Universidade de São Paulo – São Carlos
Instituto de Ciências Matemáticas e de Computação
Redes Complexas para a Ciência da Computação

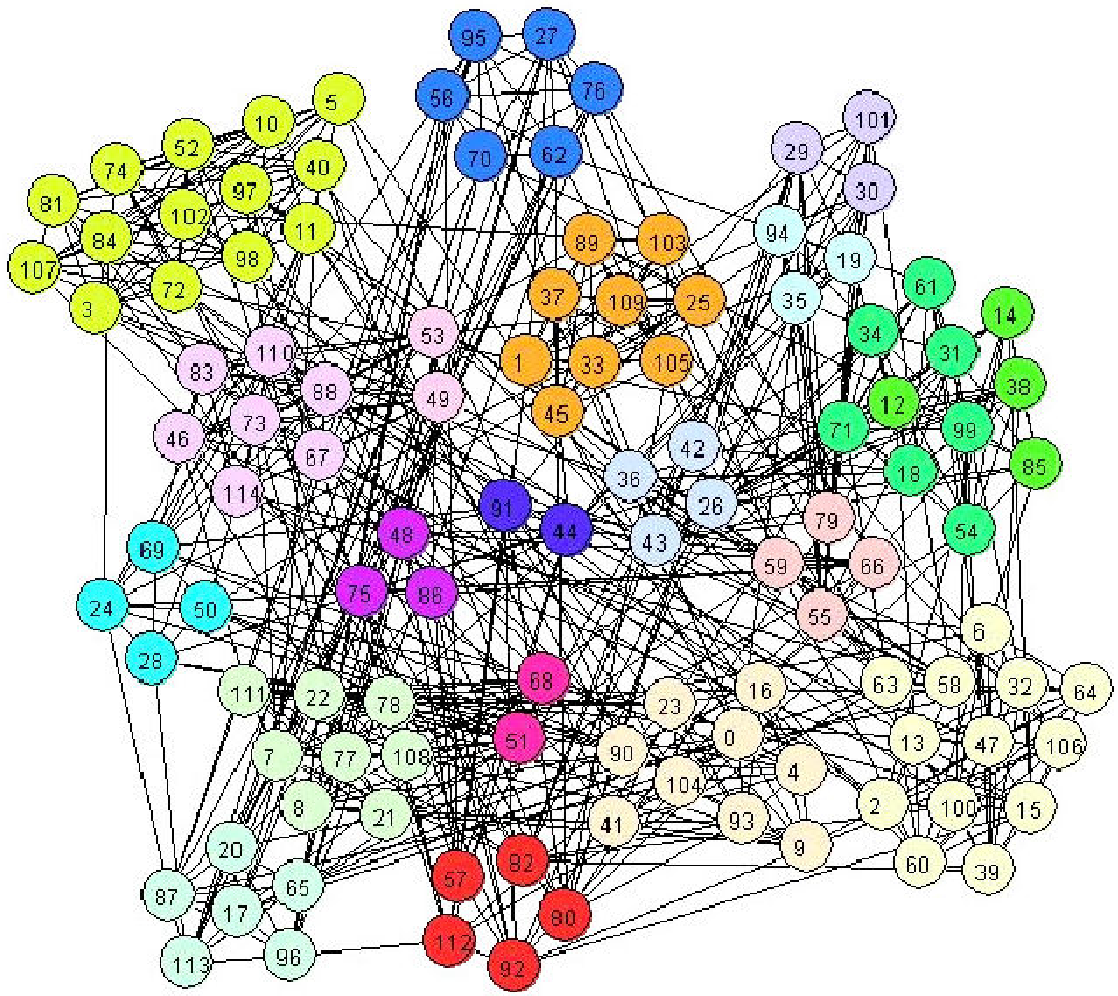
Detecção de Comunidades

Glenda Michele Botelho
Fabiano Berardo de Sousa

Roteiro

- Introdução
- *Clustering*
- Abordagens para Detecção de Comunidades
  - Métodos Divisivos
  - Métodos Aglomerativos
  - Métodos Espectrais
  - Métodos Locais
  - Otimizações da Modularidade
- Conclusões

Introdução



Estrutura de Comunidades em Redes Complexas
[Newman and Girvan 2004]

Introdução

- Detecção de Comunidades
  - Aprendizado de Máquina
    - Aprendizado Não-Supervisionado.
    - Técnicas de *Clustering*.

- Relevância
  - Redes Complexas modelam sistemas reais
    - Extração de características específicas.
    - Estudo da organização e evolução dinâmica da rede.

Clustering

- Algoritmos Particionais
- Algoritmos Baseados em Densidades
- Algoritmos Hierárquicos

[Jain and Dubes 1988]

Clustering

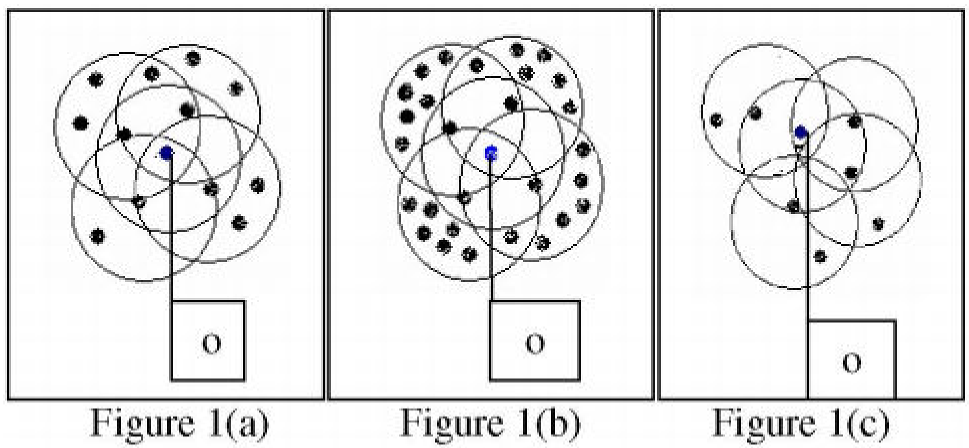
- Algoritmos Particionais
  - Obtenção de subgrafos (comunidades)
  - *K-Means* (e otimizações)
  - Não adequados!
    - Deve-se saber, a priori, o número de comunidades existentes na rede.

Clustering

- Algoritmos Baseados em Densidade
  - Regiões de alta densidade de objetos separadas por regiões de baixa densidade.
  - DBSCAN
  - *Chameleon*

Clustering

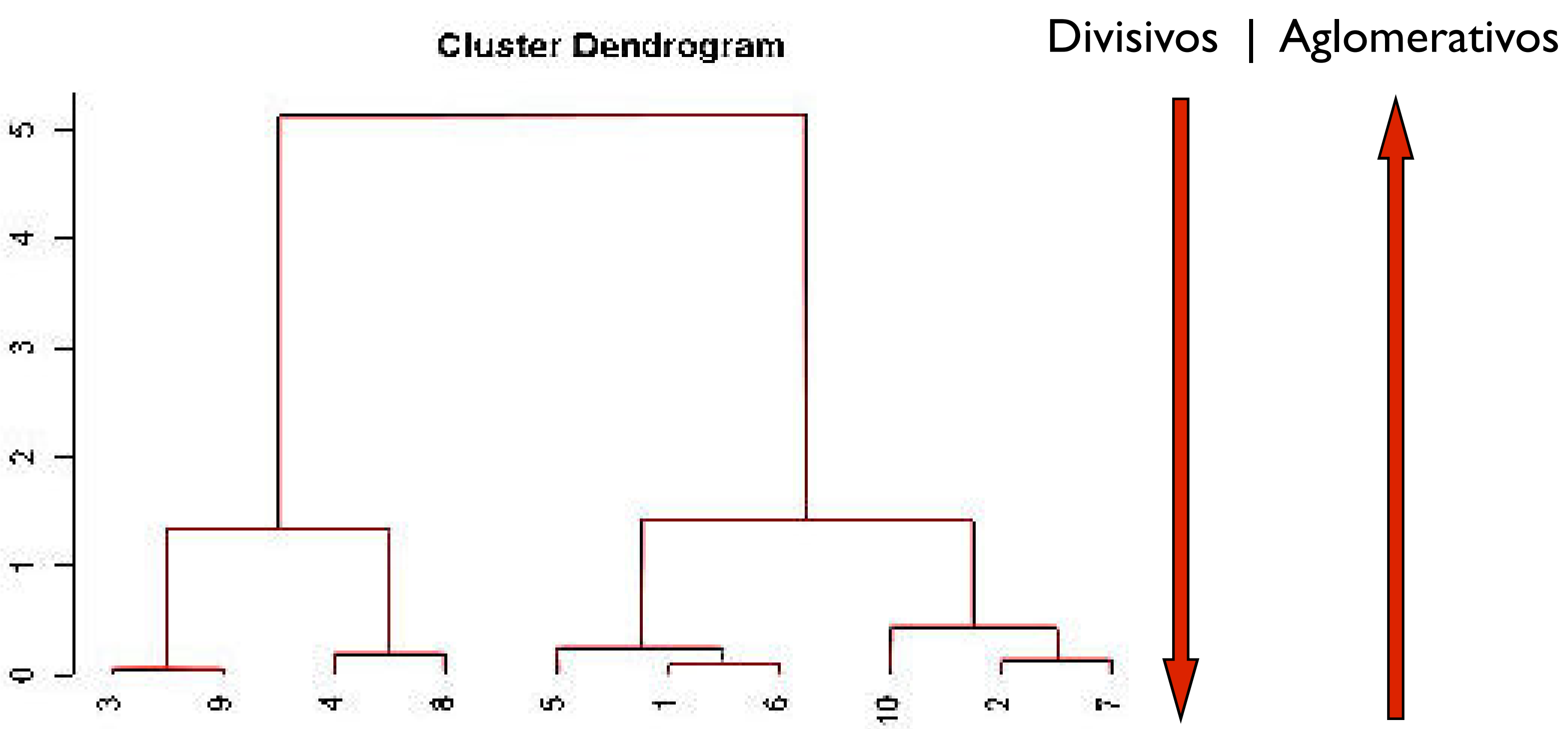
- Algoritmos Baseados em Densidade
  - DBSCAN (Density-Based Spatial Clustering of Application with Noise)



    - Baseado no conceito de *objetos densamente alcançáveis*.
    - Ineficiente quando há variação de densidade.
    - Pior caso: $O(n^2)$, para *n* vértices.

Clustering

- Algoritmos Hierárquicos
  - Baseados em *Similaridade*
  - *CURE, ROCK*



Detecção de Comunidades

- Métodos Divisivos
  - Centralidade *Betweenness*
  - Coeficiente de Agrupamento de Arestas

Detecção de Comunidades

- Métodos Divisivos
  - Centralidade *Betweenness* [Girvan and Newman 2002]

$$B_u = \sum_{ij} \frac{\sigma(i,u,j)}{\sigma(i,j)}$$

- $\sigma(i, u, j)$ = números de caminhos mínimos entre os vértices i e j e que passam pela aresta u.
- $\sigma(i, j)$ = número total de caminhos mínimos entre os vértices i e j.

- Arestas com alto valor da medida são removidas iterativamente.
- Pior caso: $O(m^2n)$, m arestas, n vértices.

Detecção de Comunidades

- Métodos Aglomerativos
  - Medida de Similaridade
  - Medida de Modularidade

Detecção de Comunidades

- Métodos Aglomerativos
  - Medida de Modularidade [Newman 2004]

$$Q = \sum_i (e_{ii} - a_i^2)$$

- e_{ii} = fração das arestas inseridas no grupo *i*.
- a_i^2 = fração das arestas inseridas aleatoriamente no grupo *i*.

- Diferença entre o real e o aleatório.
- $Q \geq 0{,}3$ indica resultado significativo.

Detecção de Comunidades

- Métodos Aglomerativos
  - Maximização da Modularidade [Newman 2004]

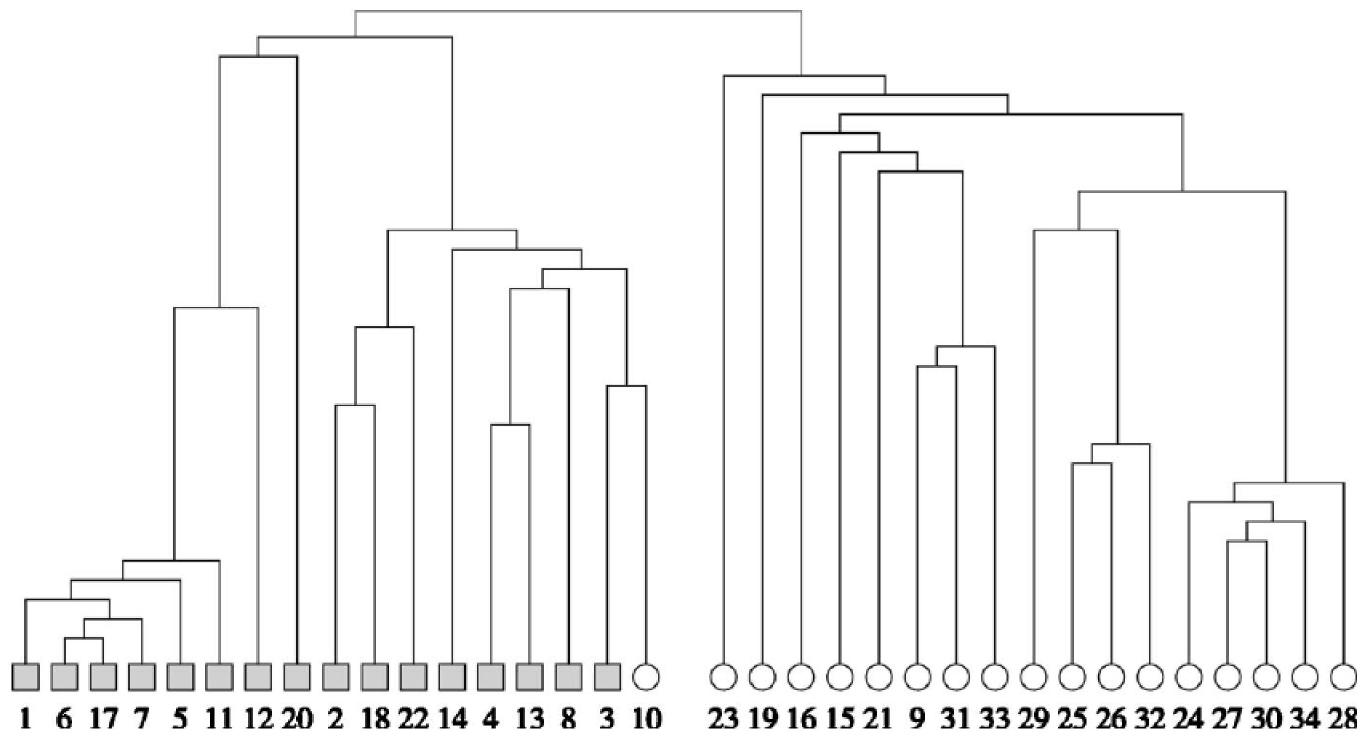
$$\Delta Q = 2(e_{ij} - a_i a_j)$$

- e_{ij} = fração das arestas que conectam o grupo *i* ao grupo *j*.
- a_i = fração das arestas que conectam o grupo *i* aos demais grupos da rede. (o mesmo para a_j).

- Algoritmo de otimização <u>guloso</u>!
- Pior caso: $O((m+n)n)$, *m* arestas, *n* vértices.

Detecção de Comunidades

- Métodos Aglomerativos
  - Medida de Modularidade - Maximização



Detecção de Comunidades

- Métodos Espectrais
  - Análise dos autovetores das matrizes derivadas das redes.
  - Matriz de modularidade $B^{(g)}$ [Newman 2006].
  - Elementos de $B^{(g)}$, para cada subgrafo *g*, dados por:

$$b_{ij}^{(g)} = a_{ij} - \frac{k_i k_j}{2M} - \delta_{ij} \sum_{u \in g} [a_{iu} - \frac{k_i k_u}{2M}]$$

    - a_{ij} corresponde ao número de arestas entre os vértices *i* e *j*.
    - $\delta_{ij} = v_i^T v_j$
  - Encontra-se autovalor mais positivo de $B^{(g)}$ com seu correspondente autovetor.
  - Sinais do elemento do vetor dividem a rede em 2 partes.
  - Processo repetido para cada comunidade.
    - Divisão da rede onde modularidade total é 0 ou negativa.

Detecção de Comunidades

- Métodos Locais
  - Comunidades conectadas localmente.
  - Modularidade local [Muff et al. 2005] → modularidade para cada comunidade *i* é calculada para uma subrede (comunidade i e suas comunidades vizinhas).

$$LQ = \sum_{i=1}^{K} [\frac{L_i}{L_{iC}} - \frac{(L_i)_{in}(L_i)_{out}}{(L_{iC})^2}]$$

    - L_i = total de arestas da comunidade *i*.
    - L_{iC} = total de arestas contidas em cada comunidade $C \in \Delta_i$ (conjunto de comunidades vizinhas de *i*).
  - Quanto mais comunidades localmente conectadas maior será *LQ* (não é limitado por 1).
  - Função de *fitness* em processo de otimização.

Detecção de Comunidades

- Maximização da Modularidade
  - Otimização extrema [Duch and Arenas 2005]
    - Otimização da variável global (modularidade) através do melhoramento de variáveis locais (relacionadas com a contribuição do vértice *i* para o somatório *Q*, dada uma divisão em *c* comunidades).

$$q_i = K_{c(i)} - k_i a_{c(i)}$$

    - $K_{c(i)}$ = n° arestas que o vértice $i \in c$ tem com vértices na mesma comunidade.
    - $a_{c(i)}$ = fração das arestas que possui pelo menos o vértice *i* dentro da comunidade *c*.

Detecção de Comunidades

- Maximização da modularidade
  - Otimização extrema

$$Q = \frac{1}{2M}\sum_i q_i$$

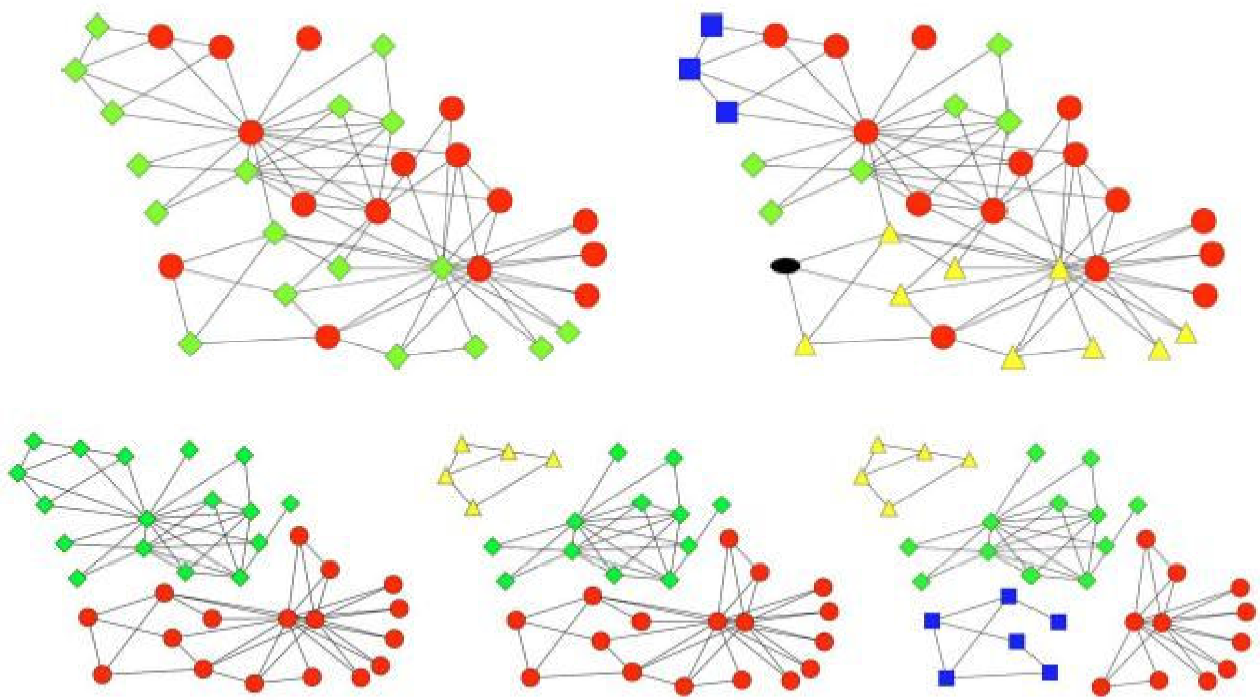
    - Redimensionado q_i através do grau do vértice k_i obtemos o *fitness* de *i*.

$$\lambda_i = \frac{q_i}{k_i} = \frac{K_{c(i)}}{k_i} - a_{c(i)}$$

    - Processo de busca:
      - Dividir rede em duas partições com mesmo n° de vértices.
      - A cada passo, movimenta-se o vértice com menor *fitness* de uma partição para outra até obter maior *Q*.
      - Deleta todas as arestas entre ambas as partições
      - Procede-se com todos os componentes conectados resultantes.
      - Até *Q* não puder ser melhorado.

Detecção de Comunidades

- Maximização da modularidade
  - Otimização extrema



Q=0.3718 Q=0.4020 Q=0.4188

Detecção de Comunidades

- Maximização da modularidade
  - Otimização extrema
    - Vértice selecionado $\rightarrow$ pior *fitness*
    - Seleção probabilística (τ-EO):
      - Ordena vértices baseado no *fitness*.
      - Seleciona vértice *i*: $P(i) \propto i^{-\tau}$ com $\tau \sim 1 + \frac{1}{\ln(N)}$
      - $\alpha N \rightarrow$ n° de passos de auto-organização para decidir que *Qmax* tem poucas chances de ser melhorado ($\alpha = 1$).
      - $O(N^2 \ln^2 N)$ ($N \ln N$ = custo de processo de ordenação)
      - *Heap* $\rightarrow$ O(*N*) $\rightarrow$ $O(N^2 \ln N)$
      - Resultados melhores que os obtidos pela otimização da modularidade proposta por Newman.

Detecção de Comunidades

- Monte Carlo com *Simulated Annealing* e *Basin Hopping* [Massen and Doye 2005]
  - Monte Carlo com *Simulated Annealing*
    - A cada passo, um vértice e uma comunidade são selecionados aleatoriamente.
    - Move-se o vértice da comunidade inicial para nova comunidade ($Q \rightarrow \Delta Q$).
      - $\Delta Q>0$ $\rightarrow$ movimento aceito.
      - $\Delta Q \leq 0$ $\rightarrow$ movimento aceito com probabilidade : $\exp(\beta \Delta Q)$
        - Critério Metropolis $\rightarrow$ β temperatura inversa.
      - Altas temperaturas: muitos movimentos aceitos e muitas divisões diferentes de comunidades
      - Baixas temperaturas: poucas divisões são experimentadas.

Detecção de Comunidades

- Monte Carlo com *Simulated Annealing* e *Basin Hopping* [Massen and Doye 2005]
  - Monte Carlo com *Simulated Annealing*
    - O algoritmo é iniciado com altas temperaturas e vai diminuindo até Q se tornar constante.
    - Ótimos locais.
    - *Quenches* periódicos $\rightarrow$ aumenta probabilidade de sucesso
      - ΔQ é calculado movendo todos os vértices da comunidade selecionada para todas as outras comunidades e o movimento com maior ΔQ é aceito.
      - Processo repetido até o maior $\Delta Q \leq 0$

Detecção de Comunidades

- Monte Carlo com *Simulated Annealing* e *Basin Hopping* [Massen and Doye 2005]
    - Monte Carlo com *Basin Hopping*
        - A cada passo, uma série de vértices são selecionados aleatoriamente e movimentados para outras comunidades.
        - Depois de cada passo, aplica-se *quenches* a nova partição.
            - Submete-se os valores de modularidade da partição ao critério de aceitação de Metropolis.
        - Se o passo é aceito, a partição corrente é atualizada para partição resultante do processo de *quenches*.
        - Algoritmo mais lento, mas, com altos valores de modularidade.

Detecção de Comunidades

- Monte Carlo com *Simulated Annealing* e *Basin Hopping* [Massen and Doye 2005]

    - Ponto inicial: qualquer partição dos vértices em comunidades (N comunidades de 1 vértice).
    - Algoritmos mais rápidos → partição inicial obtida de método guloso.
        - Checa todas as possíveis arestas, uma única aresta é escolhida aleatoriamente e inserida se ΔQ satisfaz critério de Metrópolis.
    - *Simulated Annealing* → melhor com partições obtidas de um algoritmo guloso.
        - Parâmetros iniciais: taxa de resfriamento e quantidade de *quenches*.
    - *Basin Hopping* → condições iniciais tem pouco efeito no *Qmax*.
        - Pode ser iniciado com partições aleatórias.

Conclusões

- Importância da detecção de comunidades.
  - Vértices da mesma comunidade tem maiores chances de compartilharem propriedades e dinâmicas.
- Diversas redes reais: Internet, *WWW*, redes sociais, biológicas, organizacionais, de negócio etc.
- Principais desafio da área:
  - Alto custo computacional apresentado pelos algoritmos.
  - Não se conhece a *priori* o número e o tamanho das comunidades.
- Precisa-se:
  - Maior precisão na detecção de comunidades.
  - Menor custo computacional.

Bibliografia

- [Duch and Arenas 2005] Duch, J. and Arenas, A. (2005). Community Detection Complex Networks using Extremal Optimization. Physical Review E, 72.
- [Girvan and Newman 2002] Girvan, M. and Newman, M. E. J. (2002). Community Structure in Social and Biological Networks. Proceedings of the National Academy of Science USA, 99(12):7821–7826.
- [Jain and Dubes 1988] Jain, A. K. and Dubes, R. C. (1988). Algorithms for Clustering Data. Prentice Hall.
- [Massen and Doye 2005] Massen, C. P. and Doye, J. P. K. (2005). Identifying Communities within Energy Landscapes. Physical Review E, 71(046101).

Bibliografia

- [Muff et al. 2005] Muff, S., Rao, F., and Caflisch, A. (2005). Local Modularity Measure for Network Clusterizations. Physical Review E, 72(056107).
- [Newman 2004] Newman, M. E. J. (2004). Fast Algorithm for Detecting Community Structure in Networks. Physical Review E, 69(066133).
- [Newman 2006] Newman, M. E. J. (2006). Finding Community Structure in Networks using the Eigenvectors of Matrices. Physics/0605087.

Detecção de Comunidades

Perguntas?