Agrupamento de Alunos dada suas respostas

Redes Complexas - CPS765 - 2023/3

PESC/COPPE/UFRJ

Filipe Prates - 116011311

Motivação

Sabemos que diferentes alunos se interessam e interagem de maneira diferente à diferentes estímulos acadêmicos.

Alguns preferem (e aprendem mais com) aulas práticas e interativas, enquanto outros livros físicos e vídeos expositivos. Alguns se interessam mais por uma ou outra disciplina, outros por outras.

A Jovens Gênios possui um dataset com milhões de respostas de alunos do ensino público e privado do Brasil à questões em todas as disciplinas do ensino fundamental. Seria possível usar esses dados para agrupar e entender esses diferentes tipos de alunos para melhor otimizar seus aprendizados?

Problema

Com os dados em rede disponibilizados pela empresa, gostaríamos de agrupar os alunos de acordo com suas relações e propriedades de seus vizinhos de diferentes maneiras, metrificando o quão “bem agrupado” está o resultado,

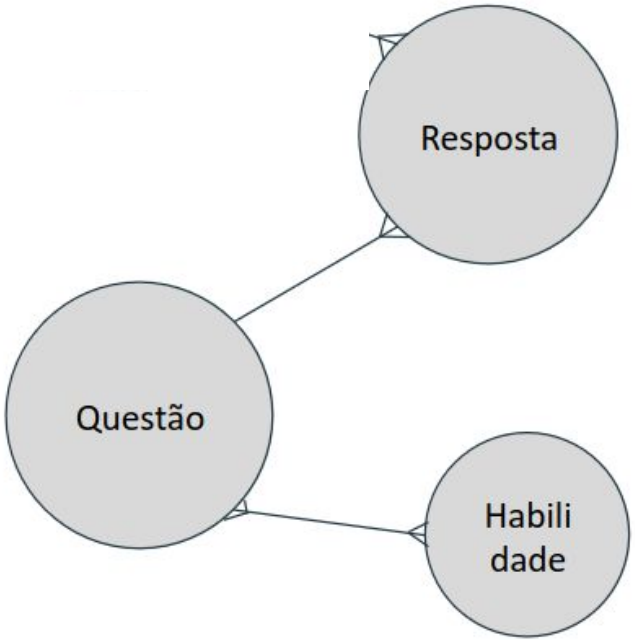
Tentando assim identificar os diferentes tipos de alunos que podem receber diferentes interfaces e estímulos acadêmicos.

Dataset (1)

Respostas para Questões por Alunos do Ensino Fundamental (1º ano - 9º ano)

2000 respostas por cada disciplina para cada ano do ensino fundamental, totalizando 90000 respostas.

Cada resposta contém os dado sobre seu acerto e o tempo que o aluno demorou para selecionar tal resposta.



id_topico nome_topico id_questao habilidades_bncc id_resposta acerto_resposta tempo_resposta datahora_resposta disciplina ano_escolar

Dataset (2)

A empresa rapidamente disponibilizou um segundo dataset para estudo, adicionando também “id_aluno” à resposta, possibilitando estudar e agrupar os usuários.

Porém o dataset (2) possui apenas 2000 respostas, sendo todas em questões de 7º ano de Matemática - contendo 881 alunos únicos, os quais responderam apenas 4 questões únicas. Parece não conter dados suficientes para entender profundamente o aluno em si - poucas perguntas distintas/dimensões entre as quais separar os alunos. Usaremos ambos os datasets no projeto.

id_aluno, id_resposta, id_questao, ids_abilidade, acerto_resposta, tempo_resposta, curso_ano x 2000 rows

Metodologia

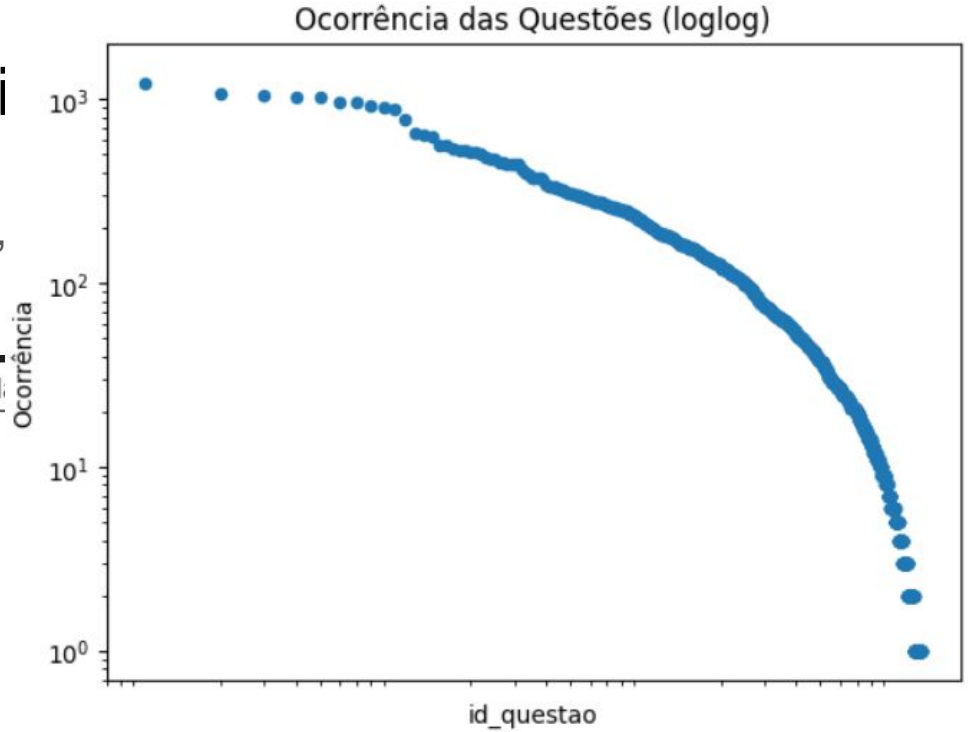
Usaremos então o dataset (1) para melhor entender as respostas , e então usaremos o dataset (2) para criar uma **rede de alunos**, onde uma aresta entre dois alunos tem peso proporcional **à similaridade entre suas respostas**.

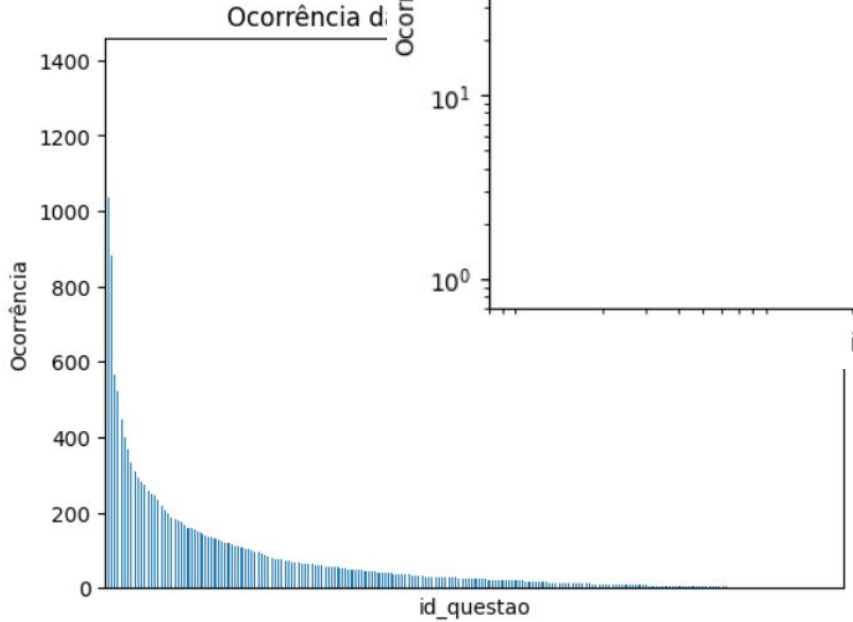
- Sabemos que será uma **rede conexa**. Já que todo aluno para termos seu id no dataset precisa estar relacionado à pelo menos uma resposta.
- A informação dos tópicos e habilidades das questões são redundantes, já que todas as 4 questões únicas compartilham tópicos e abilidades.
-

Metodologi

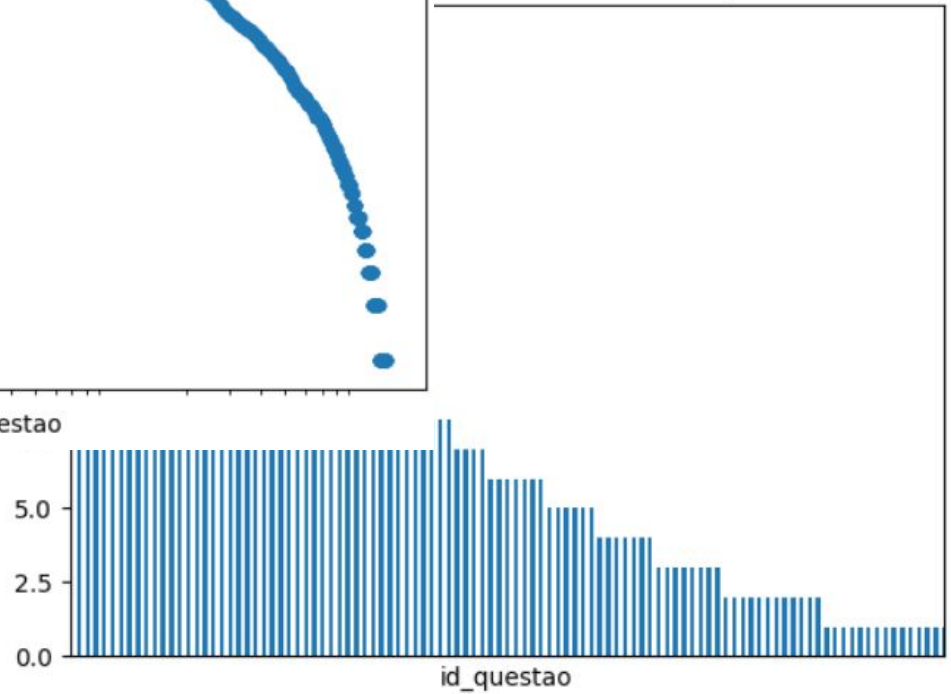
Primeiramente,

No Dataset (1)



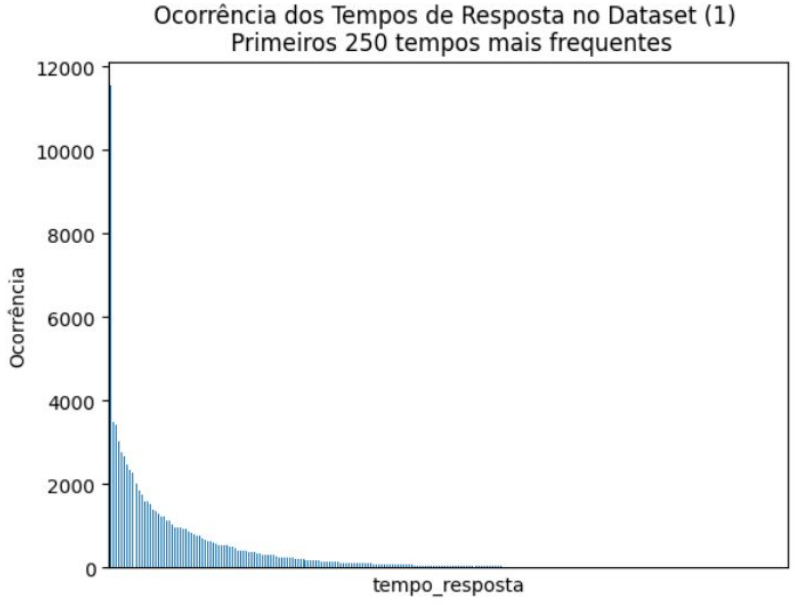


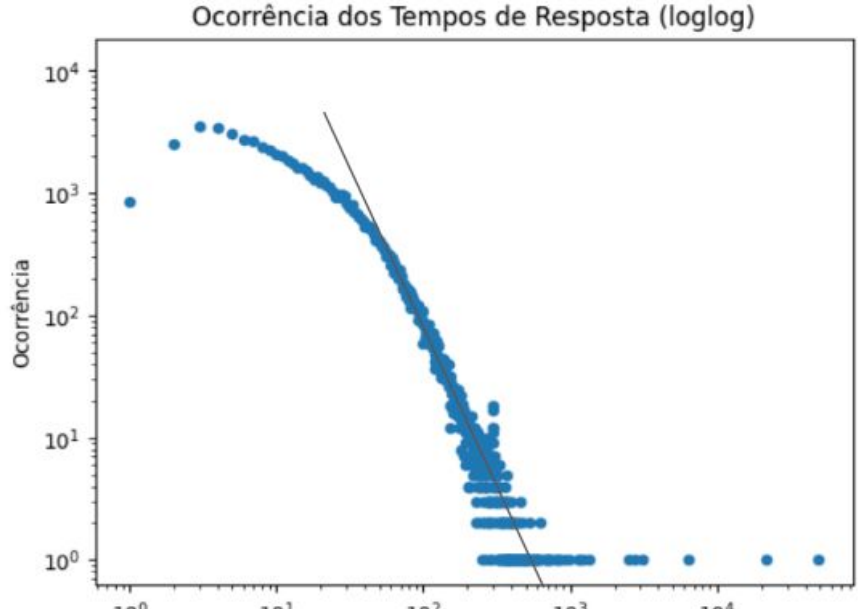
Questões no Dataset (1)
)uestões menos frequentes



Metodologia

Já a distribuição dos tempo de resposta (em segundos) é interessante, possui com maior frequência valores mais baixos, mas também com probabilidade não insignificante valores maiores, quanto maior o valor menos intensamente decresce sua ocorrência - parece seguir mais fielmente uma lei de potência.

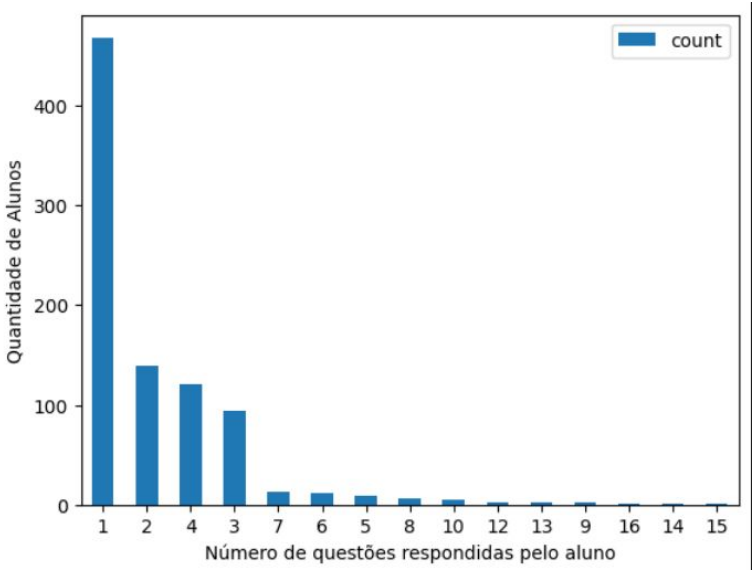




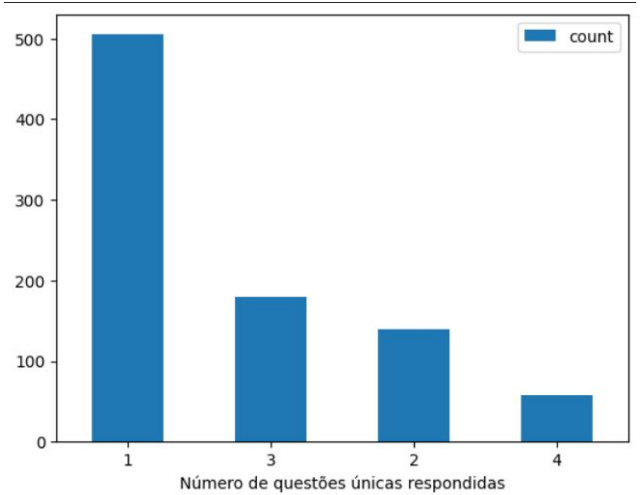
Metodologia

No Dataset (2) conseguimos olhar pros alunos individualmente:

882 alunos únicos respondendo 2000 respostas:



Uma porcentagem pequena porém significativa ~7% (60/881) responderam mais vezes que existem questões únicas.



Metodologia

Similaridade entre alunos dada suas respostas:

$$\alpha(r_i, r_j) = (r_i.acerto \oplus r_j.acerto)!$$(r_i.questao_id \xnor r_j.questão_id)

$$\beta(r_i, r_j) = Sim_{tr}(r_i, r_j)$$

$$Sim_a(a_i, a_j) = \sum_{r_i \in a_i.respostas} \sum_{r_j \in a_j.respostas} \alpha(r_i, r_j)\beta(r_i, r_j)$$

Metodologia - Distância entre respostas

Como notamos dada as distribuições no dataset(1), a distribuição de tempos de resposta das respostas parece seguir uma lei de potência, então podemos melhor comparar dois tempos de resposta,

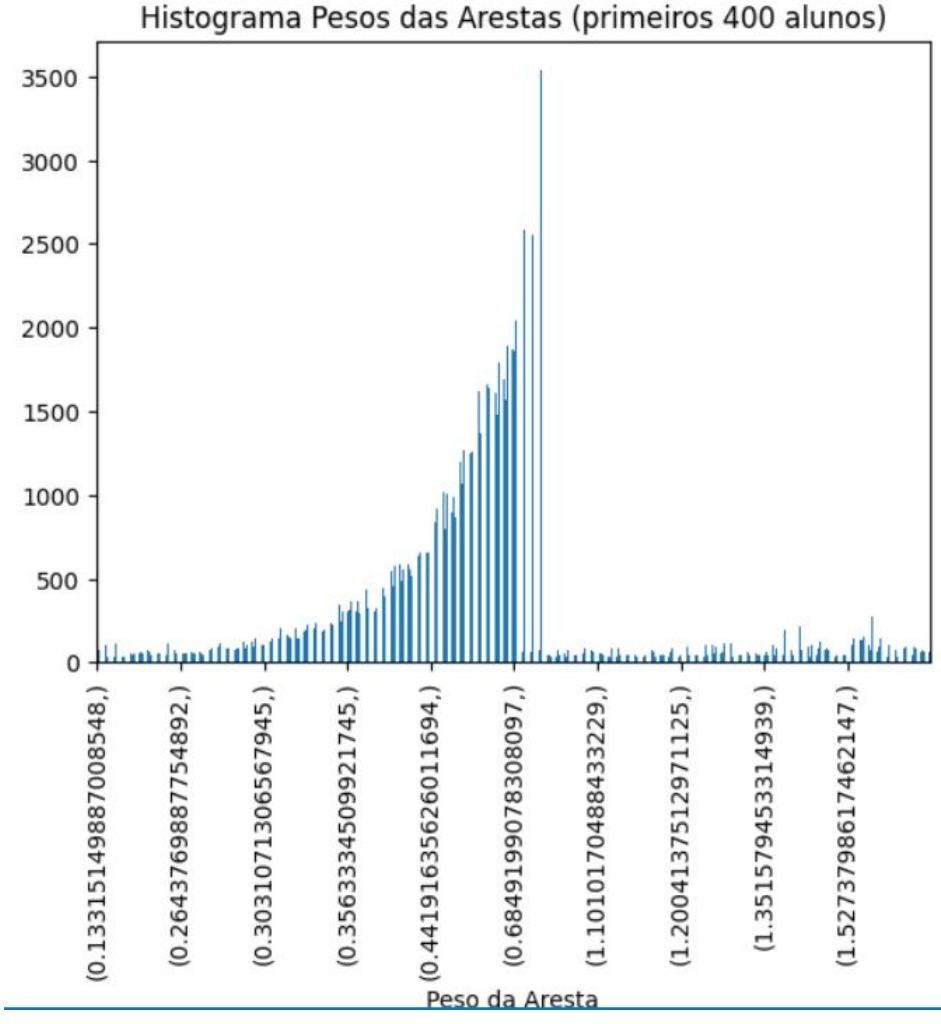
ao invés de:

$$Sim_{tr_0}(r_i, r_j) = 1 - \frac{|r_j.tempo\_resposta - r_i.tempo\_resposta|}{maxT}$$

onde assumimos que a distribuição de valores é uniforme em [0, maxT], metrificando 1 e 2 como tão distantes quanto 1001 e 1002, porém sabemos que na distribuição real suas frequências de ocorrência são mais similares entre 1001 e 1002 do que 1 e 2, e gostaríamos que a métrica representasse isso.

Temos então:

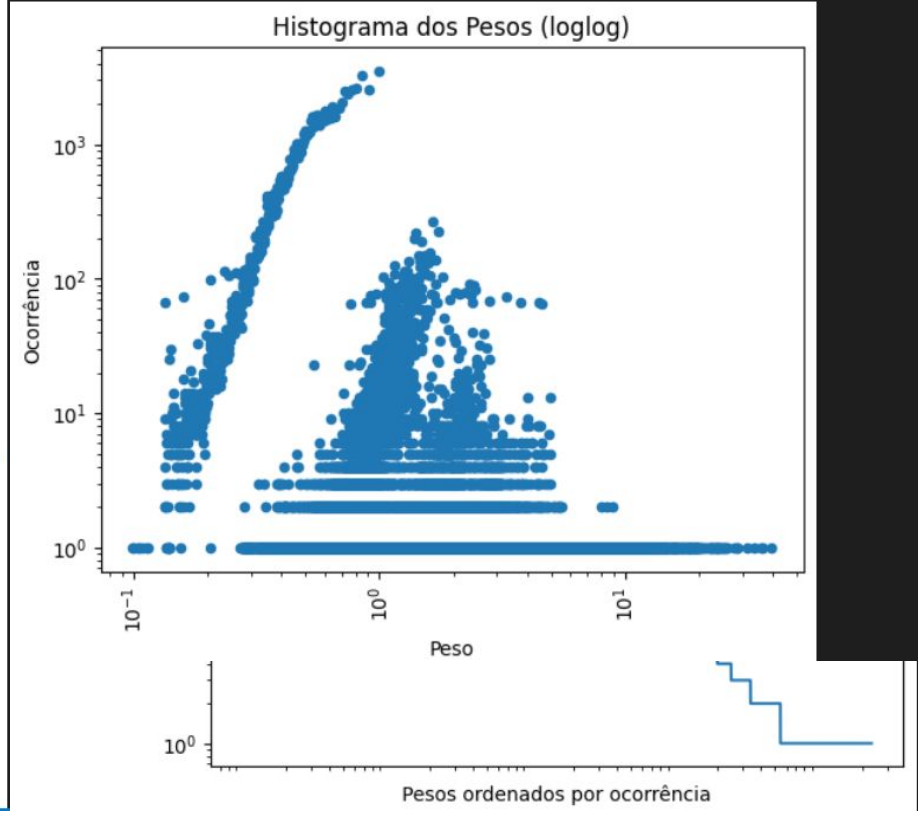
$$Sim_{tr}(r_i, r_j) = 1 - \frac{log(|r_j.tempo\_resposta - r_i.tempo\_resposta| + 1)}{log(maxT + 1)}$$




```
df3 = pd.DataFrame(pd.DataFrame(weights).value_counts()).reset_index()
df3.plot(kind='scatter',x=0,y="count",
         title="Histograma dos Pesos (loglog)",
         ylabel="Ocorrência", xlabel="Peso", loglog=True, rot=90)
```

✓ 0.5s

```
Axes: title={'center': 'Histograma dos Pesos (loglog)'}, xlabel='Peso', ylab
```



Metodologia - Rede resultant

A rede resultante de Alunos com peso de
referente à Sim_a(a_i,a_j), possui então:

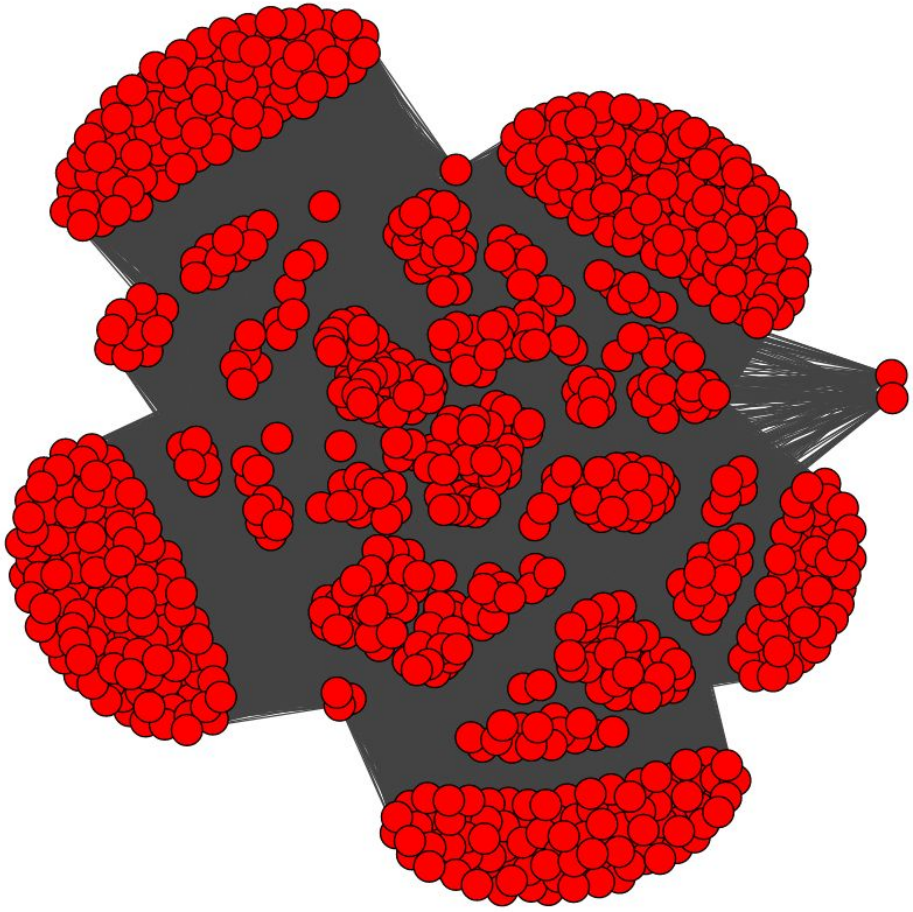
N = **882**

M = **164219**,

(logo 224302 pares alunos não com
acerto/erro em nenhuma questão)

Peso médio de aresta 351.726704

Peso máximo de aresta 2970.576825



Metodologia - Agrupamento

Agrupamos a rede de aluno com diversos algoritmos disponíveis na biblioteca python-igraph:

- community_infomap
  - 'Clustering with 882 elements and 1 clusters'
- community_fastgreedy
  - 'Clustering with 882 elements and 4 clusters'
- community_label_propagation
  - 'Clustering with 882 elements and 1 clusters'
- community_leading_eigenvector
  - 'Clustering with 882 elements and 4 clusters'

.modularity:

0.0

0.2425873265749233

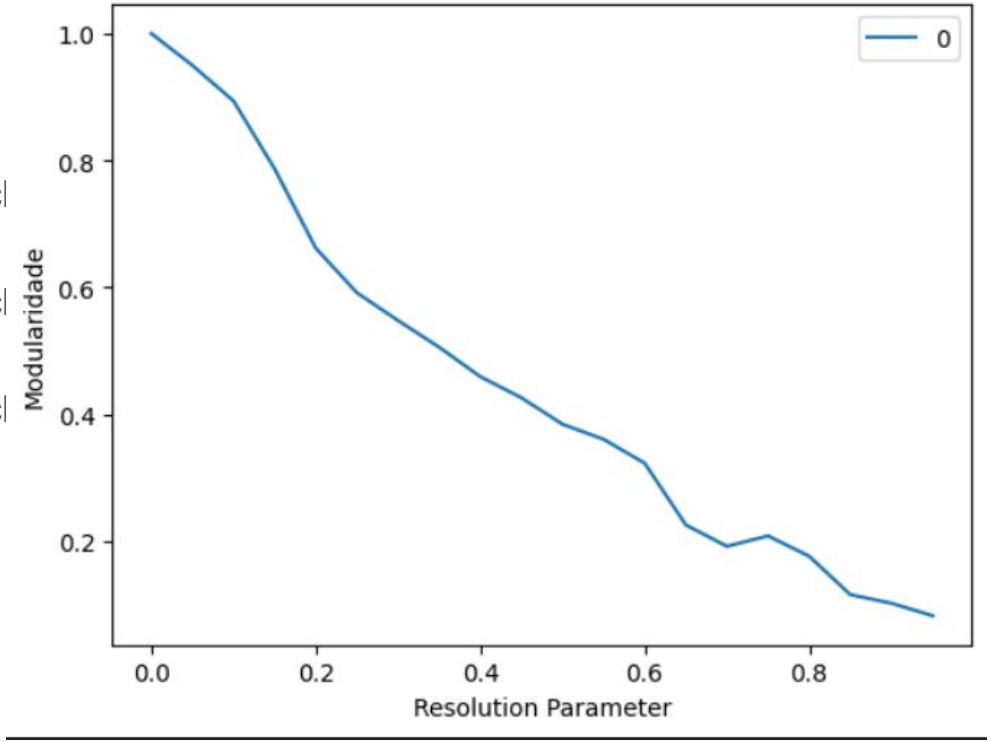
0.0

0.2604914269488371

Resultados

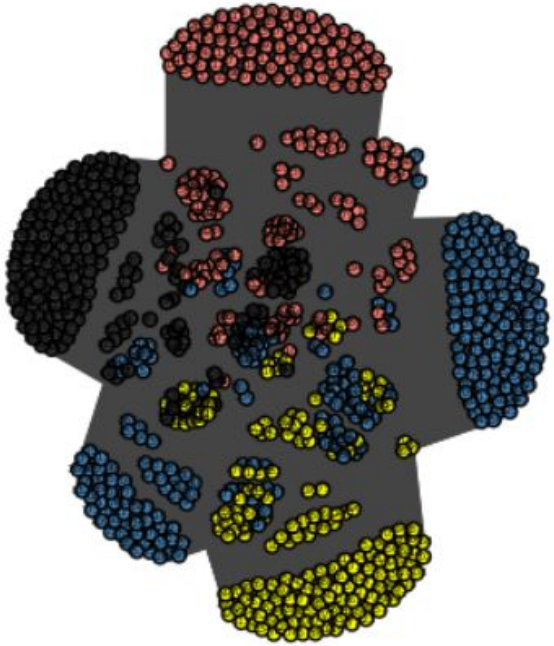
Agrupamos a rede de aluno com diversos algoritmos disponíveis na biblioteca python-igraph:

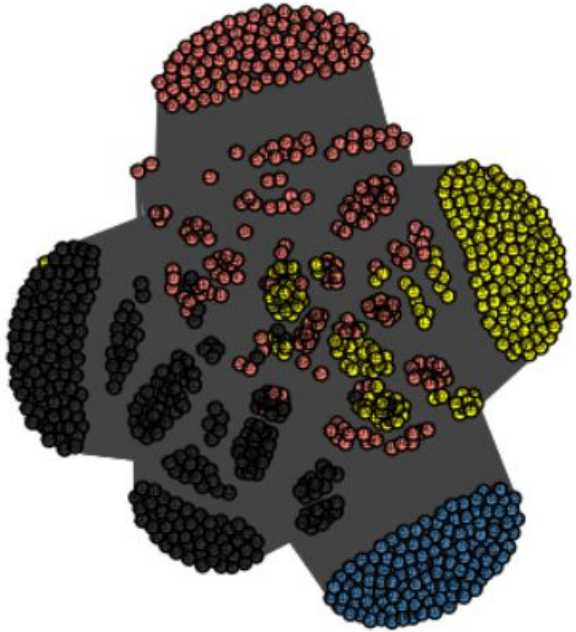
- community_leiden
  - 'Clustering with 882 elements and 8 cl
- community_multilevel
  - 'Clustering with 882 elements and 5 cl
- community_walktrap
  - 'Clustering with 882 elements and 5 cl
- community_spinglass
  - 3 minutos + rodando
- community_optimal_modularity
  - chrash



Resultados

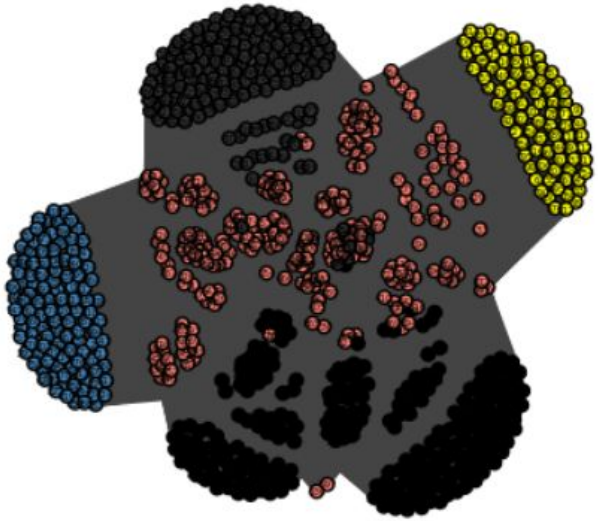
Leading Eigenvector
|C| = 4, M = 0.26





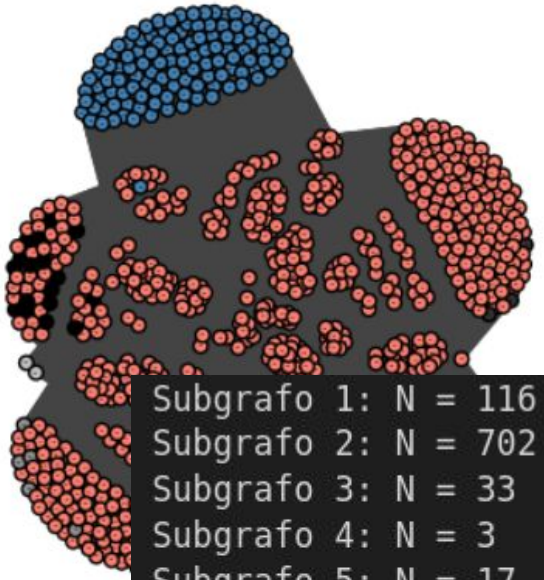
Fast and Greedy
|C| = 4, M = 0.24

Resultados

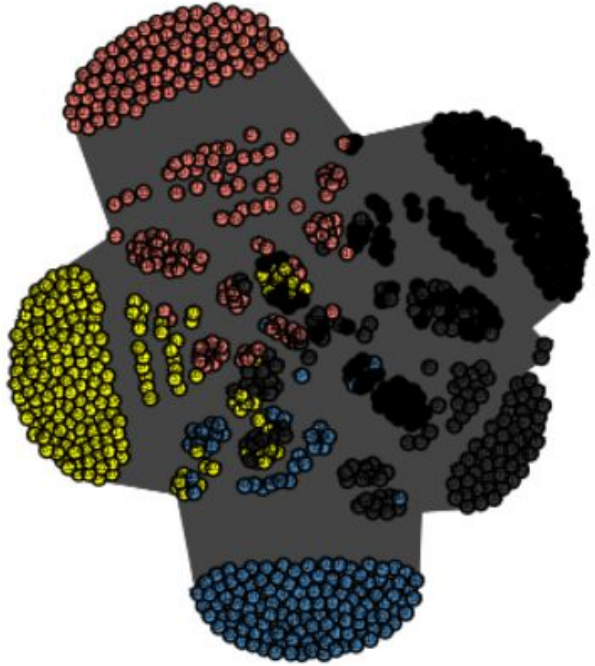


Walktrap
|C| = 5, M = 0.176

Leiden R=0.17
|C| = 8, M = 0.73



```
Subgrafo 1: N = 116
Subgrafo 2: N = 702
Subgrafo 3: N = 33
Subgrafo 4: N = 3
Subgrafo 5: N = 17
Subgrafo 6: N = 1
Subgrafo 7: N = 8
Subgrafo 8: N = 2
```



Multilevel
|C| = 5, M = 0.265

Resultados

Detecção de comunidades:

A alta densidade do grafo, e o fato dos alunos serem comparados em poucas questões, dificulta o agrupamento que resulta em alta modularidade.

Detecção de anomalias:

Alunos {142, 948} - Responderam múltiplas vezes as mesmas questões similarmente, sempre identificados como mesmo cluster, possivelmente de apenas 2 elementos. Provável uso indevido da plataforma - exploit de interface.

Resultados

Para geração dos pesos estou comparando:

para cada par de alunos:

todas as respostas de um aluno com

todas as respostas de outro

~ $n^2 r\_mean^2$, demais para utilizar em datasets maiores.

Para todos os pares de respostas à uma mesma questão somamos algum valor maior que 0

Isso fortalece muito arestas de alunos que ambos repetiram uma mesma questão, se cada aluno do par respondeu uma mesma questão k vezes temos k^2 comparações, aumentando muito o peso destas arestas,

Estudos futuros

- Remover mais arestas (threshold de peso mínimo para acrescentar aresta no grafo?) - observar se modularidade melhora indicando clusters mais bem definidos.
- Estudar alguns casos de cada subgrafo de cada cluster para tentar entender melhor as características de cada grupo.
- Obter mais dados com uma maior variedade de questões e tópicos representados, tendo apenas respostas à poucas questões limita o espaço de similaridade que temos para separar os clusters. Especialmente disciplinas diferentes é interessante para identificar as preferências e aptidões naturais dos alunos.

Referências

fastgreedy:
A. Clauset, M. E. J. Newman and C. Moore: *Finding community structure in very large networks.* Phys Rev E 70, 066111 (2004).
infomap:
M. Rosvall and C. T. Bergstrom: *Maps of information flow reveal community structure in complex networks*. PNAS 105, 1118 (2008). http://arxiv.org/abs/0707.0609M. Rosvall, D. Axelsson and C. T. Bergstrom: *The map equation*. Eur Phys J Special Topics 178, 13 (2009). http://arxiv.org/abs/0906.1405
labelpropagation:
Raghavan, U.N. and Albert, R. and Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76:036106, 2007. http://arxiv.org/abs/0709.2938.
leadingeigen:
MEJ Newman: Finding community structure in networks using the eigenvectors of matrices, arXiv:physics/0605087
multilevel:
VD Blondel, J-L Guillaume, R Lambiotte and E Lefebvre: Fast unfolding of community hierarchies in large networks. J Stat Mech P10008 (2008), http://arxiv.org/abs/0803.0476
walktrap:
Pascal Pons, Matthieu Latapy: Computing communities in large networks using random walks, http://arxiv.org/abs/physics/0512106.
powerlaw:
M. E. J. Newman, Power laws, Pareto distributions and Zipf's law. Contemporary Physics 46, 323-351 (2005)M. Mitzenmacher, A Brief History of Generative Models for Power Law and Lognormal Distributions. Internet Mathematics, Vol 1, No. 2, pp. 226-251, 2004.