Proceedings of the Federated Conference on Computer Science and Information Systems pp. 329–338
DOI: 10.15439/2017F84
ISSN 2300-5963 ACSIS, Vol. 11

Document Clustering using a Graph Covering with Pseudostable Sets

Jens Dörpinghaus*, Sebastian Schaaf†, Juliane Fluck and Marc Jacobs
Fraunhofer Institute for Algorithms and Scientific Computing,
Schloss Birlinghoven, Sankt Augustin, Germany
Email: *jens.doerpinghaus@scai.fraunhofer.de, †sebastian.schaaf@scai.fraunhofer.de

***Abstract*—In text mining, document clustering describes the efforts to assign unstructured documents to clusters, which in turn usually refer to topics. Clustering is widely used in science for data retrieval and organisation. In this paper we present a new graph theoretical approach to document clustering and its application on a real-world data set. We will show that the well-known graph partition to stable sets or cliques can be generalized to pseudostable sets or pseudocliques. This allows to make a soft clustering as well as a hard clustering. We will present an integer linear programming and a greedy approach for this NP-complete problem and discuss some results on random instances and some real world data for different similarity measures.**

I. Introduction

DOCUMENT Clustering is usually not perceived as a graph problem. But following [1] we would like to split the process in two steps. At first we need to define a similarity measure appropriate to the data domain. Then the technical clustering process can be done using a graph theoretical approach. Jain et al. also suggested a last step called "assessment of output" and we will show that this can also be solved using graph theory and building the graph visualization proposed in this paper.

We will now define the problem. For technical terms we refer to [2]. The Cluster Hypotheses is essential: “Documents in the same cluster behave similarly with respect to relevance to information needs.” We are not trying to do K-Clustering, where we have a given number of K clusters. Thus we define the document clustering as follows:

Given a similarity function for the Document Space D as $sim : D \times D \rightarrow \mathbb{R}^+$ and an $\epsilon \in \mathbb{R}^+$. We search for a minimal number of clusters, so that every two documents x, y in one cluster have $sim(x, y) \geq \epsilon$. We will use this approach as definition II.1.

A *hard clustering* defines, that every document belongs to only one cluster, whereas *soft clustering* allows documents to be belong to one or more clusters, even with a distinct probability. We will introduce a novel new graph structure that can also handle soft clustering.

A lot of research to the topic of document clustering in the last years focused on methods and heuristics. The authors of [3] for example try to cluster documents from MEDLINE by using evolutionary algorithms, whereas [4] use machine learning approaches. Only few authors like [5] use graph-based approaches. Some authors, like [6] cover related problems like clustering in the context of search queries, whereas [7] work on the field of hierarchical clusterings.

This paper tries to use a novel reformulation of document clustering as a graph partition problem to get new insights to the problem itself. We hope that this leads to new heuristics and a deeper understanding of the problem. Thus, after considering some preliminaries we will introduce pseudostable sets and pseudocliques which are deeply related to graph coloring and stable sets. We will reformulate soft document clustering as a graph problem, where we seek a minimal partition in pseudeostable sets. After introducing a greedy and integer linear programming approach we will make a proof of concept on some real world data.

II. Preliminaries

A. Document Clustering

Using a Graph Partition for Clustering has been widely discussed in literature. Schaeffer points out that “the field of graph clustering has grown quite popular and the number of published proposals for clustering algorithms as well as reported applications is high” [8]. Usually directed or weighted graphs are subject of research. But we would like to point out that for problem complexity reasons it is suitable to focus on simple graphs. The work reported in [9] explains that a graph partition in cliques or stable sets is most common.

But we could also imagine – and find in literature – approaches that discuss somehow defined subgraphs or other partitions. As [8] points out unfortunately, “no single definition of a cluster in graphs is universally accepted, and the variants used the in literature are numerous”. We will start with this definition:

Definition II.1. *(Hard Document Clustering) Given a set of documents $D = \{d_1, ..., d_N\}$ and a similarity measure $sim : D \times D \rightarrow \mathbb{R}^+$ as well as a bound $\epsilon \in \mathbb{R}^+$. We search for a minimal number of clusters, so that for every two documents x, y sharing the same cluster $sim(x, y) \geq \epsilon$ holds.*

We would like to suggest a slightly different approach to cover both hard as well as soft clustering. A graph partition into stable sets or cliques can be generalized to be universal in such a way that it can handle hard clustering as well as soft clustering.

We argue that a simple graph for a representation of documents for the purpose of document clustering is not a

limitation. The graph does not need to be directed, since for two documents d_i, d_j always $sim(d_i, d_j) = sim(d_j, d_i)$. Since every clustering algorithm needs to decide, if two documents are in one cluster there is no need to assign a weight to the edge. If a previous measurement algorithm decides that two documents cannot be in the same cluster, the value should be set that way that there is an edge.

B. Graph Theory

Given a Graph $G = (V, E)$ with nodes or vertices in a set V and a set of edges E. Two nodes $u, v \in V$ are adjacent, if an edge $(u, v) \in E$ exists. The graph coloring problem is to assign a color to each node so that every two nodes that are adjacent have a different color. The minimal number of colors needed to color a graph is called chromatic number and denoted with $\chi(G)$.

This problem has many applications and has been studied extensively. It is on most graphs NP-complete, see [10].

For every feasible coloring of G all nodes sharing the same color imply a stable set in G. S is a *stable set* in G if $(u, v) \notin E \ \forall u, v \in S$. Thus we have a partition of G in stable sets.

But it is also possible to use a set covering approach, where the set of vertices has to be covered by a minimum number of stable sets, see [11]. This is very useful in the context of linear programming. As Hansen et al. mentioned this approach involves an exponential number of variables which makes the problem complex. Many optimization problems on graphs can be formulated as set covering problems.

III. Pseudostable Sets and Pseudocliques

We will now discuss novel graph structures. Pseudostable sets were first introduced in [12] as a graph partition problem in the context of the Train Marshalling Problem covering the rearrangement of cars of an incoming train in a hump yard. They are still under research in several contexts. In this paper we will apply pseudostable sets in a total new context and also introduce pseudocliques and the corresponding graph covering problem. Thus the whole approach presented in this paper is novel.

We now consider a simple Graph $G = (V, E)$ with a subgraph $B \subset G$ of so called blue nodes and edges. B can be chosen absolutely arbitrary. For example it is also possible that $B = \emptyset$ or $B = G$.

A. A set covering approach

At first we need to define two different subsets of the graph G to create a set covering:

Definition III.1. *(Pseudostable Tuple) $T \subset G$ is a* pseudostable Tuple*, if it is the union of two stable sets D_1 and D_2 and a path p such that*

$$T = D_1 \cup p \cup D_2$$

The intersection of D_1 and p as well as p and D_2 consists of one node. The set p is pairwise disjunct and consists of three nodes and two edges in B. That means, $p_j \subset B(G)$, $|V(p_j)| = 3$ and p_j is connected and circle-free. T can also be stable if D_1 is stable and $p = D_2 = \emptyset$. Then the value of T is $\zeta(T) = 1$, otherwise $\zeta(T) = 2$.

It is also possible to allow more than one path between D_1 and D_2, see figure 1 for an illustration.

Definition III.2. *(Multiple pseudostable Tuple) $M \subset G$ is a* Multiple pseudostable Tuple*, if it is the union of two stable sets D_1 and D_2 and paths $p_1, ..., p_i$ such that*

$$M = D_1 \cup p_1 \cup ... \cup p_i \cup D_2$$

The intersection of D_1 and p_j as well as p_j and D_2 ($j \in \{1, ..., i\}$ consists of one node. The sets p_i are pairwise disjunct and consist of three nodes and two edges in B. That means, $p_j \subset B(G)$, $|V(p_j)| = 3$ and p_j connected and circle-free. T can also be stable if D_1 is stable and $i = 0$ and $D_2 = \emptyset$. Then the value of T is $\zeta(M) = 1$, otherwise $\zeta(M) = 2$.

Since we usually have more than one M or T we will use indices to denote them. In the following, M_i or T_i are an arbitrary chosen M or T. We denote for M_i or T_i both stable sets with D_1^i or D_2^i.

It is possible that $D_2^i = \emptyset$, but it is always $D_1^i \neq \emptyset$. We define that $Pf(T)$ or $Pf(M)$ is the union of all paths in T or M. $Pf(T_i) = \emptyset$ or $Pf(M_i) = \emptyset$ if, and only if $D_2^i = \emptyset$. Every pseudostable Tuple is a multiple pseudostable Tuple. We usually search for a minimal set cover S of G with $S = \{T_1, ..., T_n\}$ or $S = \{M_1, ..., M_n\}$. We define the weight w as

$$w(S) = \sum_{i=1}^{n} \zeta(S_i) + \sum_{i=1}^{n} \sum_{j \in \{1,...,n\} \setminus \{i\}} w_{i,j} \qquad \text{(EQ)}$$

$$w_{i,j} = \begin{cases} -1 & M_i \cap S_j = D_1^i = D_2^j \\ 0 & otherwise \end{cases}$$

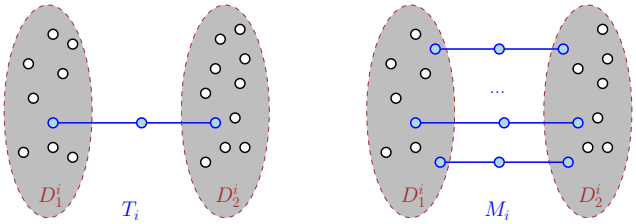
The first condition ensures that two stable sets D in two different tuples which are identical are not weighted two times. All other cases can be ignored. This weight holds for multiple pseudostable tuples as well as pseudostable tuples. With a weight we can define a minimization problem.

For a given Graph $G = (V, E)$ with a blue subgraph $B \subset G$ we define $\mathfrak{T} = \{T_1, ..., T_n\}$ as the subset of all pseudostable tuples in G with B.

With $\mathfrak{P}(T)$ we denote all inner nodes of paths within T, which means

$$\mathfrak{P}(T_i) = T_i \setminus \{D_1^i \cup D_2^i\}$$

Or, it is also possible to define it according to $Pf(T_i)$ as $Pf(T_i) \setminus \{D_1^i \cup D_2^i\}$ which is the same.



(a) (b)

Fig. 1: A pseudostable tuple T_i in (a) and a multiple pseudostable tuple M_i in (b). Both sets D_1 and D_2 are stable and some blue paths of length 3 exist between both. The sets $\mathfrak{P}(T_i)$ and $\mathfrak{P}(M_i)$ consist of all blue nodes which are neither in D_1 nor in D_2.

The Definition of the optimization problem can now be written as:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} t_i \zeta(T_i) + \sum_{i=1}^{n} t_i \sum_{j=1}^{n} t_j w_{i,j} \\ \text{subject to} \quad & \sum_{T \in \mathfrak{T}: v \in P f(T)} t_i = 1, \forall v \in V \\ & \sum_{T \in \mathfrak{T}: v \in T} t_i \geq 1, \forall v \in V \\ & t_i \in \{0,1\} \end{aligned} \tag{IP1}$$

The variable t_i indicates, if set T_i is chosen for this set covering. The minimization term refers to the weight given in equation EQ. The next line ensures that every node $v \in V$ is assigned to exactly one node within a path of a pseudostable tuple. The last condition ensures that every node $v \in V$ is covered by at least one set.

If we want to allow intersections between inner nodes of paths p we can simply skip the second condition. Thus our minimization problem is as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} t_i \zeta(T_i) + \sum_{i=1}^{n} t_i \sum_{j=1}^{n} t_j w_{i,j} \\ \text{subject to} \quad & \sum_{T \in \mathfrak{T}: v \in T} t_i \geq 1, \forall v \in V \\ & t_i \in \{0,1\} \end{aligned} \tag{IP2}$$

Both IP1 and IP2 hold for pseudostable tuples T as well as multiple pseudostable tuples M.

A set covering of a graph $G = (V, E)$ with a subset $B \subset G$ of blue nodes and edges with a set of T or M also induces the Graph of this set covering. In this graph every stable set D within the covering of G induces a node and every path an edge:

Definition III.3. *(Graph of a set covering) Given a set covering $S = \{S_1, ..., S_n\}$ of a graph $G = (V, E)$ with a subset $B \subset G$ of blue nodes and edges with pseudostable tuples $T_1, ..., T_n$ or multiple pseudostable tupels $M_1, ..., M_n$. Then we define $G_S = (V, E)$ as the Graph of the set covering with*

$$V = \{D \subset S_1, ..., S_n\}$$

$$E = \{(D_1^i, D_2^i)\ i \in \{1, ..., n\} \text{ if } D_2^i \neq \emptyset\}$$

Now we can define the minimization problem as follows. We will continue using the naming introduced in [12].

Definition III.4. *(minPS) We search for a minimal set covering S of the graph $G = (V, E)$ with a subset $B \subset G$ of blue nodes and edges with pseudostable tuples T according to IP1 where G_s is acyclic and $\delta(v) \in \{0, 1, 2\}$ for all $v \in V(G_S)$.*

Definition III.5. *(minMPS) We search for a minimal set covering S of the graph $G = (V, E)$ with a subset $B \subset G$ of blue nodes and edges with multiple pseudostable tuples M according to IP1 where G_s is acyclic and $\delta(v) \in \{0, 1, 2\}$ for all $v \in V(G_S)$.*

We denote minPS' and minMPS' as the corresponding minimization problem according to IP2. minPS-a and minMPS-a are the corresponding minimization problems without restrictions on the graph G_S. This means

Definition III.6. *(minPS'-a) We search for a minimal set covering S of the graph $G = (V, E)$ with a subset $B \subset G$ of blue nodes and edges with pseudostable tuples T according to IP2.*

Definition III.7. *(minMPS'-a) We search for a minimal set covering S of the graph $G = (V, E)$ with a subset $B \subset G$ of blue nodes and edges with multiple pseudostable tuples M according to IP2.*

Now we have a definition as set covering problem. This is also useful to proof the NP-completeness of this problem. Now we will make a definition using a graph partition approach.

B. A graph partition approach

The formulation of minPS or minMPS as graph partition problem is very clear and concrete but it gets unhandy when handling the variants minMPS-a or minMPS'. But since we need to proof that our new approach using set covering is equivalent to the work described in [12], we will introduce the graph partition approach.

Given a simple Graph $G = (V, E)$ and a subgraph $B \subset G$ of blue edges and nodes. We name a subset of G with $i \in \mathbb{N}^+$ as $P_i \subset G$. See figure 2 for an illustration of the following definitions.

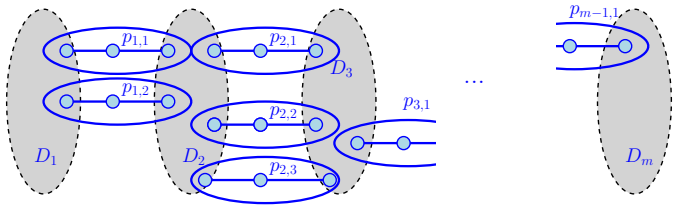


Fig. 2: Example partition $D_1, p_{1,1}, p_{1,2}, D_2, , p_{2,1}, p_{2,2}, p_{2,3}, ..., p_{m-1}, D_m$ in multiple pseudostable sets.

Definition III.8. *P_i is called a* pseudostable set *if and only if $P_i = D_i$ is stable or there exist stable sets D^i_j so that P_i is partitioned in*

$$D^i_1, p^i_1, D^i_2, p^i_2, D^i_3, ..., p^i_{m_i-1}, D^i_{m_i}$$

with $m_i \geq 2$. The intersection of following stable sets D_j and sets p_{j+1} as well as p_j and D_{j+1} consist only of one node. The sets p_j are pairwise disjunct and consist of three nodes and two edges in B. That means, $p_j \subset B(G)$, $|V(p_j)| = 3$ and p_j connected and circle-free. The value of this set P_i is m_i.

Now again we have some nodes that are not in stable sets, but in pseudostable sets. This means, we allow documents to lie in between clusters. To allow more than one node in between stable sets, we define multiple pseudostable sets:

Definition III.9. *P_i is called a* multiple pseudostable set *if and only if $P_i = D_i$ is stable or there exist stable sets D^i_j so that P_i is partitioned in*

$$D^i_1, p^i_{1,1}, ..., p^i_{1,n_1}, D^i_2, p^i_{2,1}, ..., p^i_{2,n_2},$$
$$D^i_3, ..., p^i_{m_i-1,1}, ..., p^i_{m_i-1,n_{m_i-1}}, D^i_{m_i}$$

with $m_i \geq 2$. The intersection of following stable sets D_j and sets p_{j+1} as well as p_j and $D_{j+1,n}$ consists only of one node. The sets $p_{j,n}$ are pairwise disjunct and consist of three nodes and two edges in B. That means, $p_j \subset B(G)$, $|V(p_j)| = 3$ and p_j is connected and circle-free. The value of this set P_i is m_i.

Without loss of generality it is of course possible to store the possible paths in a list and not as a subset of the graph G. Both formulations are equivalent and searching for a minimum set covering of G will provide a minimum graph partition. We will show exemplarily the following lemma. All other proofs can be done the same way.

Lemma III.10. *Every set covering S of a graph $G = (V, E)$ with a subgraph $B \subset G$ of blue edges and notes with multiple pseudostable tuples according to definition III.5 is equivalent to a graph partition of G in multiple pseudostable sets according to definition III.9.*

Proof. "⇒" Given a minimal set covering S of the graph $G = (V, E)$ with a subset $B \subset G$ of blue nodes and edges with multiple pseudostable tuples M according to IP1 where G_s is acyclic and $\delta(v) \in \{0, 1, 2\}$ for all $v \in V(G_S)$.

Since G_S is acyclic we can handle each connected component $Z \subset G_S$. This either has only one node and is thus equivalent to a stable set D^i. We then create a stable set D'^i. Or it has at least two nodes v_1 and v_j with $\delta(v) = 1$. Then we consider each stable set in sequence v_1 till v_j. Analogously we create stable sets (D^i_1, D^i_2) $i \in \{1, ..., n\}$ if $D^i_2 \neq \emptyset$. This means $D'^1_1, D'^1_2, D'^2_1, ..., D'^j_1, D'^{ij}_2$. But every time $D'^i_2 = D'^{i+2}_1$ holds, since otherwise no edge would be possible in G_S. We adjust all paths according to that, see figure 3.

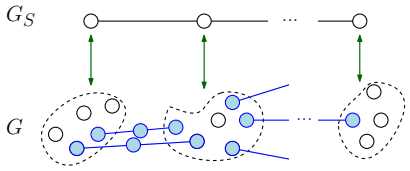


Fig. 3: Illustration of G and G_S according to the proof of lemma III.10.

Every intersection of stable sets D'^i and D'^j is either empty or we adjust all nodes according to definition III.9. Since equation IP1 holds, this is true for all paths. All other nodes can be arbitrarily assigned to one stable set that covers this node. If we eliminate one stable set, this set covering was not minimal.

"⇐" Since every graph partition is a graph covering we have to show that every pseudostable set according to lemma III.9 fulfils the definition III.5. It is obvious that two following stable sets in a pseudostable set are a pseudostable tupel. Each pseudostable set is a connected component of G_S. The value of this connected component is the same as in equation EQ. We can do this successively for every pseudostable set in the graph partition. Thus every partition holds the conditions for III.5. □

We will now introduce pseudocliques and show that they will solve the same problem on the complementary graph.

C. Pseudocliques

It is also possible to define the problem on the complement graph $\overline{G}$. This graph is defined by $G = (V, E)$ with $\overline{G} = (V, E')$ where $e \in E' \Leftrightarrow e \notin E$. Since $B \subset G$ now all blue edges are not in $\overline{G}$ any more and $B \not\subset \overline{G}$.

Definition III.11. *Q_i is a* Pseudoclique *if and only if $Q_i = C_i$ is a clique or there exist stable sets C^i_j so that Q_i is partitioned in*

$$C_1^i, p_1^i, C_2^i, p_2^i, C_3^i, ..., p_{m_i-1}^i, C_{m_i}^i$$

with the same conditions as mentioned above. For multiple Pseudoclique *this condition holds with several paths* $p_{j,k}^i$ *between the stable sets.*

A minimal Partition of $G = (V, E)$ and a subgraph $B \subset G$ in multiple pseudostable sets (minMPS) has a value of $\zeta(G)$. A minimal Partition of $\overline{G}$ with $B \not\subset G$ in multiple Pseudocliques (minMPC) has the value $\overline{\zeta}(G)$. We can conclude that both approaches are polynomial equivalent:

Lemma III.12. *Every minimal partition of a Graph* $G = (V, E)$ *with a subgraph* $B \subset G$ *in multiple pseudostable sets with value* $\zeta(G)$ *implies a partition of* $\overline{G}$ *with* $B \not\subset G$ *in multiple Pseuoclique with the value* $\overline{\zeta}(\overline{G})$ *and vice versa. This implies*

$$\zeta(G) = \overline{\zeta}(\overline{G})$$

Both approaches can be converted in polynomial time and have the same solutions and complexity. This is why we first focus on pseudostable sets and try to get some improvements by considering the problem on the complementary graph.

IV. A NEW CLUSTERING APPROACH WITH PSEUDOSTABLE SETS

We will now create a Graph $G = (V, E)$. Every document in our document set is one node $n \in V$. We would like to follow [8] and restrict our similarity measure on $[0, 1]$, "where one corresponds to a 'full' edge, intermediate values to 'partial' edges, and zero to there being no edge between two vertices." Now we can define a limit and define edges between nodes if they are not similar enough.

Given a set of documents $D = \{d_1, ..., d_N\}$, a similarity measure

$$sim: \ D \times D \to \mathbb{R}^+$$

and an $\epsilon \in \mathbb{R}^+$. The function is limited to $[0, 1]$. If not, we normalize it as $sim': \ D \times D \to [0, 1]$ as

$$sim'(x, y) = \frac{sim(x, y)}{\max sim(x, y)}$$

Our graph G is now defined as

$$G = (V, E) \ V = D$$

$$E = \{(d_i, d_j) \mid sim(d_i, d_j) \leq \epsilon\}$$

Edges between documents exist only if they are less similar than ϵ. A graph coloring approach would now create a graph partition into stable sets. This would result in a hard clustering. To achieve a soft clustering we can define another bound ι with $0 < \iota < \epsilon$ and another set of edges $B = (V, E')$ with

$$E' = \{(d_i, d_j) \mid \iota \leq sim(d_i, d_j) \leq \epsilon\}$$

We can see that $B \subset G$. We have two kinds of edges, those edges $e \subset G$ but not in B. We call them black. These refer to documents which are not similar. But those edges $e \subset B$ called blue refer to documents that are also not similar, but less not similar then those edges not in B. If we set $\iota = \epsilon$ then $B = \emptyset$ and we have a hard clustering. If $B \neq \emptyset$ we have a soft clustering if we use the following defintion:

Definition IV.1. *(PS-Document Clustering) Given a graph* G *with* $B \subset G$ *according to the definition above. A solution of minMPS'-a gives a Document Clustering in multiple pseudostable sets with* $\zeta(G)$ *Cluster and Documents that are in between those clusters* D.

Before continuing, we will create the weighted Graph of the clustering. This definition is highly related to definition III.3. Every node refers to a document cluster and every edge refers to the number of paths between both clusters.

Definition IV.2. *The* weighted Graph of the Clustering *is a Graph* $G_c = (V_c, E_c)$ *with*

$$V_c = \{D_j^i \in P_i\}, \ d(D_j^i) = |D_j^i|$$

$$E_c = \{(D_j^i, D_k^i), d(D_j^i, D_k^i) > 0\}$$

The weight $d(D_j^i, D_k^i)$ can be defined in multiple ways. The easiest way is to sum all paths between both stable sets:

$$d_s(D_j^i, D_k^i) = |P| \text{ with}$$

$$P = \{p \mid p \cap D_j^i \neq \emptyset \text{ and } p \cap D_k^i \neq \emptyset\}$$

but more intuitive is the following weight:

$$d(D_j^i, D_k^i) = \sum_p \frac{|N(v) \cap D_j^i| + |N(v) \cap D_k^i|}{|D_j^i| + |D_k^i|} / |p|$$

$$\forall p = (u, v, w) \text{ with } p \cap D_j^i \neq \emptyset \text{ and } p \cap D_k^i \neq \emptyset$$

This weight counts all inner nodes v within a path $p = (u, v, w)$ the number of neighbours in one of the stable sets. We can use this as a measure for the similarity of this node with the given stable set. If there is no edge from u to one node in the set, it might also be assigned to that stable set. Each such edge decreases this possibility. We normalize with the number of paths and thus have a value in between $[0, 1]$.

Example IV.3. *Given three documents with some similarity, see figure 4. We set* $\iota = 2, 5$ *and* $\epsilon = 5$. *Now we have a graph with blue nodes and two blue edges. One edge is black. If we partition into pseudostable sets, we find two clusters with one document and one document in between both. The weighted graph of this clustering is also shown in figure 4. Every cluster is associated with a node in* G_c.

If we precisely use the Definition of pseudostable sets given by graph partition approach, this Graph needs to be acyclic. But we will follow the definition given in the first chapter and just notice that the definition by set covering approach is more clear. This Graph is important for visualization and assessment.

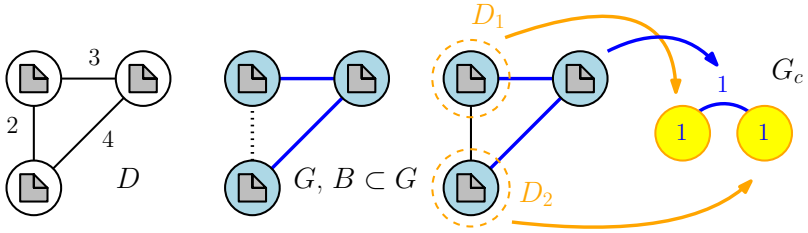


Fig. 4: Figure explaining the example IV.3. It illustrates the documents D with their similarity, the resulting Graph G, its partition into pseudostable sets D_1, D_2 and the weighted graph G_C of that clustering.

V. New Approaches

The main problem is that minMPS'-a is NP-complete. First of all, we will describe an Integer Linear Programming approach for calculating optimal solutions. Afterwards, we will discuss our Greedy-Approach for solving minMPS'-a. We want to show a small example on how all approaches solve the problem. Afterwards we will discuss some real-world data and the output.

A. Integer Linear Program

Given a graph $G = (V, E)$ with a subset $B \subset G$ of blue nodes and edges. T is the list of all paths with length three within B.

y_k denotes the variable, which indicates that a color k is used. Is $y_k = 0$ color k will not be used. $x_{i,k}$ indicates, if a node $i \in G$ is colored with color k. Color $k = 0$ will be used for those nodes which are in a path p.

$$\text{[minMPS'-a-IP]} \quad \min \sum_{k=1}^{n} y_k$$

$$\sum_{k=1}^{n} x_{i,k} = 1 \qquad \forall i = 0, ..., n \tag{1}$$

$$x_{i,k} - y_k \leq 0 \qquad \forall i = 0, ..., n, \forall k = 1, ..., n \tag{2}$$

$$x_{i,k} + x_{j,k} \leq 1 \qquad (i,j) \in E(G), \forall k = 1, ..., n \tag{3}$$

$$x_{i,0} \leq 0 \qquad \forall i \notin B(G) \tag{4}$$

$$x_{i,k} \geq 0 \tag{5}$$

$$y_k \leq 1 \tag{6}$$

$$x_{i,k} + x_{j,k} + x_{v,0} - 2 \leq 0 \qquad (i,v,j) \in T, \forall k = 1, ..., n \tag{7}$$

$$x_{i,0} + x_{j,0} + x_{v,0} \leq 1 \qquad (i,v,j) \in T, \forall k = 1, ..., n \tag{8}$$

$$x_{i,k}, y_k \in \mathbb{Z}$$

Condition 1 ensures that every node has a color or color $k = 0$. For each node i and every color k $x_{i,k} - y_k \leq 0$ is necessary. Is node i not in color k, inequality 2 holds. But if it is in color k, $y_k = 1$ and thus the inequality holds. Two connected nodes i, j must not share the same color $k > 0$. Thus $x_{i,k} + x_{j,k} \leq 1$, see condition 3. Condition 4 ensures that no node which is not within B can be assigned to color $k = 0$. The last conditions ensure that if a node v is within color $k = 0$ all within B connected nodes to v have a different color.

In practise we can only apply minMPS'-a-IP to small instances because of the exponential runtime.

B. Greedy-Approach

Given a graph $G = (V, E)$ with a subset $B \subset G$ of blue nodes and edges. We run on a (not necessary minimal) graph coloring $f : V \to F$ with $F \subset \mathbb{N}$ and implement a greedy algorithm that puts every possible path in between two stable sets. Since we do not have perfect graphs for documents clustering we need to use heuristics to get an approximate graph coloring. Alternatively we can use the complement graph $\overline{G}$ and use a partition into cliques which results in a coloring of G.

We will iteratively try to eliminate stable sets D given by the graph coloring heuristic and thus use the properties and characterizations of pseudostable sets:

- For each color i we consider node u in it:
  - Is this node not an endpoint of a path p (which ist stored in $ende$) check if there exist two nodes $v, w \in G$ which are connected by blue nodes with u and are in different color classes.
  - Is this true, remove u from i and create a new path $p = [v, u, w]$.

See algorithm 1 for pseudo code. We can not give an approximation guarantee and we will show that this heuristic does usually not provide an optimal solution.

We have used the following heuristics to start the graph coloring:

- Coloring using the *greedy independent sets* (GIS) approach with a runtime in $O(mn)$, see [13].
- Coloring using the SLF-Approach with a linear runtime $O((m+n)\log n)$ (see [13] and [14]).
- Clique Partition on $\overline{G}$ using the TSENG clique-partitioning algorithm described in [15] with a worst case runtime $O(n^3)$.

We assume to get a better solution by the third approach for instances where we have a huge amount of edges and it might

Algorithm 1 GREEDY-DC

Require: Graph G with a coloring f and a list $T = (t_1, ..., t_{t_C})$ of all paths.
Ensure: Partition P of G in MPS'-a

```
Sort all color classes f_1, ..., f_|F| increasingly by size
for each color class f_i in F do
    T_i ← all t ∈ T with a middle node in f_i
    for each t_i = (a, b, c) in T_i do
        if f(a) ≠ f(c) and ende(b) = false then
            ende(a) ← true
            ende(c) ← true
            f(b) = 0
        end if
    end for
end for
return P, where f denotes the stable sets and f_0 all paths.
```

be less complex to solve the clique partition problem on the complement graph.

We will generate some random instances using the model of Gilbert, see [16]. This creates a simple undirected graph $G = (V, E)$ with $(n(n-1))/2$ possible edges as a model $\mathscr{G}(n, p)$. Edges will be added with probability $0 < p < 1$.

Erdös and Rényi designed a similar approach $\mathscr{G}(n, m)$, were all Graphs with exactly n nodes and $0 < m < (n(n-1))/2$ edges are equal probable, see [17].

Both algorithms have a quadratic runtime. For small p Batagel and Brandes described a linear time approach with a runtime in $O(n + m)$, where m is the number of created edges, see [18].

We will chose $p = 0.75$ and a second probability $p' = 0.2$ which decides if edges are colored blue. This refers to the instances we have seen on real world data.

We will show the results for different random instances with 15 nodes in figure 5 and with 100 nodes in figure 6. We have also added the results of the integer linear program for small instances.

As we can see in both figures, the clique approach gives the worst partition into stable sets for large instances but the greedy approach eliminates most stable sets. SLF gives in general better results than GIS and also has a better runtime.

VI. DOCUMENT CLUSTERING ON MEDLINE

We apply this new approach to perform document clustering over some subsets of MEDLINE data. MEDLINE (Medical Literature Analysis and Retrieval System Online) is a bibliographic database maintained by the National Center for Biotechnology Information and covers a large number of scientific publications from medicine, psychology, and the health system. For the clustering use case, we study MEDLINE abstracts and associated metadata that are processed by ProMiner, a named entity recognition system ([19]), and indexed by the semantic information retrieval platform SCAIView ([20]). SCAIView also offers an API that allows programmatic access to the data. Currently, we only use meta information like title, journal, publishing year and the MeSH terms for our experiments.

We extract subset D of MEDLINE documents from SCAIView. Every document on MEDLINE should have a list M of keywords, so called MeSH terms. We may use them to calculate the Tanimoto similarity, also known as Jaccard similarity

$$sim(a, b) = \frac{|M_a \cap M_b|}{|M_a \cup M_b|} \; \forall a, b \in D$$

with $sim : M \times M \to [0, 1]$. This first approach is not suitable for all applications as we will show in the next section. This is why we postulate a distance model based on the vector of weighted words using NLP techniques.

We then build a graph G according to the bounds ϵ and ι. Following this, we create the directed graph of that partition by applying the Greedy approach. We also store further metadata like years and journals in nodes and edges.

We will now describe the result of one input set given by [21] and discussed by [22]. In both publications the first dataset consisted of 1660 documents obtained from two different queries 'escherichia AND pili' and 'cerevisiae AND cdc*'. Both returned the same number of 830 documents. We had a similar result with 1628 documents trying to reproduce this query with data till 2001. This dataset covers two different topics, whereas the second dataset is related to the developmental axes of Drosophila. We will now discuss several outputs of our new approach.

Consequently, we have $n = 1628$ nodes (documents). The number of edges e and blue edges b depend on the different values of ι and ϵ and the priorly used approach for similarity. We will discuss the following three measures: First an approach using a distance model d_V based on the vector of weighted words using NLP techniques for the abstracts. In addition a distance according to the journal, which is $d_J(x, y) = \{0, 1\}$. Thus we have

$$d_1(x, y) = \frac{d_V(x, y) + d_J(x, y)}{2}$$

The second approach is the usage of $d_2 = d_V$. The third approach uses only the Tanimoto similarity on MeSH terms described above, thus $d_3 = sim$.

We wanted to compare our results with those given by [21] and [22]. We will show that the comparability of clusterings with previous studies is highly dependent on the choice of this distance measurement. Every clustering produces unique details with the same heuristic running in the background. Thus it is not totally clear to connect clusters to topics. But first of all we want to proof our new approach and reproduce the results of both [21] and [22] which we will discuss for every distance measure.

Distance measure d_1: The results of our clustering approach with distance measure d_1 are shown in figure 7 and table I. We got 13 clusters (Cluster 0 to 12) with documents between 5 (Cluster 11) and 359 (Cluster 8) documents.

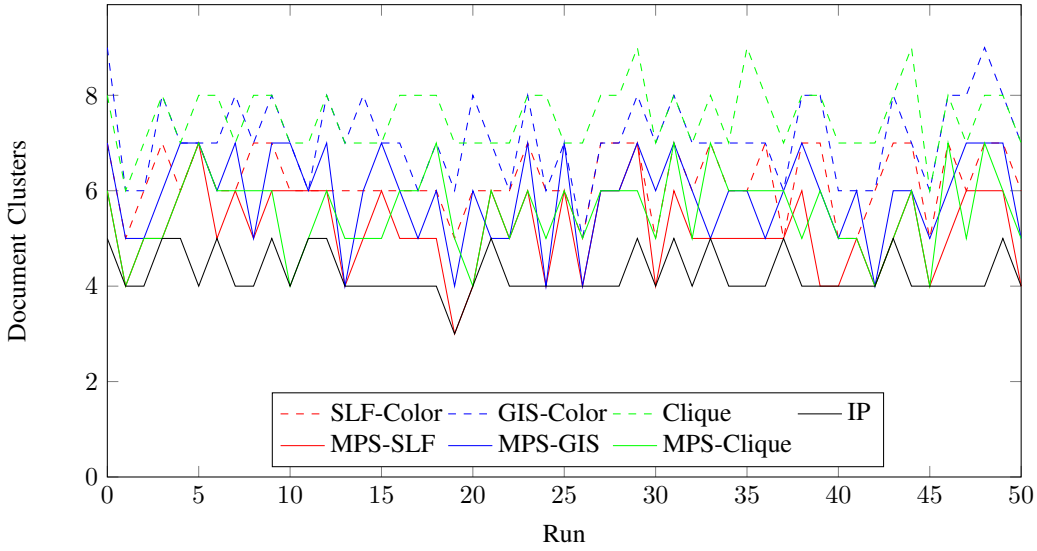


Fig. 5: Results for random instances with $n = 15$ nodes.

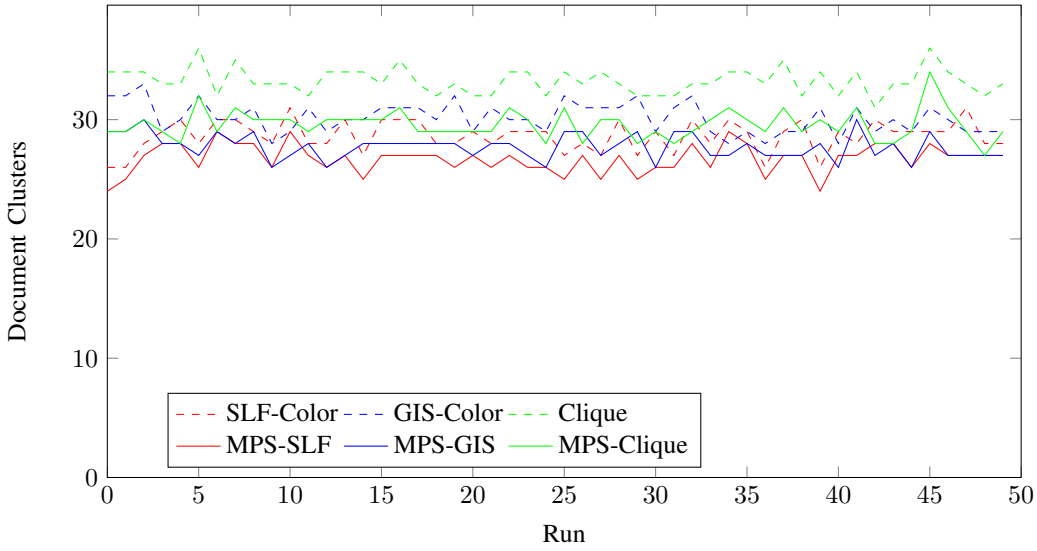


Fig. 6: Results for random instances with a node count $n = 100$

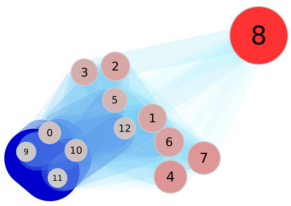


Fig. 7: The partition of the first dataset with distance d_1. The numbers identify the clusters. The size of a node is related to the number of documents included. The edges and their widths and color describe their weight. A darker blue edge has a greater weight.

Our clustering heuristic is able to produce clusterings of variable detail by choosing different values for ι and ϵ. We have chosen values that visualize the benefit of the new graph theoretical approach. Referring to figure 7 it is easy to see that the first cluster is given by cluster 8. It has only weak dependencies and relations to other clusters as can be seen by the edges in the graph. Clusters 0, 9, 10, 11 are highly dependent and thus form the second cluster agglomeration. The MeSH terms that describe these clusters can be found in table I. We can see a similar result to [22]: the terms of both clusters describe the general concepts that are relevant to both search queries. So our approach produces similar results with this distance measure.

Those clusters which are in between the two main clusters share topics with both. For example cluster 7 is related to

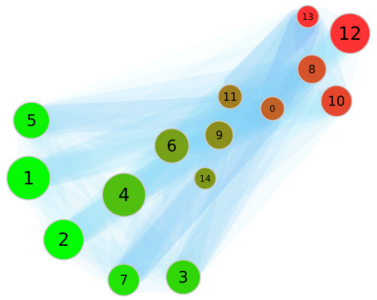


Fig. 8: The partition of the first dataset with the distance d_2. This picture shows the weighted graph of the clustering. The color of the nodes indicate a high rate of documents from the respective queries (red: 'escherichia AND pili'; green 'cerevisiae AND cdc*').

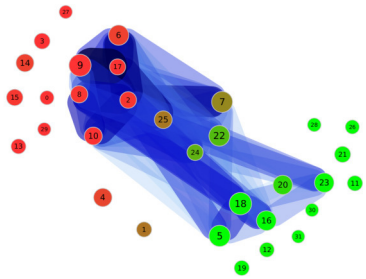


Fig. 9: The partition of the first dataset with the distance d_3. This picture shows the weighted graph of the clustering. The color of the nodes indicate a high rate of documents from the respective queries (red: 'escherichia AND pili'; green 'cerevisiae AND cdc*').

Terms	
Global Cluster 1	Global Cluster 2
Cluster [8]	Cluster [0, 9, 10, 11]
Escherichia coli	Saccharomyces cerevisiae
Fimbriae, Bacterial	Saccharomyces cerevisiae Prot.
Fimbriae Proteins	Fungal Proteins
Bacterial Adhesion	Mutation
Plasmids	Cyclins
Fimbriae, Bacterial	CDC28 Protein Kinase, S cerevisiae
	Amino Acid Sequence
	Cell Cycle Proteins

TABLE I: The MeSH terms describing a selected set of global topic clusters which consist of highly related clusters for distance d_1.

'Molecular Sequence Data' and 'Escherichia coli'. The benefit of our new graph theoretical approach is that we can visualize how much these clusters have in common and how dependent they are. We can also identify clusters that consist of different small clusters, but are highly connected.

Distance measure d_2: The results of our clustering approach with distance d_2 are shown in figure 8. The weighted graph of that clustering is now different. We got 14 clusters (Cluster 0 to 13) with documents between 2 and 5 as well as 157 and 158 documents. We now have no isolated clusters.

In this clustering it is not easy to evaluate the different topics given through the search query by evaluating the edges within the weighted graph of the clustering. Thus we have colored the graph according to the rate of documents from each query. We would expect "clean" clusters, which means the clusters should have a high fraction of documents from only one query. We see a lot of relatively clean clusters, for example 1 or 5, 2, 7 and 3. But those are not highly connected. The documents in between are mostly related to clusters which are not clearly assigned to one of both search queries. Thus we could not clearly reproduce the results from [22] with this distance measure.

Distance measure d_3: The results of our clustering approach with distance d_3 are shown in figure 9. We now have one strongly connected set of clusters. It is no longer possible to separate any of the topic clusters induced by the search query. Thus again we have colored the graph according to the fraction of documents from each query. We would expect "pure" clusters, which means the clusters should have a high fraction of documents from only one query. We get more pure clusters than with d_1 and d_2 but they are small. Most of the purest clusters are isolated and do not share documents with other clusters. Thus the result observed with d_2 gets clearer. Only those clusters which cannot be clearly assigned to one of the search queries have edges within the weighted graph of the clustering.

Since all MeSH terms are weighted equally, those terms which are not significant but shared by many of documents, are scored higher, for example 'Animals' or 'Microscopy'. And as a result, most documents have these terms in common. This explains the high connectivity of the resulting graph. Thus we could again not clearly reproduce the results from [22] with this distance measure.

VII. Conclusion and Future Work

We have shown a novel approach for document clustering considering hard clustering as well as soft clustering. We defined pseudostable sets and used the minMPS'-a approach to perform document clustering on a real-world example. We have introduced a integer linear programming and a greedy approach that gave valuable output on random instances as well as real-world data. This paper underlines that pseudostable sets have a broad application and can also be used to generalize other problems like document clustering. Since the problem is NP-complete, we could only produce and evaluate approximate solutions. Further research has to be done on evaluating the error given by the heuristics. Is it possible to find restrictions on G and B so that a solution in polynomial time is possible?

Because large graphs also increase the processing complexity, we identify the handling of such big data as an additional challenge. In the same course, it might be a good idea to

focus also on novel strategies to implement an online algorithm version of the greedy approach, which could significantly improve the scalability.

We compared three simple similarity measures using textual data given by the abstract as well as keywords. We have shown that the clustering process itself is only valuable when choosing the right similarity measure. Although we have proven that the hard clustering and soft clustering approach using pseudostable or stable sets is valid, we might need to evaluate more similarity measures. Thus further research has to be done on similarity measures. We are planning to improve document management with this novel clustering approach and do more empirical evaluation by using test sets.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 9 1999.

[2] C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[3] W. B. A. Karaa, A. S. Ashour, D. B. Sassi, P. Roy, N. Kausar, and N. Dey, "Medline text mining: an enhancement genetic algorithm based approach for document clustering," in *Applications of Intelligent Optimization in Biology and Medicine*. Springer, 2016, pp. 267–287. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-21212-8_12

[4] T. Mu, J. Y. Goulermas, I. Korkontzelos, and S. Ananiadou, "Descriptive document clustering via discriminant learning in a co-embedded space of multilevel similarities," *Journal of the Association for Information Science and Technology*, vol. 67, no. 1, pp. 106–133, 2016.

[5] L. Stanchev, "Semantic document clustering using a similarity graph," in *Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on*. IEEE, 2016, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/ICSC.2016.8

[6] L. Hirsch and A. Di Nuovo, "Document clustering with evolved search queries," 2017. [Online]. Available: http://shura.shu.ac.uk/15409/

[7] C.-J. Lee, C.-C. Hsu, and D.-R. Chen, "A hierarchical document clustering approach with frequent itemsets," *International Journal of Engineering and Technology*, vol. 9, no. 2, p. 174, 2017. [Online]. Available: http://dx.doi.org/10.7763/IJET.2017.V9.965

[8] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27 – 64, 2007.

[9] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity," *Information Processing Letters*, vol. 76, no. 4–6, pp. 175 – 181, 2000.

[10] S. O. Krumke and H. Noltemeier, *Graphentheoretische Konzepte und Algorithmen*, 2nd ed. Wiesbaden: Vieweg + Teubner, 2009.

[11] P. Hansen, M. Labbé, and D. Schindl, "Set covering and packing formulations of graph coloring: Algorithms and first polyhedral results," *Discrete Optimization*, vol. 6, no. 2, pp. 135 – 147, 2009.

[12] J. Dörpinghaus, "Über das Train Marshalling Problem," 2012. [Online]. Available: https://doi.org/10.5281/zenodo.570503

[13] A. Kosowski and K. Manuszewski, "Classical coloring of graphs," *Contemporary Mathematics*, vol. 352, pp. 1–20, 2004.

[14] D. Brélaz, "New methods to color the vertices of a graph," *Commun. ACM*, vol. 22, no. 4, pp. 251–256, Apr. 1979.

[15] J. Bhasker and T. Samad, "The clique-partitioning problem," *Computers & Mathematics with Applications*, vol. 22, no. 6, pp. 1–11, 1991.

[16] E. N. Gilbert, "Random graphs," *The Annals of Mathematical Statistics*, vol. 30, no. 4, pp. 1141–1144, 1959.

[17] P. Erdös and A. Rényi, "On random graphs, i," *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.

[18] V. Batagelj and U. Brandes, "Efficient generation of large random networks," *Phys. Rev. E*, vol. 71, p. 036113, Mar 2005.

[19] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck, "ProMiner: rule-based protein and gene entity recognition." *BMC bioinformatics*, vol. 6 Suppl 1, p. S14, 2005.

[20] E. Younesi, L. Toldo, B. Müller, C. M. Friedrich, N. Novac, A. Scheer, M. Hofmann-Apitius, and J. Fluck, "Mining biomarker information in biomedical literature," *BMC medical informatics and decision making*, vol. 12, no. 1, p. 148, 2012.

[21] I. Iliopoulos, A. Enright, and C. Ouzounis, "Textquest: Document clustering of medline," *Biocomputing 2001*, p. 384, 2000.

[22] T. Theodosiou, N. Darzentas, L. Angelis, and C. A. Ouzounis, "Pured-mcl: a graph-based pubmed document clustering methodology," *Bioinformatics*, vol. 24, no. 17, pp. 1935–1941, 2008. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btn318