

Análise de Influenciadores de Redes Sociais Digitais

Marina S. Martins¹, Sérgio Lifschitz¹

¹Departamento de Informática – (PUC-Rio) - Rio de Janeiro - RJ

{marina_schuler}@aluno.puc-rio.br,

{sergio}@inf.puc-rio.br

Abstract. *Digital Social Networks data enable effective analyses that facilitates the understanding of social phenomena. This work focuses on the concept of Digital Influencers, exploring implementations to identify and analyze their impact on DSNs. Our inferences include the users' influence in DSNs representative graphs and the impact and reach of their published content's engagement.*

Resumo. *Os dados das Redes Sociais Digitais permitem análises que facilitam a compreensão de fenômenos sociais. Este trabalho foca no conceito de Influenciadores, explorando implementações para identificar e analisar seu impacto nas RSDs. Nossas inferências incluem a influência de usuários em grafos representativos da rede e o impacto do engajamento e alcance de seus conteúdos.*

1. Introdução

Os dados provenientes das Redes Sociais Digitais (RSD) representam amostras volumosas, as quais possibilitam análises em diversas áreas do conhecimento e permitem a compreensão de inúmeros fenômenos sociais. As RSDs podem ser representadas por grafos cujos nós simbolizam os usuários e as arestas simbolizam o relacionamento entre eles. Duas métricas comuns aos grafos serão utilizadas nas implementações do presente artigo: Grau dos nós e *Betweenness Centrality* (BC). O grau de um nó é o número de conexões que o nó possui com outros nós do grafo. No caso de grafos direcionados, o grau de entrada é o número de arestas que chegam a um nó e o grau de saída é o número de arestas que saem de um nó. Já a BC de um nó i é o número de caminhos mínimos entre todos os pares de nós em um grafo que passam pelo nó i . Nós com maior BC estão presentes em um número grande de caminhos mínimos e, portanto, são mais importantes para a estrutura da rede [Benevenuto et al. 2011].

Dada a relevância desses grandes volumes de dados das RSDs, foi realizada anteriormente uma vasta pesquisa na literatura visando identificar conceitos chave que pudessem gerar análises específicas para RSDs [Martins 2023]. Em seguida, foi oferecida uma abordagem de implementação para a aplicação dessas avaliações em estudos específicos da área. Dentre as implementações desenvolvidas, este artigo explora em detalhes aquelas análises relativas ao conceito de Influenciadores.

2. Influenciadores

Influenciador ou *digital influencer* é um termo que se refere a usuários com muitas conexões em redes sociais, cujas interações e produção de conteúdos possuem grande poder de influência [Jun and Yi 2020]. A missão do influenciador é manter o alto engajamento

em seus conteúdos e conquistar uma ampla base de conexões, portanto, métricas como o número de seguidores são comumente usadas para identificá-los e categorizá-los.

Embora essa categorização forneça uma visão geral sobre a possível influência de um nó, apenas o número de conexões não é suficiente para uma análise mais profunda. Portanto, tal análise pode ser complementada utilizando-se de outros fatores, como a identificação de usuários cujos conteúdos possuem alto engajamento médio e de usuários que sejam nós centrais da rede, ou seja, que possuem grau e *betweenness* elevados.

As implementações desse estudo foram aplicadas sobre o banco de dados relacional ¹ construído conforme o Esquema Conceitual Genérico e Global para Redes Sociais Digitais (RSD) de [Salgueiro 2023]². O esquema proposto fornece semântica aos dados das RSDs e define como suas entidades se relacionam, estabelecendo uma base consistente para esse estudo. Tal esquema facilita a adaptação das implementações desenvolvidas para redes específicas, ao mesmo tempo em que permite a realização de análises simultâneas sobre dados de redes distintas, já que trata as múltiplas redes utilizando os mesmos conceitos.

3. Implementação das Análises

As funções foram desenvolvidas em Python e SQL e visaram uma abordagem generalista, oferecendo flexibilidade para futuras pesquisas e permitindo fácil adaptação para incorporar mais interações às análises, alterar métricas e ajustar-se a contextos específicos.

Primeiramente, é construído o grafo **G** cujos nós são todas as *accounts* da rede escolhida e as arestas são todas as conexões entre esses nós. Assim, a Função 2 recebe como parâmetro as *connections* do tipo *Follow* entre as *accounts* da amostra de teste, coletadas através da Função 1. O grafo gerado no estudo é direcionado, já que a RSD em análise foi o Instagram cujas *connections* entre *accounts* (*Follow*) possuem direção.

Função 1. Função que retorna as *connections* de determinado tipo entre as *accounts* de determinada plataforma

```
1 SELECT fk_Account_platform_1, fk_Account_id_1, fk_Account_platform_2, fk_Account_id_2
2 FROM Connects WHERE fk_Account_platform_1 = %s AND fk_Account_platform_1 =
   fk_Account_platform_2 AND type = %s;
```

Função 2. Função que gera o grafo direcional para as *accounts* e *connections* recebidas

```
1 def gen_diGraph(self, rows, rows2):
2     try:
3         G = nx.DiGraph()
4         nodes = set()
5         for row in rows:
6             node1 = (row[1])
7             if node1 not in nodes: nodes.add(node1)
8             if not rows2:
9                 node2 = (row[3])
10                if node2 not in nodes: nodes.add(node2)
11                G.add_nodes_from(nodes)
12            if rows2:
13                for row in rows2:
```

¹Outros modelos poderiam ser adequados; contudo, o presente trabalho visa explorar os limites e capacidades do modelo relacional.

²Recomenda-se sua leitura para maiores detalhes sobre o esquema.

```

14         node1 = (row[1])
15         node2 = (row[3])
16         if node1 in nodes and node2 in nodes: G.add_edge(node1, node2)
17     else:
18         for row in rows:
19             node1 = (row[1])
20             node2 = (row[3])
21             G.add_edge(node1, node2)
22     return G
23 except Exception as e:
24     return None

```

O grafo **G**, por sua vez, é passado como parâmetro para a Função 3 que calcula a influência das *accounts* na rede representada por **G**.

Função 3. Função que calcula a influência das *accounts* na rede (grafo **G**)

```

1 def analyze_accounts_as_influencers(self, G):
2     bc_and_degree = self.betweenness_and_degree(G)
3     accounts = [el["node"] for el in bc_and_degree]
4     avg_eng_by_account = self.average_engagement_by_account(accounts)
5     max_bc = max(bc_and_degree, key=lambda x: x["bc"])["bc"]
6     min_bc = min(bc_and_degree, key=lambda x: x["bc"])["bc"]
7     max_in_dg = max(bc_and_degree, key=lambda x: x["in_degree"])["in_degree"]
8     min_in_dg = min(bc_and_degree, key=lambda x: x["in_degree"])["in_degree"]
9     max_avg_eng = max(avg_eng_by_account, key=lambda x: x["avg_eng"])["avg_eng"]
10    min_avg_eng = min(avg_eng_by_account, key=lambda x: x["avg_eng"])["avg_eng"]
11
12    influencer_analysis = []
13    for bd_item in bc_and_degree:
14        account_id = bd_item["node"]
15        for ae_item in avg_eng_by_account:
16            if ae_item["account"] == account_id:
17                betweenness = bd_item["bc"]
18                normalized_bc = (betweenness - min_bc) / (max_bc - min_bc) if max_bc
19                    != min_bc else 0.5
20                in_degree = bd_item["in_degree"]
21                normalized_in_degree = (in_degree - min_in_dg) / (max_in_dg -
22                    min_in_dg) if max_in_dg != min_in_dg else 0.5
23                out_degree = bd_item["out_degree"]
24                avg_eng = ae_item["avg_eng"]
25                normalized_avg_eng = (avg_eng - min_avg_eng) / (max_avg_eng -
26                    min_avg_eng) if max_avg_eng != min_avg_eng else 0.5
27                influencer_analysis.append({
28                    "account": account_id,
29                    "betweenness": round(betweenness, 3),
30                    "in_degree": in_degree, "out_degree": out_degree,
31                    "average_engagement_of_contents": avg_eng,
32                    "normalized_betweenness": round(normalized_bc, 2),
33                    "normalized_in_degree": round(normalized_in_degree, 2),
34                    "normalized_average_engagement": round(normalized_avg_eng, 2),
35                    "influence": round(statistics.mean([normalized_bc,
36                        normalized_in_degree, normalized_avg_eng]), 2)
37                })
38        break
39
40    return sorted(influencer_analysis, key=lambda x: x['influence'], reverse=True)

```

A Função 3 utiliza a Função 4, para calcular *betweenness centrality* e graus de entrada e saída dos nós do grafo **G**, e a Função 5 e suas auxiliares, Função 6 e Função 7, para calcular o engajamento médio dos conteúdos de cada nó de **G**.

Função 4. Função que calcula *betweenness* e graus de entrada e saída dos nós do grafo

```

1 def betweenness_and_degree(self, G):
2     betweenness centrality = nx.betweenness centrality(G)

```

```

3  betweenness_and_degree = []
4  for node, bc in betweenness centrality.items():
5      betweenness_and_degree.append({
6          "node": node, "in_degree": G.in_degree(node), "out_degree": G.out_degree
          (node), "bc": bc
7      })
8  return betweenness_and_degree

```

Função 5. Função que calcula o engajamento médio dos conteúdos criados pelas *accounts* recebidas

```

1  def average_engagement_by_account(self, accounts):
2      dao = DBdao(self.db_host, self.db_name, self.db_user, self.db_pwd, self.db_port)
3      all_social_media_contents = []
4      social_media_contents_by_account = []
5
6      for account in accounts:
7          social_media_contents = dao.get_social_media_contents_created_by_account(
            account)
8          if social_media_contents:
9              all_social_media_contents.extend(social_media_contents)
10             contents_id = [smc[0] for smc in social_media_contents]
11             social_media_contents_by_account.append({"account": account, "
                social_media_contents": contents_id})
12         else:
13             social_media_contents_by_account.append({"account": account, "
                social_media_contents": None})
14
15         e = Engagement()
16         eng = e.calculate_engagement(all_social_media_contents)
17         eng_map = {item['content']: item['engagement'] for item in eng}
18         average_engagement_by_account = []
19         for account in social_media_contents_by_account:
20             smcs = account['social_media_contents']
21             if smcs:
22                 total_engagement = sum(eng_map[smc] for smc in smcs if smc in eng_map)
23                 average_engagement = total_engagement / len(smcs)
24             else:
25                 average_engagement = 0
26             average_engagement_by_account.append({'account': account['account'],
27                 'avg_eng': round(average_engagement, 2)})
28
29         return average_engagement_by_account

```

Função 6. Função que retorna o valor para o engajamento dos conteúdos recebidos

```

1  def calculate_engagement(self, rows):
2      share_sum = sum(row[1] for row in rows)
3      quote_sum = sum(row[2] for row in rows)
4      like_sum = sum(row[3] for row in rows)
5      reply_sum = sum(row[4] for row in rows)
6      react_sum = sum(row[5] for row in rows)
7
8      eta = share_sum + quote_sum + like_sum + reply_sum + react_sum
9      n = 5 # types of interactions considered (share, quote, like, reply and react)
10     share_weight = self.f_weight(share_sum, eta, n)
11     quote_weight = self.f_weight(quote_sum, eta, n)
12     like_weight = self.f_weight(like_sum, eta, n)
13     reply_weight = self.f_weight(reply_sum, eta, n)
14     react_weight = self.f_weight(react_sum, eta, n)
15
16     engagement = [{"content": row[0], "share": row[1], "quote": row[2], "like": row
        [3], "reply": row[4], "react": row[5], "engagement": round(row[1]*
        share_weight + row[2]*quote_weight + row[3]*like_weight + row[4]*
        reply_weight + row[5]*react_weight, 2)} for row in rows]
17
18
19     return sorted(engagement, key=lambda x: x['engagement'], reverse=True)

```

Função 7. Função que calcula o peso de determinada interação

```

1 def f_weight(self, interaction_sum, eta, n):
2     return 1 / ((interaction_sum / eta) * n)

```

Para o cálculo do engajamento dos conteúdos nesse estudo utilizou-se como base a fórmula para definir engajamento da pesquisa de [Silva et al. 2021], que propõe um cálculo utilizando pesos não arbitrários para as interações. A fórmula sugerida pelos autores foi adaptada para considerar as interações: *Share*, *Quote*, *Like*, *Reply* e *React*.

Além do uso das métricas apresentadas, outra abordagem interessante para análise de influência na rede é verificar o alcance de um conteúdo a partir da sua origem, já que esse pode alcançar e receber interações de *accounts* que não tem ligação direta com seu criador, saindo da esfera de conexões diretas do mesmo. Nesses casos, conteúdos que atingem vários níveis a partir da sua origem podem aumentar sua influência na rede. Para essa abordagem, foi construído um grafo utilizando a Função 2, a qual recebeu as *accounts* que interagiram com o conteúdo em análise e as *connections* entre essas *accounts*, através das consultas realizadas pelas Funções 1 e 8.

Função 8. Função que retorna as *accounts* que interagiram com determinado conteúdo

```

1 (SELECT r.fk_account_platform, r.fk_account_id
2 FROM Reacts r WHERE r.fk_Social_Media_Content_id = %s)
3 UNION
4 (SELECT cr.fk_account_platform, cr.fk_account_id
5 FROM References_Table rt LEFT JOIN Creates cr
6 ON cr.fk_Social_Media_Content_id = rt.fk_Social_Media_Content_1
7 WHERE rt.fk_Social_Media_Content_2 = %s)
8 UNION
9 (SELECT cr.fk_account_platform, cr.fk_account_id
10 FROM Creates cr WHERE cr.fk_Social_Media_Content_id = %s);

```

O grafo G gerado é direcionado e suas arestas saem da *account* que segue e chegam à *account* seguida. Para essa análise é necessário inverter o sentido dessas arestas, já que para percorrer os caminhos do conteúdo as arestas precisam sair da *account* seguida e chegar às *accounts* que a seguem. Após invertido o grafo G , deve-se submeter o grafo $G^{\text{invertido}}$ à uma busca em largura (*Breadth-First Search*)³ partindo do nó relativo à *account* criadora do conteúdo, para identificar quantos níveis foram alcançados por ele.

4. Resultados Obtidos

No presente estudo, o grafo G , ilustrado na Figura 1, foi construído a partir das conexões de dez *accounts* com *connections* do tipo *Follow*. O grafo G' , gerado com base nas *connections* de G , foi utilizado para determinar o alcance do conteúdo x criado pela *account* 1, e está ilustrado na Figura 2.

Como identificado na Figura 2, o alcance do conteúdo x nesse experimento é de 3 níveis a partir da *account* 1. Nesse cenário, um dos caminhos possíveis para que a *account* 9 (nível 3) tenha entrado em contato com o conteúdo x e interagido com o mesmo é através da *account* 5 (nível 2) que está conectada à *account* 2 (nível 1). A *account* 2 pode ter repostado o conteúdo, sendo assim visto pela *account* 5, ou interagido com o

³A busca em largura foi implementada em [Martins 2024], considerando especificamente o grafo apresentado, mas foi omitida devido ao limite de páginas, já que trata-se de um algoritmo amplamente conhecido.

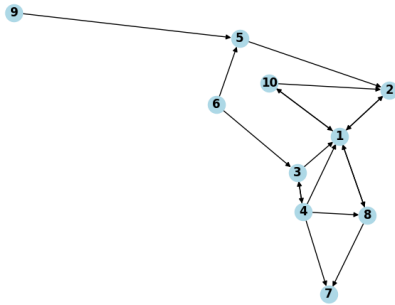


Figura 1. Grafo G das connections entre todas as accounts

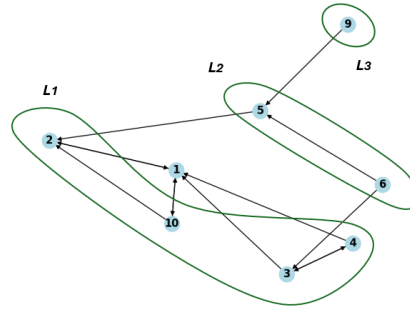


Figura 2. Grafo G' das connections entre as accounts que interagiram com o conteúdo criado por 1

mesmo, fazendo com que o algoritmo da RSD tenha recomendado-o para a *account* 5. Análogamente para a *account* 9 em relação à *account* 5.

Por fim, o cálculo da influência das *accounts*, detalhado na seção 3, é definido pela média entre os valores normalizados do engajamento médio dos conteúdos criados pela *account*, da *betweenness centrality* da *account* e do grau de entrada (*followers*) da *account*. Os resultados do teste de influência utilizando a implementação apresentada realizado sobre o grafo **G** descrito estão exibidos na tabela da Figura 3.

Account	Betweenness	In Degree	Out Degree	Average engagement of contents	Normalized Betweenness	Normalized In degree	Normalized average engagement	Influence
1	0.264	5	3	581.27	1.0	1.0	1.0	1.0
2	0.111	3	1	276.0	0.42	0.60	0.28	0.43
5	0.083	2	1	198.36	0.32	0.40	0.09	0.27
3	0.069	2	2	191.75	0.26	0.4	0.08	0.25
7	0.0	2	0	305.36	0.0	0.4	0.35	0.25
8	0.069	2	2	159.82	0.26	0.4	0.0	0.22
4	0.042	1	4	252.47	0.16	0.2	0.22	0.19
10	0.0	1	2	263.55	0.0	0.2	0.25	0.15
6	0.0	0	2	250.25	0.0	0.0	0.21	0.07
9	0.0	0	1	215.35	0.0	0.0	0.13	0.04

Figura 3. Resultados dos testes de influência das accounts

5. Conclusões

As implementações desenvolvidas permitem múltiplas inferências acerca da influência de usuários numa rede, desde a força de nós centrais em relação a rede modelada em grafo até o impacto do engajamento e do alcance de seus conteúdos nessa influência. Essas implementações e informações adicionais estão disponíveis no GitHub⁴, oferecendo base a análises pertinentes ao contexto das RSDs para pesquisadores da área.

As implementações para outros conceitos como Viralizar, Engajamento e Consumidores Leais foram desenvolvidas em [Martins 2024], algumas de suas funções foram sobretudo utilizadas nas análises deste artigo. Em trabalhos futuros serão publicados outros programas como este para tais conceitos, não mostrados no presente artigo por conta do limite de espaço.

⁴Disponível em: <https://github.com/mschuler10/analises-rsd.git>.

Referências

- Benevenuto, F., Almeida, J. M., and Silva, A. (2011). Coleta e análise de grandes bases de dados de redes sociais online. In *Jornadas de Atualização em Informática (JAI)*, pages 11–57.
- Jun, S. and Yi, J. (2020). What makes followers loyal? the role of influencer interactivity in building influencer brand equity. *Journal of Product & Brand Management*, 29(6):803–814.
- Martins, M. S. (2023). Estudo e identificação de buscas e análises sobre dados em redes sociais digitais. *Anais do XXXI Seminário de Iniciação Científica e Tecnológica da PUC-Rio*.
- Martins, M. S. (2024). Implementação de buscas e análises específicas das redes sociais digitais. *Anais do XXXII Seminário de Iniciação Científica e Tecnológica da PUC-Rio*.
- Salgueiro, M. (2023). Uma abordagem para uma especificação conceitual de redes sociais digitais. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro.
- Silva, I. O. d., Gouveia, F. C., et al. (2021). Engajamento informacional nas redes sociais: como calcular? *AtoZ: Novas Práticas em Informação e Conhecimento*.