

Grafos Encontram Redes: Aprendendo Representação Vetorial para Vértices utilizando Redes Neurais

Ciclo de Palestras – DME/IM/UFRJ
30 / 8 / 2023

Daniel Rattón Figueiredo
PESC/COPPE/UFRJ



Grafos e Redes

- **Grafos:** abstração para codificar relacionamento entre pares de objetos
 - ferramenta matemática
- **Redes:** grafos gerados de dados reais
 - redes sociais, de informação, tecnológicas, etc
- **Ciência de Redes x Teoria dos Grafos**
 - dois lados de uma mesma moeda

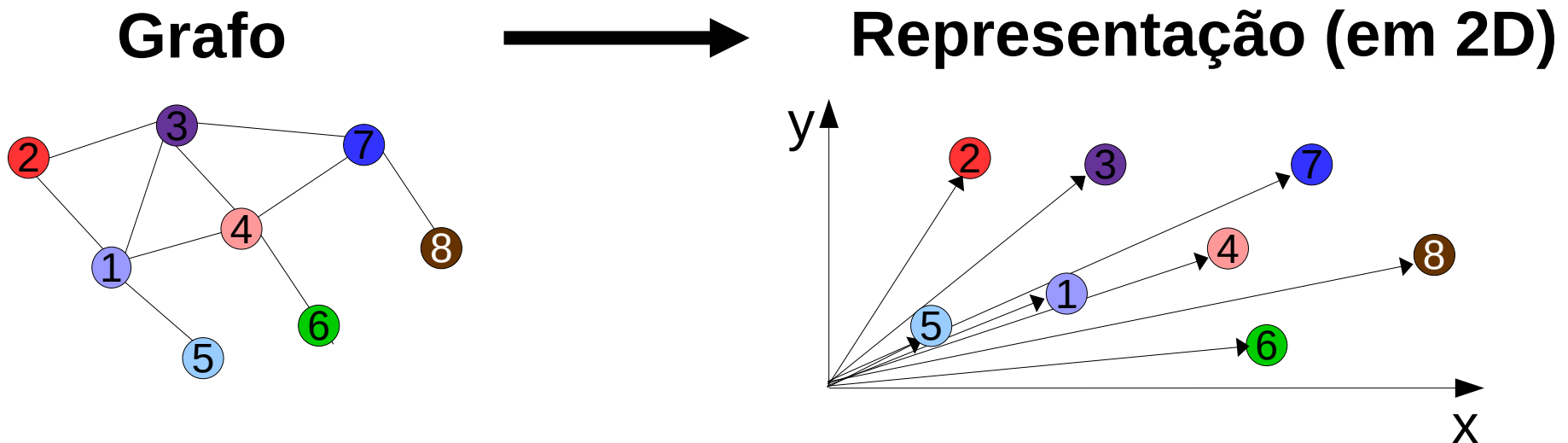
Fábio Botler

Grafos = Redes



Representação de Vértices

- **Problema:** mapear vértices de um grafo no espaço Euclidiano
 - cada vértice é associado a um vetor
 - vetor é a representação (*embedding*) do vértice



Por que fazer isso?
Como fazer isso?

Por que?

- Representação coloca grafo no espaço Euclidiano
 - dimensão d (fixa) bem menor do que n (varia)
- Vértices se tornam vetores
 - permite uso de técnicas de espaço contínuo para analisar o grafo
 - ex: K-means para clusterização, K-NN para vizinhos mais próximos, cosseno para similaridade
- Vetores se tornam atributos dos vértices
 - usado em algoritmos de aprendizado
 - ex: classificação dos vértices

Mais fácil trabalhar com vetores!

Ganhando Aderência



Stanford Computer Forum 2021 Affiliates Meeting

Stanford Graph Learning Workshop

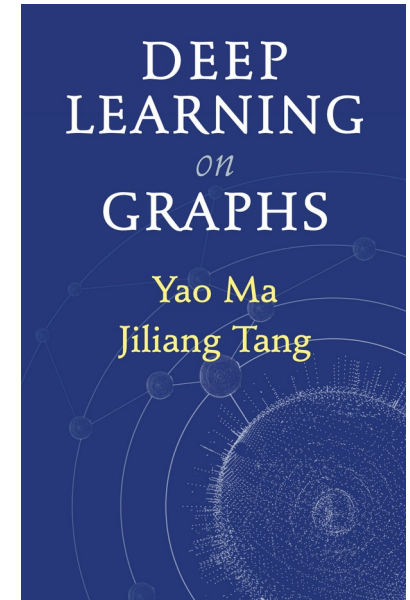
10:45 - 12:00 Industry panel [\[Video\]](#) [\[Liv 16:15 - 17:00\]](#) Industry panel [\[Video\]](#) [\[Livestream\]](#)

- Andrew Zhai, Pinterest
- Jaewon Yang, LinkedIn
- Benedek Rozemberczki, AstraZeneca
- Hatem Helal, Graphcore
- Nadia Fawaz, Pinterest (moderator)
- Kim Branson, GlaxoSmithKline
- Marinka Zitnik, Harvard University
- Naren Chittar, JP Morgan Chase
- Yu Liu, Facebook AI
- Hema Raghavan, LinkedIn (moderator)

IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, VOL. 2, NO. 2, APRIL 2021

Graph Learning: A Survey

Feng Xia , Senior Member, IEEE, Ke Sun , Shuo Yu , Member, IEEE, Abdul Aziz , Liangtian Wan , Member, IEEE, Shirui Pan , and Huan Liu , Fellow, IEEE



Cambridge Univ Press 2021

184 referências !

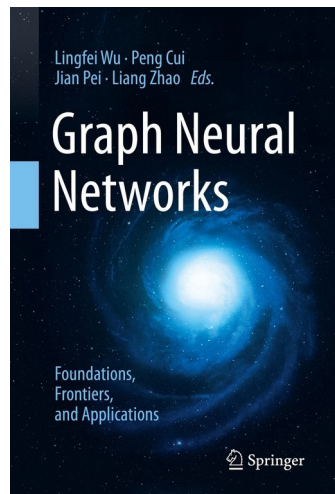
Learning on Graphs Conference



LoG is a new annual research conference that covers areas broadly related to **machine learning on graphs** and **geometry**, with a special focus on review quality.

Springer
2022

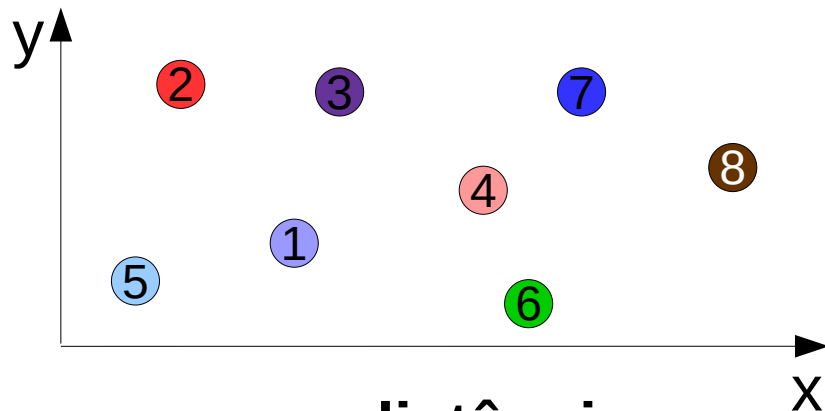
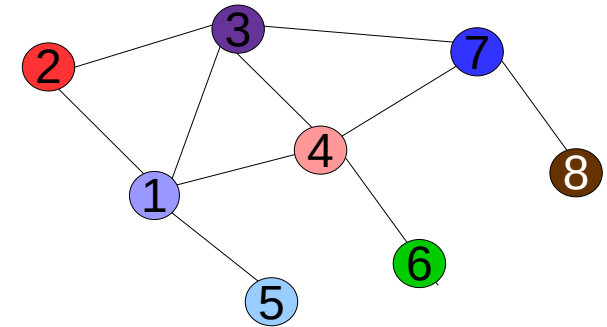
DME/IM 2023



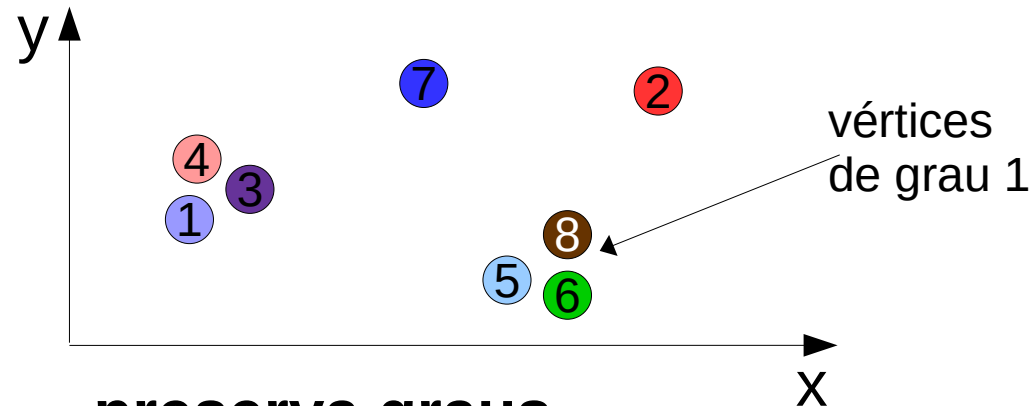
Boas Representações



- Qual a melhor representação para um grafo?



preserva distâncias
no grafo



preserva graus
no grafo

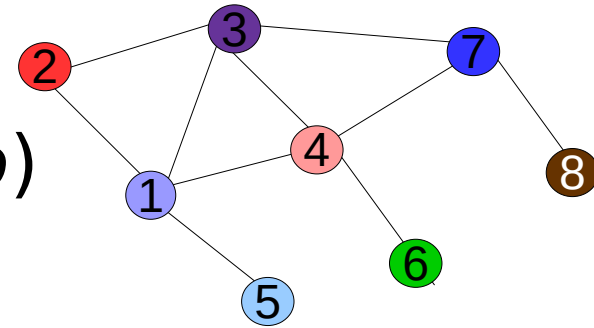
Depende dos objetivos!

- Qual a finalidade da representação?
- Não há bala de prata!

Como ?

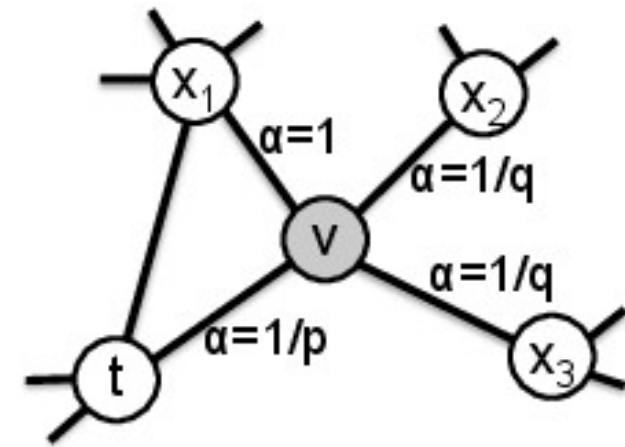
- Duas abordagens para gerar representações
 - 1) fatoração/otimização: mais antiga e menos flexível
 - 2) aprendizado: mais recente e mais flexível
- Muitas técnicas em cada abordagem
 - técnicas capturam diferentes aspectos, gerando diferentes representações
 - não supervisionadas, mas podem ser supervisionadas
 - em geral, técnicas possuem aleatoriedade

DeepWalk

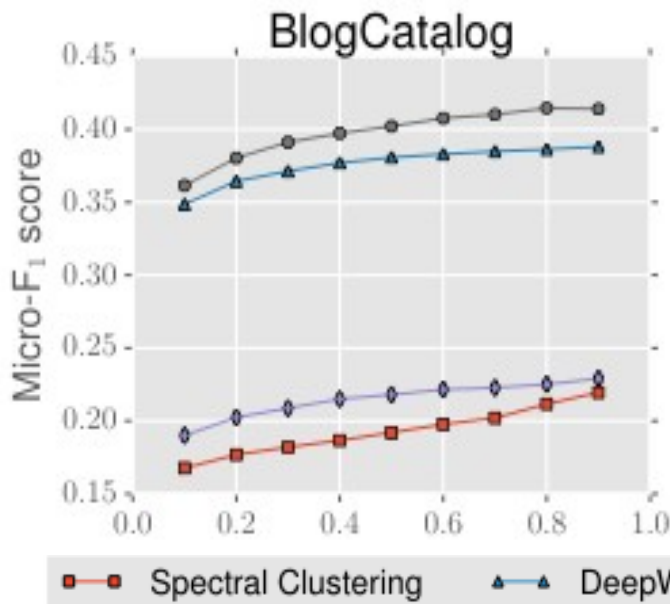


- Gera sequência de vértices (*contexto*) usando passeios aleatórios curtos
 - ex: 2,1,3,2,3 – 2,3,4,1,2 – 2,3,4,7,3, ...
- Contextos tratados como frases de uma linguagem natural
 - rótulos dos vértices são as palavras
- Usar *word2vec* para gerar representação para palavras (vértices)
 - maximizar probabilidade do contexto dado a palavra
 - rede neural gera vetor para palavras em função do contexto

node2vec



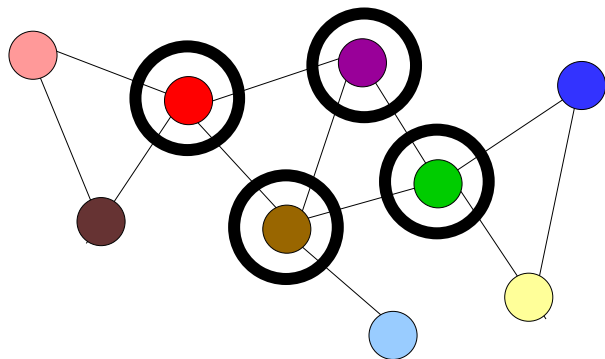
- Mesma abordagem que DeepWalk
- Contexto gerado por passeios aleatórios enviesados
 - parâmetros p , q controlam viés do passeio
- Enfatizar estrutura local quando necessário
 - ajusta representação para diferentes problemas



- Classificação de Bloggers usando rede social (10K nós, 39 classes)
- Regressão logística treinada com representações dos vértices

Identidade Estrutural

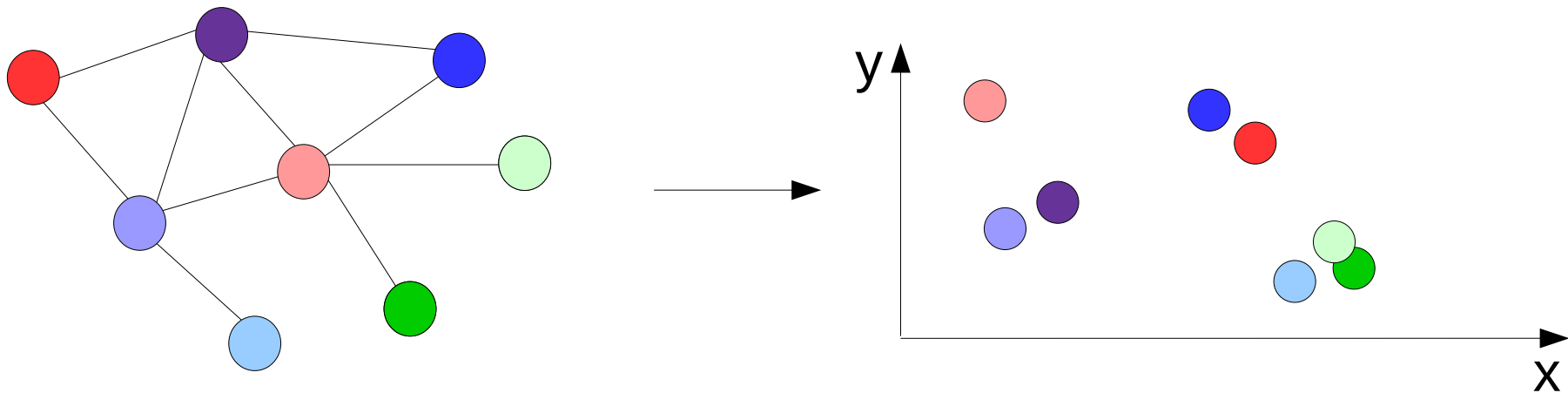
- Vértices da rede possuem diferentes papéis
 - pessoas, páginas web, proteínas, etc
- Identidade estrutural
 - identificação dos vértices baseado na estrutura da rede (e nenhum atributo)
 - relação direta com o papel do vértice
- Automorfismo: forte equivalência estrutural



- Vermelho, Verde: automorfismo
- Roxo, Marrom: estruturalmente parecidos

Encontrando Identidades

- Representar vértices no espaço Euclidiano
- Vértices estruturalmente similares ficam próximos no espaço



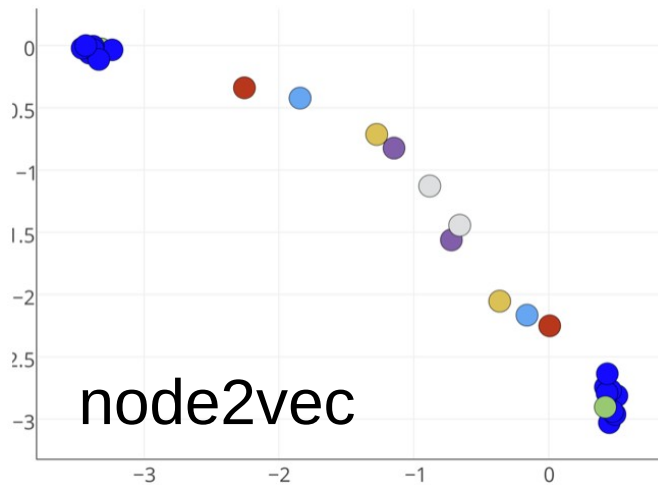
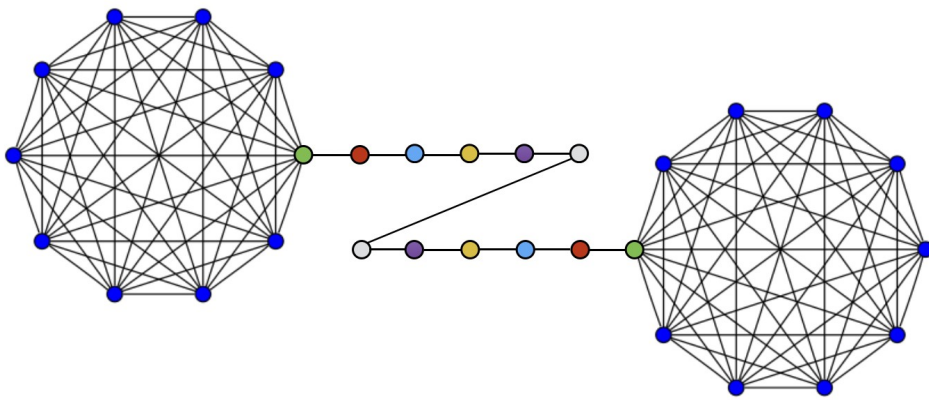
- Usar técnicas de classificação/clusterização convencionais a partir da representação

struc2vec

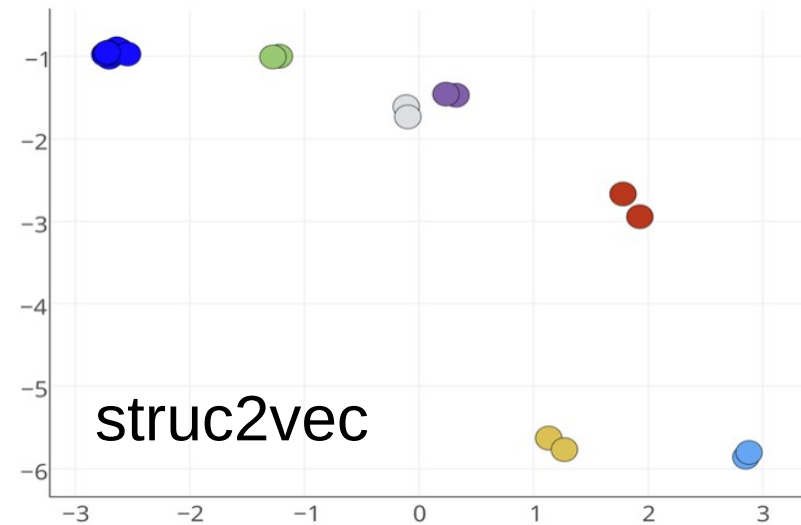
- Primeira técnica para atacar este problema utilizando aprendizado
- Construir um outro grafo: arestas indicam similaridade estrutural
 - não utiliza os rótulos dos vértices (apenas os graus)
- Gerar contexto usando passeios aleatórios enviesado no novo grafo (multi-camadas)
- Treinamento idêntico ao node2vec
 - geração do contexto é diferente
- Nova metodologia para avaliação

Exemplo

- Projeção em 2D das representações



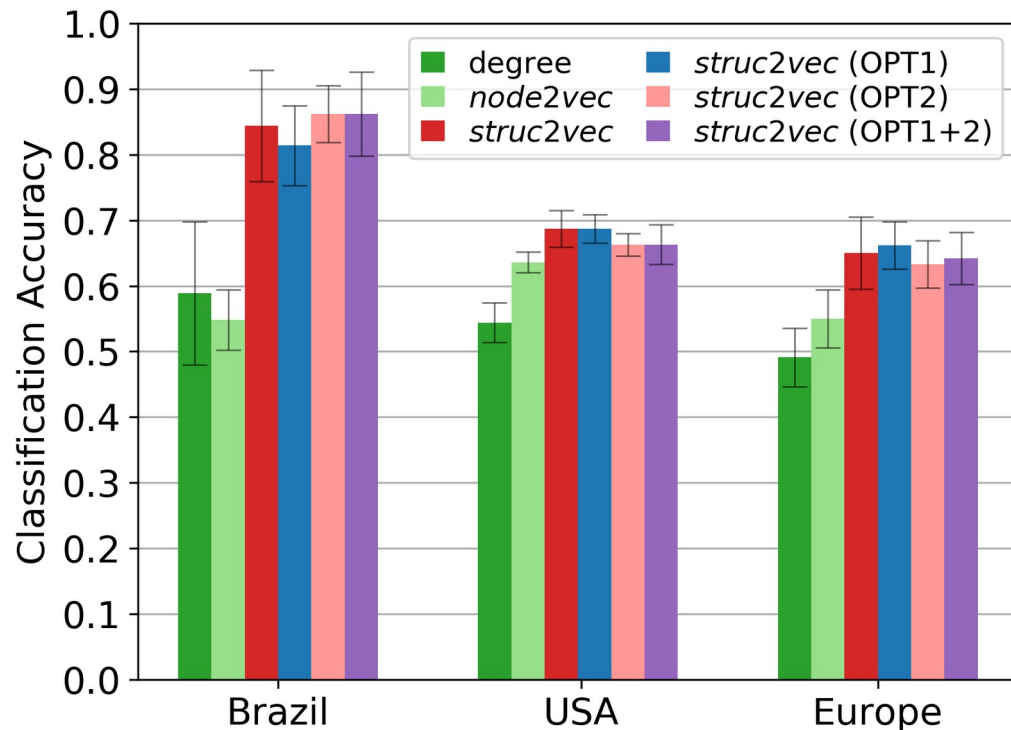
- Captura proximidade no grafo



- Captura similaridade estrutural
 - automorfismos bem próximos

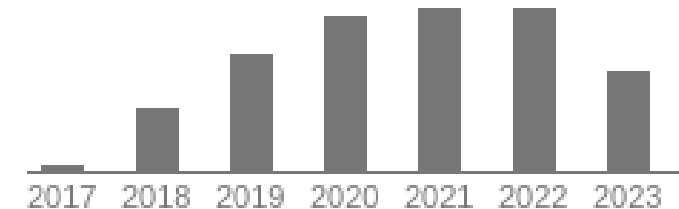
Classificação de Aeroportos

- Representação dos vértices usada para treinar um classificador (movimentação dos aeroportos)
- 4 classes, regressão logística



- Artigo bem recebido pela comunidade

Total citations Cited by 1163



- 1o. lugar no CTDIAC
- tese de mestrado do Leonardo

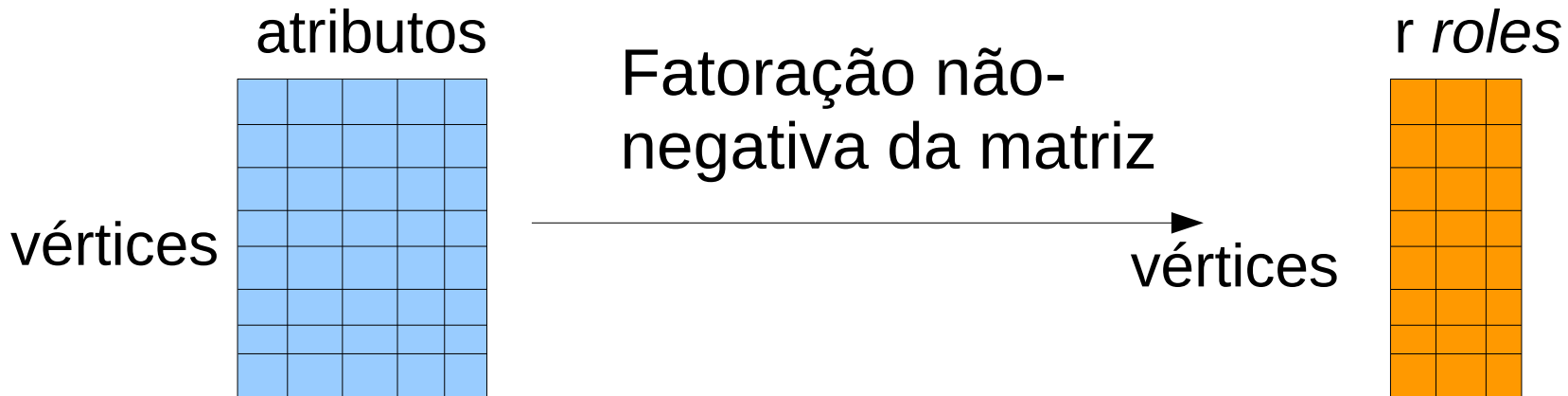
Alerta de Troca de Tema

Redes com Atributos

- Vértices (e arestas) de redes reais possuem atributos
 - rede social: nacionalidade, idade, preferência, etc
 - rede de proteínas: composição molecular, etc
- Dados reais coletados com cada vez mais atributos
- Vizinhaça e atributos são indicativos do papel do vértice
- **Ideia:** usar atributos para gerar representações!
 - métodos anteriores são puramente estruturais

Rolx

- Matriz de atributos e vértices
 - estruturais (grau, triângulos, etc)
 - exógenos (nacionalidade, idade, etc)
- Centenas de atributos



- Representação indica papéis do vértice
- Alto custo computacional

Graph Neural Networks

- Métodos de aprendizagem anteriores
 - puramente estruturais, não utilizam atributos
 - baseados na geração de contexto (aleatório)
 - representação apenas para vértices presentes na rede

Graph Neural Networks

- Metodologia para gerar representações misturando estrutura e atributos
 - inspirado em Convolutional Neural Networks (usado em imagens)

GNN: Principais Ideias

- Representação do vértice é função da representação dos vizinhos
 - conjunto de vizinhos (ou amostra)
- Iterativo: vértice possui uma representação para cada iteração/camada
 - representação inicial são atributos dos vértices
- Rede neural gera a representação
 - representação de entrada em saída
- Função objetivo depende das representações
 - exemplos positivos e negativos, com ou sem supervisão

Formalizando

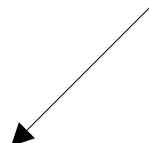
- $x_{v,k}$: representação do vértice v na iteração k
- $y_{v,k}$: representação (média) dos vizinhos de v na iteração k
- W_k, S_k : matrizes que transformam representação da iteração $k-1$
- σ : função de ativação
- N_v : vizinhos do vértice v
- x_0 : representação inicial (função dos atributos)

$$x_{v,k} = \sigma(h_{v,k})$$

$$h_{v,k} = W_k * y_{v,k-1} + S_k * x_{v,k-1}$$

$$y_{v,k} = \frac{1}{|N_v|} \sum_{u \in N_v} x_{u,k}$$

Outras funções de soma podem ser usadas!



Outras funções de agregação podem ser usadas!


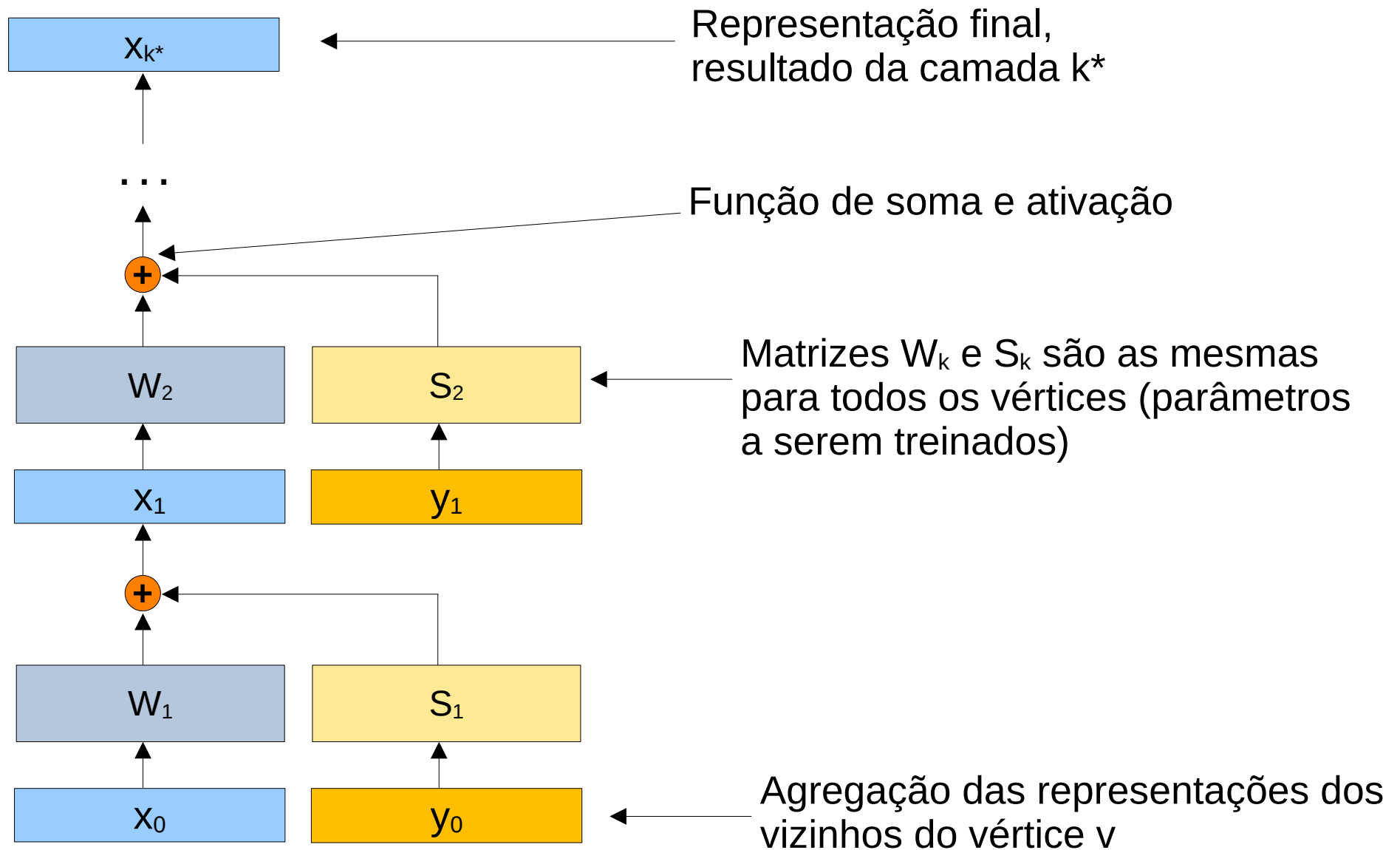


Diagrama da Rede Neural



■ Mesma rede usada para todos os vértices

Função Objetivo

- Em geral, *triplet loss*
 - amostras positivas, negativas, e margem
- Usando métrica entre representações
 - ex. distância Euclideana, produto interno

$$f = \sum_{(u,v) \in P} \|x_u - x_v\| - \sum_{(u,v) \in Q} \|x_u - x_v\| + \Delta$$

Conjunto de pares positivos

Conjunto de pares negativos

Margem

- Diferentes formas de construir P e Q
 - ex. P = pares vizinhos, Q = pares ao acaso

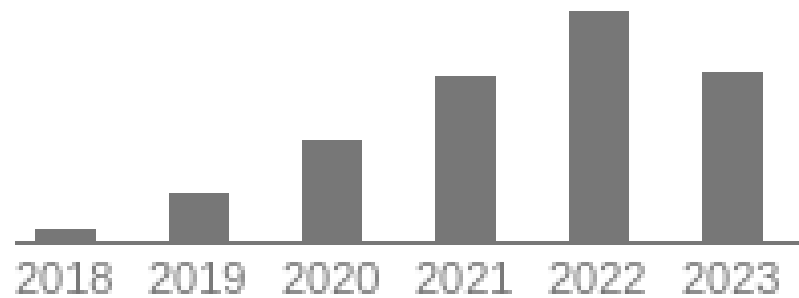
Características

- Não precisa ser treinada com todos vértices
 - amostragem, inclusive da vizinhança
- Rede treinada gera representação para qualquer vértice
 - mesmo vértices nunca antes vistos, ou vértices de outras redes
- Metodologia muito flexível
 - atributos de entrada, função de agregação, função de fusão, função objetivo, técnica de amostragem, número de camadas, dimensão da representação, etc

GraphSage

- Uma das primeiras GNNs
 - definiu essência da metodologia
- Abordagem rompeu com paradigma anterior

Total citations Cited by 11401

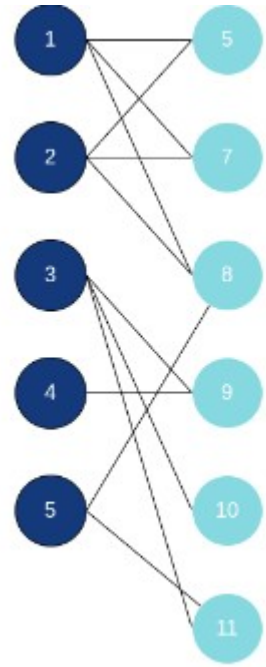


Enorme sucesso!

- Centenas de variações e melhorias e aplicações

Clusterização em Grafos

- Uma das principais aplicações de GNN
 - particionar vértices do grafo: graph clustering / community detection
- 1) GNN gera representação dos vértices
- 2) Clusterização das representações
 - vetores no espaço vetorial
- Representação é função dos atributos dos vértices e estrutura da rede
- Etapas 1 ou 2 podem ser supervisionadas ou não-supervisionadas
 - supervisionado: classe influencia representação, classe influencia clusterização



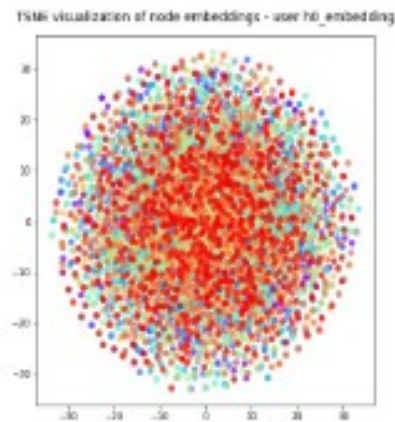
Recomendação

- Rede bipartida: usuários e itens
 - arestas indicam interação
- Clusterização de usuários e itens
 - utilizando atributos (de usuários e itens) e estrutura da rede
- Classes indentificadas usadas para fazer recomendações aos usuários
- GNN para representação em redes bipartidas
 - espaço de usuários e espaço de produtos
- Vizinhaça de dois saltos preserva tipo
 - metodologia com amostragem e pesos

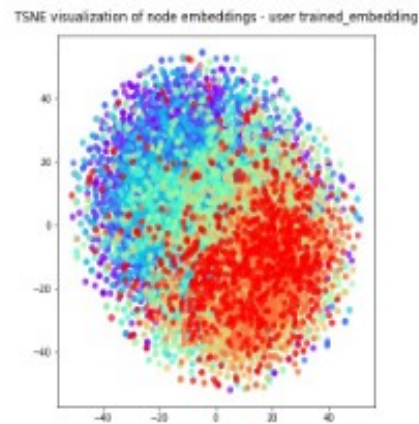
Exemplo

- Modelo SBM adaptado para redes bipartidas
- 8 classes de usuários, 8 classes de itens
- Metodologia não supervisionada
- Projeção em 2D (t-SNE) das representações
 - função do treinamento da GNN

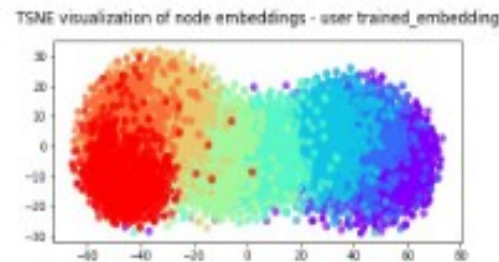
Inicial



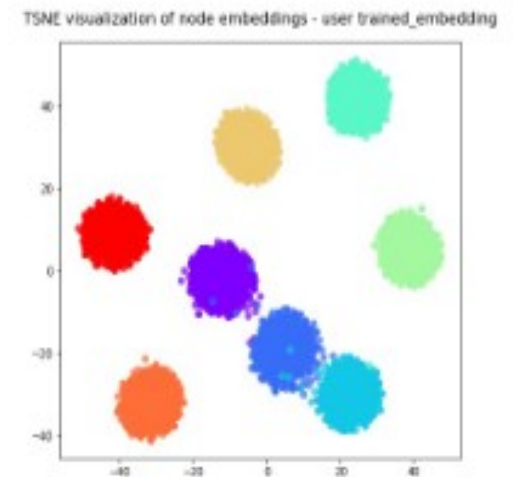
5 épocas



10 épocas

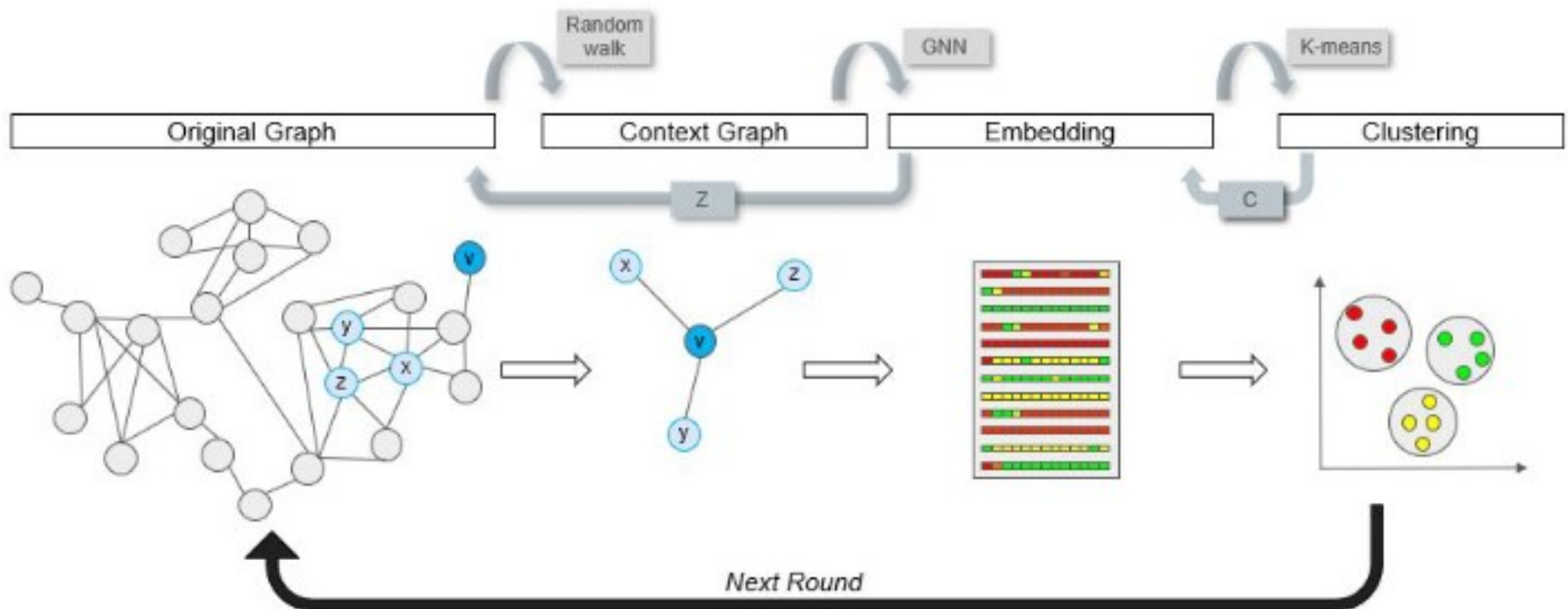


30 épocas



Clusterização em Grafos

- Importância da informação de rede e atributos
 - arestas/atributos podem ser +/- informativos
- Metodologia iterativa, auto supervisionada, utilizando uma rede intermediária (contexto)

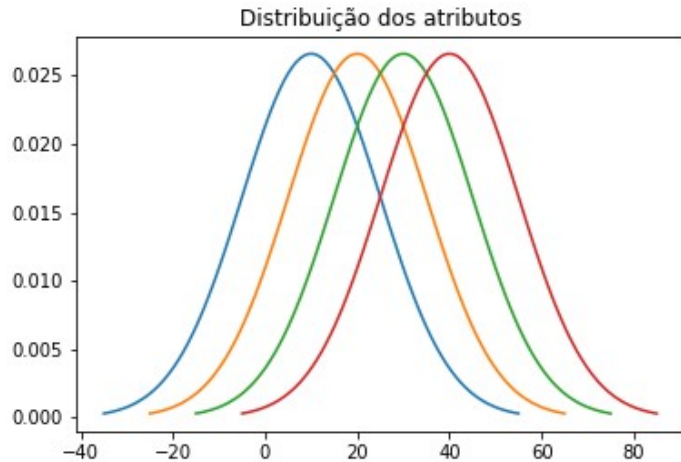


Acoplando Algoritmos

- Metodologia iterativa, auto supervisionada
 - 1) GNN gera representações para rede de contexto, que depende da clusterização
 - 2) K-means gera clusters, que depende da representação gerada
- Ideia: algoritmos se ajudam mutuamente!
 - juntos conseguem extrair melhor a informação

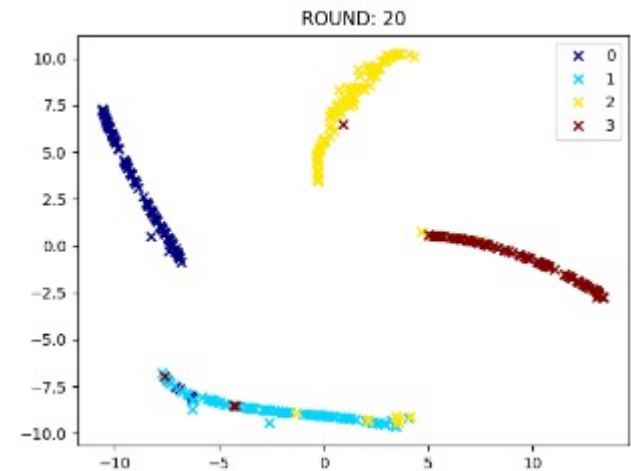
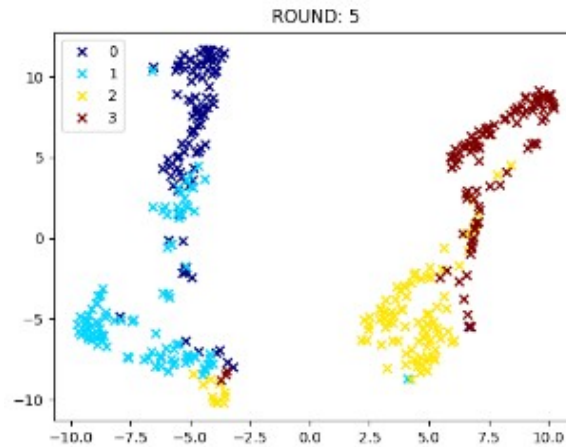
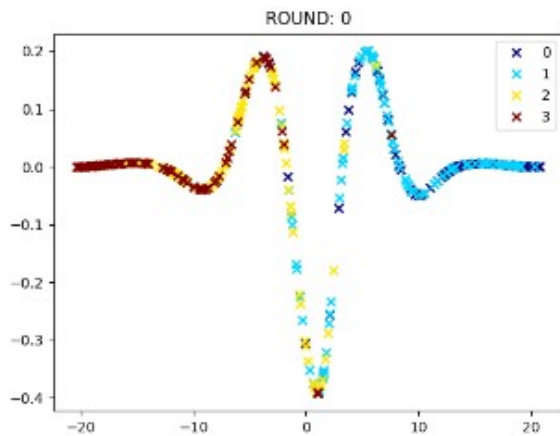
Exemplo

- Modelo SBM com quatro classes, simétricas



- Atributo: amostra de distribuição normal para cada classe

- Projeção em 2D (t-SNE) das representações por rodadas

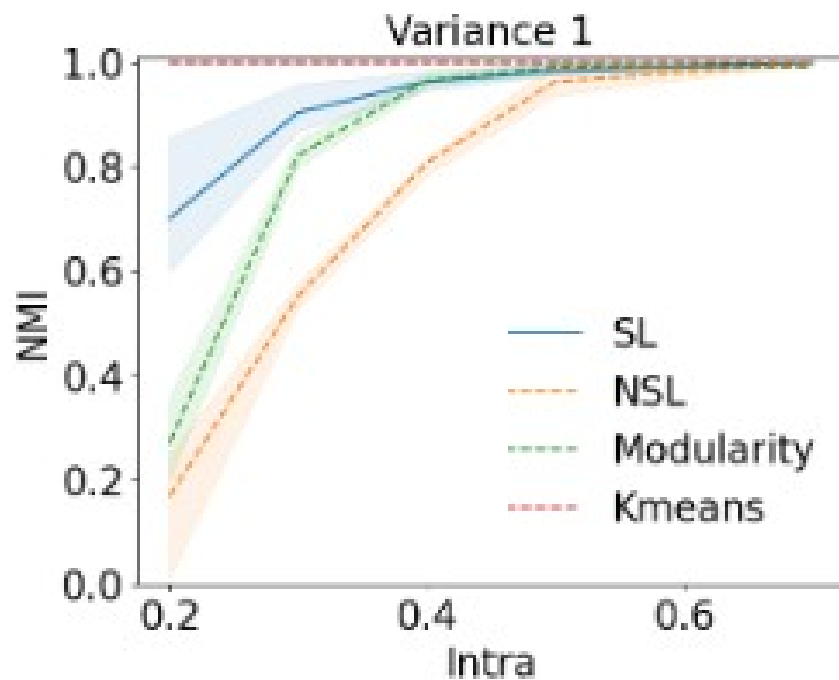


- Clusters melhor indentificados com mais rodadas

Exemplo

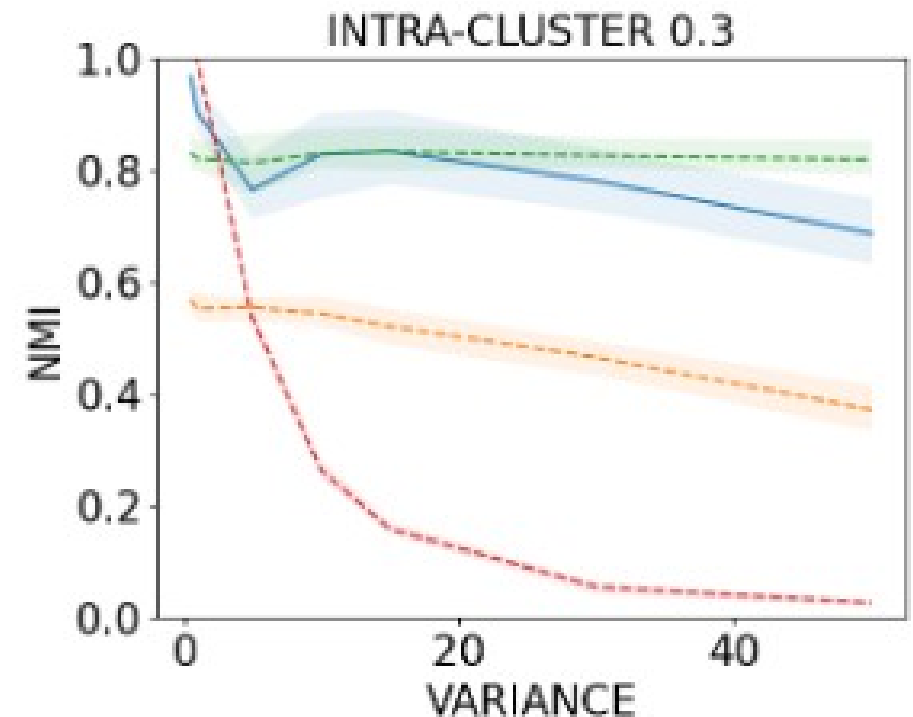
- Variando informação da rede

- prob. intra-cluster



- Variando informação dos atributos

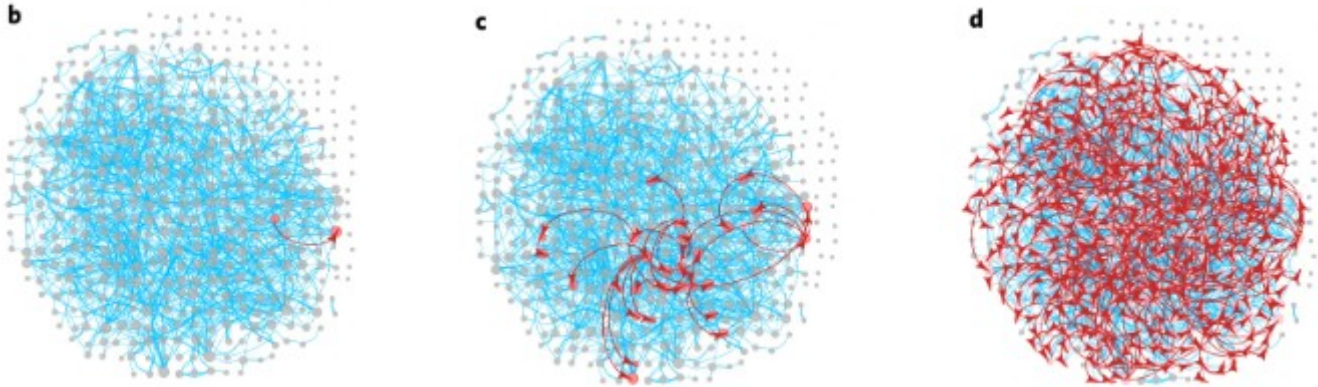
- var. da Gaussiana



- Self-Learning sempre superior a NSL
- SL superior a metodos de rede ou atributo (dependendo dos parâmetros)

Origem das Epidemias

- Epidemia em redes: processo de contágio



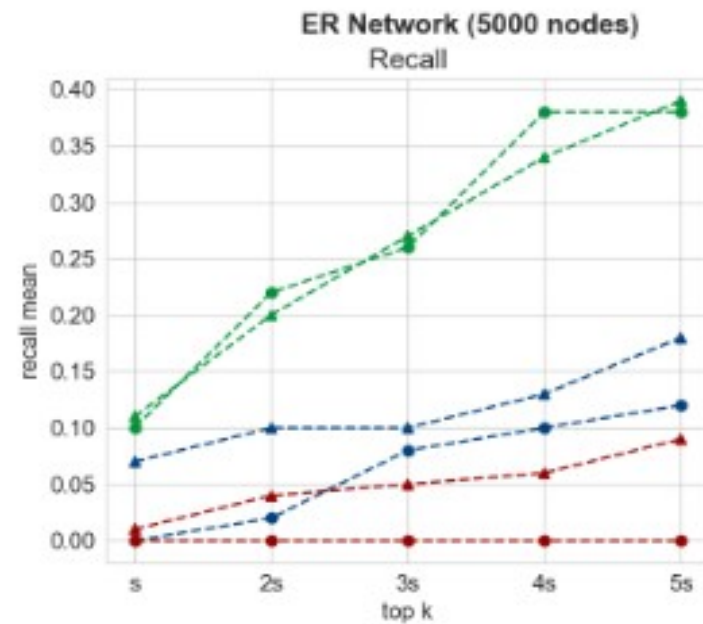
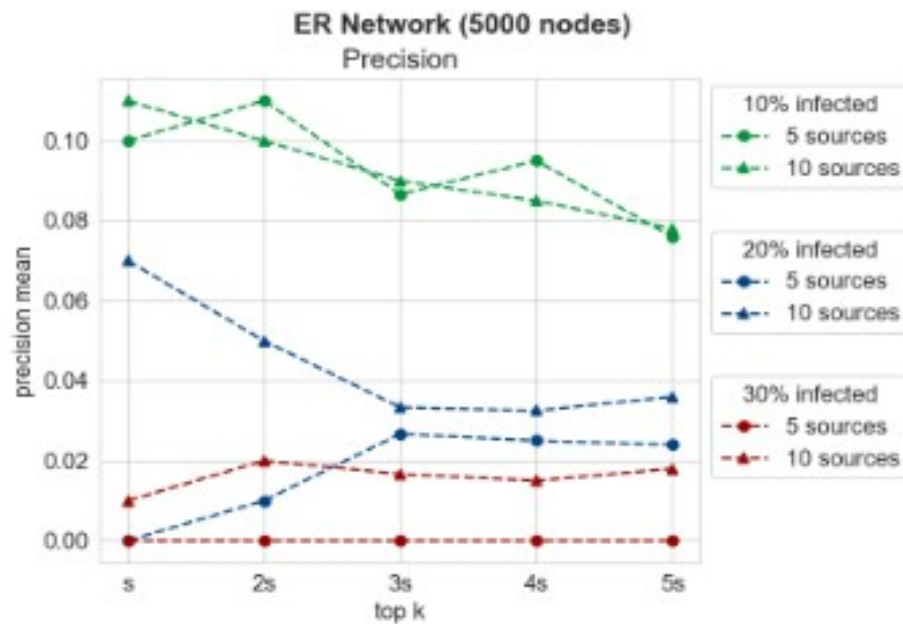
- Epidemia tem vértices de origem
- Problema: identificar vértices de origem ao meio (ou final) da epidemia
 - observação limitada do processo (ex. apenas vértices infectados)
 - problema clássico (e difícil) em epidemia em redes

Origem das Epidemias

- Múltiplos vértices de origem
- Observação apenas dos infectados
- Nenhuma informação sobre processo de contágio
- **Ideia:** construir atributos locais aos vértices em função da observação
 - ex. fração de vértices infectados no anel a distância k
- Usar GNN para classificar vértices (origem ou não)
 - mistura atributos com estrutura da rede
 - modelo supervisionado

Resultados

- Precisão e cobertura no top k^*s (s = número real de origens)



- Mais acertos para observação mais cedo (menos infectados)
- Mais acerto para epidemias com mais origens (mais verdadeiro positivos)

Desafios

- Como gerar a melhor representação para seu problema?
 - artigos mostram que representação gerada por técnica A é melhor que B para problema X
- Como escolher atributos e codificar representação inicial?
 - artigos agregam atributos de forma arbitrária
- Como lidar com grafos que evoluem?
 - vértices novos, vizinhança evoluindo, etc
- Como caracterizar erros e limitações?
 - “erros” na representação em função do modelo de GNN

Conclusão

- Graph Neural Network (GNN)
 - metodologia flexível para gerar representação vetorial para vértices misturando atributos e estrutura da rede
- Muitas variações, muitas aplicações
 - sendo testadas e usadas por grupos de pesquisas

Ferramenta fundamental

- Do saco de ferramentas do cientista de dados

Obrigado!



- Dúvidas, perguntas, ou comentários?

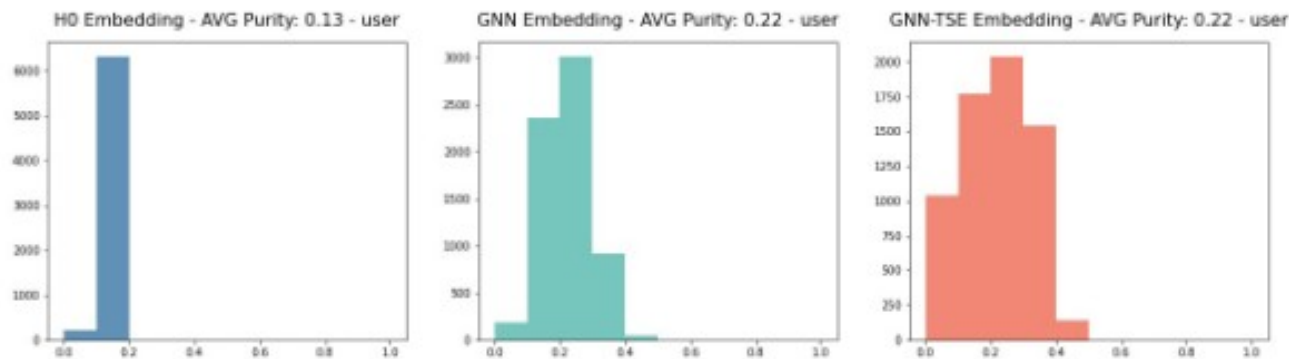
- Contato

- daniel@cos.ufrj.br

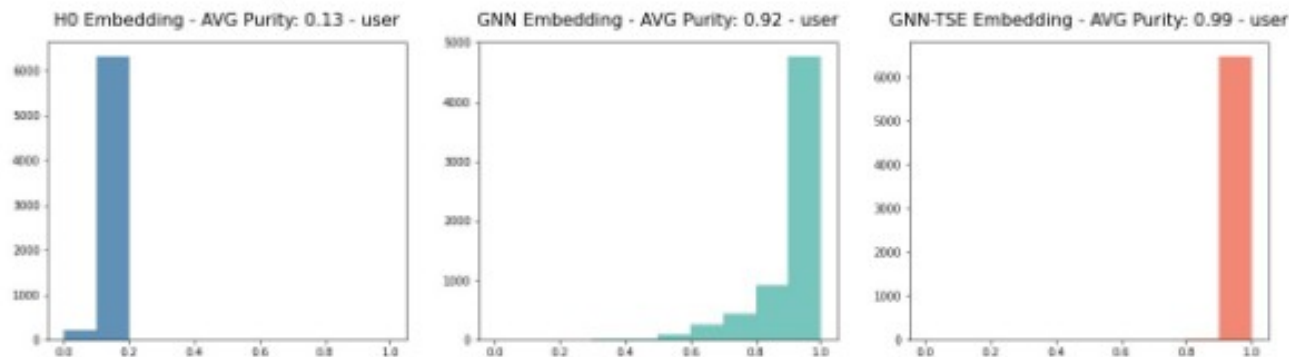
- www.cos.ufrj.br/~daniel

Distribuição da Pureza

- Precisão@100: precisão considerando os 100 pontos mais próximos de cada ponto (vértice)
- distribuição de todas as precisões



(c) Purity distribution for users at epoch 5.

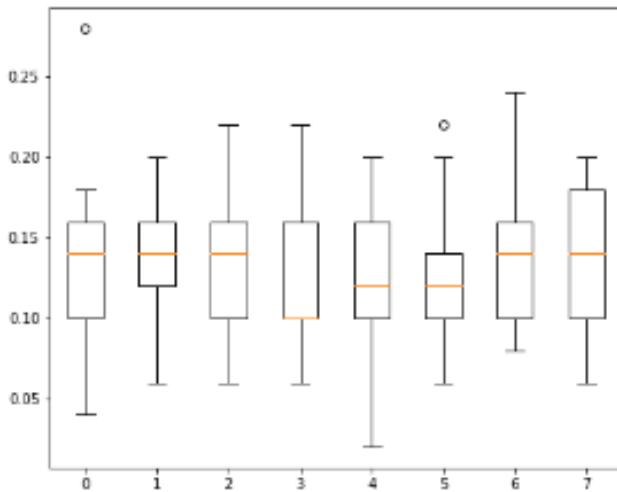


(d) Purity distribution for users at epoch 30.

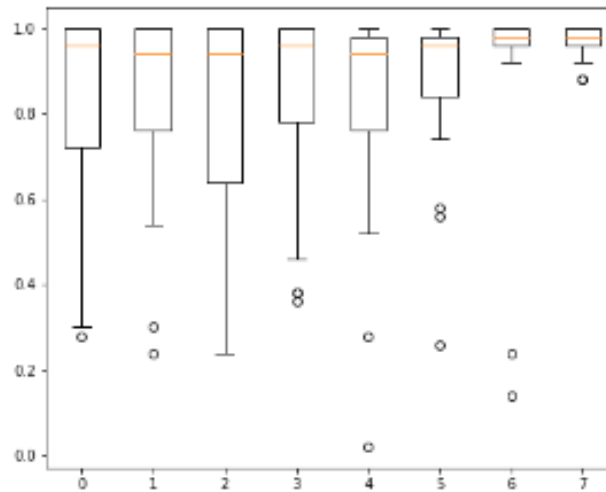
Fairness entre Grupos

■ Precisão@100: por classe, para usuários e itens

Purity distribution by group before training - user

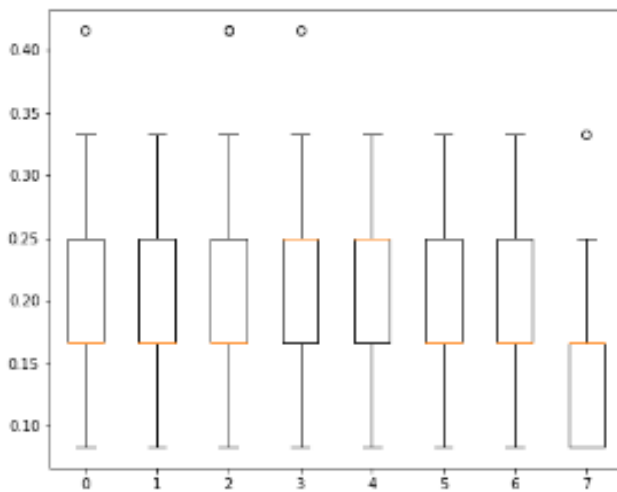


Purity distribution by group after training - user

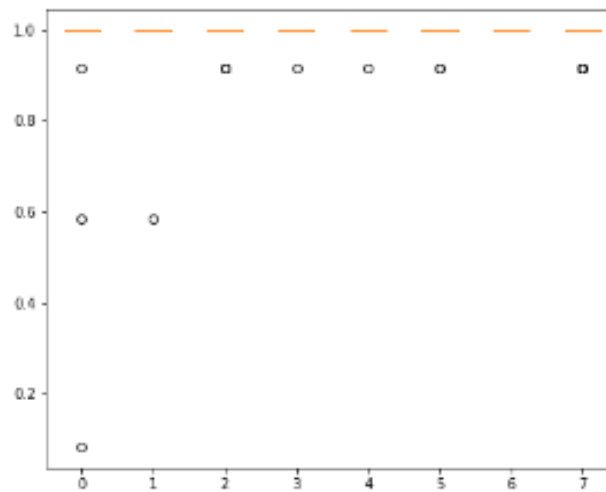


■ Fairness: todos os grupos possuem o mesmo (bom) desempenho (na média)

Purity distribution by group before training - item

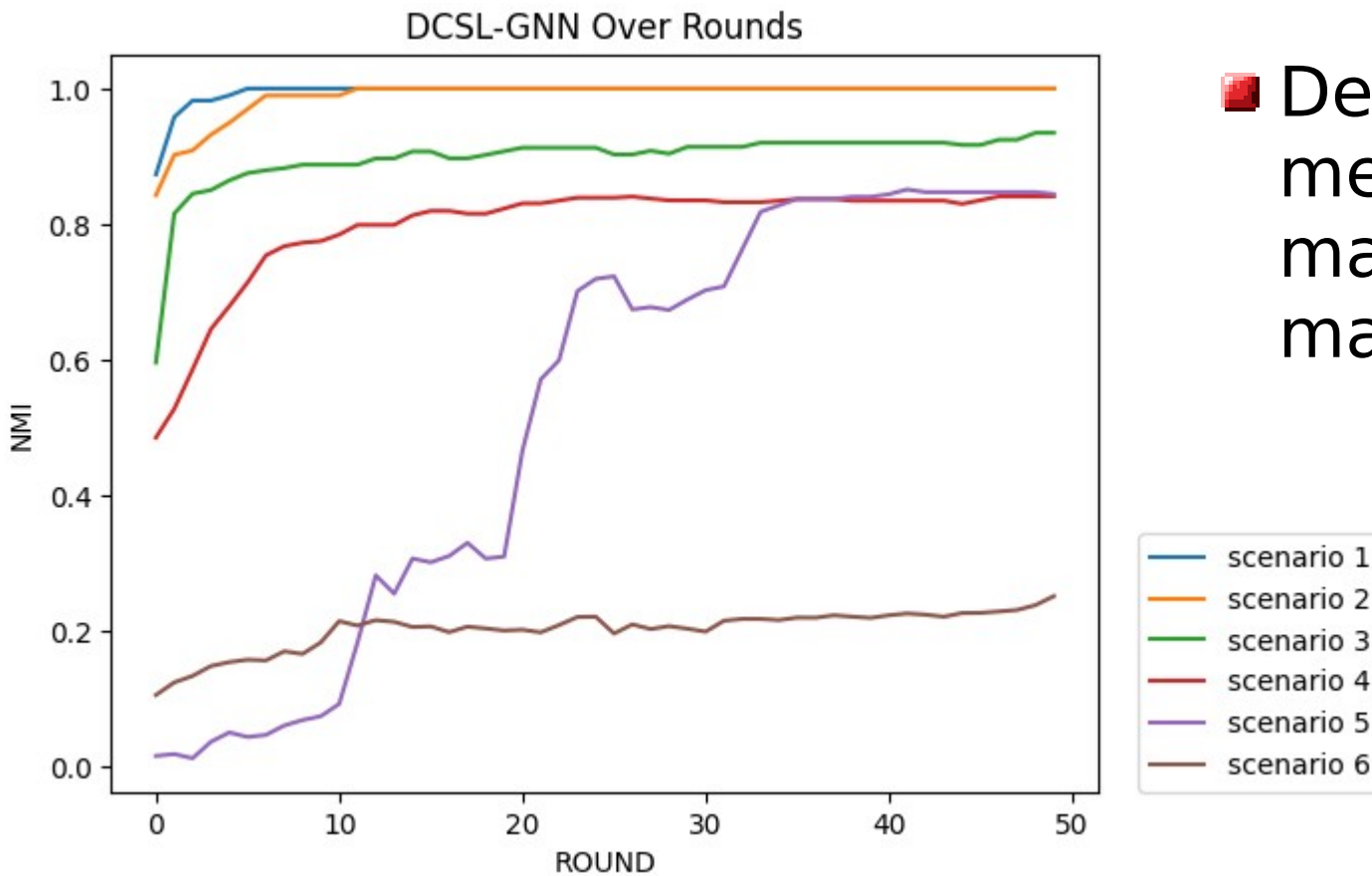


Purity distribution by group after training - item



Clusterização

- Influência das rodadas na acurácia (NMI) para diferentes cenários



- Desempenho melhora, e melhora mais em cenários mais desafiadores

	Attribute Means (μ)	Attribute Variance (σ^2)	Intra-Cluster	inter-cluster
Scenario 1	[10, 20, 30, 40]	1	0.5	0.1
Scenario 2	[10, 20, 30, 40]	50	0.5	0.1
Scenario 3	[10, 20, 30, 40]	1	0.3	0.1
Scenario 4	[10, 20, 30, 40]	50	0.3	0.1
Scenario 5	[10, 20, 30, 40]	1	0.2	0.1
Scenario 6	[10, 20, 30, 40]	50	0.2	0.1

Resultados c/ Vizinhos

- Considerando vizinhos das origens como verdadeiro positivos (na avaliação)
 - desempenho muito superior!

