PSM-Flow: Probabilistic Subgraph Mining for Reusable Workflow Fragment Identification

Ken Cheong * Daniel Garijo † William K. Cheung ‡ Yolanda Gil §

Abstract

Scientific workflows define computational processes needed for carrying out scientific experiments. Existing workflow repositories contain hundreds of scientific workflows, where scientists can find rich knowledge to facilitate workflow design for running related experiments. Identifying reusable fragments in growing workflow repositories has become increasingly important. In this paper we present PSM-Flow, a probabilistic subgraph mining algorithm designed to discover commonly occurring fragments in a workflow corpus using a modified version of the Latent Dirichlet Allocation algorithm. We propose two approaches to encode the geodesic distance between workflow steps into the model for implicitly modeling the fragments. PSM-Flow can capture variations of frequent fragments while being more efficient than graph based frequent subgraphs mining techniques, which require isomorphic test and candidate generation. For evaluation, we apply PSM-Flow to three real-world scientific workflow datasets with more than 750 workflows for neuroimaging analysis, and find it outperforms two existing frequent subgraph mining techniques. We also discuss other potential future works of the proposed method.

1 Introduction

Scientific workflows describe computational experiments which typically involve computational steps, along with the datasets used and generated by those steps. Scientific workflows are created in workflow systems that manage their execution in the required computational resources [9]. Representing workflows explicitly improves the reproducibility of scientific experiments [8].

Scientific workflow repositories contain collections of recorded scientific workflows [17]. Users may explore and reuse workflows created by others to facilitate the development of their computational experiments. While one can directly reuse an existing workflow, only a portion or fragment of a workflow is often reused. In addition, identifying commonly used fragments of workflows facilitates overviewing and exploring the contents of a workflow repository.

In [7], the authors formulated reusable workflow fragment identification as a frequent subgraph mining problem, and applied frequent subgraph mining algorithms to detect the subgraphs with high support count (number of occurrences) as candidate fragments. However, frequent subgraph mining techniques present several limitations. First, these techniques typically involve a candidate fragment generation process and a subgraph isomorphism test. Both have time and space complexities that are exponential in the worst case. Second, frequent fragments may appear in different workflows with small variations (e.g., with a node label changed, or with an additional node). Conventional frequent subgraph mining techniques use exact matching for counting fragment occurrences, and thus do not take those variations into account. Adopting stochastic models for fragment discovery can implicitly group structurally similar subgraphs together for more robust results.

To address these limitations, we propose a topic modeling approach that modifies the Latent Dirichlet Allocation (LDA) algorithm so that those latent topics to be inferred correspond to different workflow fragments. Specifically, we encode the geodesic distance of nodes in workflows into LDA and introduce a soft constraint into the model formulation so that closer nodes will have a higher probability to share the same topic label. Therefore, a topic is represented by a probability distribution of node labels in which the node labels co-occur not only frequently but also closely. Under our formulation, a topic can be interpreted as a kind of abstract subgraph, or a cluster of subgraphs in which the subgraphs are structurally similar to each other. In other words, stochastic variations of subgraphs are captured. We use two modifications of LDA. One is a straightforward modification of the Gibbs sampling procedure[10] with geodesic distance. Another one includes node embeddings into LDA for taking the geodesic dis-

*CS Department HK Baptist University Hong Kong. kencheong@gmail.com

†ISI U. of Southern California USA

‡CS Department HK Baptist University Hong Kong

§ISI U. of Southern California USA

tance into account. We have implemented both approaches in PSM-Flow (Probabilistic Subgraph Mining for Workflows)[1] for detecting reusable workflow fragments.

We evaluate our approach using three workflow corpora created by scientists using the LONI Pipeline workflow system [5] for neuroimaging analysis. We compare the quality of fragments extracted using PSM-Flow to the ones from two different subgraph mining algorithms (gSpan [19] and SUBDUE [3]) by measuring whether the extracted fragments can capture sub-workflows defined by users. Promising results are obtained.

The remainder of the paper is organized as follows. First, we introduce related work on workflow fragment mining and topic modeling in Section 2. Section 3 presents the problem formulation along with details about PSM-Flow. The experiment setup and our evaluation results can be found in Section 4. Section 5 concludes the paper with possible future work.

2 Related Work

Scientific workflow reuse is common when workflows are shared in a repository. Garijo et al.(2012) found that at least 20% of the analyzed workflows from different workflow systems and domains were composed of other workflows. Other efforts [18] confirm this practice in community workflow repositories, such as MyExperiment [17].

Graph mining algorithms have been used to automatically detect common workflow fragments in a corpus of workflows. Workflows are represented as a graph where the nodes represent computational steps and the edges correspond to the dataflow among them. Garijo et al.(2014) proposed the FragFlow framework, which makes use of two graph-based frequent subgraph mining algorithms, namely gSpan [19] and SUBDUE [3]. Other work has also applied SUBDUE to workflow corpora [4].

Frequent subgraph mining aims to extract frequent substructures in a set of graphs. It usually consists of two steps: candidate subgraph generation and subgraph isomorphism detection. These algorithms can be categorized into two types - exact and inexact match [11]. Exact match algorithms find all frequent subgraphs in a data set. Most of them perform efficiently only on sparse graphs with a large amount of labels for nodes and edges [14]. There are many exact match algorithms with different strategies for improving efficiency such as FSG [13], GASTON [16], and gSpan [19]. In contrast, inexact match algorithms calculate the similarity between two subgraphs. In these cases, a subgraph in the corpus would match a given candidate subgraph even though they are slightly different in their structure. Examples of inexact match algorithms include SUBDUE [3], GREW [14] or gApprox [2].

Our work builds on Latent Dirichlet Models (LDA) [1], first proposed for text where each document is modeled as a mixture of topics. A topic is represented as a multinomial distribution over word labels. Words that have a high probability mass in a topic tend to co-occur frequently in different documents. Griffiths and Steyvers (2004) applied Gibbs sampling for learning LDA. Here we build on their work to detect frequent workflow fragments.

A related notion similar to this work is called stochastic network motif [12]. A network motif is formed by patterns of interactions (or subgraph patterns) which appear in different parts of a network more frequently than those found in a randomized network. A stochastic network motif takes the uncertainty of edges into account, where each motif is a subgraph in which edges in the subgraph are assigned a probability that represent the absence of edges. The resulting patterns turns out to be more robust than deterministic motifs, as rare situations are also captured. Liu, Cheung, and Liu (2015) also generalize the work to multiple motifs detection.

Our method is different to stochastic network motif in several aspects. First, stochastic network motif models the absence of edges explicitly while we only encode the geodesic distance between two nodes. Also, stochastic network motif deals with relational pattern where each node in graph data is a distinct object of homogeneous type while in our workflow setting, nodes have node labels (name of computational steps) which repeatedly appear in the data set.

3 Methodology

This section presents our problem formulation and PSM-Flow, our proposed algorithm for workflow fragment identification. We leverage Latent Dirichlet Allocation (LDA) [1] which is a generative model widely used for topic modeling in a text corpus, and extend it to model reusable fragments in a workflow corpus.

3.1 Latent Dirichlet Allocation The goal of LDA is to extract topics from a text corpus. LDA assumes that each document is generated from a mixture of topics called *topic distribution*, and each topic is represented as a distribution of words called *word distribution*. The graphical representation of LDA is shown in Figure 1. Let θ_d be the topic distribution of document d, ϕ_k the word distribution of topic k, z_n the topic assignment of word instance n and w_n the word label of word instance n, and α and β the hyperparameters for the conjugate priors (Dirichlet) of the multinomial distribu-

[1]Available at https://doi.org/10.5281/zenodo.888391

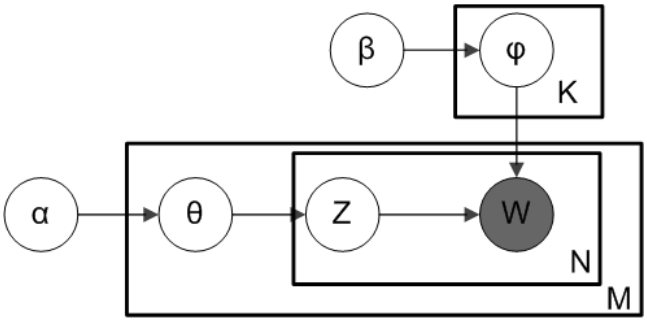

Figure 1: Graphical representation of the LDA model

tion over topics and words respectively. The generative process of LDA is as follows:

1. Draw ϕ_k from Dir(β) for each topic.
2. Draw θ_d from Dir(α) for each document.
3. Draw topic z from Multi(θ_d) for each word instance.
4. Draw word w from Multi(ϕ_k) for each word instance.

where $Dir()$ and $Multi()$ denote the Dirichlet distribution and multinomial distribution respectively. We use Gibbs sampling to infer the model parameters, which is a Markov chain Monte Carlo algorithm for estimating the posterior probability [10].

PSM-Flow extends LDA as follows. We consider a "workflow" as a "document", and the type (label) of a "workflow step" as a "word". Instead of assuming that generation of word instances are independent given the topic as in LDA, PSM-FLow assumes that "nearby" node instances in a workflow tend to have identical topic labels (as detailed in the next section). Therefore, those node instances assigned with the same topic are more likely to be connected as subgraphs in a workflow. We use the word "fragment" to refer to "subgraph" in a workflow.

3.2 Encoding Geodesic Distance in PSM-Flow

The fundamental idea of PSM-Flow is adding a soft constraint to the Gibbs sampling of LDA so that if the distance of the shortest (undirected) path (i.e., geodesic distance) between two nodes in a workflow is smaller, there will be a higher probability that the two nodes share the same topic label. Therefore, nodes sharing a topic will tend to be grouped together. As an illustration, Figure 2 shows a simple example with three workflows in the corpus where nodes with the same labels (indicated by colors) inferred using PSM-Flow do co-occur in all three workflows and are connected in each

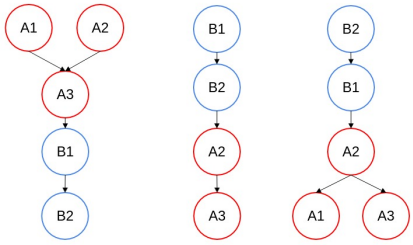

Figure 2: A synthetic corpus with three workflows. The colors show different topics inferred for the nodes.

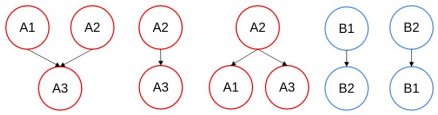

Figure 3: Fragments extracted from Figure 2. Each color represents a different topic.

workflow. We can then extract all the connected nodes with the same topic labels as the workflow fragments for reuse. Notice that a topic may end up with multiple fragments as shown in Figure 3. Fragments under the same topic vary with each other by having some different labels or some topological differences.

In our problem formulation, we incorporate the geodesic distance between nodes only and ignore other topological information of the workflow. We argue that this simplification suffices for scientific workflows. For example, assume that D1 is a data filtering component and D2 is a data analysis component. It will be natural for D1 to be followed by D2 and it is less likely to have D2 to be followed by D1. At least, it is less likely for the output format of D2 to be consistent with the input format required by D1.

In the following sections, we present two different approaches for encoding the geodesic distance in PSM-Flow. The first approach modifies directly the Gibbs sampling formula. The second approach makes use of the embedding method to modify the LDA model where the corresponding Gibbs sampling procedure is derived.

3.3 Distance Encoded Gibbs Sampling

One key step in the Gibbs sampling algorithm for LDA is to sample a topic label for word instance j at each iteration based on the following conditional probability:

$$P(z_i = j | z_{-i}, w_i, d_i, ..; \alpha, \beta) \propto \frac{C^{WT}_{w_i,j} + \beta}{\sum_W^{w=1} C^{WT}_{w,j} + W\beta} \frac{C^{DT}_{d_i,j} + \alpha}{\sum_T^{t=1} C^{DT}_{d_i,t} + T\alpha} \tag{3.1}$$

Symbol	Description
D	# of workflows or documents in a corpus.
W	# of distinct node labels or words in the vocabulary.
N	Total number of node or word instances in the corpus.
T	Total number of topics.
w_i	i-th observed node label or word.
d_i	the workflow (document) that contains the i-th observed node (word) instance.
z_i	topic label of w_i.
C_{wk}^{WT}	Count of node label (word) w assigned to topic k.
$C_{d_ik}^{DT}$	Count of topic label k in a workflow (document) d_i.
ϕ_k	Distribution of node labels (words) for topic k.
θ_{d_i}	Distribution of topics for document d_i.

Table 1: Notation used in the paper.

where the notations adopted are tabulated in Table 1. LDA assumes that word instances in a document are generated independently given a topic assignment.

Our first attempt is to modify directly the original Gibbs sampling formula by generalizing the second factor (i.e., the topic portion in document d_i) with a similarity function, given as:

(3.2)

$$P(z_i = j|z_{-i}, w_i, d_i, ..; \alpha, \beta) \propto \frac{C_{w_i,j}^{WT} + \beta}{\sum_W^{w=1} C_{w,j}^{WT} + W\beta} \cdot \frac{\sum_{\forall l \in NS(d_i)} s(i,l) I(tp(l) = j) + \alpha}{\sum_{\forall l \in NS(d_i)} s(i,l) + T\alpha}$$

where $NS(d_i)$ is the set of nodes in d_i, $tp(l)$ is the topic id for node l, $s(i,l)$ is a similarity function defined based on the geodesic distance between nodes i and l. There are many choices for $s(i,l)$, such as the reciprocal of the geodesic distance. Intuitively, the $s(i,l)$ function counts the contribution of neighboring nodes to the topic update of the current node. The closer the two nodes, the higher is the contribution of the neighbor node. It is worth mentioning that when $s(i,l)$ is constant (i.e., all neighboring nodes have the same contribution), equation 3.2 degenerates back to the original Gibbs sampling formula.

3.4 Gaussian Cluster LDA The geodesic distance is explicitly encoded in our first approach. For our second approach, we make use of the geodesic distance between each pair of nodes in the workflow to compute a dissimilarity matrix δ. Then, we apply multidimensional scaling to δ to obtain the vector representations (embedding) $\{x_j \in \Re^p\}$ for each node j, so that $\min_{x_1...x_I} \sum_{i<j} (||x_i - x_j|| - \delta_{dij})^2$. So, in principle, this will make neighboring nodes to have similar embeddings.

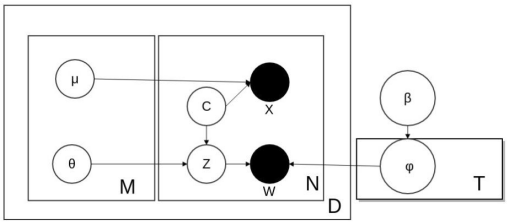

Figure 4: Gaussian cluster LDA

Given the node embeddings, we define a new graphical model as shown in Figure 4. We introduce the notion of Gaussian clusters for modeling clustered node embeddings to indicate that they often co-occur as neighbors in the workflow corpus. Let μ_i be the mean of Gaussian cluster i, θ_i the topic of Gaussian cluster i, c_j the index of Gaussian cluster that generates node j, $x_j \in \Re^p$ the embedding of node j, d_j the workflow that contains node j, and d_i the workflow that contains Gaussian cluster i.

The generative process of this new graphical model is given as follows:

1. Choose $\phi_i \sim Dir(\beta)$, where $i \in \{1, ..., T\}$ to get node label distribution of T topics.

2. For each workflow d, generate parameters for M_d Gaussian clusters. For Gaussian cluster i where $i \in \{1...M_d\}$, μ_i is drawn from the Gaussian distribution $N(0, I)$ where I is the identity matrix,[2] and the topic label θ_i is drawn uniformly from a set of T topics.

3. Generate a set of N_d nodes in workflow d. For each node j where $j \in \{1...N_d\}$, we first draw a Gaussian cluster label (c_j), and then we draw an embedding of (x_j) from $N(\mu_{c_j}, I)$. We then set topic label z_j to be the same as the topic label of c_j (θ_{c_j}), and draw node label w_i from Multi(ϕ_{z_j}).

To learn the model using Gibbs sampling, the following posterior probabilities of the latent variables

[2]We adopt the identity matrix to make it computationally more efficient.

μ,θ,z,c are to be used for the sampling.

$$P(\boldsymbol{\mu_i}|\boldsymbol{x},\boldsymbol{c}) \sim N(\boldsymbol{\mu_{base}},\boldsymbol{\sigma_{base}})$$
$$\boldsymbol{\mu_{base}} = \sum_j \frac{\boldsymbol{x_j} I\{\text{node j belongs to cluster i}\}}{n_i}$$
$$\boldsymbol{\sigma_{base}} = I * (n_i + 1)^{-1}$$

where n_i is the number of points in cluster i.

$$P(c_j = i|w_j,\boldsymbol{x_j},\boldsymbol{\theta},\boldsymbol{\mu}) \propto P(w_j|\theta_i)\cdot P(\boldsymbol{x_j}|\boldsymbol{\mu_i})\cdot P(d_j,d_i)$$
$$P(w_j|\theta_i) \propto \frac{C^{WT}_{w_j,\theta_i} + \beta}{\sum_W^{w=1} C^{WT}_{w_j,\theta_i} + W\beta}$$
$$P(\boldsymbol{x_j}|\boldsymbol{\mu_i}) \propto \frac{exp(-\frac{1}{2}(\boldsymbol{x_j}-\boldsymbol{\mu_i})^T(\boldsymbol{x_j}-\boldsymbol{\mu_i}))}{\sqrt{2\pi}}$$
$$P(d_j,d_i) \propto I\{d_j = d_i\}$$
$$P(\theta_i = t|\boldsymbol{w}) \propto \prod_{\forall j \in \text{ cluster } i} \frac{C^{WT}_{w_j,t} + \beta}{\sum_W^{w=1} C^{WT}_{w,t} + W\beta}$$
$$z_j = \theta_{c_j}.$$

In our experiment, we set the number of Gaussian clusters for each workflow as the number of nodes in each workflow, which is more than enough and eventually only some of the clusters will be sampled. Also, we need to set the dimension of node embedding. In general, the higher the dimension, the better the distance information can be preserved but at the expense of the computational complexity.

3.5 Extracting Frequent Fragments from Topic Assignments Using either one of the proposed approaches for inferring in PSM-Flow, the output will be the assignments of different latent topic labels to all the nodes in all the workflows of a given corpus. To obtain the reusable fragments, we partition the workflow corpus by topic labels as shown in Algorithm 3.1. The algorithm removes all edges in a workflow corpus which are incident with nodes with different topic labels. The final output produces a set of connected components where all nodes in each connected component have the same topic. Thus they form the set of reusable workflow fragments.

In some cases, there are outlier nodes which have topics different from the topic shared by all their neighbors, which in turn will break a fragment into multiple pieces. In order to handle this situation, before applying Algorithm 3.1, we detect such outliers and flip the topic the outlier node to have the same topic of their neighbors'.

After extracting the fragment instances, we filter them to obtain more meaningful fragments. For example, we remove all the repeated or one-step fragments in the extracted fragment set, as we are interested in fragments with at least two steps.

ALGORITHM 3.1. function PartitionCorpus(G,Topic)
G, the workflow corpus
Topic, topic labels of nodes
for $(inNode, outNode) \in$ *edges in G* **do**
  if $Topic(inNode) \neq Topic(outNode)$ **then** remove edge (inNode,outNode) in G
  end if
end for
return *Partition of G*

3.6 Hierarchical extraction of fragments It is common that users reuse subgraphs of frequent fragments rather than the fragments directly because different tasks may prefer fragments at different granularity levels. Such insight suggests the opportunity of extending the method to hierarchically extracting fragments.

We tried a particular top-down approach to apply PSW-Flow to the corpus level by level. At each level, two topics are first learned and corpus is divided to two new corpora and then PSW-Flow is applied to these new corpora recursively. Fragments extracted at all levels are collected as the final fragment set.

Algorithm 3.2 implements this idea by making use of PSW-Flow as subroutine to extract fragments level by level so that sub-fragments which occur frequently inside other fragments will also be extracted.

ALGORITHM 3.2. function HierarchicalExtraction(G,l,L)
G, the workflow corpus
l, current levels
L, maximum number of levels
Fragments, set of extracted fragments
Topic, topic labels of nodes
if $l > L$ **then return** $\emptyset$
end if
Topic $\leftarrow$ *Sample from PSW-Flow with two topics*
Glist=SeparateFragments(PartitionCorpus(G,Topic),Topic,2)
for $g \in$ *Glist* **do**
  Fragments=Fragments $\bigcup$ HierarchicalExtraction(g,l+1,L)
end for
return *Fragments*

ALGORITHM 3.3. function SeparateFragments(F,Topic,K)
F, set of extracted fragments
Topic, topic labels of fragments
K, number of topics
Glist$\leftarrow$ *list of* $\emptyset$ *with size K*

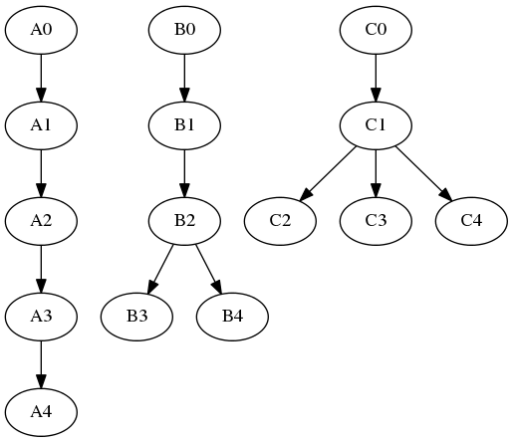

Figure 5: Base graphs for generating a synthetic dataset

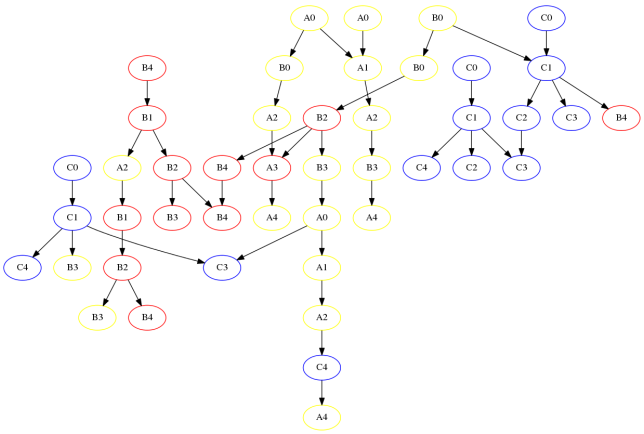

Figure 6: A sample graph where each base subgraph in Figure 5 repeats three times with one node label change. The different colors represent those topics learned by PSM-Flow.

for $f \in F$ **do**
  $Glist[Topic(f)] = Glist[Topic(f)] \bigcup f$
end for
return *Glist*

4 Experiment Setup

In order to evaluate the effectiveness of our approach and compare the performance of PSM-Flow against frequent graph mining techniques, we test PSM-Flow on synthetic data and apply our approach to a data set with scientific workflows created by scientists in their day to day work.

4.1 Synthetic data We generate a synthetic corpus from the three base subgraphs shown in Figure 5. Each base subgraph is repeatedly generated several times to obtain multiple subgraph instances. Also, for each subgraph instance, a randomly selected node label is flipped to a randomly chosen node label to form different variants of base subgraph. All subgraph instances are randomly connected together to form one big connected graph, as shown in Figure 6.

After extracting all fragments, we compare them to variants of subgraph instances in the corpus to obtain precision and recall. In order to see how the extracted subgraph from our approach can recover the subgraph instances underlying in the graph corpus, we test our method with different repetitions of base subgraphs as shown in Table 2 (only results from distance encoded Gibbs sampling are shown as gaussian cluster LDA are very similar). For each repetition time setting, we run our algorithm ten times to get mean and standard deviation (shown in parentheses).

The results show low precision and recall. This is due to several reasons. For example, if subgraph instances from same topic happen to be connected together, the algorithm will detect them as one subgraph instance rather than multiple subgraphs. Also, if the change of label appears in the tail or head of subgraph instance, it is difficult for algorithm to detect this variation as part of a subgraph.

4.2 Workflow Corpora We use three workflow corpora from LONI Pipeline [5], a widely-used workflow system for neuroimaging analysis. These corpora were used to evaluate gSpan and SUBDUE in [7]:

1. Workflow corpus 1 (WC1): A set of 441 workflows designed mostly by a single user. Some of the workflows are products of collaboration with other users. The domain of the workflows is in general medical imaging (brain image understanding, 3D skull imaging, genetic modeling of the face, etc.).
2. Workflow corpus 2 (WC2): A set of 94 workflows from one user, sometimes done in collaboration with others. Most of the workflows have been made public.
3. Workflow corpus 3 (WC3): A set of 269 workflows, submitted to the LONI pipeline for execution by 62 different users.

4.3 Workflow Data When designing workflows, users tend to consider several related computational steps as a unit. Here we refer to this unit as a *user grouping*. User groupings tend to occur frequently in workflow corpora as users reuse them. In order to assess PSM-Flow, we compare the groupings created by

repeat time	5	10	15	20	25
precision	0.26(0.14)	0.36(0.09)	0.33(0.09)	0.37(0.08)	0.36(0.06)
recall	0.25(0.14)	0.32(0.08)	0.29(0.08)	0.33(0.07)	0.31(0.05)

Table 2: Evaluation on synthetic data set

users of LONI Pipeline for the workflows in the evaluation corpora to the fragments automatically extracted by PSM-Flow, following a similar approach in [7].

We compare PSM-FLow to two popular frequent graph mining techniques, namely, gSpan and SUBDUE. GSpan uses an exact match and depth first search strategy to discover all possible frequent fragments, while SUBDUE uses inexact match approximation to hierarchically discover a set of frequent fragments. For SUBDUE, two heuristics, Minimum Description Length (MDL) and size are used for hierarchically reducing the search space when mining for fragments [3].

For a fair comparison, we filter the candidate fragment set proposed by gSpan and SUBDUE by removing single-step fragments and non-closed fragments (i.e., all those fragments that are included in another fragment with the same number of occurrences). For PSM-Flow, we filter outlier nodes before each partition and we remove single step and duplicate fragments.

4.4 Evaluation Metrics Since our extracted fragments are in many cases highly similar to user groupings but not exactly the same, we define *soft version* of precision and recall metrics for the measuring the performance. Let F denote the extracted fragments set, U the user grouping set, and $|X|$ the cardinality of the set X.

$$\text{Soft precision} = \frac{\sum_{f \in F} \max_{u \in U} overlap(f,u)}{|F|}$$

$$\text{Soft recall} = \frac{\sum_{u \in U} \max_{f \in F} overlap(f,u)}{|U|}$$

where the overlap between a fragment and a grouping is defined as the number of common node labels between them. Thus, the overlap value will be one if the fragment and the grouping are the same. Notice that the overlap value will be high as far as the node labels are the same, even though they may have different topologies. The validity of the proposed metric relies on the assumption that in a workflow corpus node labels in a fragment imply a certain topology (due to the compatibility between workflow steps). To take both precision and recall into account, we also calculate the F score for comparison where $F\text{-}score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$.

4.5 Evaluation Results Table 3 shows our evaluation results. For PSM-Flow, the number of topics are empirically selected to be 20, 35 and 30 for WC1, WC2 and WC3 respectively. For hierarchical PSM-Flow (HPSM-Flow), the maximum level is set as 5 for corpus 2 and 3 and set as 3 for corpus 1. Also, we run PSM-Flow for ten times for each data set and report the average and standard deviation (shown in parenthesis) of performance measures including precision, recall and the number of fragments for all corpora. In addition, the frequency of gSpan is normalized according to the size of the dataset (i.e., we show the percentage instead of the number of occurrences). The distance encoded Gibbs sampling (DEGS) and Gaussian Cluster LDA (GCLDA) tend to perform similarly in all corpora. In terms of $F-score$, PSM-Flow performs best in WC1 and WC3 when removing all candidate fragments with a single occurrence.

PSM-Flow outperforms the rest of the algorithms in terms of recall as variations of fragments are captured. But it performs slightly worse than gSpan in terms of precision. This may be due to the exact matching nature of gSpan. One limitation of PSM-Flow is that when fragments within same topic are connected together, the algorithm will detect them as a single fragment instead of multiple ones.

For F-score, HPSM-Flow tends to perform better in two of corpus. This implies that the hierarchical nature of the algorithm can better capture fragments of different granularity levels, giving better overall results.

5 Conclusions And Future Work

In this paper we introduced PSM-Flow, a probabilistic subgraph mining algorithm which adopts a topic modeling approach for robust frequent fragment mining in scientific workflows. We implemented two versions of PSM-Flow, one based on Gibbs sampling and another one based on Gaussian cluster LDA. Both implementations tend to improve the robustness of discovered fragments by capturing their variations in a probabilistic framework. Promising results have been obtained using both synthetic and real workflow data. For future work, we are exploring how to further extend the evaluation of PSM-Flow in other graph mining tasks such as social networks. Also, another direction is to extend PSM-Flow with Bayesian non-parametric methods, which can

Corpus	Approach	Frequency	Fragment number	Precision	Recall	F score
WC1	SUBDUE MDL	2 occur	264	0.506	0.704	0.589
	SUBDUE SIZE	2 occur	381	0.471	0.749	0.578
	gSpan	2.00%	637	0.677	0.645	0.661
		10.00%	110	**0.714**	0.385	0.5
		15.00%	33	0.689	0.213	0.325
	PSM-Flow DEGS	1 occur	747.3(16.2)	0.548(0.007)	0.784(0.017)	0.645(0.01)
		2 occur	474.3(28.9)	0.600(0.008)	0.760(0.016)	**0.671(0.01)**
	PSM-Flow GCLDA	1 occur	834.3(31.1)	0.540(0.01)	**0.790(0.019)**	0.642(0.012)
		2 occur	546.3(13.2)	0.586(0.02)	0.761(0.017)	0.662(0.015)
	HPSM-Flow	1 occur	925.7(65)	0.509(0.009)	0.788(0.009)	0.618(0.009)
WC2	SUBDUE MDL	2 occur	95	0.366	0.435	0.398
	SUBDUE SIZE	2 occur	88	0.383	0.469	0.422
	gSpan	2.00%	127	0.545	0.44	0.487
		10.00%	14	**0.559**	0.151	0.238
		15.00%	2	0.389	0.062	0.107
	PSM-Flow DEGS	1 occur	154.5(5.39)	0.347(0.014)	0.497(0.016)	0.408(0.013)
		2 occur	70.2(7.79)	0.491(0.029)	0.446(0.025)	0.467(0.024)
	PSM-Flow GCLDA	1 occur	215.5(8.81)	0.345(0.011)	0.561(0.019)	0.427(0.013)
		2 occur	84.7(4.27)	0.454(0.018)	0.471(0.019)	0.462(0.018)
	HPSM-Flow	1 occur	387.5(40.2)	0.392(0.014)	**0.655(0.019)**	**0.490(0.016)**
WC3	SUBDUE MDL	2 occur	186	0.237	0.487	0.319
	SUBDUE SIZE	2 occur	178	0.217	0.456	0.294
	gSpan	2.00%	108	**0.391**	0.444	0.416
		5.00%	29	0.255	0.052	0.086
		10.00%	9	0.226	0.016	0.029
	PSM-Flow DEGS	1 occur	327.8(8.07)	0.320(0.009)	0.614(0.019)	0.420(0.017)
		2 occur	159.0(5.44)	0.374(0.011)	0.519(0.013)	0.435(0.009)
	PSM-Flow GCLDA	1 occur	345.8(4.07)	0.306(0.005)	0.609(0.014)	0.407(0.007)
		2 occur	171.9(7.31)	0.363(0.004)	0.518(0.012)	0.427(0.005)
	HPSM-Flow	1 occur	509.5(27.8)	0.348(0.011)	**0.668(0.019)**	**0.458(0.013)**

Table 3: Comparison results between PSM-Flow, SUBDUE and gSpan on three workflow corpora

help automatically learn the number of fragments that better describe the workflow corpus. In addition, hierarchical version of PSM-Flow may provide new insights on the organization of a workflow corpus worth further investigation.

References

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3 (Mar. 2003), pp. 993–1022.

[2] C. Chen et al. "gApprox: Mining Frequent Approximate Patterns from a Massive Network". In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. Oct. 2007, pp. 445–450.

[3] Diane J. Cook and Lawrence B. Holder. "Substructure Discovery Using Minimum Description Length and Background Knowledge". In: *J. Artif. Int. Res.* 1.1 (Feb. 1994), pp. 231–255.

[4] Claudia Diamantini, Domenico Potena, and Emanuele Storti. "Mining Usage Patterns from a Repository of Scientific Workflows". In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. SAC '12. Trento, Italy: ACM, 2012, pp. 152–157.

[5] Ivo D. Dinov et al. "Efficient, Distributed and Interactive Neuroimaging Data Analysis Using the LONI Pipeline". In: *Frontiers in Neuroinformatics*. Vol. 3. 22. 2009.

[6] D. Garijo et al. "Common motifs in scientific workflows: An empirical analysis". In: *2012 IEEE 8th International Conference on E-Science*. Oct. 2012, pp. 1–8.

[7] D. Garijo et al. “FragFlow Automated Fragment Detection in Scientific Workflows”. In: *2014 IEEE 10th International Conference on e-Science*. Vol. 1. Oct. 2014, pp. 281–289.

[8] Yolanda Gil. “From Data to Knowledge to Discoveries: Artificial Intelligence and Scientific Workflows”. In: *Sci. Program.* 17.3 (Aug. 2009), pp. 231–246.

[9] Yolanda Gil et al. “Wings: Intelligent Workflow-Based Design of Computational Experiments”. In: *IEEE Intelligent Systems*. 2011.

[10] T. L. Griffiths and M. Steyvers. “Finding scientific topics”. In: *Proceedings of the National Academy of Sciences* 101.Suppl. 1 (Apr. 2004), pp. 5228–5235.

[11] Chuntao Jiang, Frans Coenen, and Michele Zito. “A Survey of Frequent Subgraph Mining Algorithms”. In: 000 (Jan. 2004), pp. 1–31.

[12] Rui Jiang et al. “Network motif identification in stochastic networks”. In: *Proceedings of the National Academy of Sciences* 103.25 (2006), pp. 9404–9409.

[13] M. Kuramochi and G. Karypis. “An efficient algorithm for discovering frequent subgraphs”. In: *IEEE Transactions on Knowledge and Data Engineering* 16.9 (Sept. 2004), pp. 1038–1051.

[14] M. Kuramochi and G. Karypis. “GREW - a scalable frequent subgraph discovery algorithm”. In: *Data Mining, 2004. ICDM ’04. Fourth IEEE International Conference on*. Nov. 2004, pp. 439–442.

[15] Kai Liu, William K. Cheung, and Jiming Liu. “Detecting multiple stochastic network motifs in network data”. In: *Knowledge and Information Systems* 42.1 (Jan. 2015), pp. 49–74.

[16] Siegfried Nijssen and Joost N. Kok. “A Quickstart in Frequent Structure Mining Can Make a Difference”. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’04. Seattle, WA, USA: ACM, 2004, pp. 647–652. ISBN: 1-58113-888-1.

[17] David De Roure, Carole A. Goble, and Robert Stevens. “The design and realisation of the myExperiment Virtual Research Environment for social sharing of workflows”. In: *Future Generation Comp. Syst.* 25.5 (2009), pp. 561–567.

[18] Johannes Starlinger, Sarah Cohen-Boulakia, and Ulf Leser. “(Re)Use in Public Scientific Workflow Repositories”. In: *Scientific and Statistical Database Management*. Ed. by Anastasia Ailamaki and Shawn Bowers. Vol. 7338. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 361–378.

[19] Xifeng Yan and Jiawei Han. “gSpan: graph-based substructure pattern mining”. In: *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* 2002, pp. 721–724.