Image Count Support Measure

In [3], the authors proposed an antimonotonic support measure named *min-image-based support*. For notational consistency, we give a slightly alternative definition.

$$minImage(P, G) = \min_{v \in V(P)} |\{\pi(v) \mid \pi \in \mathrm{Emb}(P, G)\}| \tag{10.1}$$

This support counts for every vertex, the number of nodes in the data graph to which the vertex can be mapped in an embedding. The measure is the minimum of this number over all vertices of the pattern graph. The antimonotonicity of this support is obvious, and it can be computed very efficiently. It has, however, several drawbacks as we demonstrate next.

First, from a statistical point of view, *minImage* overestimates the evidence. In particular, as Figure 10.2 shows, a vertex can be counted arbitrarily many times.

Second, *minImage* is not additive. Given a subgraph pattern P, if a database graph G has n ($n \geq 2$) connected components, that is, $G = \bigcup_{1 \leq i \leq n} G_i$, then $minImage(P, G) \geq \sum_{1 \leq i \leq n} minImage(P, G_i)$. For many realistic database graphs, strict inequality holds. In this case, it is unclear how much a connected component contributes to the whole support. Figure 10.3 shows an example.

10.4 Overlap-Graph-Based Measures

In this section, we give an overview of the most important results class of the single-graph-support measures that are the key focus of this chapter: the overlap-graph-based measures. First, we introduce the notion of an overlap graph. Then different measures based on the overlap graph and the important characterization of the monotone overlap-graph-based measures by Vanetik et al. [17] and its extension by Calders et al. [4] are discussed. After that, we extend to overlap-hypergraph measures that capture more subtle differences in how instances overlap than plain overlap graphs.

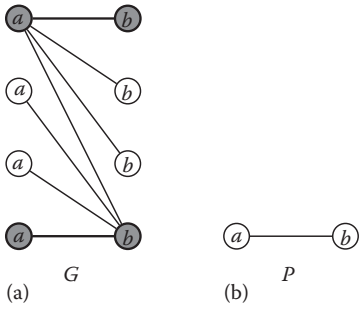


FIGURE 10.2: (a) Database graph G contains two independent images of (b) the subgraph pattern P. However, $minImage(P, G) = 4$ (and we can make this value arbitrarily large by adding more vertices with label b (resp. a) and link them to the top-left vertex with label a (resp. bottom-right vertex with label b). As a consequence, if we remove just a single vertex (the top-left or bottom-right one) the support of the pattern in the network can suddenly drop to one.

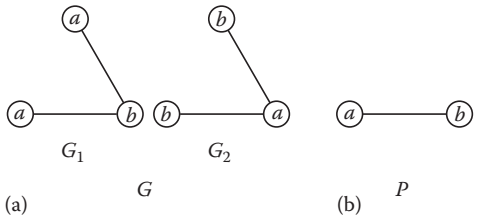


FIGURE 10.3: A database graph G has two connected components G_1 and G_2. $minImage(P, G_1) = minImage(P, G_2) = 1$, but $minImage(P, G) = 3 > minImage(P, G_1) + minImage(P, G_2)$.

10.4.1 Pairwise Overlap Graph

An important class of antimonotonic measures are the ones that are based on the notion of an *overlap* graph G_P [11,17].* An overlap graph summarizes not only the images of the pattern in the database graph but also how they overlap:

Definition 10.4.1 *Let $G \in \mathcal{G}$ be a database graph, P a pattern, and $g_1, g_2 \in Img(P, G)$ be two instances of P. g_1 and g_2 of G have a* vertex overlap *if $V(g_1) \cap V(g_2) \neq \emptyset$ and an* edge overlap *if $E(g_1) \cap E(g_2) \neq \emptyset$.*

For clarity of presentation, we will restrict ourselves to vertex overlap and isomorphic embeddings in this survey, but as shown in [4], all notions and results can be extended to other graph classes and overlap and morphism types.

* Vanetik et al. [17] uses the term *instance* graph instead of overlap graph. The term *instance* suggests the use of isomorphisms, and we consider support measures based on any kind of morphism. Therefore, we follow the terminology of [11] to avoid confusion.

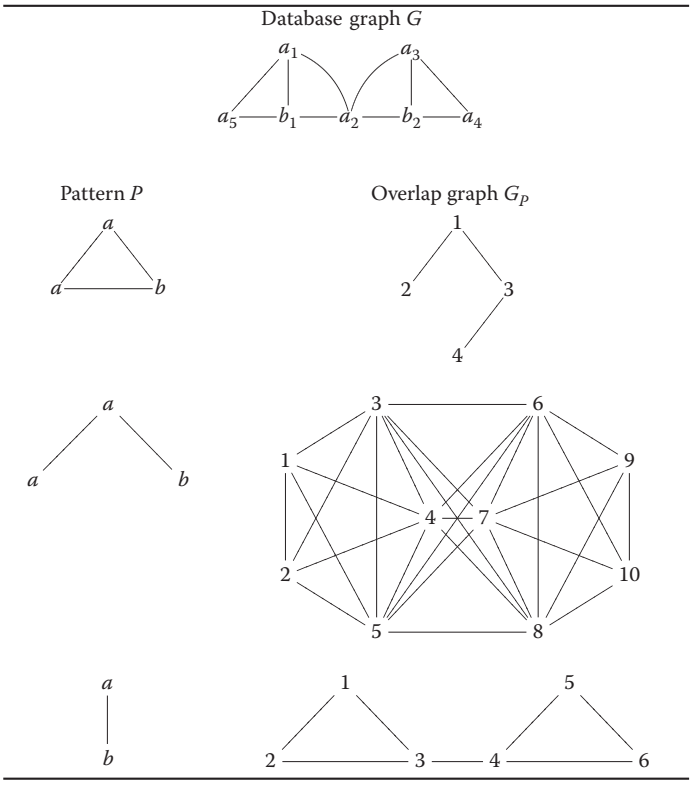


FIGURE 10.4: Database graph G, and three patterns, with their corresponding pairwise overlap graphs G_P.

Definition 10.4.2 *The* pairwise overlap graph *(POG)* G_P *of a pattern* P *in the database graph* G *is an undirected, unlabeled graph in which each vertex corresponds to an instance of the pattern* P*. Two vertices are adjacent in* G_P *if the corresponding embeddings overlap.*

Figure 10.4 gives examples of POGs.

10.4.2 Overlap-Graph-Based Support Measure

We are now ready to define a POG-based support measure.

Definition 10.4.3 *A support measure* f *on graphs is a* pairwise overlap graph-based support measure *if there exists a graph measure* $\hat{f}$ *such that* $\forall P, G \in \mathcal{G} : f(P, G) = \hat{f}(G_P)$.

Informally, a POG-based support measure is a support measure that only depends on the POG. Consider, for example, the following measure based on the *maximal*

independent set (MIS) of the POG. An *independent set* of a graph G is a subset I of $V(G)$ such that $\forall v, w \in I : \{v, w\} \notin E(G)$. A *MIS* of G is an independent set of maximal cardinality and its size is notated as $mis(G)$. The MIS-based POG support measure assigns to every pattern P the size of the MIS [17] of its POG G_P:

$$MIS(P, G) := mis(G_P).$$

Notice that $MIS(P, G)$ can intuitively be interpreted as the maximal number of instances that fit in G without overlap. This measure is antimonotonic.

Example 10.4.1 *Consider the example given in Figure 10.4. The POG of the triangular pattern in the data graph G consists in one path of length* 3*. The MISs in the POG in Figure 10.4 are* $\{2, 3\}$ *and* $\{1, 4\}$*. Hence,* $MIS(P, G) = 2$*.*

10.4.3 Alternative Characterization for Antimonotonicity

Vanetik et al. [17] have shown an alternative characterization of antimonotone POG-based measures. They consider three operations on the overlap graph G_P: clique contraction, edge removal, and vertex addition, as defined in the following.

Definition 10.4.4 *Let* $K \subseteq G$ *be a clique in* $G = (V, E)$*. The* clique contraction $\mathsf{CC}(G, K)$ *yields a new graph* $G' = (V', E')$ *in which the subgraph* $K \subseteq G$ *is replaced by a new vertex* $k \notin V$ *adjacent to* $\{w \mid \forall v \in V(K) : \{v, w\} \in E\}$*:*

$$V' = V \setminus V(K) \cup \{k\}$$
$$E' = E \setminus \{\{v, w\} \mid \{v, w\} \cap V(K) \neq \emptyset\} \cup \{\{k, w\} \mid \forall v' \in V(K) : \{v', w\} \in E\}.$$

The edge removal $\mathsf{ER}(G, e)$ *of the edge* $e = \{v, w\}$ *in the graph* $G = (V, E)$ *yields a new graph* $G' = (V, E \setminus \{\{v, w\}\})$*.*

The vertex addition $\mathsf{VA}(G, v)$ *of the vertex* $v \notin V$ *in the graph* $G = (V, E)$ *yields a new graph* $G' = (V \cup \{v\}, E \cup \{\{v, w\} \mid w \in V\})$*.*

The rationale behind these operations is that the overlap graph of a pattern P can be transformed into the overlap graph of a subpattern p of P by means of these operations.

Property 10.4.1 (Vanetik, Shimony, and Gudes)*. Let G be a database graph,* $p \subseteq P$ *two patterns.* G_P *can be transformed into* G_p *with a sequence of* CC*,* VA*, and* ER *operations.*

Example 10.4.2 *In the POGs in Figure 10.1, we can transform the overlap graph of the third pattern (consisting of one edge between a node labeled a and a node labeled b) to the POG of the second pattern by a series on node additions and edge removals. These two operations together make it possible to transform a graph into any of its supergraphs. From the second overlap graph to the first, we need a series of clique contractions. We could contract subsequently* $\{1, 2\}$*,* $\{3, 4, 5\}$*,* $\{6, 7, 8\}$*, and* $\{9, 10\}$*.*

A direct result of this property is the following theorem of Vanetik et al. [17] that restates the antimonotonicity of f in function of $\hat{f}$ being nondecreasing in function of the three operations specified earlier.

Theorem 10.4.1 (Vanetik, Shimony, and Gudes). *Any overlap-graph-based support measure f is antimonotonic if and only if the associated graph measure $\hat{f}$ is nondecreasing under clique contraction, edge removal, and vertex addition.*

The proof of sufficiency, that is, that any overlap support measure f is antimonotonic if the associated graph measure $\hat{f}$ is nondecreasing under CC, VA, and ER follows immediately from the fact that G_P can be transformed into G_p by these operations.

To prove necessity, Vanetik, Shimony, and Gudes construct for every unlabeled graph H and every operation o a triple (P, p, G), where P is a superpattern, p a subpattern, and G a database graph such that $G_P \cong H$ and $G_p \cong o(H)$. Henceforth, if f would be increasing under some $o \in \{\mathsf{CC}, \mathsf{ER}, \mathsf{VA}\}$, then there would be a H such that $f(H) > f(o(H))$ and one could construct a G, P, and p such that $f(G, P) > f(G, p)$, which would mean that f is not antimonotonic.

10.4.4 Bounding Theorem

The result of Vanetik et al. [17] was later extended by Calders et al. [4] to all combinations of iso-, homo-, and homeomorphisms; edge/vertex-overlap graphs; directed/undirected; and labeled/unlabeled graphs. An important consequence of the alternative characterization of the antimonotonicity of f in terms of $\hat{f}$ being nondecreasing is the bounding theorem proven by Calders et al. [4]. This theorem states that the different antimonotone measures are bounded by a natural minimal and maximal support measure; every *normalized* overlap support measure will always be between these two extremes. A normalized support measure is defined as follows.

Definition 10.4.5 *Let G be an undirected graph and $\overline{K_k}$ the graph composed of k isolated vertices.*

We call an overlap support measure f normalized *if it is antimonotonic and assigns the frequency k to k nonoverlapping images, that is, $\hat{f}(\overline{K_k}) = k$.*

Before we state the bounding theorem, we first introduce the minimum clique partition (MCP) measure.

10.4.4.1 MCP Measure

The first antimonotonic, normalized overlap-graph-based support measure was the MIS measure *MIS*. The *MIS* measure is defined as the size of the MIS of the overlap graph and was introduced and proven to be antimonotonic in [17]. A more compact proof of the antimonotonicity can be found in [8]. This measure was shown to be a lower bound on all normalized, antimonotonic overlap-graph-based measures. Later on, Calders et al. [4] introduced two more normalized antimonotone

overlap-graph-based measures, being MCP and the Schrijver measure. We will review the Schrijver measure in Section 10.4.5.

The MCP measure is inspired by the CC-operation:

Definition 10.4.6 *A* clique partition *of an undirected graph G is a partitioning of $V(G)$ into $\{V_1, \ldots, V_k\}$ such that each V_i induces a complete graph in G. A* MCP *is a clique partition of minimum cardinality. Its cardinality is denoted $mcp(G)$.*

The MCP measure *is defined by $MCP(P, G) : (P, G) \mapsto mcp(G_P)$.*

Theorem 10.4.2 *[4] The MCP measure is an antimonotonic and normalized.*

It is interesting to compare *MCP* with *MIS*. Let $\chi(G)$ be the *chromatic number* of G, that is, the minimal number of colors needed to color the vertices of G such that no two vertices with the same color are adjacent, and let $\omega(G)$ be the *clique number*, the size of the largest clique in G.

First, it is known that $mcp(G) = \chi(\overline{G})$ and $mis(G) = \omega(\overline{G})$ (see, e.g., [9], Section 5.5.1). Consequently, $mcp(G) \geq mis(G)$, for all undirected graphs G, since the size of a maximum clique is a lower bound for the chromatic number.

Informally, it is easy to see why this is so: let $\{V_1, \ldots, V_k\}$ be an MCP and I a MIS for G. We know that I contains at most one vertex v_i of each V_i, $1 \leq i \leq k$. In other words, to decide whether we can include a image of V_i, *MIS* forces us to choose either no image or exactly one image v_i, which must be independent of all chosen $v_j \in V_j$. *MCP*, however, allows us to count a image in V_i as soon as there is *a* image in V_i, which does not overlap with *a* image in V_j. That is, we can make another choice for each (V_i, V_j) pair.

Example 10.4.3 *Let us look at an example: consider pattern P and the graph G as shown in Figure 10.5. The 5 images of P are the induced subgraphs of the database with, respectively, the nodes $\{a, b, c, d, e\}$, $\{i, f, d, k, l\}$, $\{i, f, g, k, l\}$, $\{e, h, j, m, l\}$, and $\{g, h, j, m, l\}$. The POG G_P of P in G is shown on the right in Figure 10.5 and is isomorphic to a pentagon. The white vertices mark the MIS $\{2, 5\}$ and the dashed ellipses*

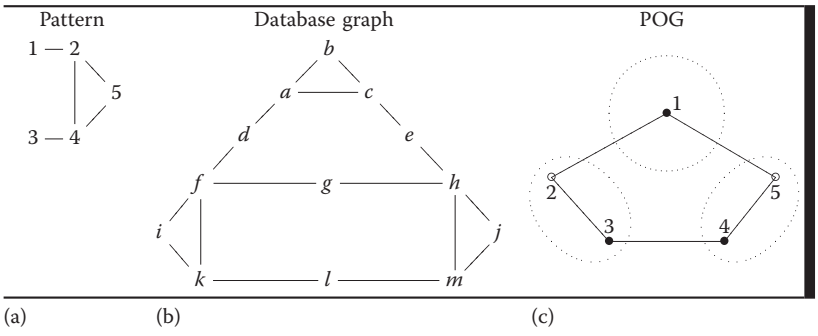


FIGURE 10.5: (a) A pattern P and (b) a graph G. (c) Overlap graph G_P with a *MCP* (dashed ellipses) and a *MIS* (white vertices).

mark a MCP consisting of the three cliques $\{1\}$, $\{2,3\}$, *and* $\{4,5\}$ *of* G_P. *Hence, if we count image 2 with MIS, we can only take image 4 or image 5 as second independent image, because 1 and 3 overlap, leading to a MIS support of* 2. *This is a bit unnatural, because each of the* 3 *images of the triangle can be extended to a image of P in a way that they do not overlap with each other, which would lead to a support of* 3 *of P.*

This more natural notion of counting independent images is exactly what MCP support allows us to do: we do not count individual images, but groups of images of P sharing a image of a subpattern p (a triangle), and allow to "switch" images to decide whether a group is independent of another. In this example, the group $\{1\}$ *is independent of the groups* $\{2,3\}$ *and* $\{4,5\}$, *because it does not overlap with image* 3 *(respectively image* 4*) and the group* $\{2,3\}$ *is independent of the group* $\{4,5\}$ *because, for instance, image* 2 *and image* 5 *do not overlap.*

10.4.4.2 Theorem

Interestingly, *MIS* and *MCP* turn out to be the minimal and the maximal possible normalized overlap measures. The following theorem is one of our main results:

Theorem 10.4.3 *For every normalized overlap measure f, and every pattern P and database graph G, it holds that*

$$MIS(P,G) \leq f(P,G) \leq MCP(P,G).$$

This bounding theorem still leaves a lot of room to define support measures, as there can be an arbitrarily large gap between MIS and MCP [2]. The Lovász and Schrijver measures that will be discussed in Section 10.4.5 is one such measure.

Example 10.4.4 *Consider again the example given in Figure 10.5.* $mis(G_P) = 2$ *and* $mcp(G_P) = 3$. *Hence, every antimonotonic normalized overlap support measure must assign a value between* 2 *and* 3 *for P in G. Indeed, as illustrated in Figure 10.6,* $\overline{K_2}$ *can be transformed into* G_P *and* G_P *can on its turn be transformed into* $\overline{K_3}$.

10.4.5 Lovász and Schrijver Graph Measures as Support Measures

The first function that was shown to be a normalized antimonotonic POG-based support measure computable in polynomial time was the Lovász ϑ value of the overlap graph [4]. A similar argument can be used for the Schrijver measure [18]. Both measures are studied in depth in the graph theory literature and often also relations to the size of the maximum independent set (MIS) and other important measures are

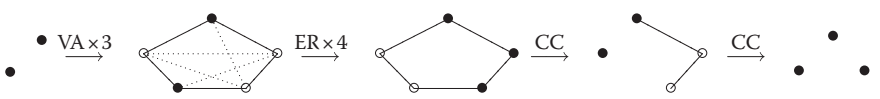


FIGURE 10.6: Illustration of a sequence of operations to move from a MIS to the overlap graph of Figure 10.5 to the minimal clique partition.

considered. We believe that it may be valuable for the data mining community to further explore this literature. Here, we briefly defined both measures. Interested readers can consult [4,18].

Let G be a graph. In the following, we will assume that $V(G) = \{1, \ldots, n\}$ so that we can use a vertex as an index of a vector or matrix, for example, we will denote cell i of a vector x by x_i. A *Lovász feasible matrix* A for G is a symmetric positive semidefinite matrix with (i) $A_{u,v} = 0$ for all u and v such that $\{u, v\} \in E(G)$ and (ii) $Tr(A) = 1$.

Given a graph G, the *Lovász* ϑ value of G is

$$\vartheta(G) = \max \left\{ \sum_{i,j} A_{i,j} \mid A \text{ is a Lovász feasible matrix of } G \right\}.$$

The *Schrijver graph measure* of G is

$$SGM(G) = \max \left\{ \sum_{i,j} A_{i,j} \;\middle|\; \begin{array}{l} A \text{ is a Lovász feasible matrix of } G \\ \text{and } \forall i, j : A_{i,j} \geq 0 \end{array} \right\}.$$

The Schrijver graph measure has nearly the same computational complexity as the Lovász ϑ value but is closer to the MIS support measure. The latter can be an advantage for certain statistical tasks in which we want to stay as close as possible to an independent set of images.

10.5 Object-Specific Overlap Hypergraphs

The overlap-based measures we discussed up to now viewed overlap as a binary property: two instances either overlap or not. In this section, we extend this idea further with support measures taking into account *how* instances overlap. We call this object-specific overlap and will represent the overlap relationships with a hypergraph instead of a simple graph [18].

As earlier, several types of overlap exist. First, we should select the *objects* of overlap. These can be vertices, edges, or both. For a database graph G, we define

$$\begin{aligned} Obj_{vertex}(G) &= V(G) \\ Obj_{edge}(G) &= E(G) \\ Obj_{ev}(G) &= V(G) \cup E(G) \end{aligned}$$

These objects induce cliques in the pairwise overlap graph G_P. For instance, when considering vertex overlap of instances (as we did in the previous sections), all instances containing a vertex v of G will form a clique in G_P.

Definition 10.5.1 (Overlap hypergraph) *Let* $\gamma \in \{vertex, edge, ev\}$. *The* γ*-ins-*overlap hypergraph *of* P *in* G, *denoted by* $H^{G,ins}_{P,\gamma}$ *or more briefly* H^G_P *if the rest is*

clear from the context, is a hypergraph whose vertices are the instances $Img(G,P)$ *and for each object* $x \in Obj_\gamma(G)$*, there is a hyperedge* $e_x \in E\left(H_P^G\right)$ *such that* $e_x = \{g \in V\left(H_P^G\right) \mid x \in Obj_\gamma(g)\}$.

The γ*-emb-*overlap hypergraph *of P in G, denoted by* $H_{P,\gamma}^{G,emb}$ *or more briefly* H_P^G *if the rest is clear from the context, is a hypergraph whose vertices are the embeddings* $Emb(G,P)$ *and for each object* $x \in Obj_\gamma(G)$ *and object* $y \in Obj_\gamma(P)$*, there is a hyperedge* $e_x \in E(H_P^G)$ *such that* $e_x = \{\pi \in V\left(H_P^G\right) \mid x = \pi(y)\}$.

In an overlap hypergraph $H_{P,\gamma}^{D,\delta}$, we say that a hyperedge e is *dominated* by another hyperedge e' if $e \subset e'$, and a hyperedge e is *dominating* if it is not dominated by any other hyperedge. For any D and P, we define the reduced overlap hypergraph $\tilde{H}_{P,\gamma}^{D,\delta}$ to be the hypergraph for which $V\left(\tilde{H}_{P,\gamma}^{D,\delta}\right) = V\left(H_{P,\gamma}^{D,\delta}\right)$ and $E\left(\tilde{H}_{P,\gamma}^{D,\delta}\right)$ is the set of all dominating hyperedges of $H_{P,\gamma}^{D,\delta}$. In the sequel, we only refer to $\tilde{H}_{P,\gamma}^{D,\delta}$, omitting δ and γ when they are clear from the context. We will abuse terminology and simply call $\tilde{H}_P^D$ the overlap hypergraph. See Figure 10.7 for an example.

We henceforth refer to the overlap hypergraph measures, which we denote by $f'\left(\tilde{H}_P^D\right)$, instead of referring to the induced support measure $f(D,P)$. Such induced support measures are called overlap-hypergraph-based support measures (OHSM).

10.5.1 Support Measure Based on Relaxed Maximal Independent Set

Given an overlap hypergraph $\tilde{H}_P^D$, we can derive the corresponding overlap graph G_P^D by replacing every hyperedge with a clique. Therefore, we can rephrase the definition of the MIS measure using overlap hypergraphs. Suppose D is a database graph and P is a subgraph pattern:

$$MIS(D,P) = MIS\left(\tilde{H}_P^D\right) = \max\left|\left\{I \subseteq V\left(\tilde{H}_P^D\right) \mid \forall e \in E\left(\tilde{H}_P^D\right) : |e \cap I| \leq 1\right\}\right| \tag{10.2}$$

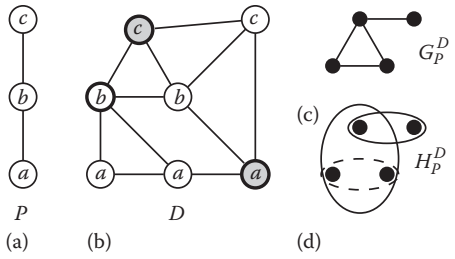


FIGURE 10.7: Overlap graph and overlap hypergraph. Given (a) a subgraph pattern P, (b) a database graph D, (c) the overlap graph G_P^D and (d) the overlap hypergraph H_P^D are shown on the right. In the overlap hypergraph, the (dominating) hyperedges are determined by the highlighted vertices in the database graph, and a dominated hyperedge is given in a dashed ellipse.

The MIS measure requires that a vertex of an overlap (hyper)graph is either in the independent set I or not. The s measure [18] we consider in this section is a relaxation of the MIS support measure by allowing for counting vertices of an overlap hypergraph only partially.

Let $\tilde{H}_P^D$ be an overlap hypergraph. We start by assigning to each vertex v of $\tilde{H}_P^D$ a variable x_v. We then consider vectors $x \in \mathbb{R}^{V\left(\tilde{H}_P^D\right)}$ of variables where for every $v \in V\left(\tilde{H}_P^D\right)$, x_v denotes the variable (component of x) corresponding to v. x is *feasible* if and only if it satisfies

$$\begin{aligned} &\text{(i)} \quad \forall v \in V(\tilde{H}_P^D) : 0 \leq x_v \\ &\text{(ii)} \quad \forall e \in E(\tilde{H}_P^D) : \textstyle\sum_{v \in e} x_v \leq 1. \end{aligned} \tag{10.3}$$

We denote the feasible region (the set of all feasible $x \in \mathbb{R}^{V\left(\tilde{H}_P^D\right)}$) by $\mathfrak{R}\left(\tilde{H}_P^D\right)$, which is a convex polytope.

Definition 10.5.2 (s support measure) *The measure* s *is defined by*

$$\mathsf{s}\left(\tilde{H}_P^D\right) = \max_{x \in \mathfrak{R}\left(\tilde{H}_P^D\right)} \sum_{v \in V\left(\tilde{H}_P^D\right)} x_v \tag{10.4}$$

Clearly, s is the solution to a linear program.

We will call an element $x \in \mathfrak{R}\left(\tilde{H}_P^D\right)$, which makes $\sum_{v \in V\left(\tilde{H}_P^D\right)} x_v$ maximal a *solution* to the LP of s.

There are very effective methods for solving LPs, including the simplex method, which is efficient in practice although its complexity is exponential, and the more recent interior-point methods [1]. The interior-point method solves an LP in $O(n^2 m)$ time, where n (here $\min\{|V\left(\tilde{H}_P^D\right)|, |E\left(\tilde{H}_P^D\right)|\}$) is the number of variables and m (here $|V\left(\tilde{H}_P^D\right)| + |E\left(\tilde{H}_P^D\right)|$) is the number of constraints. Usually, subgraph patterns are not large, so the LPs for computing s are sparse. Almost all LP solvers perform significantly better for sparse LPs.

10.5.1.1 Conditions for Antimonotonicity

The conditions for antimonotonicity of OHSMs are similar to the ones discussed earlier based on normal overlap graphs. In particular, we can show that an OHSM is antimonotonic if and only if it is nondecreasing under three operations on the overlap hypergraph. We begin by defining these three operations. They are similar to those on overlap graphs.

Definition 10.5.3 (Hypergraph operators) *For $H \in \mathcal{H}$, we define*

- *Vertex addition: A new vertex v is added to every existing hyperedge: $VA(H, v) = (V(H) \cup \{v\}, \{e \cup \{v\} \mid e \in E(H)\})$.*

- *Subset contraction: Let $K \subseteq V(H)$ be a set of vertices of the hypergraph such that $\exists e \in E(H) : K \subseteq e$. Then, the subset contraction operation contracts K into a single vertex k, which remains in only those hyperedges that are supersets of K. Formally, $SC(H, K, k) = (V(H) - K \cup \{k\}, E_1 \cup E_2)$ where $E_1 = \{e - K \cup \{k\} \mid e \in E(H)$ and $K \subseteq e\}$ and $E_2 = \{e - K \mid e \in E(H)$ and $K \nsubseteq e\}$).*

- *Hyperedge split: This operation splits a size k hyperedge into k hyperedges of size $(k-1)$ each: $HS(H, e) = (V(H), E(H) - \{e\} \cup \{e - \{v\} \mid v \in e\})$, where $e \in E(H)$.*

For example, suppose H_0 is a hypergraph, $V(H_0) = \{v_1, v_2, v_3, v_4\}$, and $E(H_0)$ contains two hyperedges $\{v_1, v_2, v_3\}$ and $\{v_1, v_4\}$. Let $H_1 = VA(H_0, v_5)$, then $V(H_1) = \{v_1, v_2, v_3, v_4, v_5\}$ and $E(H_1)$ contains hyperedges $\{v_1, v_2, v_3, v_5\}$ and $\{v_1, v_4, v_5\}$. Let $H_2 = SC(H_1, \{v_1, v_3\}, v_6)$, then $V(H_2) = \{v_2, v_4, v_5, v_6\}$ and $E(H_2)$ contains hyperedges $\{v_2, v_5, v_6\}$ and $\{v_4, v_5\}$. Let $H_3 = HS(H_2, \{v_2, v_5, v_6\})$, then $V(H_3) = V(H_2)$ and $E(H_2)$ contains four hyperedges $\{v_2, v_5\}$, $\{v_2, v_6\}$, $\{v_5, v_6\}$, and $\{v_4, v_5\}$.

10.5.1.2 Sufficient Condition

We present a sufficient condition for support measure antimonotonicity in terms of the three operations on the overlap hypergraph that we have defined (Figure 10.8).

AQ4

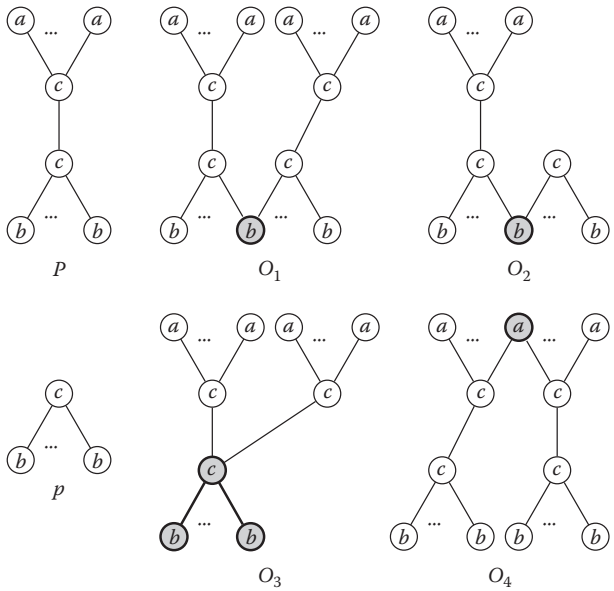


FIGURE 10.8: Patterns and different types of overlap. The highlighted parts show the ways two images overlap.

Theorem 10.5.1 *Let $f' : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ be a support measure and $f : \mathcal{H} \to \mathbb{R}$ with $f'(D, P) = f\left(\tilde{H}_P^D\right)$ be the induced OHSM. If f is nondecreasing under VA, SC, and HS, then f' is an antimonotonic support measure.*

Theorem 10.5.2 $\mathsf{s}(D, P) = \mathsf{s}(\tilde{H}_P^D)$ *is a normalized antimonotonic support measure.*

10.5.1.3 Necessary Condition

We show that the condition for antimonotonicity mentioned earlier is not only a sufficient but also a necessary condition.

Theorem 10.5.3 *Let $f' : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ be a support measure and $f : \mathcal{H} \to \mathbb{R}$ with $f'(D, P) = f\left(\tilde{H}_P^D\right)$ be the induced OHSM. If f' is antimonotonic, then f is nondecreasing under VA, SC, and HS.*

10.5.2 Bounding Theorem

In [4], the authors showed that all normalized antimonotonic OGSMs are bounded (between the MIS size and the MCP). Similarly, we prove that all normalized antimonotonic OHSMs are also bounded. We first introduce another OHSM, MSC, the size of a minimum set cover of overlap hypergraphs:

$$MSC(D, P) = MSC\left(\tilde{H}_P^D\right) = \min\left|\left\{S \subseteq E\left(\tilde{H}_P^D\right) \mid \bigcup_{e \in S} e = V\left(\tilde{H}_P^D\right)\right\}\right| \tag{10.5}$$

It is not difficult to verify that MSC is normalized and antimonotonic. Computing MSC is an NP-hard problem. The MIS size (Equation 10.2) and minimum vertex cover (Equation 10.5) are the minimally and maximally possible normalized antimonotonic OHSMs.

Theorem 10.5.4 *Given a database graph D, and a subgraph pattern P, it holds that $MIS(D, P) \leq f(D, P) \leq MSC(D, P)$ for every normalized antimonotonic OHSM $f(D, P) = f'\left(\tilde{H}_P^D\right)$.*

10.5.3 Relaxation of the OGSM MIS

One may ask whether the s support can be defined by relaxing the OGSM MIS instead of the OHSM MIS. In other words, is the concept of overlap hypergraphs really necessary?

Our answer is that the concept of overlap hypergraph is needed for the definition of the s support measure because it carries additional information on the overlap graph. In particular, the hyperedges show which overlaps have a common cause. If we did not have this information, we would not be able to reconstruct it. For instance, if we see a triangle in an overlap graph, we do not know whether this triangle originates from one vertex shared by the three images or from three vertices, each shared by two of the images. This additional information is needed for the definition of s, and for its mathematical properties.

10.6 Bounding the Variance of Sample Estimates Using the s Measure

An important motivation for investigating support measures is the need to perform statistical analysis on datasets. Statistical theory often assumes that data points are drawn independently. In networked data, however, where vertices are connected with edges, this is not the case anymore. In this section, we relate the statistical power of a set of observations to its s-measure. In particular, if a pattern has a number of overlapping embeddings in a database graph and every embedding has some properties, one can estimate the distribution of these properties (or its mean, variance, moment) from a sample. We are interested in bounding the variance of such estimates.

When performing statistics on a particular type of observations, we first have to define the properties that the observations of interest will need to satisfy, thus creating a subgraph pattern. For instance, suppose we want to analyze the satisfaction of clients with their first lawsuit where they are assisted by a pro-deo lawyer, that is, a lawyer paid by the government or by an association to offer legal aid services to those who cannot afford a lawyer. Then, the subgraph pattern representing the observation type of interest would consist of a client node, a lawyer node, a judge node, and a lawsuit node to which the former three are connected.

Next, let us assume that the occurrence of these observations occurs independently from the properties that are relevant for our statistical analysis. In our example, in order to ensure impartiality, the court randomly assigns judges to cases and the lawyer association randomly assigns pro-deo lawyers to cases. Hence, in order to explore the relationships between the properties of the case and its outcome, we do not need to take into account the dependency between occurrences of the subgraph pattern and its properties.

This simplifying assumption does not imply, however, that we can treat the properties of the nodes of the embeddings of the pattern as independent, since embeddings may share nodes. In our example, the same parties, lawyers, or judges may participate in different lawsuits.

Consider the simple task of estimating the expected value of a function over the properties of nodes participating in a random embedding. In particular, let $f(\cdot)$ be a function on embeddings and let μ be its expected value and σ its standard deviation. Consider also a sample, that is, a set of possibly overlapping embeddings, and the problem of estimating μ as accurately as possible. In our example, f could be the measurement of client satisfaction with the outcome of the lawsuit depending on properties of the client, the lawsuit, the lawyer, or the judge.

We will now present two approaches for deriving a relation between sample size and the variance on the estimate obtained.

In a first approach, we take a maximal independent set S_{MIS} of vertices of the overlap hypergraph H_p^D. As we assumed that the embeddings are independent from the properties of the nodes they connect, all elements in S_{MIS} are independent and the values $f(v)$ of the observations v in S_{MIS} are distributed independently with $\mathbb{E}\left[(f(v) - \mu)^2\right] = \sigma^2$. Consider now the estimator

$$\hat{\mu}_{\text{MIS}} = \frac{\sum_{u \in S_{\text{MIS}}} f(u)}{|S_{\text{MIS}}|}.$$

As the terms in the sum are independent random variables,

$$\mathbb{E}\left[\left(\hat{\mu}_{\text{MIS}} - \mu\right)^2\right] = \frac{\sigma^2}{|S_{\text{MIS}}|}. \tag{10.6}$$

We will now present a second approach based on our s measure. Suppose that we have a set $V\left(H_p^D\right)$ of observations (embeddings of the pattern p in the database graph D), whose overlaps are given by the overlap hypergraph H_p^D, and a vector x of weights x_v for the $v \in V\left(H_p^D\right)$, which is a feasible solution to the s measure related linear program (10.3). We define the estimator:

$$\hat{\mu}_{\mathsf{s}}(f, V\left(H_p^D\right), x) = \frac{\sum_{v \in V\left(H_p^D\right)} x_v f(v)}{\sum_{v \in V\left(H_p^D\right)} x_v} \tag{10.7}$$

We will now prove the following:

Theorem 10.6.1 *Let p be a pattern graph with $V(p) = \{i\}_{i=1}^k$ and $D = \cup_{i=1}^k D_i$ be a database with $k = |V(p)|$ domains. Let the set of embeddings of p in D be k-partite, that is, $Emb(D, p) \subseteq D_1 \times \cdots \times D_k$. Let the overlap hypergraph H_p^D represent this set of embeddings and their overlaps, that is, two vertices $u, v \in V\left(H_p^D\right)$ overlap if and only if $u(i) = v(i)$ for some $i \in \{1 \ldots k\}$. Assume the nodes in D have properties that are independent of these embeddings. Let x be a vector of weights for the embeddings satisfying (10.3). Let f be a function on the properties of the nodes participating in an embedding. Assume that for a randomly chosen embedding u, $\mathbb{E}\left[(f(u) - \mu)^2\right] = \sigma^2$. Then,*

$$\mathbb{E}\left[\left(\hat{\mu}_{\mathsf{s}}(f, V\left(H_p^D\right), x) - \mu\right)^2\right] \leq \frac{\sigma^2}{\sum_{v \in V\left(H_p^D\right)} x_v}$$

In conclusion, if we choose x such that $\sum_v x_v = \mathsf{s}\left(H_P^D\right)$, we get

$$\mathbb{E}\left[\left(\hat{\mu}_{\mathsf{s}}(f, V\left(H_p^D\right), x) - \mu\right)^2\right] \leq \frac{\sigma^2}{\mathsf{s}\left(H_p^D\right)}.$$

Because $\mathsf{s} \geq |MIS|$, the second approach yields a better estimate of μ. Even though we had to make a number of assumptions, this first result linking s and the statistical power of a sample suggests that closer analysis of its properties may be a valuable direction for further research. Note that the assumptions on which the first method (using a MIS) relies are not necessarily much weaker than those made for the method using s.

10.7 Discussion and Conclusion

Next to the types of overlap we considered in this paper, other types of overlap may be of interest. For example, the following notions of overlap could also be considered:

- *Two-vertex (edge) overlap*: Two images overlap if and only if they share two or more common vertices (edges).
- *Label-specific overlap*: Two images overlap if and only if they share a common vertex (edge), which has a label in a certain set.
- *Distance-based overlap*: Two images u and v overlap if u has a vertex x and v has a vertex y such that the distance between x and y is smaller than a specified constant *min_dist*.

In each of these cases, small patterns need to be treated with caution, but an anti-monotonic support measure is obtained for patterns of minimal size.

The choice of overlap notion may be inspired by several factors, one of the main ones being the statistical assumptions made and the task to be performed. For instance, in Section 10.6, we presented a derivation for the statistical power of a sample assuming that the property of interest only depends on the properties of the nodes participating in the embedding. Suppose now that this assumption does not hold. For instance, in our lawsuit example, clients belonging to the same family might share common properties or be influenced by each other and hence might not be independent. We could then add family relations to the graph and say that two embeddings $(client1, lawyer1, judge1, case1)$ and $(client2, lawyer2, judge2, case2)$ overlap if $client1$ and $client2$ are members of the same family (i.e., have a distance of at most 1 in the family relationship graph). In this way, we can relax our assumptions by strengthening our notion of overlap.

Frequency is not a perfect indicator of a patterns interestingness. In most cases, frequent subgraphs are trivial patterns. Therefore, Milo et al. [13] proposed methods based on statistical hypothesis testing to filter out insignificant patterns. They first assume a null model that generates networks by preserving the network degree distribution. Subsequently, every frequent subgraph is checked against the null model in a randomization test. This approach effectively filters out a lot of trivial frequent patterns. Importantly, it is computationally demanding because randomization tests require generation of random samples of the entire network and perform frequency counting on these large samples.

In this chapter, we studied support measures in the single-graph context. Most of the results in this chapter concerned the so-called overlap-graph-based support measures for which alternative characteristics, a bounding theorem, and extensions to hypergraphs were shown. We also show one example of an application, being bounding the variance of sample estimates.

References

1. S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, U.K., 2004.
2. V.E. Brimkov. Clique, chromatic, and Lovasz numbers of certain circulant graphs. *Electronic Notes in Discrete Mathematics*, 17:63–67, 2004.
3. B. Bringmann and S. Nijssen. What is frequent in a single graph? In *Proceedings of Mining and Learning with Graphs (MLG) 2007*, Florence, Italy, 2007.
4. T. Calders, J. Ramon, and D. Van Dyck. All normalized anti-monotonic overlap graph measures are bounded. *Data Mining and Knowledge Discovery*, 23(3):503–548, 2011.
5. L. De Raedt. Logical settings for concept learning. *Artificial Intelligence*, 95:187–201, 1997.
6. L. De Raedt and S. Kramer. The levelwise version space algorithm and its application to molecular fragment finding. In B. Nebel, ed., *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pp. 853–862. Morgan Kaufmann, San Francisco, CA, 2001.
7. R. Diestel. *Graph Theory*. Springer-Verlag, New York, 2000.
8. M. Fiedler and C. Borgelt. Support computation for mining frequent subgraphs in a single graph. In *Proceedings of the Fifth Workshop on Mining and Learning with Graphs (MLG'07)*, Firenze, Italy, 2007.
9. J.L. Gross and J. Yellen. *Handbook of Graph Theory*. CRC Press, Boca Raton, FL, 2004.
10. H. He and A.K. Singh. Efficient algorithms for mining significant substructures in graphs with quality guarantees. *Data Mining, IEEE International Conference on*, 0:163–172, 2007.
11. M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph. *Data Mining and Knowledge Discovery*, 11(3):243–271, 2005.
12. M. McGlohon, J. Leskovec, C. Faloutsos, M. Hurst, and N. Glance. Finding patterns in blog shapes and blog evolution. In *Proceedings of the International Conference on Weblogs and Social Media*, pp. 26–28, Boulder, CO, March 2007.
13. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.
14. C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.

15. J. Ramon, T. Francis, and H. Blockeel. Learning a Tsume-Go heuristic with Tilde. In *Proceedings of CG2000, the Second International Conference on Computers and Games*, volume 2063 of Lecture Notes in Computer Science, pp. 151–169, Hamamatsu, Japan. Springer-Verlag, Heidelberg, Germany, 2000.

16. H. Tong, C. Faloutsos, B. Gallagher, and T. Eliassi-Rad. Fast best-effort pattern matching in large attributed graphs. In *KDD'07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 737–746, New York. ACM, New York, 2007.

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

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

AUTHOR QUERIES

[AQ1] Please specify the part labels in figure caption for Figure 10.1, 10.3.

[AQ2] Please confirm the correctness of part labels and captions inserted in Figure 10.2, 10.5, 10.7.

[AQ3] Please provide closing parenthesis for sentence starting "However, $minImage(P, G) = 4$…" in the caption of Figure 10.2.

[AQ4] Please check the inserted citation of Figure 10.8 for correctness.