Learning to Explore Scientific Workflow Repositories

Julia Stoyanovich
Drexel University, USA
Skoltech, Russia
stoyanovich@drexel.edu

Paramveer Dhillon
University of Pennsylvania
Philadelphia, PA, USA
dhillon@cis.upenn.edu

Brian Lyons
University of Pennsylvania
Philadelphia, PA, USA
lyonsb@seas.upenn.edu

Susan B. Davidson
University of Pennsylvania
Philadelphia, PA, USA
susan@cis.upenn.edu

ABSTRACT

Scientific workflows are gaining popularity, and repositories of workflows are starting to emerge. In this paper we describe *TopicsExplorer*, a data exploration approach for *myExperiment.org*, a collaborative platform for the exchange of scientific workflows and experimental plans. Our approach uses a variant of topic modeling with *tags* as features, and generates a browsable view of the repository. *TopicsExplorer* has been fully integrated into the open-source platform of *myExperiment.org*, and is available to users at `www.myexperiment.org/topics`. We also present our recently developed personalization component that customizes topics based on user feedback. Finally, we discuss our ongoing performance optimization efforts that make computing and managing personalized topic views of the *myExperiment.org* repository feasible.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications

General Terms

Algorithms

Keywords

scientific workflows, topic mining, data exploration

1. INTRODUCTION

Workflows have been gaining popularity in life sciences, where users deal with large amounts of data and perform sophisticated *in silico* experiments. A scientific workflow is an encoding of a sequence of steps that progressively transform one or several data products. Workflows help automate repetitive tasks and make experiments reproducible.

On-line workflow repositories are emerging in support of sharing and reuse. The largest public repository is *myExperiment.org* [3], with close to 2500 workflows and thousands of registered users at the time of this writing. Workflows in the repository implement various types of functionality, with most focusing on bioinformatics. Users of *myExperiment.org* post workflows to the site and often search for existing workflows, with the goal of re-using them, or of using them as examples to help develop new workflows. To help users find workflows of interest *myExperiment.org* supports tagging — a workflow may be tagged by its author or by other users. These tags are used for keyword search and for browsing, through a tag cloud (i.e., one tag at a time).

In a recent paper [7] we considered several data exploration approaches for workflow repositories. Here, we follow up on [7] and present *TopicsExplorer*, a data exploration system that is fully integrated into the open-source framework of *myExperiment.org*. *TopicsExplorer* uses tags annotating workflows to derive browsable workflow categories. Figure 1 presents a screenshot of the *TopicsExplorer* interface, available to users at `www.myexperiment.org/topics`.

Topics derived by *TopicsExplorer* are combinations of several tags that together represent a conceptual unit of functionality — a goal that the author had in mind when building the workflow, e.g., sequence alignment, visualization of an isosurface, etc. Topics form the *semantic structure* of a workflow collection, with each workflow belonging to one or several topics. If the set of topics in the collection were known, together with a workflow-to-topic assignment, then the collection could be organized around the topics for browsing. However, this semantic structure is typically hidden and must be learned. *TopicsExplorer* uses Latent Dirichlet Allocation (LDA), a popular topic mining technique. In Section 2 we describe LDA and explain how it is used for mining topics in *myExperiment.org*.

Having implemented *TopicsExplorer*, we asked 10 active users of *myExperiment.org* to evaluate the derived topics. We asked users to rate the topics on a scale of 0 through 3, with 0 assigned to a topic that is outside of user's area of expertise (and so the user's judgment on such a topic was excluded from our analysis), 1 assigned to a topic that is incoherent (tags do not fit together), 2 assigned to a somewhat coherent topic (a mix of 2 or more natural topics), and 3 assigned to a coherent topic. Preliminary user feedback was encouraging. We found that users assigned a score of 2 or 3 to a significant portion of the topics, meaning that these were useful for data exploration. Interestingly, users often assigned a higher score to topics corresponding to their research interests. This motivated us to develop a person-

.

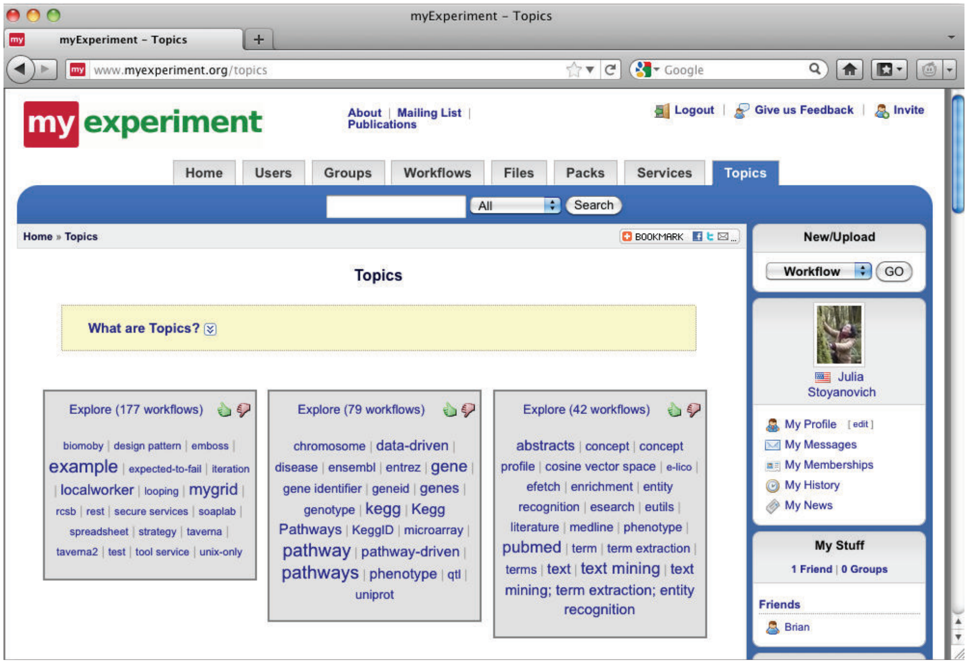
myExperiment – Topics
myExperiment – Topics
www.myexperiment.org/topics
Google
my experiment
About | Mailing List | Publications
Logout | Give us Feedback | Invite
Home
Users
Groups
Workflows
Files
Packs
Services
Topics
All
Search
Home » Topics
BOOKMARK
Topics
What are Topics?
Explore (177 workflows)
biomoby | design pattern | emboss | example | expected-to-fail | iteration | localworker | looping | mygrid | rcsb | rest | secure services | soaplab | spreadsheet | strategy | taverna | taverna2 | test | tool service | unix-only
Explore (79 workflows)
chromosome | data-driven | disease | ensembl | entrez | gene | gene identifier | geneid | genes | genotype | kegg | Kegg Pathways | KeggID | microarray | pathway | pathway-driven | pathways | phenotype | qtl | uniprot
Explore (42 workflows)
abstracts | concept | concept profile | cosine vector space | e-lico | efetch | enrichment | entity recognition | esearch | eutils | literature | medline | phenotype | pubmed | term | term extraction | terms | text | text mining | text mining; term extraction; entity recognition
New/Upload
Workflow
GO
Julia Stoyanovich
My Profile [edit]
My Messages
My Memberships
My History
My News
My Stuff
1 Friend | 0 Groups
Friends
Brian

Figure 1: Screenshot of the Topics Explorer UI.

alized data exploration solution that incorporates user feedback and tailors topics to interests of a particular user or a group of users. We describe our personalization approach in Section 3. Our personalization framework has been implemented; it is currently in prototype stage and has not yet been made part of the live *myExperiment.org* site.

Personalization introduces interesting efficiency challenges. First, storing as many sets of topics as there are users may result in unreasonable space overhead. We observe that some of the topics generated for users, even if users differ on part of their feedback, may be the same or very similar. We thus implement a *post-processing step* that analyzes pair-wise similarity between topics and only stores one copy if two or more topics are similar. This step is fully implemented in our system and discussed in Section 4. Second, generating a different set of topics for each user who gives feedback may become prohibitively expensive, even if done off-line. We observe that users may be grouped together based on similar feedback during *pre-processing*. The system would then generate a single set of topics for a group of users. Developing inter-user similarity metrics that successfully manage space overhead while faithfully accounting for feedback is part of our ongoing work.

We stress that the focus of our work is not on comparing the performance of various topic mining or clustering algorithms in workflow repositories. Rather, we aim to build a practical data exploration tool for *myExperiment.org* that lends itself well to personalization. The fact that *TopicsExplorer* has been incorporated into *myExperiment.org* is testament to the appropriateness of LDA for this task.

2. GENERATING TOPICS

Topic models are probabilistic models for uncovering the semantic structure of a dataset based on hierarchical Bayesian analysis. We use a particular kind of a topic model, called latent Dirichlet allocation (LDA) [2]. We now briefly describe LDA, and refer the reader to [2] for details.

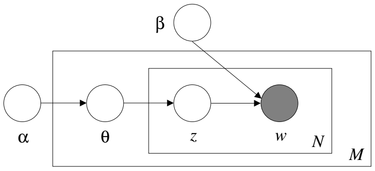

Figure 2: Plate representation of the LDA model.

LDA is a generative model that explains sets of observations by unobserved (latent) random variables. In our case, observations are the presence or absence of a particular tag in the description of a workflow. LDA assumes that the number of topics in a collection is given, and that topics are drawn *independently* from a multinomial distribution with a Dirichlet prior. Figure 2 represents LDA using plate notation, with N tags and M topics. The outer plate represents workflows, while the inner plate represents the repeated choice of topics and tags within a workflow. α is the parameter of the uniform Dirichlet prior on the per-workflow topic distributions, while β is a vector, specifying parameters of the uniform Dirichlet priors on the per-topic word distributions.

LDA assumes the following generative process for each workflow. First, choose $\theta \sim Dirichlet(\alpha)$. Next, for each of the N tags, choose a topic $z_n \sim Multinomial(\theta)$. Finally, choose a tag w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n. Assignments of tags to workflows are the only observable variables (w); all other variables are latent and are estimated by LDA. Also given is the number of topics M, and an initial setting for the parameter α. To determine appropriate values for M and α

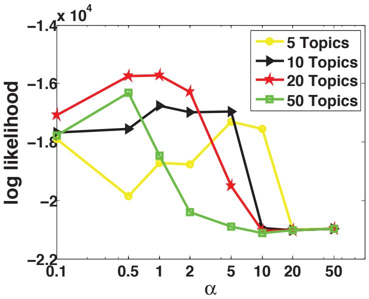

Figure 3: LDA hyperparameter tuning.

for *myExperiment.org*, we performed hyperparameter tuning, maximizing the log-likelihood of the model. Figure 3 presents results of the tuning, with 5, 10, 20 and 50 topics, and with α ranging from 0.1 to 50. We found that $\alpha = 2$ and $M = 20$ give the best model, and we use these parameter values in our implementation.

As is typically done in topic modeling literature, topics are described by their most probable features. We describe each topic using 20 most probable tags shown as a tag cloud, with font size of each tag representing its relative frequency among the workflows that belong to the topic. Figure 1 shows three topics, with the corresponding tag clouds. While a more sophisticated selection and presentation of features is possible, e.g., one based on *tf-idf* weighting, we use tag clouds here for consistency with the rest of the *myExperiment.org* site.

We chose LDA in our implementation for the following reasons. In [7] we compared the performance of LDA to that of frequent itemset mining and of distance-based clustering [6], and concluded that LDA gives superior results for the *myExperiment.org* corpus. We also considered more sophisticated methods, such as correlated topic mining (CTM), but found that CTM does not derive categories of better quality for our corpus compared to LDA, yet is more expensive to compute. Importantly, several efficient open-source implementations of LDA exist, making its use in a live system practical. We opt for the MALLET [5] implementation, which is robust and easy to extend. Another point in favor of LDA is that, as we describe in Section 3, LDA, and its MALLET implementation, lend themselves well to our novel approach of incorporating user feedback.

3. REFINING TOPICS USING FEEDBACK

We have been collecting user feedback since *TopicsExplorer* was incorporated into *myExperiment.org*. Users provide feedback by clicking on the thumbs-up / thumbs-down buttons inside each topic description, see Figure 1. This feedback can then be aggregated (using, e.g., majority voting) to improve topics for all users. Alternatively, the system may consider feedback from one user at a time, generating personalized topics. Both aggregated and per-user feedback may be used for topic refinement, which we now describe.

Following a recent approach for incorporating domain knowledge into LDA [1], we translate per-topic feedback into pair-wise constraints over tags. Positive feedback is translated into *must-link* and negative — into *cannot-link* constraints. A must-link constraint between tags a and b means that a and b should both have a high probability of belonging to the same topic or a low probability (i.e., either both of them are strongly associated with the topic or neither one is). A cannot-link constraint states that a and b cannot both have a high probability of belonging to the same topic. Constraints are soft rather than hard, and so checking the consistency of a set of constraints is not required.

While our modeling of constrains is similar to that of [1], our way of enforcing them differs. In [1], the uniform Dirichlet prior on the probability of a tag given a topic was replaced with a *forest of Dirichlet priors*. This makes the computation of LDA significantly more demanding, and no standard implementation of this approach exists. An LDA model is learned using Gibbs sampling. Our procedure enforces constraints by combining Gibbs sampling with *deterministic annealing*. We use insights from [4], where annealing was used in conjunction with Gibbs sampling, although not in the context of enforcing constraints. Our approach is described in some detail in the rest of this section.

We start with the set of LDA topics $T_1 \dots T_M$, and collect user feedback on these topics. Feedback on topic T, denoted $feedback(T)$, has values $+1$ for positive, -1 for negative, and 0 for no feedback. If $feedback(T) \neq 0$, we define a constraint for all pairs of tags $a, b \in ImportantTags(T)$, where $ImportantTags(T)$ is the set of tags corresponding to 80% of the topic's probability mass; the constraint is denoted $(a, b)_T$. Pair-wise constraints may be positive or negative. *Strength* of a constraint models the importance of the pair of tags a and b to topic T. Strength is not given by the user; rather, it is derived from LDA output, and represents the joint probability of tags a and b in T, i.e., the probability that a and b will co-occur in a document generated by T. Constraint strength models the reasonable assumption that a user's judgment about the quality of the topic is based on the high-probability keywords within that topic.

Importantly, while a positive (resp. negative) judgment about the relationship of a and b was made by a user w.r.t. a particular topic T, we must assume, by definition of must-link and cannot-link constraints above, that this judgment is universal, i.e., that a and b must (resp. cannot) both have a high probability in *any topic*. Because multiple conflicting, or mutually reinforcing, constraints may exist on a pair of tags a and b, and because ultimately constraints are applied over all topics, we aggregate constraints for distinct pairs of tags, over all topics. We refer to aggregated positive constraints as $\mathcal{C}^+$, and to negative as $\mathcal{C}^-$.

Our goal is to combine user feedback with the probabilistic model learned by LDA. As we noted above, LDA uses Gibbs sampling to estimate the probability distribution over all tags for each topic. At each round of sampling, and for a given tag a, the sampler draws a topic T_i based on its current estimate of *frequency scores*: $score(a, T_1), \dots score(a, T_M)$. Our idea is to adjust the value for each $score(a, T_i)$ by combining current frequency scores with applicable constraints. We do this in scope of a deterministic annealing framework, where the magnitude of adjustment (either positive or negative) increases with increasing round number r. At each annealing round $r : 0 \dots R$, we execute multiple rounds of Gibbs sampling (similar to what is done in traditional LDA); at each round of sampling tags are considered in some fixed order. Given a tag a and a topic $T_i : T_1 \dots T_M$, denote by $score_r(a, T_i)$ the frequency score of T_i for a, at annealing round r. Suppose that tags b and c are currently as-

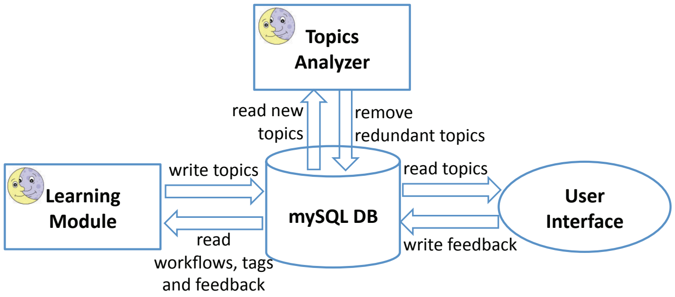

Figure 4: System architecture of *TopicsExplorer*.

signed to topic T_i, and that there exist constraints $C_1 : (a, b)$, $C_2 : (a, c)$ and $C_3 : (b, c)$, with the corresponding strength scores. Of these, C_1 and C_2 are applicable, because they involve a and one of the other tags already assigned to T_i. From these, we select one constraint to apply, randomly and according to its strength. Suppose that constraint C_j was chosen. We compute the frequency score of topic T_i at annealing round $r + 1$ as:

$$score_{r+1}(a, T_i) = \begin{cases} score_r(a, T_i)^{\gamma} & if \quad C_j \in \mathcal{C}^+ \\ score_r(a, T_i)^{\frac{1}{\gamma}} & if \quad C_j \in \mathcal{C}^- \end{cases}$$

Here, γ is a parameter for which an appropriate value is determined experimentally. We use $\gamma = 2$ in our implementation. Developing principled evaluation metrics for a personalized solution such as ours, which would allow for automatic tuning of γ, is part of our on-going work.

4. SYSTEM ARCHITECTURE

Figure 4 presents the system architecture of *TopicsExplorer*. Data is stored in a mySQL database, and is accessed by the user interface, implemented in Ruby on Rails. The Learning Module computes topics based on workflow tagging information and, in the feedback-aware version of our system, on user feedback. Topics Analyzer compares multiple topic models computed by the Learning Module, deriving a compact representation for a collection of models.

Users access topics through the Topics tab, where they see 20 topics, each described by a tag cloud (see Figure 1). The user may access the workflows that belong to a particular topic by clicking on the *Explore* link. Workflows will be displayed in order of probability of belonging to a topic, and may belong to multiple topics. Users are encouraged to submit feedback on the quality of the topics, by clicking on a thumbs up / thumbs down icon inside each topic description. Feedback is then used to refine topics (see Section 3).

Topics are computed by the Learning Module, which is executed nightly, and re-computes topics for the entire workflow repository, incorporating newly-added workflows and tags. The version of *TopicsExplorer* that is running on *myExperiment.org* computes a single *global model* based on LDA (Section 2). Our feedback-aware prototype also computes multiple *personalized models* on demand (Section 3).

Following the execution of the Learning Module, Topics Analyzer is invoked; this module reduces the space overhead of maintaining multiple alternative topic models. Topics Analyzer starts by comparing the global model derived in the current run to that derived in the previous run, topic by topic. If a pair of redundant topics is detected, i.e., if a topic in the current model is the same or very similar to a topic from last night's run, only a single copy of the topic is kept. Next, Topics Analyzer compares each personalized model to the current global model, topic by topic, and similarly removes redundant topics. Maintaining a single copy of redundant topics is extremely important in a system such as ours, where potentially many alternative personalized models must be maintained, and these must be updated daily to account for new workflows and new user feedback.

In the current version of *Topics Explorer*, we consider topics T_1 and T_2 to be redundant if the same tags make up 80% of the topics' probability mass. An alternative is to compare the probability distributions over the tags that T_1 and T_2 define, using, e.g., KL-divergence.

The Learning Module computes a single topic model in under a minute on the complete *myExperiment.org* dataset, and so is feasible to execute as part of an off-line nightly process. Augmenting *TopicsExplorer* with personalization, and making this functionality part of the live *myExperiment.org* site, requires additional optimizations. As discussed in the Introduction, we are currently working on grouping together users who agree on their feedback. This will allow up to compute a single personalized set of topics per group, rather than per user, resulting in manageable computation time and space overhead. An alternative that we are also considering is computing personalized topics on the fly, e.g., by combining several existing global topic models.

5. CONCLUSION

In this paper, we discussed a data exploration approach for repositories of scientific workflows that is based on topic mining, and showed how topics may be personalized based on user feedback. We presented *TopicsExplorer*, an implementation of our proposed approach for *myExperiment.org*, the largest public workflow repository. Our system draws on, and non-trivially extends, techniques that were recently proposed in the machine learning literature. In particular, our proposed method for topic personalization is novel. Our techniques are useful beyond scientific workflow repositories, and we are currently working on identifying other datasets, and on evaluating wider applicability.

6. REFERENCES

[1] D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *ICML*, 2009.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[3] D. De Roure, C. A. Goble, and R. Stevens. The design and realisation of the myExperiment virtual research environment for social sharing of workflows. *Future Generation Comp. Syst.*, 25(5):561–567, 2009.

[4] J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, 2005.

[5] A. K. McCallum. MALLET: A machine learning for language toolkit, 2002.

[6] E. Santos et al. A first study on clustering collections of workflow graphs. In *IPAW*, 2008.

[7] J. Stoyanovich, S. Davidson, and B. Taskar. Exploring repositories of scientific workflows. In *WANDS*, 2010.