An ontology and frequency-based approach to recommend activities in scientific workflows

Adilson Khouri[1☯], Luciano Digiampietri[2☯]

1 School of Arts, Sciences and Humanities, University of São Paulo, Brazil
2 School of Arts, Sciences and Humanities, University of São Paulo, Brazil

☯These authors contributed equally to this work.

Abstract

The number of activities provided by scientific workflow management systems is large, which requires scientists to know many of them to take advantage of the reusability of these systems. To minimize this problem, the literature presents some techniques to recommend activities during the scientific workflow construction. This project specified and developed a hybrid activity recommendation system considering information on frequency, input and outputs of activities and ontological annotations. Additionally, this project presents a modeling of activities recommendation as a classification problem, tested using 5 classifiers; 5 regressors; a SVM classifier, which uses the results of other classifiers and regressors to recommend; and Rotation Forest , an ensemble of classifiers. The proposed technique was compared to other related techniques and to classifiers and regressors, using 10-fold-cross-validation, achieving a MRR at least 70% greater than those obtained by other techniques.

Introduction

A quantidade de projetos de pesquisa que usam computação intensiva vem crescendo em áreas que não possuem conhecimentos avançados em computação, como biologia, física e astronomia. Uma das ferramentas para auxiliar no gerenciamento e construção de experimentos de computação intensiva são os sistemas gerenciadores de *workflows*. *Workflows científicos* são processos estruturados e ordenados, construídos de forma manual, semi-automática ou automática que permitem solucionar problemas científicos utilizando atividades, que podem ser: i) blocos de código fonte; ii) serviços; e iii) *workflows* finalizados [1]. Estes sistemas facilitam a criação de novos experimentos, compartilhamento dos resultados e reutilização de atividades existentes.

Atualmente há um grande número de atividades disponíveis em repositórios como *myExperiment* que armazena mais de 2.500 *workflows* [1] e *BioCatalogue* que disponibiliza mais de 2.464 serviços [2]. O grande número de atividades e o baixo reuso de algumas atividades e *workflows* [1] motivam a construção de técnicas para recomendar atividades aos cientistas durante a composição dos *workflows*.

Dentro dos sistemas gerenciadores de *workflow*, as atividades são tipicamente representadas como ícones gráficos com função *drag and drop*. Desta forma é possível construir experimentos computacionais arrastando ícones e preenchendo parâmetros de entrada. A maioria destes sistemas fornecem conjuntos de atividades básicas que podem ser utilizadas em diferentes domínios, por exemplo, uma atividade que calcula o valor

[1] http://www.myexperiment.org/

médio de um conjunto de dados, é aplicável em biologia, física, astronomia e outras áreas. Porém, há uma pré-condição para se reutilizar e/ou criar *workflows*: conhecer quais são as atividades disponíveis.

Para minimizar o problema de conhecer um grande número de atividades foram propostas diversas técnicas para recomendar atividades ou compor workflows. No primeiro caso, cujo objetivo é atender um usuário experiente nesses sistemas, durante a construção do workflow são recomendadas atividades para ajudar a finalizar o workflow. No segundo caso, cujo objetivo é atender um usuário menos experiente nesses sistemas, diversos workflows são construídos e sugeridos para o usuário selecionar qual satisfaz mais sua necessidade.

Este artigo apresenta uma estratégia híbrida para recomendar atividades em workflows científicos baseada em frequência de atividades em conjunto com uma ontologia de domínio (*knowledge-base* híbrido, com MoC *dataflow*) para conjuntos de dados sem proveniência, sem dados de confiabilidade entre autores e sem anotações semânticas prévias. Além disso sugere uma modelagem do problema de recomendar atividades em workflows científicos para que seja solucionado por classificadores como: Suport Vector Machine (SVM), Naive Bayes (NB), K-Nearest-Neighbor (KNN), Classification and Regression Trees (CART) e Rede Neural (MLP). Também são utilizados os seguintes regressores como: Suport Vector Regression (SVR), CART, Rede Neural, Multivariate Adaptive Regression Splines (MARS) e regressão binomial (RB). E uma comparação das soluções da literatura correlata com as propostas.

Na seção *Correlatos* são dicutidas as técnicas propostas pela literatura correlata, em *Materiais e métodos* serão descritas a fonte dos dados, a adaptação do problema para ser tratado como um problema de classificação e metodologia de testes. Em *Técnica Proposta* é descrita a solução proposta nesse artigo. Na seção *Results and Discussion* serão descritos os resultados obtidos com o experimento e suas analises por fim, na seção *Conclusion* serão feitas as considerações finais.

Correlatos

A literatura correlata apresenta diversas técnicas para recomendar atividades em workflows científicos que serão descritas brevemente nessa seção. Os trabalhos de 3 e 4, que consideram a mineração sequencial de atividades como *itemsets* desconsideram a ordem das atividades e a semântica das mesmas. A proposta de 5 desconsidera apenas a semântica das atividades. Esta proposta de mestrado considera a ordem de atividades que é um fator importante na recomendação conforme visto no capítulo de conceitos fundamentais.

Os trabalhos de 6–14 consideram a ordem das atividades, entrada e saída e proveniência dos dados. Suas limitações são a necessidade de dados de proveniência, pois nem todo SGWC armazena essas informações, além de desconsiderar informação semântica dos *workflows* e atividades. Este projeto não necessita de informações de proveniência e considera a semântica da informação por meio de uma ontologia hierarquizada e validada por um especialista da área.

O trabalho de 15 usa apenas um mapeamento entre atividades e ontologia desconsiderando a entrada e saída, o que potencialmente gera recomendações ineficientes. Neste projeto são consideradas às entradas e saídas de cada atividade individualmente, além do uso de uma ontologia de domínio.

16, 17 desconsideram o uso de semântica das atividades e da frequência de suas ocorrências em pares. Nesse projeto de mestrado são considerados esses dois fatores.

O trabalho de 18 exige dados que permitam calcular a confiança dos usuários e dos

seus *workflows*. Repositórios como *myExperiment*[2] não exigem dos usuários o preenchimento de todos os seus dados, de forma que grande parte das informações relacionadas a este aspecto não são preenchidas pelos usuários. Além disso, os autores desconsideram a semântica das atividades e *workflows*. Este projeto de mestrado considera a semântica de *workflows* e não necessita da informação sobre a confiança dos usuários.

Os trabalhos de 19, 20 e 21 desconsideram o uso de semântica de dados para recomendar, o que é um limitante conforme discutido por 22, 23. No presente mestrado, a frequência é considerada em conjunto com a ontologia de domínio.

Os trabalhos de 24–26 desconsideram o uso de uma ontologia hierarquizada e validada por um especialista. Dessa forma, a qualidade das anotações semânticas é questionável. Nesse projeto foi construída uma ontologia usando uma metodologia e esta foi validada por um especialista.

Os trabalhos de 22, 23 consideram o uso de frequência e ontologia, como neste projeto, porém recomendam *subworkflows* o que limita as recomendações de atividades. Apenas atividades usadas em fragmentos comuns de *workflows* poderão ser recomendadas. Em outras palavras, se a atividade se encontra no "meio" de um *subworkflow* esta nunca poderá ser recomendada individualmente. No presente mestrado, todas as atividades tem possibilidade de ser recomendadas, mesmo que no final da lista de recomendação. Além disso, apresenta uma recomendação mais abrangente, pois trata o caso de atividades simples, *subworkflows* e *Shims* (atividades conversoras de tipos de dados e/ou adpatadores).

Materiais e métodos

Os *workflows* foram obtidos no repositório *myExperiment* [27], por meio do programa *wget* [28]. Após efetuar o *download* dos 2481 *workflows* em formato *xml*, foi utilizado o analisador de código *Beautiful Soup* [29], para organizar o conjunto de dados em uma base de dados relacional.

Os dados foram armazenados em uma matriz simples usada para as técnicas que não usam ordem das atividades. E também em uma matriz adaptada para a modelagem como problema de classificação (binária) e regressão. As matizes serão descritas nas próximas seções.

Matriz simples

Os *workflows* da área de bioinformática (totalizando 73) em conjunto com suas atividades (totalizando 280) foram convertidos em uma matriz $M_{i,j}$ em que cada linha i representa um *workflow*, cada coluna j representa uma das 280 atividades e cada célula da matriz M representa a existência $M_{i,j} = 1$, ou não $M_{i,j} = 0$, da atividade da coluna j no *workflow* i. A tabela 1 apresenta um exemplo, fictício, de matriz M. Para a realização dos testes, para cada linha da tabela 1 é removida uma atividade e é recomendada uma lista de possíveis atividades. O objetivo do sistema de recomendação é identificar corretamente qual a atividade está faltando no workflow (isto é, aquela que foi removida).

Matriz adaptada

Para usar técnicas de classificação e regressão foram propostas algumas alterações no conjunto de dados original, descrito na tabela 1, as quais podem ser visualizadas na tabela 2. Cada *workflow* foi replicado 118 vezes. Destes, 59 são uma cópia idêntica ao

[2] http://www.myexperiment.org/

Table 1. Exemplo de matriz de entrada.

Workflow	**Ativ 01**	**Ativ 02**	...	**Ativ 280**
01	1	0	...	0
02	1	1	...	1
03	1	0	...	1
⋮	⋮	⋮	⋮	⋮
73	1	0	...	0

original, enquanto que dos outros 59 foi removida uma mesma atividade para todos os *workflows*, e foi adicionada uma nova atividade representando uma possível recomendação. Dessa forma, para cada *workflow* original haverá 59 instâncias corretas e 59 instâncias incorretas e este tipo de informação será utilizada para treinar os classificadores ou regressores.

Table 2. Exemplo de matriz de entrada para técnicas de classificação e regressão

#	*Workflow*	Ativ 01	Ativ 02	...	Ativ 279	Ativ 280	Rótulo
1	01	1	0	...	0	0	T
2	01	1	0	...	0	0	T
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
59	01	1	0	...	0	0	T
1	01	0 (removida)	1 (adicionada)	...	1	0	F
2	01	0 (removida)	0	...	1 (adicionada)	0	F
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
59	01	0 (removida)	0	...	0	1 (adicionada)	F
	⋮						
1	73	1	1	...	0	0	T
2	73	1	1	...	0	0	T
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
59	73	1	1	...	0	0	T
1	73	1 (adicionada)	0 (removida)	...	1	0	F
2	73	1	0 (removida)	...	1 (adicionada)	0	F
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
59	73	1	0 (removida)	...	0	1 (adicionada)	F

A escolha de 59 atividades a serem recomendadas foi feita por duas razões. A primeira é selecionar as 59 atividades com maior frequência na base de dados. A segunda é a limitação computacional: replicar as 280 possíveis recomendações poderia ser inviável em termos de treinamento. Foram replicadas 59 instâncias de *workflows* idênticas consideradas corretas, isto é com a atividade correta não removida, para garantir o balanceamento entre classes. A última alteração foi adicionar uma coluna indicando se a recomendação da atividade proposta é a correta, isto é, a pertencente ao respectivo *workflow* (T) ou não (F).

Validação de Resultados

Para a validação será utilizada a técnica cruzada considerando 10 subconjuntos (10-*fold cross validation*). Nessa técnica, o conjunto de dados é dividido em 10 subconjuntos (*folds*) e são realizadas dez execuções. Em cada uma, 10% dos *workflows* são separados para teste e 90% para treinamento. Assim, para cada execução, o sistema treina com 90% dos dados e o resultado do treinamento é testado para os 10% restantes.

Deve-se ressaltar que 100% do conjunto de dados é rotulado (isto é, fica explícito ao sistema qual atividade foi removida) e assim é possível verificar o desempenho de cada uma das execuções. O teste apresenta os 10% de *workflows*, sem informar os rótulos (a atividade removida), para os sistemas de recomendação que já foram treinados. Ao

término das dez execuções são calculadas as médias das métricas: i) *Sucess at rank k* ($S@k$); e ii) *Mean Reciprocal Rank* (MRR).

A métrica $S@k$ calcula a probabilidade de um item de interesse estar localizado entre as k primeiras posições da lista de atividades recomendadas. Seus valores residem entre zero e um. Os resultados dessa métrica são cumulativos para valores crescentes de k, isto ocorre pois se uma atividade de interesse estiver entre as cinco primeiras posições da lista de recomendações, ela também encontra-se entre as dez primeiras posições. No limite, a atividade sempre estará entre as L primeiras posições, sendo L o tamanho total da lista de recomendações. Assim, valores elevados para $S@k$ são considerados bons, especialmente para valores baixos de k. Essas métricas são calculadas por:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{n_i}\right) \tag{1}$$

$$S@k = \frac{1}{N}\sum_{i=1}^{N}(I(n_i \leq k)) \tag{2}$$

em que N é o número de listas recomendadas, n_i é a posição do item desejado na lista de recomendações i, k é uma posição da lista determinada como parâmetro de entrada da equação (2) e a função I, indica se a atividade n_i ocorre em uma posição (x) menor ou igual ao parâmetro de entrada k, e é dada por

$$I(x,k) = \begin{cases} 1 \text{ se } x \leq k \\ 0 \text{ caso contrário} \end{cases} \tag{3}$$

Técnica Proposta

A solução proposta neste artigo recomenda atividades usando três conceitos importantes na área de *workflows* científicos: i) frequência de atividades; ii) compatibilidade entre entrada e saída; e ii) semântica de atividades. Para explicar esta proposta, será usada a figura 1 como exemplo. Nela é possível observar seis *workflows* com suas anotações, que simulam uma base de dados de *workflows* científicos.

Fig 1. Exemplo de banco de dados de workflows científicos. Wokflows científicos com anotações ontológicas usados para exemplificar a solução proposta.

A solução proposta começa calculando a frequência de ocorrência de cada par de atividades existentes, que é o número de vezes que uma atividade *W* ocorre imediatamente após uma outra atividade *Z*. Ao considerar somente atividades que já foram conectadas, previamente na base de *workflows*, a compatibilidade de entrada e saída é garantida por consequência.

Após calcular a frequência é necessário anotar todos os *workflows* da figura 1, usando os conceitos da ontologia construída (ver figura 2). Essa etapa é feita manualmente (de forma não automatizada). Por fim, o algoritmo anota todas as atividades com as mesmas anotações de seus respectivos *workflows*; isto é, se a atividade *X* (da figura 1) está dentro de dois *workflows* com anotações distintas então esta atividade receberá duas anotações. O resultado final é a tabela 3, que apresenta as frequências e anotações de atividades, nesse ponto o sistema está treinado e pronto para uso do cientista.

Fig 2. Ontologia. Ontologia construída para anotar Wokflows científicos com anotações ontológicas.

Para compreender o mecanismo de recomendação treinado será usado outro exemplo, cujo objetivo é simular a interação do usuário com o sistema de recomendação. Suponha que durante a construção do *workflow* 1 (ver figura 2) um cientista insira a atividade Z e solicite uma recomendação. O sistema vai procurar na lista das atividades posteriores a Z ordenadas por frequência e conceito ontológico e irá retornar a lista de recomendação apresentada na tabela 3. A ordenação por conceito ontológico, além de ser estável serve como critério de desempate, quando duas atividades tiverem a mesma frequência. Neste exemplo, de acordo com a lista de recomendação da tabela 3, a atividade W seria recomendada em primeiro lugar ao cientista, o que representa um acerto.

Table 3. Recomendação para a atividade Z ordenada por frequência e conceito ontológico

Posição na Lista	Ativ	Frequência	Anotação Atividade
1	W	3	BLAST
2	X	2	FAST, CLUSTAL
3	Q	1	SNAP DRAGONS
⋮	⋮	⋮	⋮
280	⋮	⋮	⋮

As atividades são anotadas com a mesma anotação dos *workflows* que as contém. Dessa forma, é possível que haja pelo menos uma atividade com mais de uma anotação. Isso gera um novo caso de recomendação a ser considerado. Suponha que ambas as atividades W e X contenham dentro de suas listas de anotação o conceito *BLAST*. Nesse caso, seria recomendada a atividade com menor número de anotações, por ser considerada mais específica para o experimento em questão. Caso ambas as atividades tenham o mesmo número de anotações, é utilizada a ordem alfabética de conceitos como critério de desempate. Se ocorrer um novo empate é usado um seletor aleatório.

Resultados e discussão

A tabela 4 exibe os resultados de cada sistema recomendador usado. As técnicas que possuem a letra C em subscrito são classificadores; as que possuem letra R em subscrito são regressores; e as que não tem nada são da literatura correlata. Cada sistema efetua suas recomendações de acordo com seus diferentes critérios em uma lista inicial. Em seguida, as atividades não recomendadas são acrescentadas ao final da lista inicial. Dessa forma, a atividade correta sempre será encontrada, e o fator que diferencia os sistemas de recomendação é a posição em que as atividades ocupam na lista de atividades final que contém 280 posições.

O sistema baseado em *Aleatoriedade* não precisou de treinamento. O algoritmo apenas selecionava aleatoriamente as atividades formando uma lista de atividades recomendadas. Esse sistema recomendou menos de 3% das atividades corretas entre as dez primeiras posições. A maioria das atividades corretas foram classificadas próximas a posição 140 que é a posição média das listas recomendadas. Os valores das métricas $S@280 = 1$ e $S@100 = 0,0400$ indicam que a maior parte dos itens corretos foi encontrado após a centésima posição. Esse sistema foi proposto como um marco de comparação.

O sistema que usa a técnica *Apriori* obteve seu melhor desempenho quando os parâmetros *confiança* e *suporte* foram definidos como *sem limitação*, isto é, não foi estabelecido um valor de confiança ou suporte mínimo para considerar possíveis regras de associação criadas. Todas as regras foram consideradas válidas. Mesmo sem

Table 4. Resultados dos sistemas de recomendação

#	**Técnica**	**S@1**	**S@5**	**S@10**	**S@50**	**S@100**	**S@280**	**MRR**
1	Aleatório	0,0037	0,0260	0,0280	0,0300	0,0400	1,0000	0.033
2	*Apriori*	0,0037	0,0385	0,0559	0,0568	0,0570	1,0000	0,037
3	KNN_C	0,0037	0,0685	0,0959	0,5068	1,0000	1,0000	0,040
4	Rede neural_C	0,0137	0,1507	0,1781	0,8082	1,0000	1,0000	0,089
5	CART_C	0,0274	0,1233	0,3699	0,7671	1,0000	1,0000	0,113
6	CART_R	0,1370	0,1370	0,2603	0,6164	1,0000	1,0000	0,114
7	Naive Bayes_C	0,0274	0,1507	0,3425	0,6301	1,0000	1,0000	0,114
8	Binomial_R	0,0822	0,1918	0,2055	0,8493	1,0000	1,0000	0,136
9	Rede neural_R	0,1096	0,2603	0,2603	0,2603	1,0000	1,0000	0,154
10	MARS_R	0,1233	0,2055	0,2192	0,7260	1,0000	1,0000	0,167
11	SVM_R	0,1233	0,3151	0,4932	0,8493	1,0000	1,0000	0,238
12	FES	0,1474	0,2603	0,3699	0,8671	1,0000	1,0000	0,196
13	SVM_C	0,2425	0,4658	0,4932	0,7123	1,0000	1,0000	0,244
14	SVM composto_C	0,2515	0,4458	0,5232	0,7623	1,0000	1,0000	0,314
15	Rotation Forest_C	0,2925	0,4558	0,5432	0,7723	1,0000	1,0000	0,324
16	FESO	0,3425	0,4658	0,5932	0,8123	1,0000	1,0000	0,334

restringir esses valores, os resultados desse sistema foram superiores apenas ao sistema baseado em Aleatoriedade. Recomendando menos de 6% das atividades corretas entre as 50 primeiras posições, sua precisão ainda é baixa com valor de $MRR = 0,037$. Os baixos resultados dessa técnica acontecem devido ao fato de desconsiderar a ordem das atividades durante a geração das regras e, consequentemente, da recomendação.

O sistema baseado em *KNN* foi treinado para diferentes valores do parâmetro $k = [1 : 100]$ que representa o número de vizinhos mais próximos (de acordo com a distância Euclidiana) que serão considerados para classificar. Este sistema apresentou os melhores resultados de recomendação para o valor de $k = 2$. Mesmo assim, menos de 10% dos itens corretos foram encontrados entre as dez primeiras posições da lista e 50% dos itens entre os 50 primeiros itens. De acordo com a métrica MRR, a posição média dos itens recomendados foi distante da primeira posição da lista $MRR = 0,040$. Esses resultados indicam que classificar atividades de acordo com a distância entre grupos de vizinhos próximos não é uma abordagem adequada para o problema.

O sistema que usa uma rede neural MLP como classificador teve uma melhoria de quase quatro vezes na métrica $S@1$ de $0,0037$ para $0,0137$ em relação ao *KNN*. Para o treinamento da rede foram usados os parâmetros: i) número de neurônios η (variando entre $1 : 40$); ii) taxa de aprendizagem α (variando entre $10^{-7} : 10^{+1}$); iii) duas camadas escondidas; e iv) arquitetura totalmente conectada. Os melhores resultados de classificação foram obtidos para $\eta = 18$ e $\alpha = 10^{-4}$ obtendo 17% de itens classificados entre as dez primeiras posições da lista, e 80% entre as 50 primeiras posições, o que representa uma melhoria de 30% em relação a técnica *KNN*. O valor da métrica $MRR = 0,089$ apresentou uma taxa duas vezes mais elevada que a do *KNN*, esse aumento de precisão indica que o poder de generalizar da rede neural para solucionar problemas não lineares foi mais eficiente que a capacidade de generalização das técnicas anteriores.

O sistema baseado em CART como classificador, que tem como característica tratar dados categóricos, apresentou um resultado superior ao da rede neural. O treinamento usou os parâmetros: i) valor mínimo de divisão $\gamma = [0 : 30]$; ii) tamanho máximo da árvore final $\delta = [0 : 10000]$; iii) valor mínimo de variação para realizar uma divisão $cp = [10^{-7} : 10^{+1}]$; iv) função de divisão (ξ) como índice de Gini ou ganho de informação. O melhor resultado foi para $gamma = 0$, $\delta = 30$, $cp = 10^{-3}$ e $\xi =$ Ganho de informação.

Os resultados desse sistema foram aproximadamente duas vezes melhores que os da rede neural. Isso indica uma tendência de bons resultados para técnicas que lidem com dados categóricos por natureza. Essa melhoria indicou um aumento de 26% na métrica MRR que representa um aumento da precisão do sistema, além disso posicionou 13% dos itens procurados na primeira posição e 26% nas primeiras 50 posições.

O sistema baseado em CART como regressor, teve seu melhor valor com os parâmetros $gamma = 2$, $\delta = 20$, $cp = 10^{-5}$ e $\xi =$ Ganho de informação. A recomendação que usou valores contínuos apresentou um resultado superior ao $CART_C$ nas métricas S@1 e S@5 e um resultado inferior para S@10 e S@50, e a precisão geral (MRR) do $CART_R$ foi levemente superior.

O sistema baseado no classificador Naive Bayes obteve resultados muito próximos ao do regressor CART. O treinamento ocorreu modificando o atributo *correção de Laplace* com valores entre $[0 : 100]$. O melhor resultado ocorreu para o valor zero obtendo 34% dos itens recomendados entre as dez primeiras posições e 63% entre as 50 primeiras posições. Em contrapartida, o valor de MRR não sofreu grande variação.

O sistema baseado em regressor binomial apresentou melhoria em relação ao Naive Bayes e à rede neural (técnicas que apresentaram resultados próximos). O treinamento dessa técnica ocorre por máxima verossimilhança de um modelo generalizado linear aproximado por uma distribuição binomial. Os resultados para S@5 e S@50 foram superiores que das técnicas anteriores e o valor da métrica MRR melhorou em aproximadamente 19% em relação a técnica Naive Bayes. Isto indica que aproximar a variável dependente por uma distribuição binomial e estimar seus parâmetros por verossimilhança é uma ideia potencialmente interessante para tratar este problema.

A rede neural como regressor, que utiliza o peso da rede neural como saída, foi treinada de forma análoga à rede neural usada como classificador. O melhor resultado foi obtido para os valores de $\eta = 10$ e $\alpha = 10^{-2}$ recomendando 26% dos itens corretos entre as dez primeiras posições da lista. A precisão do sistema (MRR) melhorou 13% em relação ao regressor binomial. Esses resultados indicam que usar um regressor ao invés de um classificador apresenta um resultado melhor para esse tipo de problema, quando solucionado com redes neurais.

O sistema que usou o algoritmo MARS como regressor apresentou um resultado superior à rede neural (usada como regressor) em $12,5\%$ na métrica S@1, três vezes mais atividades recomendadas entre as 50 primeiras e um aumento de precisão geral (MRR) de 8%. Esse resultado mostra que as curvas criadas pelas diversas funções conectadas do MARS obtiveram uma generalização melhor que da rede neural. O treinamento dos parâmetros foi por verossimilhança.

O regressor SVM apresentou resultados duas vezes melhores que o algoritmo MARS para a medida S@10, pois em 49% das recomendações o item correto estava entre as dez primeiras posições da lista de recomendações. O valor de MRR também foi superior (42%). O treinamento foi feito por otimização de margem com os valores de $c = [10^{-7} : 10^{2}]$, $\epsilon = [10^{-7} : 10^{2}]$, valores de tolerância $\beta = [10^{-7} : 10^{2}]$, funções de *kernel*: i) linear; ii) sigmoide; iii) polinomial; e iv) radial, os parâmetros do *kernel* polinomial são: i) $p = [1 : 10]$ que é a potência da função. Os melhores valores encontrados foram para $c = 1$, $\epsilon = 1$, $\beta = 10^{-4}$, *kernel* polinomial com $p = 2$. Esse resultado é um indício que o problema não é linearmente separável, pois foi usada uma função de *kernel* polinomial para mapear o problema em alta dimensão e projetá-lo novamente para uma dimensão mais baixa. Os autores acreditam que esta característica foi responsável pelo bom desempenho desse regressor.

Dentre os sistemas propostos pela literatura, o sistema baseado em entrada, saída e frequência (FES) [16] é o que apresenta os melhores resultados. Nos experimentos realizados, este sistema identificou o item correto entre as dez primeiras posições da lista de recomendação em 37% dos casos, e obteve um valor de $MRR = 0,196$.

O sistema baseado no algoritmo SVM para classificação foi o único classificador que superou os resultados dos regressores. Seu treinamento foi análogo ao SVM para regressão. Sua melhor execução foi para os valores $c = 10^{-1}$, $p = 10^{-4}$ e *kernel* linear. Esta execução, para a métrica $S@1$ foi 64% melhor que a da técnica FES e o valor da precisão geral (MRR) aumentou 24%. Este resultado indica que a solução utilizando *kernel* para mapeamento em alta dimensão é uma proposta eficiente no caso de classificadores.

O sistema SVM composto, que executa sobre os resultados dos outros sistemas de recomendação, apresentou um desempenho superior ao SVM para classificação. Seu treinamento foi análogo ao do SVM_C e seu melhor desempenho foi para os parâmetros $c = 10^{-2}$, $p = 1$ e *kernel polinomial*. Houve uma melhoria de 3% na métrica $S@1$ e 28% na métrica MRR, essa melhoria é em virtude do uso do resultado de outros classificadores em conjunto com a redução de esparsidade do conjunto de dados.

O sistema utilizando *Rotation Forest* apresentou o segundo melhor resultado, seu treinamento utilizou os parâmetros: i) valor mínimo de divisão $\gamma = [0 : 30]$; ii) tamanho máximo da árvore final $\delta = [0 : 10000]$; iii) valor mínimo de variação para realizar uma divisão $cp = [10^{-7} : 10^{+1}]$; iv) função de divisão (ξ) como índice de Gini e ganho de informação; v) $K = [1 : 10]$ como número de partições; vi) $L = [1 : 10]$ como o número de classificadores; e vii) valores de corte $0,25; 0,5; 0,75$. Essa melhoria foi em virtude de usar em conjunto uma técnica de classificação do tipo *ensemble* e três limiares de corte, os quais foram estabelecidos para converter os valores numéricos (da média dos L classificadores) em valores binários.

A técnica FESO, apresentou um resultado superior às demais. Este considera o uso de frequência, entrada e saída e informações semânticas sobre as atividades. Em comparação com as demais técnicas seu resultado foi superior para todas as métricas calculadas, exceto $S@50$ para algumas técnicas. Em relação à técnica FES, seu resultado foi superior. Em particular, parte dessa melhora é justificada pelos casos em que a atividade correta teria frequência zero no conjunto de treinamento, pois ela permite recomendar baseada na ontologia (usando as atividades que contenham a ontologia do novo *workflow*). Além disso, para o caso em que há empate entre duas atividades com o critério de entrada e saída e a frequência a técnica proposta apresenta um fator a mais para ser utilizado como desempate.

Algumas tendências observadas com esses resultados foram que aumentar a informação sobre dados na recomendação melhora o seu desempenho, como o resultado dos experimentos: 2, 12 e 14 mostram. Uma segunda tendência é que o classificador SVM foi o único que obteve um melhor resultado que os regressores, indicando que soluções por maximização de espaço entre dados em alta dimensão podem ser uma área de estudo promissora. Uma terceira tendência é o uso de classificadores compostos e *ensembles*, os quais apresentaram resultados promissores. No caso do *ensemble* há um indício que técnicas desse tipo, que usem limiares para converter os valores da média dos resultados do conjunto L em valores binários, têm resultados promissores na recomendação de atividades.

Conclusão

Este trabalho desenvolveu uma técnica híbrida para recomendar atividades em *workflows* científicos, que usa compatibilidade sintática, frequência e ontologias de domínio para recomendar atividades, denominada FESO. Além disso, também modelou o problema de recomendação como um problema de regressão e classificação em inteligência artificial. Para encontrar as técnicas da literatura correlata, foi realizada uma revisão sistemática. Nessa revisão foram encontradas as técnicas, suas restrições, suas vantagens e as formas que foram validadas. O próximo passo foi implementá-las e

compará-las com as soluções propostas neste mestrado, incluindo as soluções baseadas em classificadores e regressores.

Para realizar a comparação foi organizado um banco de dados relacional de *workflows* e suas atividades. Também foi necessário estabelecer uma metodologia para comparar diferentes técnicas de recomendação de atividades para um mesmo conjunto de dados com as mesmas métricas de validação $S@k$ e MRR.

Ao comparar todas as técnicas, foram constatados determinados aspectos do conjunto de dados, como o fato das atividades não serem independentes; o problema não ser linearmente separável; e que técnicas de agrupamento não se mostraram adequadas para solucionar este problema. Com exceção do SVM, regressores apresentaram soluções mais precisas do que classificadores. Além disso, adicionar informação nos sistemas de recomendação melhorou a precisão destes.

Como trabalhos futuros pode-se usar classificadores compostos, recomendação baseada em redes sociais, obter dados sobre a proveniência das atividades para aumentar a precisão das recomendações entre outros.

Agradecimentos

Agradecemos a Pró-Reitoria de Pós-Graduação da Universidade de São Paulo (USP) e a agência CAPES que forneceram bolsas de estudo para o estudante. Permitindo completar esse mestrado com publicações na área de computação.

References

1. Wang F, Deng H, Guo L, Ji K. A Survey on Scientific Workflow Techniques for Escience in Astronomy. In: 2010 International Forum on Information Technology and Applications. vol. 1. IEEE; 2010. p. 417–420. Available from: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5634997`.

2. Bhagat J, Tanoh F, Nzuobontane E, Laurent T, Orlowski J, Roos M, et al.. BioCatalogue: a universal catalogue of web services for the life sciences; 2014. Available from: `doi:10.1093/nar/gkq394`.

3. Shao Q, Kinsy M, Chen Y. Storing and Discovering Critical Workflows from Log in Scientific Exploration. In: 2007 IEEE Congress on Services (Services 2007). IEEE; 2007. p. 209–212. Available from: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4278799`.

4. Shao Q, Sun P, Chen Y. Efficiently discovering critical workflows in scientific explorations. Future Generation Computer Systems. 2009;25(5):577–585.

5. Oliveira FTd, Braganholo V, Murta L, Mattoso M. Improving workflow design by mining reusable tasks. Journal of the Brazilian Computer Society. 2015;21(1):16.

6. Koop D. VisComplete: Automating Suggestions for Visualization Pipelines. IEEE Transactions on Visualization and Computer Graphics. 2008;14(6):1691–1698.

7. de Oliveira FT, Murta L, Werner C, Mattoso M. Using Provenance to Improve Workflow Design. In: Freire J, Koop D, Moreau L, editors. Provenance and Annotation of Data and Processes. vol. 5272 of Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2008. p. 136–143. Available from: `http://dx.doi.org/10.1007/978-3-540-89965-5_15`.

8. Wang Y, Cao J, Li M. Change Sequence Mining in Context-Aware Scientific Workflow. In: 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications. IEEE; 2009. p. 635–640. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5207868.

9. Zhang J, Liu Q, Xu K. FlowRecommender: A Workflow Recommendation Technique for Process Provenance; 2009.

10. Tan W, Zhang J, Madduri R, Foster I, De Roure D, Goble C. Providing Map and GPS Assistance to Service Composition in Bioinformatics. In: 2011 IEEE International Conference on Services Computing. IEEE; 2011. p. 632–639. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6009316.

11. Cao B, Yin J, Deng S, Wang D, Wu Z. Graph-based workflow recommendation: on improving business process modeling. In: Proceedings of the 21st ACM international conference on Information and knowledge management. CIKM '12. ACM; 2012. p. 1527–1531. Available from: http://doi.acm.org/10.1145/2396761.2398466.

12. Diamantini C, Potena D, Storti E. Mining Usage Patterns from a Repository of Scientific Workflows. In: Proceedings of the 27th Annual {ACM} Symposium on Applied Computing. SAC '12. ACM; 2012. p. 152–157. Available from: http://doi.acm.org/10.1145/2245276.2245307.

13. Garijo D, Corcho O, Gil Y. Detecting Common Scientific Workflow Fragments Using Templates and Execution Provenance. In: Proceedings of the Seventh International Conference on Knowledge Capture. K-CAP '13. New York, NY, USA: ACM; 2013. p. 33–40. Available from: http://doi.acm.org/10.1145/2479832.2479848.

14. Yeo P, Abidi SSR. Dataflow Oriented Similarity Matching for Scientific Workflows. In: 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum. IEEE; 2013. p. 2091–2100. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6651115.

15. Bomfim E, Oliveira J, de Souza JM, Strauch J. Thoth: improving experiences reuses in the scientific environment through workflow management system. In: Computer Supported Cooperative Work in Design, 2005. Proceedings of the Ninth International Conference. vol. 2; 2005. p. 1164–1170 Vol. 2. Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1504261.

16. Wang J, Han Y, Yan S, Chen W, Ji G. VINCA4Science: A Personal Workflow System for e-Science. In: Internet Computing in Science and Engineering, 2008. ICICSE '08. International Conference on; 2008. p. 444–451. Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4548305.

17. Leng Y, El-Gayyar M, Cremers AB. Semantics Enhanced Composition Planner for Distributed Resources. In: 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science. IEEE; 2010. p. 61–65. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5573302.

18. Yao J, Tan W, Nepal S, Chen S, Zhang J, De Roure D, et al. ReputationNet: A Reputation Engine to Enhance ServiceMap by Recommending Trusted Services.

In: Services Computing (SCC), 2012 IEEE Ninth International Conference on; 2012. p. 454–461. Available from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6274177.

19. Telea A, van Wijk JJ. Vission: An Object Oriented Dataflow System for Simulation and Visualization. In: PROCEEDINGS OF IEEE VISSYM; 1999. p. 95–104. Available from: http://link.springer.com/chapter/10.1007%2F978-3-7091-6803-5_21.

20. de Oliveira FT. UM SISTEMA DE RECOMENDAÇÃO PARA COMPOSIÇÃO DE WORKFLOWS. UNIVERSIDADE FEDERAL DO RIO DE JANEIRO; 2010.

21. Zhang J, Tan W, Alexander J, Foster I, Madduri R. Recommend-As-You-Go: A Novel Approach Supporting Services-Oriented Scientific Workflow Reuse. In: 2011 IEEE International Conference on Services Computing. IEEE; 2011. p. 48–55. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6009243.

22. Garijo D, Corcho O, Gil Y, Braskie MN, Hibar D, Hua X, et al. Workflow Reuse in Practice: A Study of Neuroimaging Pipeline Users. Proceedings of the 2014 IEEE 10th International Conference on eScience. 2014; p. 239–246.

23. Soomro K, Munir K, Mcclatchey R. Incorporating Semantics in Pattern-Based Scientific Workflow Recommender Systems. 2015;.

24. Zhang J, Lee C, Xiao S, Votava P, Lee TJ, Nemani R, et al. A Community-Driven Workflow Recommendations and Reuse Infrastructure. In: 2014 IEEE 8th International Symposium on Service Oriented System Engineering. IEEE; 2014. p. 162–172. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6830902.

25. Mohan A, Ebrahimi M, Lu S. 2015 IEEE International Conference on Services Computing A Folksonomy-Based Social Recommendation System for Scientific Workflow Reuse. 2015;.

26. Cerezo N, Montagnat J. Scientific Workflow Reuse Through Conceptual Workflows on the Virtual Imaging Platform. In: Proceedings of the 6th Workshop on Workflows in Support of Large-scale Science. {WORKS} '11. ACM; 2011. p. 1–10. Available from: http://doi.acm.org/10.1145/2110497.2110499.

27. Roure CG. myExperiment; 2015. Available from: http://www.myexperiment.org/.

28. Scrivano G, Niksic H. GNU Wget Introduction to GNU Wget; 2015. Available from: http://www.gnu.org/software/wget/.

29. Richardson L. Beautiful Soup; 2015. Available from: http://www.crummy.com/software/BeautifulSoup/.