Utilização de Redes Heterogêneas para Medir a Força dos Relacionamentos no GitHub

Gabriel P. Oliveira, Natércia A. Batista, Michele A. Brandão, Mirella M. Moro

Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

{gabrielpoliveira,natercia,micheleabrandao,mirella}@dcc.ufmg.br

Abstract. *Our goal is to measure the strength of the relationships between GitHub users by considering social and technical features. The contributions include a new heterogeneous graph model with different types of interactions and new metrics for the strength of such relationships. The results show the proposed metrics bring new information about the relationships.*

Resumo. *Nosso objetivo é medir a força dos relacionamentos entre usuários do GitHub considerando fatores técnicos e sociais. As contribuições incluem uma nova modelagem de grafo heterogêneo com diferentes tipos de interação e novas métricas para a força de tais relacionamentos. Os resultados mostram que as métricas propostas acrescentam informação sobre os relacionamentos.*

1. Introdução

É tarefa primordial da área de Bancos de Dados agregar ou enriquecer dados existentes a fim de extrair informações relevantes e até novos conhecimentos a partir dos mesmos. De fato, essa área (como outras da Computação) tem evoluído para abranger as mais diversas necessidades de processamento de dados atuais. Por exemplo, com a expansão da Internet e o acesso a serviços de banda larga, pessoas comuns têm à disposição uma gama de aplicativos e serviços online. Com tal acesso, geram-se diariamente grandes volumes de dados que podem ser processados com os mais variados propósitos. Entre tantas opções, um dos serviços mais populares é o de redes sociais online, que conectam pessoas a partir de seus relacionamentos pessoais e profissionais.

Neste trabalho, estudamos o GitHub, uma rede de desenvolvimento colaborativo de software. Especificamente, uma modelagem de rede heterogênea para representar diferentes tipos de colaboração e novas formas de medir a força dos relacionamentos são propostas, considerando fatores técnicos e sociais. Assim, este trabalho propõe métricas semânticas que consideram tais fatores e analisa se elas confirmam a força dos relacionamentos definida por métricas existentes ou adicionam novas informações. Em termos de aplicação, tais métricas permitem descoberta de padrões que auxiliam no estudo da formação de times, detecção de comunidades e identificação de influentes, entre outras.

2. Trabalhos Relacionados

Em redes sociais, existem diversos estudos sobre força de relacionamentos [Alves et al. 2016; Casalnuovo et al. 2015; Goyal et al. 2018]. Exemplos de métricas específicas estão nas Tabelas 1 e 2. Porém, nenhum diferencia a força dos relacionamentos em fatores técnicos e sociais em rede *heterogênea* a partir do GitHub. Tais estudos consideram apenas um fator (e.g., a intensidade de contribuições em um repositório) para determinar tal

Tabela 1. Métricas definidas por meio de propriedades topológicas.

Considere para um nó X da rede, $\mathcal{N}(X)$ como o conjunto de vizinhos de X, $w(X)$ como a soma dos pesos das arestas conectadas a X e $w(X,Y)$ como o peso da aresta entre X e Y.

Métricas topológicas	
Neighborhood Overlap (NO) - [Easley and Kleinberg 2010]	É uma maneira de medir a força das ligações entre nós por meio da similaridade de seus vizinhos: $NO_{(X,Y)} = \frac{\lvert\mathcal{N}(X)\cap\mathcal{N}(Y)\rvert}{\lvert\mathcal{N}(X)\cup\mathcal{N}(Y)-\{X,Y\}\rvert}$.
Preferencial Attachment (PA) - [Barabási and Albert 1999]	Há uma relação linear entre o número de vizinhos de um nó e a sua probabilidade de conectar-se a outro nó: $PA_{(X,Y)} = \lvert\mathcal{N}(X)\rvert\lvert\mathcal{N}(Y)\rvert$.
Adamic-Adar [2003] (AA)	Dá maior peso aos vizinhos que não se relacionam com muitos outros: $AA_{(X,Y)} = \sum_{\forall Z\in\mathcal{N}(X)\cap\mathcal{N}(Y)} \frac{1}{log\lvert\mathcal{N}(Z)\rvert}$.
Métricas topológicas ponderadas	
Tieness (T)* - [Brandão and Moro 2017]	Mede a força das relações de coautoria; no contexto do GitHub: $T_{(X,Y)} = \frac{\frac{\lvert\mathcal{N}(X)\cap\mathcal{N}(Y)\rvert+1}{1+\lvert\mathcal{N}(X)\cup\mathcal{N}(Y)-\{X,Y\}\rvert}\lvert\lvert w(X,Y)\rvert\rvert}{2}$, onde o valor do peso $w(X,Y)$ é normalizado.

** Utilizada em conjunto com cada uma das métricas semânticas da Tabela 2, cujo valor é o peso $w(X,Y)$.*

Tabela 2. Métricas definidas por meio de propriedades semânticas.

Propostas por Alves et al. [2016] e Batista et al. [2017a], consideram a semântica das relações no contexto específico do GitHub. Seja $\mathcal{R}$ o conjunto de todos os repositórios onde dois usuários X e Y colaboram.

Métricas semânticas no contexto do GitHub	
Number of Shared Repositories (SR)	Definida como o número de repositórios compartilhados entre dois desenvolvedores: $SR_{(X,Y)} = \lvert\mathcal{R}\rvert$.
Jointly Developers Contribution to Shared Repositories (JCSR)	Seja $JCSR_{(X,Y,r_i)}$ a razão entre a quantidade de desenvolvedores no par e o total de desenvolvedores em r_i, $JCSR_{(X,Y)} = \frac{\sum_{\forall r_i\in\mathcal{R}} JCSR_{(X,Y,r_i)}}{\lvert\mathcal{R}\rvert}$.
Jointly Developers Commits to Shared Repositories (JCOSR)	Sejam $NC_{(X,r_i)}$ o número de *commits* feitos por um usuário X em um repositório r_i e $NC_{(r_i)}$ o número total de *commits* no repositório r_i: $JCOSR_{(X,Y)} = \sum_{\forall r_i\in\mathcal{R}} \frac{NC_{(X,r_i)}+NC_{(Y,r_i)}}{NC_{(r_i)}}$.
Previous Collaboration (PC)	Seja $ND_{(r_i,t)}$ o número de desenvolvedores no repositório r_i no tempo t, então a quantidade de colaboração de X e Y em t: $PC_{(X,Y,t)} = \frac{\sum_{\forall r_i\in\mathcal{R}} \frac{1}{ND_{(r_i,t)}}}{\lvert\mathcal{R}\rvert}$.
Global Potential Contribution (GPC)	Seja $T_{(X,Y,r_i)}$ o intervalo de tempo em que os desenvolvedores X e Y contribuem no repositório r_i e $\mathcal{D}$ o conjunto de todos os desenvolvedores na rede: $GPC_{(X,Y)} = \frac{\sum_{\forall r_i\in\mathcal{R}} T_{(X,Y,r_i)}}{max_{\forall(D_i,D_j)\in\mathcal{D},r_i\in\mathcal{R}} T_{(D_i,D_j,r_i)}}$.

força em uma rede homogênea. Assim, a modelagem *heterogênea* proposta aqui permite uma definição mais ampla e completa da força dos relacionamentos.

3. Rede Social de Colaboração

Esta seção descreve a base de dados utilizada para criar a rede social e a modelagem da rede heterogênea de desenvolvimento colaborativo de software.

Base de Dados. A base de dados utilizada é originada do GitSED 2015 (*GitHub Socially Enhanced Dataset*) [Batista et al. 2017b], um conjunto de dados do GitHub curado, expandido e enriquecido a partir do GHTorrent [Gousios 2013]. A versão original do GitSED considera repositórios desenvolvidos em apenas três linguagens de programação. Dessa forma, expandimos a base de dados para considerar seis linguagens subdivididas

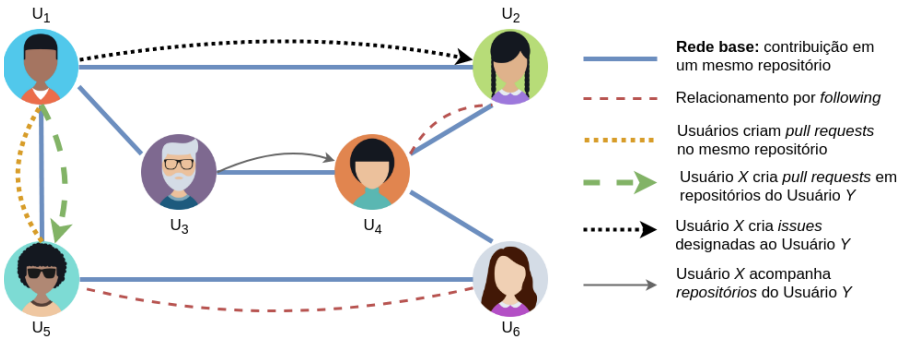

Figura 1. Exemplo da rede heterogênea de colaboração do GitHub com novos tipos de relacionamento considerados.

em dois grupos, de acordo com seu nível de colaboração definido por Rocha et al. [2016]: linguagens mais colaborativas (JavaScript, Ruby e Python) e linguagens menos colaborativas (Assembly, Pascal e Visual Basic). Utilizando a mesma metodologia de coleta e curadoria, atualizou-se o GitSED com dados do GHTorrent até maio de 2017. Portanto, tem-se uma versão *ampliada* do conjunto de dados, disponibilizada publicamente[1].

Modelagem da Rede. A partir da rede base com relacionamentos de colaboração no mesmo repositório, a nova modelagem contém seis (dos quais cinco são novos e aqui propostos) tipos de arestas: (i) colaboração no mesmo repositório (obrigatório, pois são desenvolvedores-colaboradores); (ii) seguidores; (iii) criação de *pull requests* em um mesmo repositório; (iv) criação de *pull requests* em repositório de outro usuário; (v) criação de *issues* designadas a outro usuário; e (vi) acompanhamento de repositórios favoritos. Assim, a rede se torna *heterogênea* representada por um multigrafo $\mathcal{G}' = (\mathcal{V}, \mathcal{E}_1 \cup \mathcal{E}_2 \cup \cdots \cup \mathcal{E}_6)$, onde o conjunto $\mathcal{V}$ continua com vértices para os desenvolvedores da rede; porém, cada conjunto $\mathcal{E}_k$, para $k \in \{1, \cdots, 6\}$, representa um dos seis tipos de arestas. Ademais, o peso das arestas também é calculado por métricas topológicas descritas na Tabela 1, ou semânticas na Tabela 2.

4. Novas Métricas Semânticas

A Figura 1 ilustra os seis tipos de relacionamentos presentes na nova modelagem heterogênea do GitHub. Então, a Tabela 3 apresenta as métricas propostas a partir de propriedades semânticas no contexto do GitHub para medir a força dos relacionamentos.

5. Caracterização das Redes, Análise e Correlação das Métricas

Esta seção apresenta os resultados obtidos na caracterização das redes de colaboração de cada linguagem e no estudo da correlação das diferentes métricas.

Caracterização das Redes Heterogêneas de Colaboração. As seis linguagens de programação foram analisadas para verificar se o nível de colaboração de cada uma se mantém em relação à classificação feita por Rocha et al. [2016] (note que o dataset foi *expandido*). As Tabelas 4 e 5 mostram os resultados dessa caracterização. Para as mais colaborativas (Tabela 4), as três possuem um alto índice de nós e arestas em seus maiores componentes conectados (*Giant Component* – GC). Em todas elas, mais de 80% das

[1]Projeto Apoena: `http://www.dcc.ufmg.br/~mirella/projs/apoena`

Tabela 3. Novas métricas para a força dos relacionamentos no GitHub.
Fatores técnicos conectam desenvolvedores por meio dos aspectos de desenvolvimento de software, enquanto que fatores sociais consideram aspectos da interação *direta* entre dois usuários no GitHub.

Novas métricas semânticas considerando fatores técnicos	
Unidirectional Assigned Issues (UAI)	Sejam $NI_{(X,Y)}$ o número de *issues*[2] criadas pelo usuário X que são designadas ao usuário Y e $NTI_{(Y)}$ o número total de *issues* designadas ao usuário Y: $UAI_{(X,Y)} = \frac{NI_{(X,Y)}}{NTI_{(Y)}}$.
Unidirectional Pull Requests (UPR)	Sejam $PR_{(X,Y)}$ o número de *pull requests*[3] que o usuário X cria em repositórios do usuário Y e $TPR_{(Y)}$ o número total de *pull requests* que os repositórios do usuário Y possuem: $UPR_{(X,Y)} = \frac{PR_{(X,Y)}}{TPR_{(Y)}}$.
Bidirectional Pull Requests (BPR)	Sejam $NPR_{(X,r)}$ o número de *pull requests* que o usuário X cria em um repositório r, $NTPR_{(r)}$ o número total de *pull requests* em r e $\mathcal{U}$ o conjunto universo dos repositórios existentes na base. Para cada par (X, Y) em um repositório $r \in \mathcal{U}$, se $NPR_{(X,r)} \neq 0$ e $NPR_{(Y,r)} \neq 0$: $BPR_{(X,Y)} = \sum_{\forall r_i \in \mathcal{U}} \frac{NPR_{(X,r_i)} + NPR_{(Y,r_i)}}{NTPR_{(r_i)}}$.
Novas métricas semânticas considerando fatores sociais	
Bidirectional Intensity of Followers (BIF)	A intensidade de seguidores é definida com base na relação onde um usuário X *segue* um usuário Y no GitHub. Propõe-se os seguintes valores para medir tal intensidade: $BIF_{(X,Y)} = \begin{cases} 1 & \text{se } X \text{ segue } Y \textbf{ AND } Y \text{ segue } X \\ 0,5 & \text{se } X \text{ segue } Y \textbf{ XOR } Y \text{ segue } X \\ 0 & \text{caso contrário} \end{cases}$
Unidirectional Intensity of starMarks (UIM)	Sejam $NS_{(X,Y)}$ o número de repositórios de um usuário Y nos quais X tem interesse (clicando no botão *star*) e $NRS_{(X)}$ o número total de repositórios nos quais X está interessado: $UIM_{(X,Y)} = \frac{NS_{(X,Y)}}{NRS_{(X)}}$.

arestas da rede estão presentes no GC. Então tais redes são bem conectadas, confirmando a classificação de Rocha et al. [2016]. Para menos colaborativas (Tabela 5), as três possuem comportamentos semelhantes em suas redes, todas com baixo grau médio e baixo índice de arestas no GC. Portanto, infere-se que os repositórios dessas linguagens são em sua maioria compostos por poucos desenvolvedores, que colaboram pouco entre si. Devido a restrições de espaço, as análises seguintes consideram Ruby e Visual Basic como representantes das linguagens mais e menos colaborativas, respectivamente.

Análise das Novas Métricas Semânticas. Para verificar a independência entre as novas métricas, a correlação entre elas é analisada por meio dos coeficientes de Pearson e Spearman (verificam relações lineares e monotônicas entre as métricas, respectivamente). Por limitações de espaço, os resultados das correlações são apenas discutidos. Em todas as linguagens, observa-se que a correlação entre as novas métricas é baixa ou insubstancial, com valores próximos a zero. Tal resultado pode ser explicado pelo fato de que cada métrica considera fatores diferentes para calcular a interação entre os pares. A exceção está nas métricas UPR e BPR, que analisam a interação por meio da criação de *pull requests*. A baixa correlação entre as métricas é um forte indicador de que as propriedades semânticas consideradas adicionam novas informações à rede de colaboração do GitHub.

Para entender o quanto cada tipo de relacionamento representa da rede, a Tabela 6 mostra a proporção de desenvolvedores e arestas presentes quando são retirados os pares que não contêm o tipo de relacionamento representado por cada métrica. Os resultados mostram que poucos nós e arestas permanecem na rede para métricas com fatores sociais (BIF e UIM). Ou seja, desenvolvedores do GitHub não consideram tais funcionalidades

Tabela 4. Estatísticas das redes das linguagens mais colaborativas.

	JavaScript	Python	Ruby
Número de repositórios	6.767.297	3.074.827	2.536.133
Número de nós (desenvolvedores)	854.255	519.771	279.281
Número de arestas	2.571.154	3.699.096	33.979.590
Densidade (10^{-3})	0,007	0,027	0,871
Grau médio	6,02	14,23	243,34
Coeficiente de Clusterização Médio	0,358	0,384	0,429
Número de nós no GC*	379.637 (44,4%)	259.355 (49,9%)	180.175 (64,5%)
Número de arestas no GC*	2.105.747 (81,9%)	3.282.140 (88,7%)	33.873.748 (99,7%)

** Giant Component (GC): maior componente conectado de um grafo*

Tabela 5. Estatísticas das redes das linguagens menos colaborativas.

	Assembly	Pascal	Visual Basic
Número de repositórios	35.073	20.330	33.275
Número de nós (desenvolvedores)	7.516	3.520	5.602
Número de arestas	14.906	9.377	7.205
Densidade (10^{-3})	0,528	1,514	0,459
Grau médio	3,97	5,3	2,57
Coeficiente de Clusterização Médio	0,354	0,374	0,311
Número de nós no GC*	483 (6,4%)	577 (16,4%)	95 (1,7%)
Número de arestas no GC*	3.335 (22,3%)	5.140 (54,8%)	1.368 (19%)

** Giant Component (GC): maior componente conectado de um grafo*

Tabela 6. Participação na rede a partir de métricas selecionadas.

Métrica	# de desenvolvedores		# de arestas	
	Ruby	Visual Basic	Ruby	Visual Basic
Rede completa	279.281 (100%)	5.602 (100%)	33.979.590 (100%)	7.205 (100%)
UAI - *Unidirecional Assigned Issues*	9.982 (3,57%)	124 (2,21%)	8.517 (0,02%)	82 (1,14%)
UPR - *Unidirecional Pull Requests*	14.811 (5,3%)	166 (2,96%)	10.927 (0,03%)	104 (1,44%)
BPR - *Bidirectional Pull Requests*	57.604 (20,63%)	453 (8,09%)	1.697.256 (5%)	391 (5,43%)
BIF - *Bidirecional Intensity of Followers*	58.715 (21,02%)	923 (16,48%)	102.736 (0,3%)	616 (8,55%)
UIM - *Unidirecional Intensity of starMarks*	13.248 (4,74%)	107 (1,91%)	15.577 (0,04%)	56 (0,78%)

quando colaboram em um repositório. Em relação às métricas com fatores técnicos (UAI, UPR e BPR), o baixo número de nós e arestas é explicado pelo fato de que grande parte das *issues* e *pull requests* nos repositórios provém de usuários externos.

Correlação com as Métricas Existentes. Para analisar a correlação entre as métricas existentes (NO, PA, AA, T, PC, GPC, SR, JCSR, JCOSR) e as propostas (BPR, UAI, UPR, BIF e UIM), utilizamos o coeficiente de Pearson e Spearman. Os resultados são similares para ambos em todas as linguagens, com algumas exceções (novamente, devido a limitações de espaço, os resultados são discutidos sem gráficos ou tabelas). Não há correlação significativa linear ou monotônica entre a maioria das propriedades. Uma das exceções são AA e PA com correlação acima de 0,87 considerando todas as linguagens. Se PA e AA estão correlacionadas, então é esperado que uma métrica correlacionada com AA também esteja com PA e vice-versa. Outra exceção é JCSR (contribuição conjunta em repositórios) em correlação negativa no intervalo [-0,95;-0,57] com PA e AA para todas as linguagens. Isso pode indicar que pares de desenvolvedores com intensa contribuição conjunta em repositórios podem não se conectar com muitos outros desenvolvedores.

Uma diferença significativa entre as linguagens mais e menos colaborativas é a existência de correlação negativa no intervalo [-0,9;-0,4] entre JCOSR com GPC, PA, AA e NO para Assembly, Pascal e Visual Basic. Tal resultado indica que nas linguagens menos colaborativas, os relacionamentos de um par de desenvolvedores com seus vizinhos influenciam negativamente à contribuição conjunta por *commits*. Por outro lado, a

correlação negativa entre JCOSR e GPC indica que pares de desenvolvedores tendem a não contribuir por muito tempo. Existe também forte correlação entre as métricas *Tieness* ponderadas com propriedades semânticas, ou seja, *Tieness* com diferentes pesos traz as mesmas informações à análise dos relacionamentos, e assim, pode-se escolher apenas uma delas. Nesse caso, recomenda-se o uso de métricas com baixo custo computacional, como T_BIF ou T_SR (combinação de T com BIF e T com SR, respectivamente).

6. Conclusão e Trabalhos Futuros

Este artigo apresentou uma nova modelagem heterogênea para a rede de colaboração entre desenvolvedores no GitHub, bem como novas métricas semânticas para a força dos relacionamentos. A análise revelou que todas representam informações novas sobre os relacionamentos. Ademais, altas correlações negativas entre métricas que consideram a contribuição em repositórios e métricas baseadas em vizinhos mostraram que a colaboração é mais intensa entre pares de desenvolvedores com menos vizinhos. Como trabalho futuro, pretende-se investigar a relação dessas propriedades semânticas com o ranqueamento e a influência de desenvolvedores no GitHub. Também planeja-se investigar como fatores técnicos são influenciados por fatores sociais.

Agradecimentos. Trabalho parcialmente financiado por CAPES, CNPq e FAPEMIG.

Referências

Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the Web. *Social Networks*, 25(3):211 – 230.

Alves, G. B., Brandão, M. A., Santana, D. M., da Silva, A. P. C., and Moro, M. M. (2016). The Strength of Social Coding Collaboration on GitHub. In *SBBD - Short Papers*.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Batista, N. A., Brandão, M. A., Alves, G. B., da Silva, A. P. C., and Moro, M. M. (2017a). Collaboration Strength Metrics and Analyses on GitHub. In *WI*, pages 170–178.

Batista, N. A. et al. (2017b). GitSED: Um Conjunto de Dados com Informações Sociais Baseado no GitHub. In *SBBD - Dataset Showcase Workshop*, pages 224–233.

Brandão, M. A. and Moro, M. M. (2017). The strength of co-authorship ties through different topological properties. *JBCS*, 23(1):5.

Casalnuovo, C. et al. (2015). Developer onboarding in GitHub: the role of prior social links and language experience. In *ESEC/FSE*, pages 817–828.

Easley, D. and Kleinberg, J. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press.

Gousios, G. (2013). The GHTorrent Dataset and Tool Suite. In *MSR*, pages 233–236.

Goyal, R. et al. (2018). Identifying unusual commits on GitHub. *Journal of Software: Evolution and Process*, 30(1).

Rocha, L. M. A., Silva, T. H. P., and Moro, M. M. (2016). Análise da Contribuição para Código entre Repositórios do GitHub. In *SBBD - Short Papers*, pages 103–108.