

APS de Grafos

Equipe:

Higor Luiz F. Celante
Brendow Paolillo Isidoro

2018

Resumo

Este trabalho tem como objetivo demonstrar o estudo de grafos feito com base na PokeAPI. Uma API gratuita que contém todas as informações computáveis sobre o mundo Pokemon. Com o intuito de encontrar, por meio de grafos, qual o tipo de pokemon é mais forte e o mais fraco, qual tipo de pokemon escolher em uma batalha e qual não escolher.

Palavras-chaves: PokeAPI. API. Grafos. Python. Networkx.

Introdução

No mundo do Pokemon, existe uma relação de dano entre os tipos de pokemon, que pode variar entre nenhum dano, meio dano, dano normal e dano duplo. A partir dessa relação é possível decidir qual tipo de pokemon escolher e qual não escolher em uma batalha dado certo oponente.

1 Referencial Teórico

1.1 Grafos

Um grafo $G = (V, A, \phi)$ é um conjunto não vazio de vértices V , um conjunto de arestas A , e um uma função de mapeamento $: A \hookrightarrow P(V)$ que mapeia cada aresta a um par não ordenado formado pelos vértices de G . O número de vértices de G , denotado por $|V|$ indica a ordem de G , enquanto que o número de arestas, denotado por $|A|$, indica o tamanho de G^2 .(PRESTES, 2012)

1.2 Grau de Entrada e de Saída

Em um dígrafo qualquer, cada vértice V possui um grau de entrada, $d^-(v)$, e um grau de saída $d^+(v)$. O grau de entrada de um vértice V é definido pelo número de arcos que possuem como destino V (incluindo o laço) enquanto que o grau de saída é o número de arcos que possuem como origem v (incluindo o laço). Um laço em vértice V é contado uma única vez no grau de entrada de V e uma única vez no grau de saída de V .(PRESTES, 2012)

2 A API

A API PokeAPI(POKEAPI, 2018) disponibiliza sua base de dados em seu *website* de forma totalmente gratuita, podendo consultar a base de dados diretamente pelo site e também fazer o seu *download*. A base de dados da API é toda em Json e contém todas as informações computáveis sobre o universo Pokemon. Seu uso é bastante simples e a documentação é bem completa.

3 Desenvolvimento

3.1 O Problema

A relação de dano dos tipos de Pokemons se dá da seguinte forma:

| DEFENSE → ATTACK ↘ | NOR | FIR | WAT | ELE | GRA | ICE | FIG | POI | GRO | FLY | PSY | BUG | ROC | GHO | DRA | DAR | STE | FAI |
|-----------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| NORMAL | | | | | | | | | | | | | ½ | 0 | | | ½ | |
| FIRE | | ½ | ½ | | 2 | 2 | | | | | | 2 | ½ | | ½ | | 2 | |
| WATER | | 2 | ½ | | ½ | | | | 2 | | | | 2 | | ½ | | | |
| ELECTRIC | | | 2 | ½ | ½ | | | | 0 | 2 | | | | | ½ | | | |
| GRASS | | ½ | 2 | | ½ | | | ½ | 2 | ½ | | ½ | 2 | | ½ | | ½ | |
| ICE | | ½ | ½ | | 2 | ½ | | | 2 | 2 | | | | | 2 | | ½ | |
| FIGHTING | 2 | | | | | 2 | | ½ | | ½ | ½ | ½ | 2 | 0 | | 2 | 2 | ½ |
| POISON | | | | | 2 | | | ½ | ½ | | | | ½ | ½ | | | 0 | 2 |
| GROUND | | 2 | | 2 | ½ | | | 2 | | 0 | | ½ | 2 | | | | 2 | |
| FLYING | | | | ½ | 2 | | 2 | | | | | 2 | ½ | | | | ½ | |
| PSYCHIC | | | | | | | 2 | 2 | | | ½ | | | | 0 | | ½ | |
| BUG | | ½ | | | 2 | | ½ | ½ | ½ | 2 | | | | ½ | 2 | ½ | ½ | ½ |
| ROCK | | 2 | | | | 2 | ½ | | ½ | 2 | | 2 | | | | | ½ | |
| GHOST | 0 | | | | | | | | | | 2 | | | 2 | | ½ | | |
| DRAGON | | | | | | | | | | | | | | | 2 | | ½ | 0 |
| DARK | | | | | | | ½ | | | | 2 | | | 2 | | ½ | | ½ |
| STEEL | | ½ | ½ | ½ | | 2 | | | | | | | 2 | | | | ½ | 2 |
| FAIRY | | ½ | | | | | 2 | ½ | | | | | | | 2 | 2 | ½ | |

Figura 1 – Tabela de Relação de Dano Tipos de Pokemon (POKEMONDB, 2018)

- Onde há relação de dano duplo, a célula está verde.
- Onde há relação de meio dano, a célula está vermelha.
- Onde há relação de dano nulo, a célula está cinza.

3.2 Ambientação

Para a implementação do grafo foi usado o Python 2.7 e importado três bibliotecas:

- Json (leitura da base de dados)
- Matplotlib (plotagem do grafo)
- Networkx (construção do grafo direcionado)

3.3 Instalação

Para instalar as bibliotecas necessárias basta executar o seguinte comando no terminal:

```
$ pip install matplotlib networkx json
```

4 Execução

Para executar o código basta já ter instalado as bibliotecas necessárias e executar o arquivo 'graph.py' seguindo os exemplos:

- Para as relações no qual o elemento elétrico causa **dano duplo**:

```
$ python graph.py electric dd
```

- Para as relações no qual o elemento metal causa **meio dano**:

```
$ python graph.py steel hd
```

- Para as relações no qual o elemento fogo **não causa dano**:

```
$ python graph.py fire nd
```

- Para buscar qual elemento tem mais relações de **dano duplo**, ou seja, qual o elemento **causa maior dano**:

```
$ python graph.py strg
```

- Para buscar qual elemento é o **mais fraco**, ou seja, tem o **menor grau de saída** de dano duplo e ao mesmo tempo o **maior grau de entrada** para dano duplo:

```
$ python graph.py weaker
```

5 Implementação

5.1 Função Strongest

```
def tipomaisforte(g):
    for i in g.nodes: #Percorre os nos
        for j in g.neighbors(i):
            suma +=1
            if suma > max_n: #Seleciona o que causa maior dano
                typeofpkm = i
                max_n = suma
        suma = 0
    print typeofpkm
```

5.2 Função Weaker

```
def tipomaisfraco(g):
    for i in g.nodes: #Percorre os nos
        degreeae = g.degree(i) #Pega o grau de entrada do vertice
        for j in g.neighbors(i):
            suma +=1 #Itera o grau de saida
            if (suma < min_n) and (degreeae > max_n): #Seleciona o mais fraco
                typeofpkm = i
                min_n = suma
        suma = 0
```

6 Plotagem e Resultados

Após executar a chamada corretamente e passar o tipo de pokemon e/ou a relação desejada por parâmetro, ele **imprime no terminal** o resultado e plota o gráfico de todos que têm a relação de dano escolhida da seguinte forma:

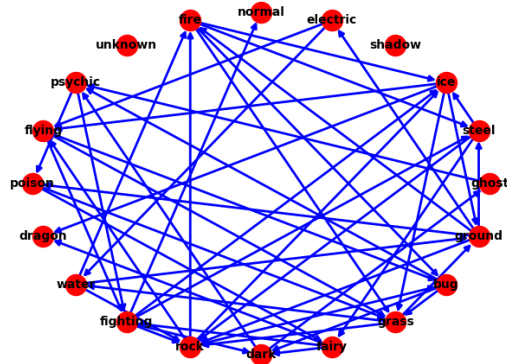


Figura 2 – Grafo de Relação de Dano Duplo

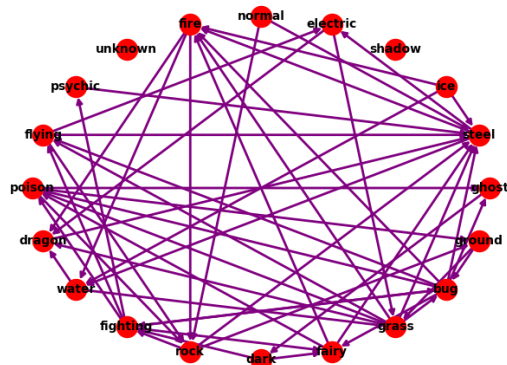


Figura 3 – Grafo de Relação de Meio Dano

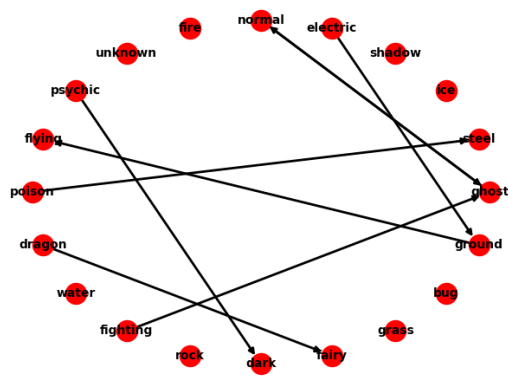


Figura 4 – Grafo de Relação de Dano Nulo

Considerações finais

Com esse estudo concluímos que é possível tirar várias informações relevantes de grandes quantidades de dados aplicando o estudo de grafos e seus métodos. Concluímos também que o tipo de pokemon mais forte é *fighting* e mais fraco *normal*, tendo em vista os resultados obtidos.

Referências

POKEAPI. *PokeAPI Restfull API*. 2018. Site. Disponível em: <www-pokeapi.co/about.html>. Acesso em: 4.12.2018. Citado na página [2](#).

POKEMONDB. *Pokemon Database*. 2018. Site. Acesso em: 4.12.2018. Citado na página [2](#).

PRESTES, E. *Introdução á Teoria dos Grafos*. [S.l.]: Universidade Federal do Rio Grande do Sul, 2012. Citado 2 vezes nas páginas [1](#) e [2](#).