

Primeiro Exercício-Programa

Prof. Luciano Antonio Digiampietri

Prazo máximo para a entrega: 12/05/2024

1 Predição de Relacionamentos em Redes Sociais

Diferentes estruturas ou sistemas do mundo real podem ser representados utilizando-se grafos.

Neste EP trabalharemos com Redes Sociais representadas via grafos não direcionados e não ponderados, em que vértices representam pessoas e arestas representam a existência de um relacionamento (por exemplo, de amizade) entre pares de pessoas.

Um dos aspectos mais estudados na Análise de Redes Sociais em relação à dinâmica das redes é a predição de relacionamentos (*Link Prediction*). A predição de relacionamentos visa a identificar/sugerir relacionamentos que não estão presentes na rede social.

Do ponto de vista da antropologia e sociologia, alguns aspectos sociais relevantes para a criação de relacionamentos são:¹

1. Homofilia: quanto mais interesses em comum as pessoas possuem, como características ou preferências, maior as chances de se relacionarem.
2. Raridade: há mais chances de relacionamento entre pessoas com características ou preferências raras em comum. Pois características difíceis de encontrar tendem a se destacar em relação às outras características.
3. Influência Social: uma característica compartilhada com muitos amigos de uma determinada pessoa pode ser útil para encontrar potenciais relacionamentos.
4. Amizades em Comum (ou vizinhos em comum): quanto mais amigos em comum duas pessoas possuem, maiores são as chances de se relacionarem.
5. Proximidade Social: pessoas localizadas próximas em um grafo social possuem um relacionamento em potencial.
6. Conexão Preferencial: pessoas populares tendem a atrair mais pessoas quando comparadas às pessoas com poucos relacionamentos.

Para este EP focaremos nos aspectos puramente estruturais da predição de relacionamentos, em que, com base na estrutura da rede (ou seja, no grafo) calcularemos diferentes

¹Yin et al. (2010), Maruyama (2016)

métricas comumente utilizadas pelos sistemas de predição de relacionamentos (ou de sugestão de relacionamentos) em redes sociais.

A ideia aqui é desenvolver funções que calculam, para um dado indivíduo (vértice) diferentes métricas que tentam mensurar a potencial relação com os demais indivíduos da rede. Cada uma dessas funções receberá o endereço de um grafo não direcionado e não ponderado (utilizando a representação de matriz de adjacências vista em aula), a indicação de um vértice v e o endereço de um arranjo que receberá a respectiva métrica calculada para cada um dos indivíduos da rede em relação ao vértice v (suas funções deverão preencher os valores desse arranjo).

Detalhamento: Para este EP, utilizaremos a mesma estrutura de dados usada nas aulas para representar grafos não direcionados e não ponderados com matrizes de adjacências.

O código fornecido para este EP já possui várias funções implementadas de gerenciamento básico de grafos, bem como para realizar alguns testes no EP.

Você deverá implementar/completar oito funções, podendo, se julgar conveniente, implementar funções adicionais/auxiliares.

Todas as funções recebem três parâmetros: o ponteiro/endereço para o grafo g , o vértice alvo v (para o qual serão calculadas as métricas em relação a todos os vértices do grafo) e o endereço de um arranjo cujo tamanho é igual ao número de vértices do grafo (e o qual deve ser preenchido por sua função). Para este EP, você pode considerar que g sempre terá o endereço de um grafo válido e que v sempre corresponderá a um vértice do grafo g .

Considere, para as fórmulas a seguir, que: $\Gamma(x)$ corresponde ao conjunto de vizinhos (ou adjacentes) de x ; $|\Gamma(x)|$ corresponde ao número de vizinhos de x ; $\Gamma(x) \cap \Gamma(y)$ corresponde ao conjunto de elementos da intersecção entre os vizinhos de x e vizinhos de y ; $|\Gamma(x) \cap \Gamma(y)|$ corresponde ao número de elementos da intersecção entre os vizinhos de x e vizinhos de y ; $\Gamma(x) \cup \Gamma(y)$ corresponde ao conjunto formado pela união dos vizinhos de x com os vizinhos de y e $|\Gamma(x) \cup \Gamma(y)|$ corresponde ao número de elementos da união entre os vizinhos de x e vizinhos de y .

- *void vizinhosEmComum(Grafo* g, int v, int* vizinhos)*: função que calcula o número de vizinhos em comum (ou adjacentes em comum) do vértice v com cada um dos vértices do grafo. Isto é, na posição 0 (zero) do arranjo *vizinhos* (isto é, *vizinhos[0]*) a função deverá atribuir o número de vizinhos em comum entre o vértice v e o vértice 0. Especificamente na posição *vizinhos[v]* deverá atribuir o número de vizinhos de v (isto é, todos os vizinhos de v são vizinhos em comum com ele mesmo).

Vizinhos em Comum (*Common Neighbors - CN*):

$$CN_{x,y} = |\Gamma(x) \cap \Gamma(y)|$$

- *void coeficienteDeJaccard(Grafo* g, int v, float* coeficientes)*: função que calcula o Coeficiente de Jaccard do vértice v em relação a cada um dos vértices do grafo. Caso o denominador da fórmula seja igual a 0 (zero), o valor -1 deverá ser atribuído na respectiva posição do arranjo.

Coeficiente de Jaccard (*Jaccard Coefficient - JC*):

$$JC_{x,y} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

- *void AdamicAdar(Grafo* g, int v, float* coeficientes)*: função que calcula a métrica Adamic Adar do vértice v em relação a cada um dos vértices do grafo. Caso o denominador da fórmula seja igual a 0 (zero), isto é, se o grau de z for igual a 1 para qualquer z vizinho em comum de x e y, o valor -1 deverá ser atribuído na respectiva posição do arranjo (posição y do arranjo)².

Adamic-Adar (*AA*):

$$AA_{x,y} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log(|\Gamma(z)|)}$$

- *void alocaoDeRecursos(Grafo* g, int v, float* coeficientes)*: função que calcula a medida Alocação de Recursos do vértice v em relação a cada um dos vértices do grafo.

Alocação de Recursos (*Resource Allocation - RA*):

$$RA_{x,y} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|}$$

- *void similaridadeCosseno(Grafo* g, int v, float* coeficientes)*: função que calcula a Similaridade Cosseno do vértice v em relação a cada um dos vértices do grafo. Caso o denominador da fórmula seja igual a 0 (zero), o valor -1 deverá ser atribuído na respectiva posição do arranjo.

Similaridade Cosseno (*SA*):

$$SA_{x,y} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)| * |\Gamma(y)|}}$$

²Isto pode acontecer quando $x = y$ e houver um vizinho de x que não é vizinho de mais ninguém.

- *void coeficienteDeDice(Grafo* g, int v, float* coeficientes)*: função que calcula o Coeficiente de Dice do vértice v em relação a cada um dos vértices do grafo. Caso o denominador da fórmula seja igual a 0 (zero), o valor -1 deverá ser atribuído na respectiva posição do arranjo.

Coeficiente de Dice (*Dice Coefficient - SD*):

$$SD_{x,y} = \frac{2 * |\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| + |\Gamma(y)|}$$

- *void HPI(Grafo* g, int v, float* coeficientes)*: função que calcula a medida HPI do vértice v em relação a cada um dos vértices do grafo. Caso o denominador da fórmula seja igual a 0 (zero), o valor -1 deverá ser atribuído na respectiva posição do arranjo.

Hub Promoted Index - HPI:

$$HPI_{x,y} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\min(|\Gamma(x)|, |\Gamma(y)|)}$$

- *void HDI(Grafo* g, int v, float* coeficientes)*: função que calcula a medida HPI do vértice v em relação a cada um dos vértices do grafo. Caso o denominador da fórmula seja igual a 0 (zero), o valor -1 deverá ser atribuído na respectiva posição do arranjo.

Hub Depressed Index - HPI:

$$HDI_{x,y} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\max(|\Gamma(x)|, |\Gamma(y)|)}$$

Observações: as funções raiz quadrada (*sqrt*) e logaritmo (*log*) a serem utilizadas neste EP são aquelas presentes na biblioteca *math* (já incluída no código do EP). Se, durante a compilação, você receber uma mensagem de erro informando que essas funções não foram declaradas, tente compilar adicionando o parâmetro *-lm*. Os arranjos que deverão ser preenchidos por suas funções podem **não estar zerados**, assim, certifique-se de preencher todas as posições do arranjo com os valores corretos.

Para saber um pouco mais sobre Predição de Relacionamentos em Redes Sociais, você pode acessar os seguintes slides: http://www.each.usp.br/digiampietri/garsc/disciplina/07_PredicaoDeRelacionamentos.pdf ou a seguinte videoaula: <https://youtu.be/vS50okgSfko?si=DQfi757E5ZVQaX2i>.

Exemplo: a seguir são apresentadas as soluções para um grafo específico.

Imprimindo grafo (vertices: 5; arestas: 5).

	0	1	2	3	4
0	0	1	1	0	0
1	1	0	0	1	1
2	1	0	0	1	0
3	0	1	1	0	0
4	0	1	0	0	0

Vizinhos em Comum de v0:

v0	v1	v2	v3	v4
2	0	0	2	1

Coefficientes de Jaccard de v0:

v0	v1	v2	v3	v4
1.00	0.00	0.00	1.00	0.50

Medida de Adamic-Adar de v0:

v0	v1	v2	v3	v4
2.35	0.00	0.00	2.35	0.91

Medida de Alocação de Recursos de v0:

v0	v1	v2	v3	v4
0.83	0.00	0.00	0.83	0.33

Similaridade Cosseno de v0:

v0	v1	v2	v3	v4
1.00	0.00	0.00	1.00	0.71

Coefficiente de Dice de v0:

v0	v1	v2	v3	v4
1.00	0.00	0.00	1.00	0.67

Índice HPI de v0:

v0	v1	v2	v3	v4
1.00	0.00	0.00	1.00	1.00

Índice HDI de v0:

v0	v1	v2	v3	v4
1.00	0.00	0.00	1.00	0.50

1.1 Material a Ser Entregue

Um arquivo, denominado *NUSP.c* (sendo NUSP o seu número USP, por exemplo: 123456789.c), contendo seu código, incluindo todas as funções solicitadas e qualquer outra função adicional que ache necessário. Para sua conveniência, *completeERenomeie.c* será fornecido, cabendo a você então completá-lo e renomeá-lo para a submissão.

Atenção!

1. Não modifique as assinaturas das funções já implementadas e/ou que você deverá completar!
2. Para avaliação, as diferentes funções serão invocadas diretamente (individualmente ou em conjunto com outras funções). Em especial, qualquer código dentro da função `main()` será ignorado.

2 Entrega

A entrega será feita única e exclusivamente via sistema e-Disciplinas, até a data final marcada. Deverá ser postado no sistema um arquivo .c, tendo como nome seu número USP:

`seuNumeroUSP.c` (por exemplo, `12345678.c`)

Não esqueça de preencher o cabeçalho constante do arquivo .c, com seu nome, número USP, turma etc.

A responsabilidade de postagem é exclusivamente sua. Por isso, submeta e certifique-se de que o arquivo submetido é o correto (fazendo seu download, por exemplo). Problemas referentes ao uso do sistema devem ser resolvidos com antecedência.

3 Avaliação

A nota atribuída ao EP será baseada nas funcionalidades solicitadas, porém não esqueça de se atentar aos seguintes aspectos:

1. Documentação: se há comentários explicando o que se faz nos passos mais importantes e para que serve o programa (tanto a função quanto o programa em que está inserida);
2. Apresentação visual: se o código está legível, indentado etc;
3. Corretude: se o programa funciona.

Além disso, algumas observações pertinentes ao trabalho, que influenciam em sua nota, são:

- Este exercício-programa deve ser elaborado individualmente;
- Não será tolerado plágio;
- Exercícios com erro de sintaxe (ou seja, erros de compilação), receberão nota ZERO.

Referências

- 1 MARUYAMA, William Takahiro. Predição de coautorias em redes sociais acadêmicas. 2016. Dissertação (Mestrado em Sistemas de Informação) - EACH-USP, São Paulo, 2016. doi:10.11606/D.100.2016.tde-07052016-232625.
- 2 YIN, Z. et al. A unified framework for link recommendation using random walks. In: Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on., 2010. p. 152–159.