GRASP para Maximização da Modularidade por Densidade com Sinais

Letícia do Nascimento, Rafael de Santiago, Álvaro Junio Pereira Franco, Pedro Belin Castellucci

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – CEP 88040-370 – Florianópolis – SC – Brazil

leticia.nascimento@grad.ufsc.br

{r.santiago,alvaro.junio,pedro.castellucci}@ufsc.br

Abstract. *We focus on a clustering problem in graphs based on maximizing modularity, known as Modularity Density Maximization. In the investigated variation, both positive and negative edges are considered to define node clusters. In this work, we propose a version of the metaheuristic Greedy Randomized Adaptive Search Procedure. The method was able to achieve the best-known objective values in the state-of-the-art.*

Resumo. *Este trabalho lida com uma versão de problema de agrupamento em grafos baseada na maximização da modularidade, chamada de problema da Maximização da Modularidade por Densidade. Na variação tratada, são consideradas arestas positivas e negativas para agrupar vértices. Para este problema, é proposto o uso da metaheurística Greedy Randomized Adaptive Search Procedure. O método proposto atinge os melhores valores objetivo conhecidos no estado da arte.*

1. Introdução

Encontrar grupos bem relacionados de entidades na vida real é um problema útil em diversos domínios, e o interesse nesses estudos vem crescendo nos últimos anos. O problema pode ser aplicado na medicina, para identificar a memória funcional envolvida no reconhecimento olfativo [Meunier et al. 2014], na astronomia, para identificação de grupos de estrelas [Schmeja 2011], na biologia, para encontrar complexos de proteínas [Nepusz et al. 2012], na identificação de comunidades em redes de transporte público [Guimerà et al. 2005], entre muitas outras aplicações.

Esses grupos são denominados na literatura de *comunidades* e são geralmente formalizados em grafos, nos quais cada vértice representa uma entidade e as relações entre elas são definidas por arcos ou arestas [Newman and Girvan 2004, Leskovec et al. 2010]. Cada relação entre entidades pode incorporar um valor numérico positivo (atração) ou negativo (repulsa). Um exemplo de aplicação dessas relações com sinal (positivo ou negativo) seria no estudo de mídias sociais, no qual elas podem representar uma relação positiva (de amizade) ou negativa (de antagonismo) entre os usuários. A atitude de um usuário em relação a outro pode ser estimada a partir de evidências fornecidas por seus relacionamentos com outros membros da rede social [Leskovec et al. 2010].

Visando encontrar comunidades em grafos, Mark Newman e Michelle Girvan definiram um problema de otimização de maximização da modularidade, cujo objetivo é

medir a diferença entre o número de arestas internas e o número esperado de arestas internas dentro de cada comunidade [Newman and Girvan 2004]. Por tratar-se de um problema NP-Difícil [Brandes 2008] e por diversas vezes as aplicações serem caracterizadas por instâncias de tamanho significativo (muitos vértices e arestas), o problema da Maximização da Modularidade é frequentemente resolvido por métodos heurísticos na literatura [Clauset et al. 2004, Blondel et al. 2008]. No entanto, a modularidade apresenta um problema chamado de limite de resolução [Fortunato and Barthélemy 2007], no qual o reconhecimento de comunidades em redes cuja quantidade total de vértices está abaixo de um número mínimo de arestas esperadas não ocorre.

Com a motivação principal sendo evitar o limite de resolução, uma reformulação para o problema da Maximização da Modularidade foi desenvolvida [Li et al. 2008], chamada de Maximização da Modularidade por Densidade. A reformulação utiliza a diferença entre a densidade interna e externa de arestas para cada comunidade, avaliando assim as soluções do problema. Sua variação para lidar com grafos com sinais foi proposta na literatura [Li et al. 2014, de Santiago and Lamb 2020]. Nesse grafos, pesos positivos nas arestas representam atração entre vértices (maior chance de estarem na mesma comunidade) e negativos representam repulsa entre dois vértices (menos chance de compartilharem a mesma comunidade). Alguns métodos exatos e heurísticos foram propostos em [Li et al. 2014].

Este trabalho propõe um *Greedy Randomized Adaptive Search Procedure* (GRASP) [Feo and Resende 1995] para o problema da Maximização da Modularidade por Densidade com Sinais a partir do estado atual da arte. Este artigo apresenta na sequência, a definição do problema; depois, descreve o método proposto, destacando os resultados obtidos logo em seguida. Por fim, são apresentadas as considerações finais e os potenciais trabalhos futuros.

2. Definição do Problema

Dado um grafo G e uma comunidade C, considere $G = (V, E^+, E^-)$, onde V é o conjunto de vértices, E^+ é o conjunto de arestas com peso de valor numérico positivo (arestas positivas), e E^- é o conjunto de arestas com peso de valor numérico negativo (arestas negativas). Considere que $w^+_{uv} = 1$ se $\{u, v\} \in E^+$, caso contrário $w^+_{uv} = 0$, e $w^-_{uv} = 1$ se $\{u, v\} \in E^-$, caso contrário $w^-_{uv} = 0$. Considere também um conjunto C uma partição dos vértices de G, na qual, cada subconjunto $c \in C$ é chamado de comunidade. Os conjuntos E^+_c e E^-_c são compostos pelas arestas positivas e negativas, respectivamente, de uma comunidade $c \in C$. Portanto, $|E^+_c|$ representa o número de arestas positivas em uma determinada comunidade c e $|E^-_c|$ é o número de arestas negativas em uma determinada comunidade c (que conectam vértices da mesma comunidade c). Dividimos o grau de cada vértice entre positivos e negativos. Logo, o grau positivo de v é dado por $d^+_v = \sum_{u \in V} w^+_{uv}$ (soma dos graus positivos), e o grau negativo é dado por $d^-_v = \sum_{u \in V} w^-_{uv}$ (soma dos graus negativos). O número de vértices na comunidade c é dado por $|c|$. Em seguida, é possível ver a função objetivo $D_\lambda(C)$ para um conjunto de comunidades C com um parâmetro de ajuste λ.

$$\max D_\lambda(C) =$$
$$\sum_{c \in C} \left(\frac{2\lambda|E_c^+| - 2(1-\lambda)(\sum_{v \in c} d_v^+ - |E_c^+|) - 2(1-\lambda)|E_c^-| + 2\lambda(\sum_{v \in c} d_v^- - |E_c^-|)}{|c|} \right).$$

O parâmetro $\lambda \in [0;1]$ é usado para calibrar a função objetivo na busca de uma estrutura de comunidades em acordo a uma determinada aplicação [Li et al. 2008]. Para obter a chamada "associação de proporção" e encontrar comunidades com poucos vértices, deve-se utilizar $\lambda > 0{,}5$. Para obter o "recorte de proporção" e encontrar comunidades com mais vértices, utilizar $\lambda < 0{,}5$.

3. Método Proposto

Um pseudocódigo para o GRASP está no Algoritmo 1. Inicialmente, o algoritmo define uma variável S^* que armazenará a melhor solução encontrada na busca. Depois, o algoritmo gera m soluções (uma para cada repetição no laço que compreende as linhas 3 e 9). Para cada solução, seleciona-se iterativamente um elemento da lista aleatória de vértices e os coloca na melhor solução possível com a função *ConstruirSolução*. Quando a solução estiver completa, uma busca local é aplicada à solução recém criada (linha 5, *BuscaLocal*) que obterá a ótima local S'. Ao final, verifica-se a solução encontrada durante a i-ésima iteração, realizando uma comparação com a solução de referência S^*, que é substituída por S' caso essa última seja melhor que a de referência (linhas 6 e 7). Ao final, a busca devolve S^* (linha 10).

Algoritmo 1 GRASP proposto

```
1: Entrada: um grafo não-dirigido e ponderado G = (V, E, w)
2: S* = {}
3: for all i ∈ {1, 2 ..., m} do
4:     S = ConstruirSolução(G)
5:     S' = BuscaLocal(G, S)
6:     if D_λ(S') melhor que D_λ(S*) then
7:         S* = S'
8:     end if
9: end for
10: return S*
```

A função *ConstruirSolução* inicia definindo uma sequência aleatória dos vértices. Para cada vértice, utiliza-se a função objetivo do problema para identificar qual o maior ganho: entrar em uma das comunidades pré-existentes ou criar uma nova. Depois, adiciona-se o vértice à opção de maior ganho. Repete-se o procedimento até que todos os vértices tenham sido atribuídos a uma comunidade. A função *BuscaLocal* é um método de busca local, que recebe uma solução inicial e, iterativamente, busca a melhor solução vizinha. Se ela for melhor que a solução atual, a mesma é substituída. Se não for, o método para e devolve a melhor solução encontrada. As soluções vizinhas são obtidas a partir da geração de soluções por modificar a solução de referência, alterando um vértice para outra comunidade existente ou uma nova.

4. Resultados

Para a realização dos experimentos, foram utilizadas instâncias reais de grafos, sendo elas ***Slovene Parliamentary Party*** e ***Gahuku-Gama Subtribes***, que serão referenciadas nesta seção como $Parlamento$ e $Gahuku$, respectivamente. A rede $Parlamento$ representa a relação entre dez partidos políticos do Parlamento Esloveno em 1994, possui 10 vértices e 45 arestas [Kropivnik and Mrvar 1996]. A rede $Gahuku$ representa a relação das tribos da Nova Guiné, possui 16 vértices e 120 arestas [Read 1964]. Três conjuntos de configurações de experimento foram definidas para aplicar a heurística, onde cada configuração é dada por um valor de λ e um valor de m (que define a quantidade de soluções que serão criadas no Algoritmo 1). Cada configuração do experimento foi repetida uma quantidade de 30 vezes, com os parâmetros: m sendo definido como 20, 30 e 40; e o valor de λ variando no intervalo $[0{,}2; 0{,}9]$ conforme a Tabela 1.

A Tabela 1 apresenta as médias de densidade obtidas por configuração, para cada rede estudada. Destacado em vermelho estão as médias de densidade mais distantes do valor ótimo. Em laranja, as médias são valores um pouco mais próximos. Em azul as médias obtidas foram ainda mais próximas e em verde os valores ideais foram alcançados.

Tabela 1. Valores médios de densidade obtidos aplicando a heurística para cada instância e parâmetro. Os valores ótimos conhecidos de cada instância e λ estão identificados com *, obtidos em [de Santiago and Lamb 2020].

λ	m	Gahuku	Parlamento	λ	m	Gahuku	Parlamento
0,2		**7,445***	**5,800***	**0,6**		**36,520***	**36,000***
	20	0,511	5,335		**20**	**36,520**	**36,000**
	30	1,047	5,793		**30**	**36,520**	**36,000**
	40	2,774	5,787		**40**	**36,520**	**36,000**
0,3		**11,854***	**11,200***	**0,7**		**55,749***	**53,999***
	20	9,120	**11,200**		**20**	**55,749**	**53,999**
	30	8,668	**11,200**		**30**	**55,749**	**53,999**
	40	9,815	**11,200**		**40**	**55,749**	**53.999**
0,4		**17,076***	**16,600***	**0,8**		**75,500***	**72,000***
	20	16,480	**16,600**		**20**	**75,500**	**72,000**
	30	16,690	**16,600**		**30**	**75,500**	**72,000**
	40	16,588	**16,600**		**40**	**75,500**	**72,000**
0,5		**24,238***	**23,000***	**0,9**		**95,466***	**89,999***
	20	**24,238**	**23,000**		**20**	**95,466**	**89,999**
	30	**24,238**	**23,000**		**30**	**95,466**	**89,999**
	40	**24,238**	**23,000**		**40**	**95,466**	**89,999**

5. Conclusões

A aplicação da heurística para duas instâncias reais alcançou os valores objetivo próximos ao ótimo. A partir de um λ de $0{,}5$, os valores obtidos foram melhores. Como trabalhos futuros, sugere-se experimentar redes maiores e outros critérios gulosos para o GRASP, para investigar sua escalabilidade.

Referências

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Brandes, U. (2008). On Modularity Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188.

Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, 70(6):066111.

de Santiago, R. and Lamb, L. C. (2020). Exact signed modularity density maximization solutions and their real meaning. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–7.

Feo, T. A. and Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133.

Fortunato, S. and Barthélemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, 104(1):36–41.

Guimerà, R., Mossa, S., Turtschi, A., and Amaral, L. A. N. (2005). The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences*, 102(22):7794–7799.

Kropivnik, S. and Mrvar, A. (1996). An analysis of the slovene parliamentary parties network. *Developments in Statistics and Methodology, Metodološki zvezki*, 12:209–216.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Signed networks in social media. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, page 1361, New York, New York, USA. ACM Press.

Li, Y., Liu, J., and Liu, C. (2014). A comparative analysis of evolutionary and memetic algorithms for community detection from signed social networks. *Soft Computing*, 18(2):329–348.

Li, Z., Zhang, S., Wang, R.-S., Zhang, X.-S., and Chen, L. (2008). Quantitative function for community detection. *Physical Review E*, 77(3):036109.

Meunier, D., Fonlupt, P., Saive, A.-L., Plailly, J., Ravel, N., and Royet, J.-P. (2014). Modular structure of functional networks in olfactory memory. *NeuroImage*, pages 264–75.

Nepusz, T., Yu, H., and Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods*, 9(5):471–472.

Newman, M. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113.

Read, K. E. (1964). Cultures of the central highlands, new guinea. *Southwestern Journal of Anthropology*, 10.

Schmeja, S. (2011). Identifying star clusters in a field: a comparison of different algorithms. *Astronomische Nachrichten*, 332(2):172–184.