HUMBOLDT-UNIVERSITÄT ZU BERLIN
MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT
INSTITUT FÜR INFORMATIK

Local Graph Patterns for Scientific Workflow Similarity Search

Exposé

eingereicht von:	David Luis Wiegandt
geboren am:	16.09.1993
geboren in:	Berlin
Gutachter:	Dr. med. Dipl.-Inf. Johannes Starlinger Prof. Dr. Ulf Leser
eingereicht am:	13.11.2015

1 Introduction

During the last years, *scientific workflows (SWFs)* have emerged as a useful means of data analysis particularly for biologists. SWFs consist of a set of nodes (also called tasks) we denote by V, with each node featuring a set of input and output ports. By connecting the output port of a node $v_1 \in V$ to the input port of another node v_2, a *directed* edge $(v_1, v_2) \in E \subseteq V \times V$ is created with E being the set of all edges [SBCBL14]. This way, we obtain a graph $G = (V, E)$.

SWFs are executed by *scientific workflow management systems (SWFMS)*. Since most SWFMS prohibit looping (i.e. cycles in the corresponding graph [Tal13]), most SWF graphs can be represented as *directed, acyclic graphs (DAGs)* [BL13].

The implemented functionality of a node appears as a black box to its user, i.e. whether the implementation calls a web service or executes a script is opaque.

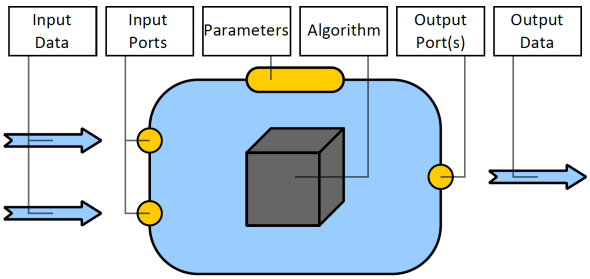

Figure 1: A labelled task of an SWF [BL13]

To optimally support non experts like biologists who cannot develop SWFs by themselves, SWFs are shared in repositories. A problem that arises with the growth of such repositories is the occurrence of (partial) duplicates. This problem has been attacked in various ways, using either global or local measures.

An approach presented in [SCBK+14] suggests to decompose SWFs into their layers and to use the best non-crossing matching between two workflows as a similarity measure. This approach empirically outperforms competing state-of-the-art algorithms.

However, a graph-theoretic approach to measure SWF similarity that has not been studied before is the comparison of the distribution of local graph patterns. Such patterns could be simple paths up to a certain length n, or more complex structures such as neighbourhoods.

2 Goal

We will study the use of local graph patterns for measuring graph similarity in the context of SWFs. To this end, we approach the problem of SWF comparison gradually, starting with the study of existing solutions leading to the more specific problem of pattern-based measures.

3 Related work

3.1 Global Measures

A common graph distance measure is the graph edit distance. Given graphs G_1, G_2 and the three edit operation *insert*, *delete* and *substitute* (all applicable to edges and nodes), each with a certain cost assigned, the graph edit distance is defined as the minimum cost sequence of edit operations to transform G_1 into G_2 or vice versa [RB09]. Despite being optimal in

terms of accuracy in finding structural differences between graphs, the graph edit distance's time complexity is exponential, making it inappropriate for large graphs.

As shown in [SBCBL14], graph comparison algorithms considering the graphs full structure appear to be too strict for fine grained assessment of scientific workflow similarity. An alternative proposed in the same work is to only focus on substructures by comparing *path sets*. Given SWF graphs G_1, G_2, we denote the sets of all paths starting at a node with no inbound edges and ending in a node with no outbound edges by PS_1, PS_2. The similarity measure is computed as the sum $\sum \text{sim}(P_1, P_2) | (P_1, P_2) \in \text{mw}(PS_1, PS_2)$ with mw being the *maximum weight matching* between two sets.

3.2 Local Measures

A similar approach that completely disregards substructures by considering *module sets* instead of path sets proved to be comparably reliable and fast [SBCBL14].

Given SWF graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$, the similarity measure is computed as the sum $\sum \text{sim}(v_1, v_2) | (v_1, v_2) \in \text{mw}(V_1, V_2)$ with mw being the *maximum weight matching* between two sets. Though, path sets turned out to provide more stable results.

3.3 Layer Decomposition

In [SCBK+14] a new approach to SWF similarity search is introduced. The previous evaluation in [SBCBL14] yields that preserving substructures leads to more reliable results, but is computationally expensive. *Layer Decomposition (LD)* aims to be a compromise between module set and path sets by decomposing SWF graphs G_1, G_2 into their layers LD_1, LD_2 and to use the *maximum weight non crossing matching* $\sum \text{sim}(L_1, L_2) | (L_1, L_2) \in \text{mwnc}(LD_1, LD_2)$ between two workflows' layers as similarity measure.

Experiments show that LD is able to outperform other algorithms in terms of correctness, including paths sets, module sets and the graph edit distance.

4 Novel Approach

We propose a new approach that compromises between only focusing on topology and only focusing on the node's various attributes. First, we introduce the concept of n-grams by defining path n-grams and neighbourhood n-grams. Therefore, we assume $\mathbb{S}$ to be the set of all SWF graphs.

Definition 4.1 (Path n-gram)**.** Given a graph $G = (V, E) \in \mathbb{S}$, we define the function $P_n : \mathbb{S} \to \mathbb{P}(V^n)$ containing all paths of length $n-1$ as

$$P_n(G) = \{(v_1, v_2, \ldots, v_n) \in V^n | (v_i, v_{i+1}) \in E, i = 1 \ldots n-1\}$$

with $P_n(G)$ being a set of n-grams of vertices.

In [ABG05], a similar approach (focussing on trees) that is based on patterns they call pq-grams has been proposed. A pq-gram is defined as a $(p+q)$-tuple consisting of an anchor node, its $p-1$ ancestors and q children. In that, computing $P_3(G)$ for some graph G resembles computing $2,1$-grams. To measure two tree's similarity, they compute a set of all pq-grams per tree and relate the cardinalities with a modified Jaccard distance. Experiments show that the pq-gram distance outperforms the tree edit distance in terms of time-complexity and competing approximations of the tree edit distance in terms of accuracy.

We apply this concept to graphs, but modify the idea of pq-grams into n-grams to depend on only one variable n besides the graph G. Therefore, we introduce a modified closed neighbourhood.

Definition 4.2 (Closed neighbourhood)**.** Given a graph $G = (V,E) \in \mathbb{S}$, we define the closed neighbourhood $\mathrm{N}_n : \mathbb{S}, v \to \mathbb{P}(V^n)$ of a vertex $v \in V$ as the set of all n-grams with the first component being v itself, followed by a predecessor $u \in \mathrm{pre}(v)$ and $n-2$ successors $w_1, \ldots, w_{n-2} \in \mathrm{suc}(v)$:

$$\mathrm{N}_n(G,v) = \{(v,u,w_1,\ldots,w_{n-2}) | u \in \mathrm{pre}(v), w_1,\ldots,w_{n-2} \in \mathrm{suc}(v)\}$$

Definition 4.3 (Neighbourhood n-gram)**.** We define the function $\mathrm{N}_n : \mathbb{S} \to \mathbb{P}(V^n)$ that computes a set of all closed neighbourhoods of a graph $G = (V,E) \in \mathbb{S}$ as

$$\mathrm{N}_n(G) = \bigcup_{v \in V} \mathrm{N}_n(G,v)$$

To derive a measure from the sets of n-grams, we introduce the n-gram distance measure.

Definition 4.4 (n-gram distance measure)**.** We define the n-gram distance between two SWF graphs G_1, G_2 as the Jaccard distance of their sets $f(G_1), f(G_2)$ with $f \in (\bigcup_{n\in\mathbb{N}} P_n) \cup (\bigcup_{n\in\mathbb{N}} N_n)$:

$$\mathrm{sim}_f(G_1,G_2) = 1 - \frac{|f(G_1) \cap f(G_2)|}{|f(G_1) \cup f(G_2)|}$$

Now, given two SWF graphs G_1, G_2, we first create a mapping between their vertices using one of the methods mentioned in [SCBK+14]. Afterwards, we retrieve sets $f(G_1), f(G_2)$ with f being one of the n-gram functions and compute their n-gram distance.

Due to the algorithm's reliance on n-grams, another modification could be to index a complete repository made up of k SWF graphs $G_1, \ldots, G_k$ with $G_i = (V_i, E_i)$: We denote the set of all vertices found in the repository by $V = \bigcup_{i=1}^{k} V_i$. Now, for each tuple $t \in f_n(G_i)$, we store the ordered pair (t,i) in the index that finally computes a function $f : V^n \ni t \mapsto i \in \mathbb{N}$.

5 Implementation and Evaluation

For our implementation, we focus on the DAG-based *Apache Taverna*, being the major SWFMS in *myexperiment*, the largest public repository for SWFs at the time of writing [DHSY13].

The benchmarking of each implementation is based on an expert-curated gold standard [SCBK+15]. All pairs of SWFs from this set are checked for similarity and the outcome is compared to the result we would have expected. Finally, we compare the performance (in terms of correctness) of the implementations to other state-of-the-art algorithms using the established measures *precision* and *recall*.

The implementation of the index is declared optional for now and depends on the results from the evaluation of the general approach that takes only two SWFs into account.

6 References

[ABG05] Nikolaus Augsten, Michael Böhlen, and Johann Gamper. Approximate matching of hierarchical data using pq-grams. In *Proceedings of the 31st international conference on Very large data bases*, pages 301–312. VLDB Endowment, 2005.

[BL13] Marc Bux and Ulf Leser. Parallelization in scientific workflow management systems. *arXiv preprint arXiv:1303.7195*, 2013.

[DHSY13] Susan B. Davidson, Xiaocheng Huang, Julia Stoyanovich, and Xiaojie Yuan. Search and result presentation in scientific workflow repositories. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, SSDBM, pages 17:1–17:12, New York, NY, USA, 2013. ACM.

[RB09] Kaspar Riesen and Horst Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, 27(7):950–959, 2009.

[SBCBL14] Johannes Starlinger, Bryan Brancotte, Sarah Cohen-Boulakia, and Ulf Leser. Similarity search for scientific workflows. *Proceedings of the VLDB Endowment*, 7(12):1143–1154, 2014.

[SCBK+14] Johannes Starlinger, Sarah Cohen-Boulakia, Sanjeev Khanna, Susan B Davidson, and Ulf Leser. Layer decomposition: An effective structure-based approach for scientific workflow similarity. In *e-Science (e-Science), 2014 IEEE 10th International Conference on*, volume 1, pages 169–176. IEEE, 2014.

[SCBK+15] Johannes Starlinger, Sarah Cohen-Boulakia, Sanjeev Khanna, Susan B. Davidson, and Ulf Leser. Effective and efficient similarity search in scientific workflow repositories. *Future Generation Computer Systems*, 2015.

[Tal13] Domenico Talia. Workflow systems for science: concepts and tools. *ISRN Software Engineering*, 2013, 2013.