

PSM-Flow: Probabilistic Subgraph Mining for Discovering Reusable Fragments in Workflows

Ken Cheong
CS Department
HK Baptist University
Hong Kong

Daniel Garijo
Information Sciences Institute
U. of Southern California
USA

William K. Cheung
CS Department
HK Baptist University
Hong Kong

Yolanda Gil
Information Sciences Institute
U. of Southern California
USA

Abstract—Scientific workflows define computational processes needed for carrying out scientific experiments. Existing workflow repositories contain hundreds of scientific workflows, where scientists can find materials and 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. The proposed model encodes the geodesic distance between workflow steps into the model for implicitly modeling fragments. PSM-Flow captures variations of frequent fragments while maintaining its space complexity bounded polynomially, as it requires no candidate generation. We applied PSM-Flow to three real-world scientific workflow datasets containing more than 750 workflows for neuroimaging analysis. Our results show that PSM-Flow outperforms three state of the art frequent subgraph mining techniques. We also discuss other potential future improvements of the proposed method.

I. 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 [11]. Representing workflows explicitly improves the reproducibility of scientific experiments [10].

Scientific workflow repositories contain collections of recorded scientific workflows [22]. 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 [9], the authors formulated reusable workflow fragment identification as a frequent subgraph mining problem, and applied frequent subgraph mining algorithms (FSM) 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 present a time complexity (combinatorial exploration of candidate fragments)

and a space complexity (large number of candidate subgraphs are generated in memory) that are exponential in the worst case. Second, frequent fragments may appear in different workflows with small variations (e.g., with changes in node labels, 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. In other words, we create a higher level abstraction for grouping fragments based on their commonality which makes the discovery of infrequent but potentially useful fragments (e.g., similar to those frequent fragments) possible.

To address these limitations, we propose PSM-Flow (Probabilistic Subgraph Mining for Reusable Workflow Fragment Identification) a topic modeling approach that modifies the Latent Dirichlet Allocation (LDA) algorithm so that those latent topics to be inferred correspond to different groups of 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.

When compared to most of frequent subgraph mining techniques, which take exponential space in the worst case, PSM-Flow has the advantage that the required space is polynomially bounded by $O(N^2)$ where N is the number of nodes (computational steps) in the workflow corpus.

We evaluate our approach using three workflow corpora created by scientists using the LONI Pipeline workflow system [6] for neuroimaging analysis. We compare the quality of fragments extracted using PSM-Flow to the three different subgraph mining algorithms (gSpan [25], SUBDUE [4] and READUM[18]) 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 a discussion future lines of work.

II. RELATED WORK

Scientific workflow reuse is common when workflows are shared in a repository. Garijo et al.(2012) [8] report that at least 20% of the analyzed workflows from different workflow systems and domains were composed of other workflows. Other efforts [23] confirm this practice in community workflow repositories, such as MyExperiment [22].

Frequent subgraph 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) [9] proposed the FragFlow framework, which makes use of two graph-based frequent subgraph mining algorithms, namely gSpan [25] and SUBDUE [4]. Other work has also applied SUBDUE to workflow corpora [5]. In general, 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 [13]. Exact match algorithms extract 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 [17]. But there also exist some efficient exact match algorithms such as FSG [16], GASTON [20], and gSpan [25]. In contrast, inexact match algorithms consider similarity between subgraphs and allow a subgraph in the corpus to match a given candidate subgraph even though they have slight structural differences. Examples of inexact match algorithms include SUBDUE [4], GREW [17], gApprox [3] and REAFUM[18].

In this paper, we formulate the workflow fragment discovery problem based on Latent Dirichlet Allocation (LDA) [2] which is a widely used model for topic discovery. LDA was 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. To determine the number of topics automatically, Dirichlet process(DP) is a Bayesian nonparametric tool [21] which has been adopted in topic modeling.[24]

A related notion similar to this work is called stochastic network motif [14]. 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 with probability values representing the presence of the edges. The resulting

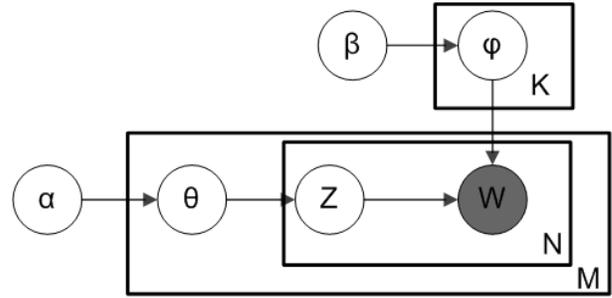


Fig. 1: Graphical representation of the LDA model

patterns turn 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 from 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. In addition, stochastic network motif deals with relational patterns where each node in graph data is a distinct object of homogeneous type. In our workflow setting, nodes have node labels (name of computational steps) which repeatedly appear in the data set.

III. METHODOLOGY

This section presents our problem formulation and PSM-Flow, our proposed algorithm for workflow fragment identification. We leverage Latent Dirichlet Allocation (LDA) [2] 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.

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

- 1) Draw ϕ_k from $\text{Dir}(\beta)$ for each topic.
- 2) Draw θ_d from $\text{Dir}(\alpha)$ for each document.
- 3) Draw topic z from $\text{Multi}(\theta_d)$ for each word instance.
- 4) Draw word w from $\text{Multi}(\phi_k)$ for each word instance.

where $\text{Dir}()$ and $\text{Multi}()$ 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 [12].

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

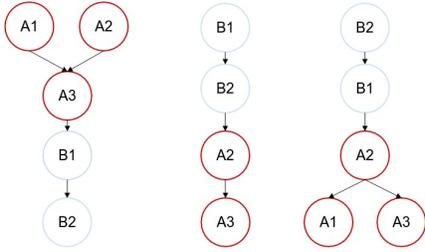


Fig. 2: A synthetic corpus with three workflows. The colors show different topics inferred for the nodes.

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. A topic, hence, is a distribution of node labels in which most probability mass is concentrated on those frequently co-occur labels. A fragment F belongs to a topic T means that all topic assignments of node instances from F are equal to topic T .

B. Encoding Geodesic Distance in PSM-Flow

The fundamental idea of PSM-Flow is adding a soft constraint to 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 a cluster which corresponds to a fragment.

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 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 for simplicity reasons. We argue that this simplification suffices for scientific workflows. For example, let’s 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 other words, even if the order of the workflow steps is ignored in our model, the effectiveness of the fragment extraction will not be significantly affected.

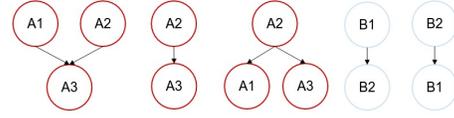


Fig. 3: Fragments extracted from Figure 2. Each color represents a different topic.

Symbol	Description
D	# of workflows in a corpus.
W	# of distinct node labels or words in the vocabulary.
N	Total number of nodes in the corpus.
T	Total number of topics.
M_d	# of Gaussian clusters in the workflow d .
G	Global topic distribution
H	Base distribution of Dirichlet process
w_i	i -th observed node label.
d_i	Workflow that contains the i -th observed node.
z_i	Topic label of w_i .
x_i	Embedding of node i .
c_i	Index of Gaussian cluster that generates node i
μ_i	Mean of Gaussian cluster i
C_{wk}^{WT}	Count of node label w assigned to topic k .
$C_{d_i k}^{DT}$	Count of topic label k in a workflow d_i .
ϕ_k	Distribution of node labels for topic k .
θ_{d_i}	Distribution of topics for document d_i .
D_{ab}	Geodesic distance between nodes a and b .

TABLE I: A summary of notations.

In the following sections we present PSM-Flow, which makes use of the embedding method to modify the LDA model where the corresponding Gibbs sampling procedure is derived. The notations we use in the paper are shown in Table I.

C. PSM-Flow

The graphical model of PSM-Flow can be seen in Figure 4. To encode the constraint defined in Section III-B, we assume that each node in the workflow has a corresponding node embedding which preserves the geodesic distance of nodes. 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. A Gaussian cluster is a multivariate Gaussian distribution with an associated topic label θ . The cluster generates embeddings for nodes and all nodes in the same Gaussian cluster share the same topic label as the cluster’s associated topic label θ . To determine the number of topics needed for the PSM-Flow model, we make use of Dirichlet Process (DP) to automatically learn the optimal number of topics. The use of DP for learning the optimal number of topics has been shown effective for LDA.

The generative process of PSM-Flow is :

- 1) Choose $\phi_i \sim Dir(\beta)$, where $i \in \{1, \dots, \infty\}$ to get node label distribution.
- 2) Choose $G \sim DP(\tau H)$.
- 3) For each workflow d , generate parameters for M_d Gaussian clusters. For Gaussian cluster i where $i \in \{1 \dots M_d\}$, μ_i is drawn from the Gaussian distribution $N(0, I)$

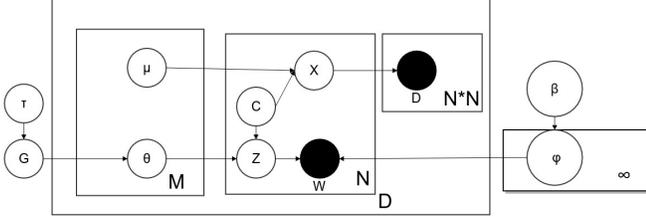


Fig. 4: PSM-Flow Model

where I is the identity matrix and the topic label θ_i is drawn from G .

- 4) Generate a set of N_d nodes in workflow d . For each node j where $j \in \{1 \dots N_d\}$, we first draw a Gaussian cluster label (c_j), and then draw an embedding of (x_j) from $N(\mu_{c_j}, I)$.¹ 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 $\text{Multi}(\phi_{z_j})$. The distance variable for nodes i and j D_{ij} is drawn from $N(\|x_i - x_j\|, 1)$ where $\|\cdot\|$ is the euclidean distance function.

To learn the model using Gibbs sampling, the following posterior probabilities of the latent variables μ, θ, z, c, x are to be used for the sampling.

$$P(\mu_i | x, c) \sim N(\mu_{base}, \sigma_{base})$$

$$\mu_{base} = \sum_j \frac{x_j I \{ \text{node } j \text{ belongs to cluster } i \}}{n_i}$$

$$\sigma_{base} = I * (n_i + 1)^{-1}$$

where n_i is the number of points in cluster i .

$$P(c_j = i | w_j, x_j, \theta, \mu) \propto P(w_j | \theta_i) \cdot P(x_j | \mu_i) \cdot P(d_j, d_i)$$

$$P(w_j | \theta_i) \propto \frac{C_{w_j, \theta_i}^{WT} + \beta}{\sum_{w=1}^W C_{w, \theta_i}^{WT} + W\beta}$$

$$P(x_j | \mu_i) \propto \frac{\exp(-\frac{1}{2}(x_j - \mu_i)^T (x_j - \mu_i))}{\sqrt{2\pi}}$$

$$P(d_j, d_i) \propto I\{d_j = d_i\}$$

$$P(\theta_i = t | w) \propto \begin{cases} \frac{\tau}{n+\tau}, & \text{if } t \text{ is a new topic} \\ \frac{\sum_{w=1}^W C_{w, t}^{WT}}{n+\tau} \prod_{\forall j \in \text{cluster } i} \frac{C_{w_j, t}^{WT} + \beta}{\sum_{w=1}^W C_{w, t}^{WT} + W\beta} \end{cases}$$

$$z_j = \theta_{c_j}$$

n is total number word tokens in corpus. The embedding x is sampled according to

$$P(x_i | x_{-i}, c_i, \mu_{c_i}, D, \dots)$$

$$\propto P(x_i | \mu_{c_i}) P(x_{-i}, D | x_i)$$

$$\propto P(x_i | \mu_{c_i}) \prod_{j \neq i \wedge d_j = d_i} N(|x_j - x_i|, 1)$$

¹We adopt the identity matrix to make it computationally more efficient.

In our experiment, we set the number of Gaussian clusters for each workflow as the number of nodes in each workflow. We consider this to be enough, as eventually only some of the clusters will be sampled. Also, we need to set the dimension for our node embedding. In general, the higher the dimension, the better the distance information can be preserved but at the expense of the computational complexity. In our experiments, we empirically selected 10 as the dimension which is enough for preserving distance.

D. Extracting Frequent Fragments After Topic Assignment

The output of Gibbs sampling is 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 1. The algorithm removes all edges which link up nodes with different topic labels in a workflow corpus. 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 1, we detect such outliers and replace the topic of the outlier nodes to have the same topic as their neighbors'.

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

Algorithm 1 Extracting fragments from a workflow corpus with topic labels

function PartitionCorpus(G, Topic)

Input: G , the workflow corpus

Output: Topic, topic labels of nodes

```

for ( $inNode, outNode$ )  $\in$  edges in  $G$  do
  if  $\text{Topic}(inNode) \neq \text{Topic}(outNode)$  then
    remove edge ( $inNode, outNode$ ) in  $G$ 
  end if
end for
return Partition of  $G$ 

```

A prototype implementation of PSM-Flow is available online [15].

IV. EXPERIMENT SETUP

We use three workflow corpora from LONI Pipeline [6], a widely-used workflow system for neuroimaging analysis. These corpora were used to evaluate gSpan and SUBDUE in [9]:

- 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

Corpus	Approach	Frequency	Frag num	Precision	Recall	F score	
WC1	SUBDUE MDL	2 occur	264	0.506	0.704	0.589	
		2 occur	381	0.471	0.749	0.578	
	SUBDUE SIZE 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	
		5.00%	208	0.606	0.462	0.525	
	REAFUM	10.00%	44	0.648	0.25	0.36	
		1 occur	708(37.7)	0.533(0.012)	0.797(0.019)	0.639(0.014)	
	PSM-Flow	2 occur	481(25.8)	0.585(0.015)	0.773(0.016)	0.666(0.011)	
		2 occur	95	0.366	0.435	0.398	
WC2	SUBDUE MDL	2 occur	88	0.383	0.469	0.422	
		2 occur	127	0.545	0.44	0.487	
	SUBDUE SIZE gSpan	2.00%	14	0.559	0.151	0.238	
		5.00%	2	0.389	0.062	0.107	
		10.00%	227	0.485	0.376	0.423	
		5.00%	123	0.607	0.163	0.257	
	REAFUM	1 occur	150(12.4)	0.382(0.019)	0.536(0.03)	0.446(0.02)	
		2 occur	46(8)	0.569(0.026)	0.418(0.042)	0.481(0.033)	
	WC3	SUBDUE MDL	2 occur	186	0.237	0.487	0.319
			2 occur	178	0.217	0.456	0.294
SUBDUE SIZE 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	
		2.00%	76	0.332	0.314	0.323	
REAFUM		5.00%	45	0.269	0.047	0.08	
		10.00%	4	0.5	0.016	0.03	
PSM-Flow		1 occur	318(9.66)	0.307(0.008)	0.586(0.014)	0.403(0.009)	
		2 occur	150(10.9)	0.374(0.018)	0.514(0.022)	0.433(0.019)	

TABLE II: Comparison results between PSM-Flow, SUBDUE and gSpan on three workflow corpora

(brain image understanding, 3D skull imaging, genetic modeling of the face, etc.).

- 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.
- Workflow corpus 3 (WC3): A set of 269 workflows, submitted to the LONI pipeline for execution by 62 different users.

The data is available at [7].

A. 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 occur frequently in workflow corpora as users reuse them. In order to assess PSM-Flow, we compare the groupings created by users of the LONI Pipeline for the workflows in the evaluation corpora to the fragments automatically extracted by PSM-Flow, following a similar approach in [9].

We compare PSM-Flow to three popular frequent graph mining techniques, namely, gSpan, SUBDUE and REAFUM. GSpan uses an exact match and depth first search strategy to discover all possible frequent fragments, while SUBDUE and REAFUM use inexact match approximation to 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 [4].

For a fair comparison, we filter the candidate fragment set 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).

B. 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 .

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

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

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 $F\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$.

C. Evaluation Results

Table II shows our evaluation results. 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 and REAFUM (i.e., the minimum number of times a common fragment has to appear to be detected) is normalized to the size of the corpus (2% meaning that any fragments appearing in more than 2% of the workflows of a corpus will be detected).

It should be noticed that looking merely on precision and recall could be misleading because different methods with different parameter settings give different number of fragments. For example, it is normal that the more number of fragments, the higher the recall, although not necessarily higher precision. Because of that, we cannot declare one method is better than the others based on either recall or precision, as they are highly affected by the number of detected fragments which can be adjusted by the minimum support.

F-score is a commonly used as performance metric to address the bias as it balances the precision and recall. In terms of F-score, PSM-Flow performs best in WC1 and WC3 when removing all candidate fragments with a single occurrence while only slightly worse than gSpan in WC2.

In addition, we also plot the precision-recall graph for each method as indicated in Figure 8. To compute that, we first sort the fragments by their support values so that we can get top- k fragments. Then we plot precision/recall/F score with different values of k for all methods.

PSM-Flow outperforms all methods for WC2 and WC3, and all except SUBDUE (SIZE) for WC1. The result implies that PSM-Flow can achieve better precision and recall trade-off. In other words, given same precision, PSM-Flow can have a higher recall and vice versa.

Sometimes, precision could be more preferable over recall in our task due to the fact that what we are interested in is in getting useful fragments, i.e., those that are similar to what users have identified, rather than high recall, which is subjective to the way users design. Figure 7 compares the precision and demonstrates that PSM-Flow tends to obtain higher precision for those most frequent fragments in all corpora.

Two extracted workflow fragments from WC2 under the same topic are shown in figure 5. These two fragments are used for graphical model based multivariate analysis (GAMMA) in neuroimaging data analysis. Compared to the first fragment, the second fragment replaces GAMMAAlgorithm component with GAMMAEL (GAMMA with ensemble learning). This is a case where two fragments share similar functionality but using different versions of a algorithm. PSM-Flow can extract both fragments and put them into the same topic showing that variations of fragments are captured.

In terms of scalability, we have executed PSM-Flow with synthetic workflow data using a procedure similar to [1]. We tested PSM-Flow with different number of workflows (D) and fixed parameters, with the number of node label types(N) equal to 100, with the number of fragments contained in a workflow (S) equal to 5, with the size of generated fragments (I) equal to

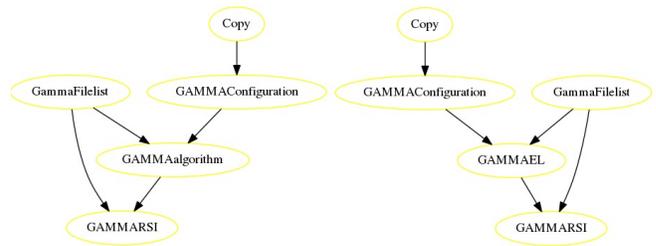


Fig. 5: Two fragments extracted fragments by PSM-Flow on WC2

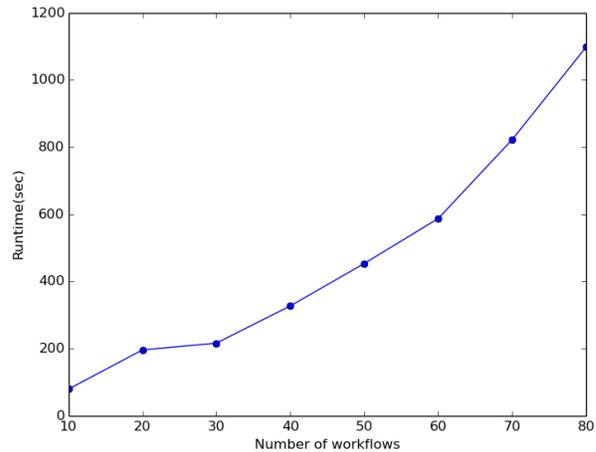


Fig. 6: Scalability on the number of workflows

15 and with the number of frequent fragment types (L) equal to 20.

The number of transactions (workflows) versus runtime is plotted in Figure 6, which shows that the runtime of PSM-Flow grows linearly with the size of workflow corpus.

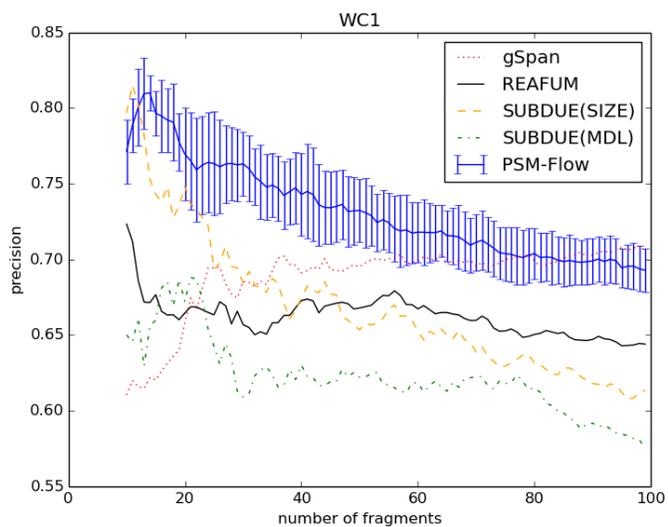
V. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced PSM-Flow, a probabilistic subgraph mining algorithm which is a novel way to adopt the topic modeling approach for robust reusable fragment discovery in scientific workflows. The method tends to improve the robustness of discovered fragments by capturing their variations in a probabilistic framework. Promising results have been obtained using workflow data.

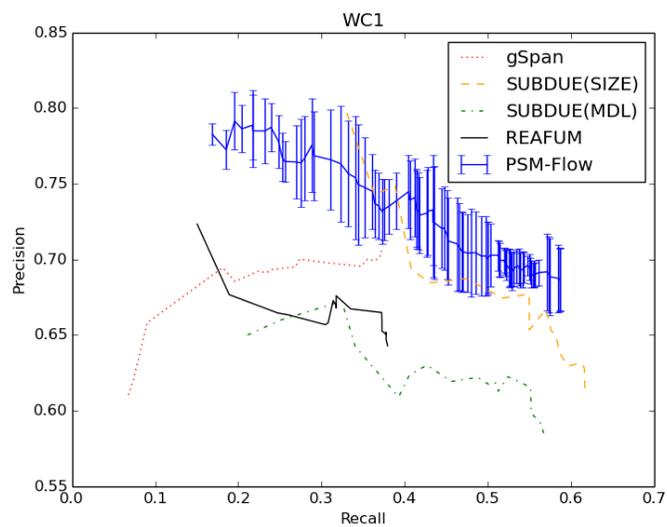
For future work, we may make use of other types of information like author information for mining workflows. It is also possible to further extend the evaluation of PSM-Flow in other graph mining tasks such as social networks.

REFERENCES

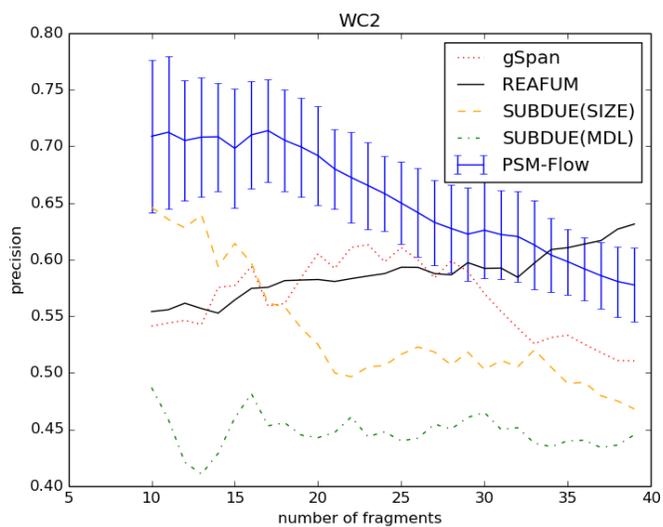
- [1] Rakesh Agrawal and Ramakrishnan Srikant. “Fast Algorithms for Mining Association Rules in Large Databases”. In: *Proceedings of the 20th International*



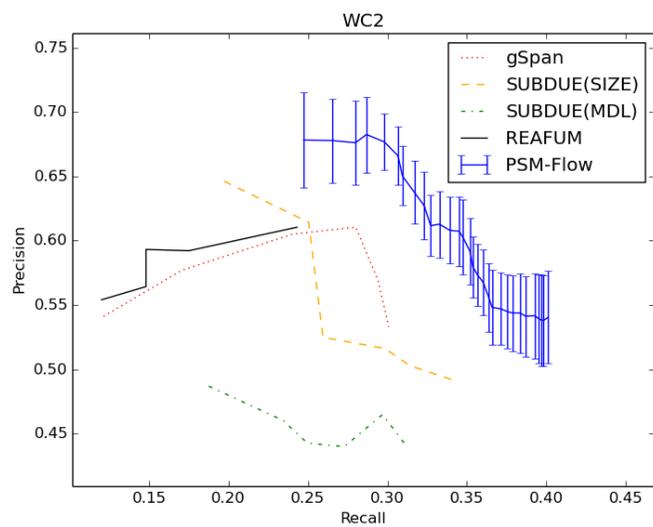
(a) WC1



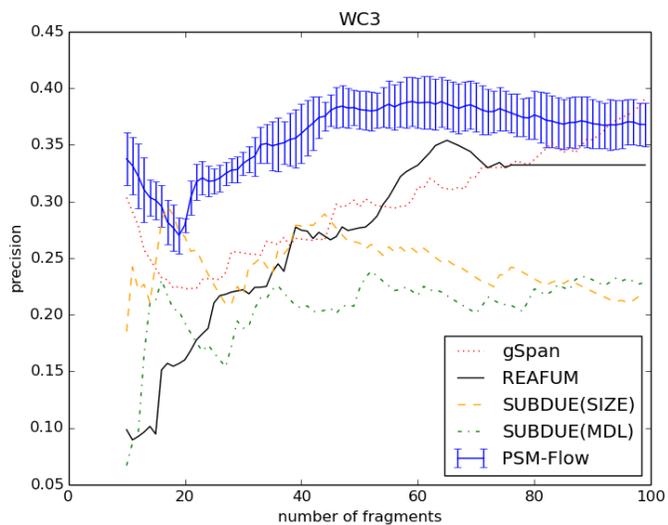
(a) WC1



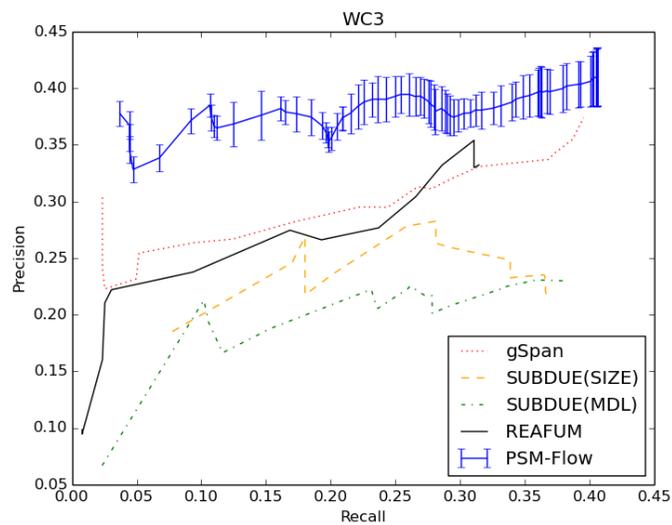
(b) WC2



(b) WC2



(c) WC3



(c) WC3

Fig. 7: Precision Curve

Fig. 8: Precision Recall curve

- Conference on Very Large Data Bases. VLDB '94.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 487–499. ISBN: 1-55860-153-8. URL: <http://dl.acm.org/citation.cfm?id=645920.672836>.
- [2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. “Latent Dirichlet Allocation”. In: *J. Mach. Learn. Res.* 3 (Mar. 2003), pp. 993–1022.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] Ivo D. Dinov et al. “Efficient, Distributed and Interactive Neuroimaging Data Analysis Using the LONI Pipeline”. In: *Frontiers in Neuroinformatics*. Vol. 3. 22. 2009.
- [7] Daniel Garijo. “The LONI Pipeline workflow inputs”. In: (Nov. 2015). DOI: 10.6084/m9.figshare.1603175.v2. URL: https://figshare.com/articles/The_LONI_Pipeline_workflow_inputs/1603175.
- [8] 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.
- [9] 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.
- [10] Yolanda Gil. “From Data to Knowledge to Discoveries: Artificial Intelligence and Scientific Workflows”. In: *Sci. Program.* 17.3 (Aug. 2009), pp. 231–246.
- [11] Yolanda Gil et al. “Wings: Intelligent Workflow-Based Design of Computational Experiments”. In: *IEEE Intelligent Systems*. 2011.
- [12] 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.
- [13] Chuntao Jiang, Frans Coenen, and Michele Zito. “A Survey of Frequent Subgraph Mining Algorithms”. In: 000 (Jan. 2004), pp. 1–31.
- [14] Rui Jiang et al. “Network motif identification in stochastic networks”. In: *Proceedings of the National Academy of Sciences* 103.25 (2006), pp. 9404–9409.
- [15] KenCheong. *KenCheong/PSM-Flow v2.0*. June 2018. DOI: 10.5281/zenodo.1289781. URL: <https://doi.org/10.5281/zenodo.1289781>.
- [16] 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.
- [17] 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.
- [18] Ruirui Li and Wei Wang. “REAFUM: Representative Approximate Frequent Subgraph Mining”. In: *SDM*. 2015.
- [19] 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.
- [20] 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.
- [21] Carl Edward Rasmussen. “The Infinite Gaussian Mixture Model”. In: *Advances in Neural Information Processing Systems 12*. Ed. by S. A. Solla, T. K. Leen, and K. Müller. MIT Press, 2000, pp. 554–560. URL: <http://papers.nips.cc/paper/1745-the-infinite-gaussian-mixture-model.pdf>.
- [22] 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.
- [23] 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.
- [24] Sinead Williamson et al. “The IBP Compound Dirichlet Process and Its Application to Focused Topic Modeling”. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning. ICML'10*. Haifa, Israel: Omnipress, 2010, pp. 1151–1158. ISBN: 978-1-60558-907-7. URL: <http://dl.acm.org/citation.cfm?id=3104322.3104468>.
- [25] 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.