

Netflix Movies & TV Shows: Estudo, análise e descoberta de conhecimento na rede – Resultados Parciais

Antônio Marcos Machado Bernardes
Ronan José Lopes

Professor: Vinícius Vieira



Base de dados

- TV Shows and Movies listed on Netflix:
<https://www.kaggle.com/shivamb/netflix-shows>
- Número de registros: 7789
- Atributos: show_id, type, title, director, cast, country, date_added, release_year, rating, duration, listed_in, description
- Ferramentas utilizadas na análise: igraph (eventualmente abandonado por questões de desempenho), networkX e gephi (comparações de valores obtidos pelas métricas) e matplotlib para plotagem de distribuições

Pré-processamento

- Para cada série/filme/show de televisão presente na base, foi agregado a nota de avaliação no IMDB
- API utilizada: The Open Movie DataBase (<http://www.omdbapi.com>)
- Dos 7.789 registros, 6.752 retornaram a nota a partir da consulta
- Cada ator pode aparecer em múltiplos registros, portanto, é efetuada uma média de notas a serem associadas ao ator

Modelagem da rede

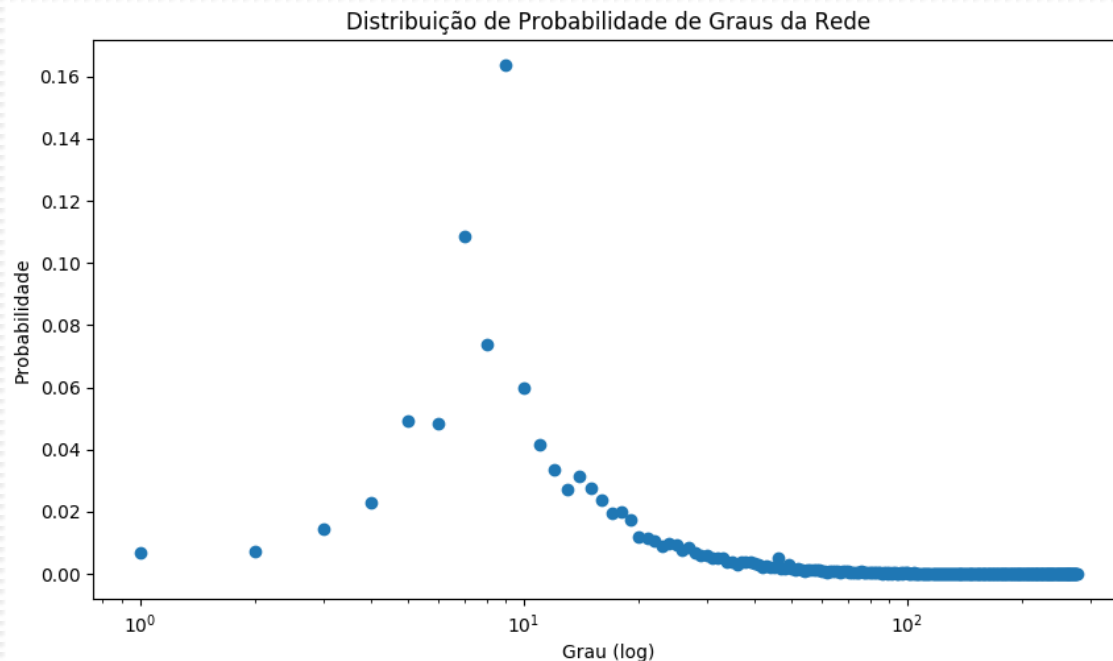
- Em termos de abstração, o grafo a ser modelado representa como interação a co-atuação entre os atores presentes em cada *cast*
- Grafo não-direcionado: vértices representam os atores, ponderados por sua nota média. Existe uma aresta entre dois vértices a_1 e a_2 se a_1 co-atuou com a_2 em algum registro da base
- Pela característica da base, cada cast contido em um registro é, por definição, um clique do grafo (todos os nós tem ligações entre si)

Características básicas da rede

- Número de nós: 32.881
- Número de arestas: 252.055
- Grau médio: 15,3313
- Número de componentes conectados: 857
- Para uma análise mais profunda e melhor utilização dos algoritmos disponíveis na literatura, tomou-se como objeto de estudo a componente gigante da rede, cuja cobertura inclui cerca de 89,5% dos vértices da rede

Componente Gigante

- Número de nós: 29.440
- Número de vértices: 241744
- Grau médio: 16,42

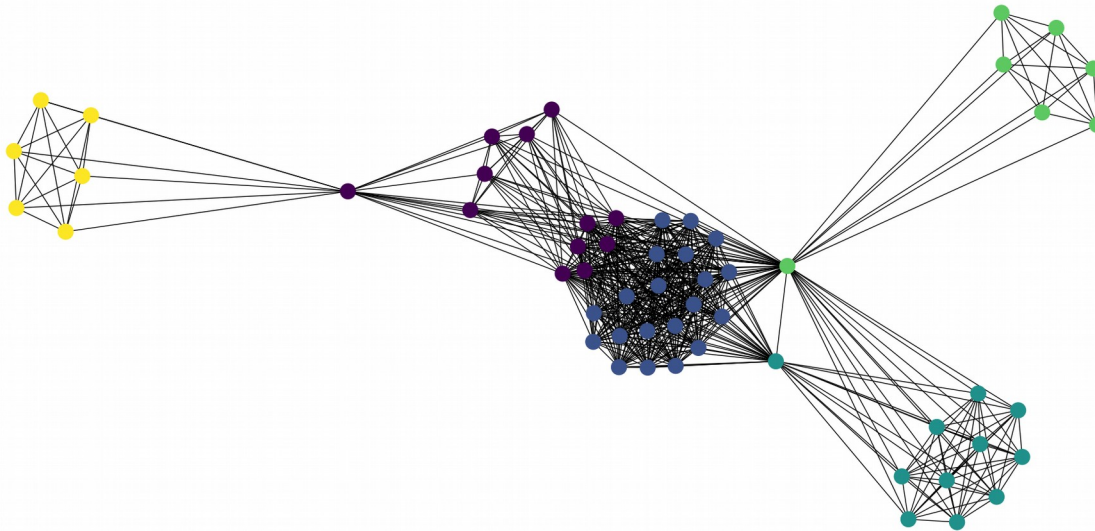


Detecção de comunidades

- No NetworkX, foi utilizado o algoritmo de melhor partição do método Louvain.
- 93 comunidades obtidas na componente gigante:
 - Maior comunidade: 4.823 vértices
 - Menor comunidade: 5 vértices
 - Média: 313,19 vértices
- No Gephi, 99 comunidades foram particionadas utilizando um algoritmo de modularidade para detecção.

Detecção de comunidades

- Exemplo de plot de sub-amostra pelo networkX (o elevado número de vértices e arestas inviabiliza a visualização da componente gigante):



É possível observar a formação dos cliques correspondendo a casts de cada registro, onde alguns nós fazem as pontes interligando os componentes

Detecção de comunidades

- No Gephi, sem as arestas e com a paleta de cores definidas pelas partições de comunidades, e vértices com tamanho proporcional a seu grau:



Componente Gigante

- Densidade: 0,000557(...) - Razão entre número de arestas existentes / possíveis
- Diâmetro: 17 (caminho mais longo possível)
- Coeficiente de Clustering médio: 0,824 (alta probabilidade devido às “aglomerações locais” do cast de cada registro)
- Comprimento médio de caminho: 5,647 (dentro da teoria de seis graus de separação)
- Fechamento triadico: 0,39849(...) - Probabilidade de formação de “triângulos” - relativamente alta pelo mesmo motivo do coeficiente de aglomeração

Topologia da rede

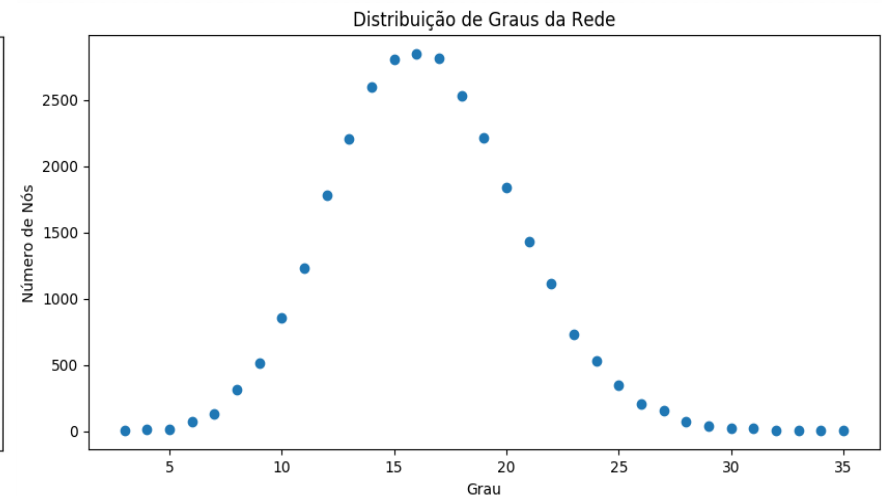
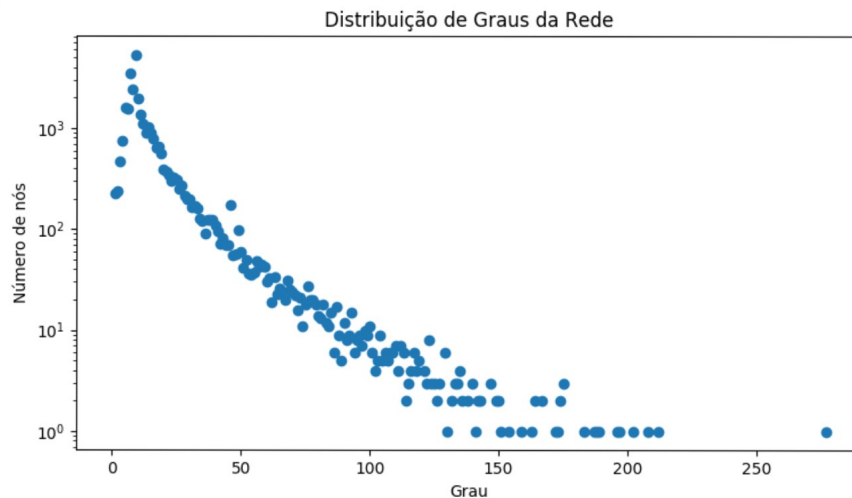
- A fim de se verificar se o processo de formação da rede se aproxima de um processo aleatório, gerou-se um modelo de Erdős-Rényi utilizando os parâmetros da componente principal para quantidade de nós e densidade
- A rede obtida, como esperado, tem 29.440 nós e um número bem próximo de arestas (241.853).
- Número de componentes conectados: 1 (componente principal contém todos os vértices)

Os coeficientes de aglomeração/triangulação são significativamente menores que a rede original:

- Coeficiente de clustering médio: 0.000587(...)
- Fechamento tríadico: 0.000582(...)

Rede modelada vs Erdos-Rényi

- A diferença na formação das redes fica evidente ao observar a distribuição de graus dos nós:



- Enquanto a rede analisada (primeira) segue uma lei de potência em sua distribuição, a rede gerada aleatoriamente (segunda) segue uma distribuição normal.

Rankeamento dos vértices

- A fim de identificar, dentre os nós da rede, aqueles que se destacam em algum critério (número de ligações, centralidade, presença em caminhos mais curtos), foram aplicadas algumas métricas para *rankeamento* dos vértices:
- Maior grau:
 - Anupam Kher – 277 interações
 - Shah Rukh Khan – 212
 - Takahiro Sakurai – 208
 - Yuki Kaji – 202
 - Fred Tatasciore – 197
 - Yuichi Nakamura – 196
 - Fred Armisen – 189
 - Akshay Kumar – 188
 - Om Puri – 187
 - Boman Irani – 183

Rankeamento dos vértices

- A fim de identificar, dentre os nós da rede, aqueles que se destacam em algum critério (número de ligações, centralidade, presença em caminhos mais curtos), foram aplicadas algumas métricas para *rankeamento* dos vértices:
- Betweenness:
 - Anupam Kher – 0.059
 - Om Puri – 0.03
 - Sahajak Boonthanakit – 0.028
 - Iko Uwais – 0.026
 - Ben Kingsley – 0.025
 - Cesar Montano – 0.025
 - Steven Yeun – 0.025
 - Kari Wahlgren – 0.022
 - Haluk Bilginer – 0.021
 - Christopher Lee – 0.02124413707284819

Rankeamento dos vértices

- A fim de identificar, dentre os nós da rede, aqueles que se destacam em algum critério (número de ligações, centralidade, presença em caminhos mais curtos), foram aplicadas algumas métricas para *rankeamento* dos vértices:
- Auto-vetor:
 - Takahiro Sakurai - 0.16
 - Yuichi Nakamura - 0.15
 - Yuki Kaji - 0.15
 - Jun Fukuyama - 0.14
 - Junichi Suwabe - 0.13
 - Katsuyuki Konishi', 0.13
 - Kana Hanazawa - 0.12
 - Eri Kitamura - 0.12
 - Daisuke Ono - 0.12
 - Hiroshi Kamiya - 0.12

Rankeamento dos vértices

- A fim de identificar, dentre os nós da rede, aqueles que se destacam em algum critério (número de ligações, centralidade, presença em caminhos mais curtos), foram aplicadas algumas métricas para *rankeamento* dos vértices:
- Closeness: (execução em andamento há quase 48 horas. Aguardando para ver se a execução é finalizada em tempo hábil)

Em Progresso

- Verificação da viabilidade de um modelo de predição do sucesso na avaliação de um autor (baseado na média do imdb – peso do vértice) de acordo com os pesos/avaliações dos vértices vizinhos
- Analisando a profundidade de nós vizinhos a serem buscados e se existe uma função linear que descreva aproximadamente a predição
- Analisando como verificar se o modelo obtido é viável e analisar sua eficácia para a rede. A ideia é utilizar uma medida de erro padrão e verificar o erro médio para a rede.