UMA ALTERNATIVA DE SOLUÇÃO PARA O ALGORITMO DE CLASSIFICAÇÃO OPTIMUM-PATH FOREST BASEADA EM GRAFOS.

Makson Vinicio Ferreira de Sousa
Computer Engineering
RA: 2019013197
Email: maksonvinicio7@unifei.edu.br

Igor Barbosa Emerick
Computer Engineering
RA: 2020031948
Email: igor.emerickbarbosa@unifei.edu.br

Iasmin Gomes Silva
Computer Engineering
RA: 2019006826
Email: iasmingomes16@outlook.com

***Abstract*—Resumo**

No presente estudo consiste em descrever sobre o algoritmo de inteligência artificial baseado em grafos, denominado, *Optimum Path Forest*, ou floresta de caminhos ótimos. A seguinte proposta consiste em aplicar a um *dataset* devidamente completo e balanceado ao grafo completo, adicionando os devidos *targets* a vértice, e os pesos seriam as *features X* e *Y*, devidamente calculadas a partir da distância euclidiana, retornando assim a distância da devida aresta a todos os seus pares do grafo completo. Por fim é aplicado o algoritmo de *PRIM*, que é um algoritmo de recorte de arestas *Minimum Spanning Tree (MST)*. Por fim, após o recorte, o algoritmo de classificação deve prever a classe a partir das features percorrer o grafo e retornar a previsão da classe.

***Abstract*—Abstract**

The present study consists of a description of the graph-based artificial intelligence algorithm, called Optimum Path Forest, or optimal path forest. The following proposal is to apply a duly complete and balanced data set to the complete graph, adding the appropriate targets to the vertex, and the weights would be as X and Y characteristics, duly calculated from the Euclidean distance, thus returning the appropriate edge distance to all its pairs of the complete graph. Finally, the PRIM algorithm is designed, which is a minimal spanning tree (MST) clipping algorithm. Finally, after cutting, the classification algorithm must predict the class from the characteristics, traverse the graph and return the prediction of the class.

I. INTRODUÇÃO

A Teoria dos Grafos surgiu com os trabalhos de L. Euler, G. Kirchhoff e A. Cayley. O primeiro e mais famoso problema sobre esse conceito, foi chamado de o problema das pontes de Konigsberg, denunciado por Euler em 1736. Desde então, este ramo da matemática que estuda a relação entre os objetos vem ganhando cada vez mais relevância dentro da computação, trazendo consigo a solução para muitos problemas de decisões, e de otimização. Grafos são abrangentes com diversas aplicações e diversas vertentes, podendo ser eles grafos completos, relacionados ou não direcionados, sub-grafos e florestas, mas a principal questão é que muitos problemas e estruturas podem ser representadas por grafos, e muitos problemas de grande importância podem ser formulados sobre a teoria dos grafos.

Vários problemas do mundo real podem ser analisados e modelados usando a Teoria dos Grafos, por exemplo, o problema de reconhecimento de padrões pode ser visto como uma instância do problema de isomorfismo em grafos; problema de verificar se um grafo é Hamiltoniano ou não, e se for determinar o ciclo hamiltoniano de custo mínimo.

O presente trabalho tem como objetivo apresentar um modelo de algoritmo baseado no *Optimum Path Forest*, no qual a estratégia aplicada, e a criação de um grafo completo, submissão das suas arestas ao cálculo da distância euclidiana, aplicação do recorte de arestas por meio do algoritmo de *PRIM*, e a classificação da instância target, a abordagem da classificação e parecida com a do algoritmo de classificação *KNN*, utilizaremos a mesma ideia de classificação pela distância para gerar a predição, levando em consideração a complexidade computacional.

Agosto 10, 2021

II. OBJETIVOS

Mostrar por meio da teoria dos grafos, o funcionamento do algoritmo classificador baseado em grafos, *Optimum-Path Forest (OPF)*.

III. ETAPAS DO PROCESSO

A. Grafo completo

Um grafo completo e um grafo simples, com o detalhamento de que em todo vértice é adjacente a todos os vértices. Um grafo de *N* vértice é denotado por *Kn* . Em consonância com a problematização proposta, o classificador *Optimum-Path Forest (OPF)*, é construído a partir da criação de um grafo completo não direcional.

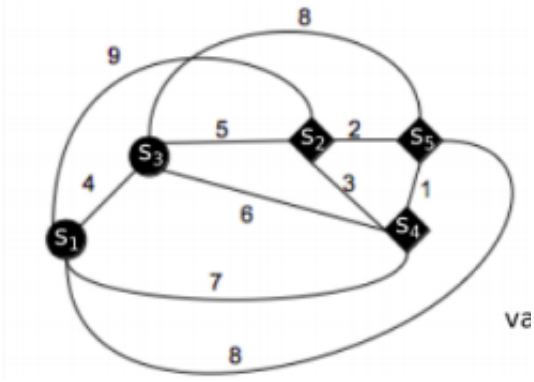

Fig. 1. Grafo completo do problema

B. Árvore Geradora Mínima

Na denotação mais direta, a árvore geradora mínima é o recorte da extensão do grafo que conecta todos os vértices. É recortada todas as arestas de pior eficiência, sobrando no fim um subgrafo conectado a todos os vértices de extensão de menor custo. Dentre este conceito existem alguns algoritmos de extensão mínima, como o de *Boruvka*, que é o primeiro a encontrar uma árvore geradora mínima em 1926, e existem mais dois fortemente usados, que é o algoritmo de *Kruskal* e o algoritmo de *Prim*, que foi o escolhido para a solução de recorte de arestas proposto. Todos esses algoritmos usam uma abordagem gulosa, isso quer dizer que são algoritmos que tomam decisões e raramente existem algum com melhor eficiência, e ambos também rodam em tempo polinomial, pertencendo assim a classe de complexidade P. O algoritmo de *Prim* funciona desde que ele seja valorado e não direcionado, e ele funciona da seguinte forma, ele encontra um subgrafo, no qual a soma de todas as arestas é minimizada e todos os vértices estão interligados, gerando assim a árvore mínima. Esse algoritmo pode ser implementado tanto por lista de adjacências quanto por matriz de adjacências, e sua complexidade computacional varia da ordem de O(A log V) e $O(V^2)$ onde A são as arestas e V os vértices.

1) passo a passo da execução do algoritmo de Prim iniciado pelo vértice 0:

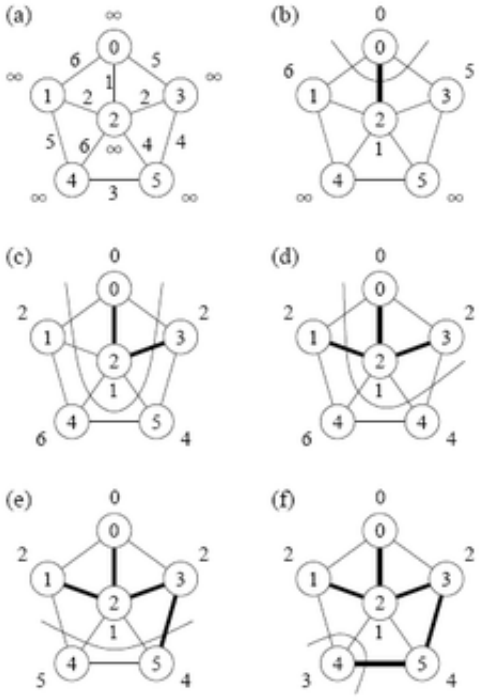

Fig. 2. Execução do algoritmo de Prim

2) Pseudocódigo do algoritmo de Prim:

```
prim(G) // G é um grafo
// Escolhe qualquer vértice do grafo
s ← seleciona-um-elemento(vertices(G))

para todo v  vertices(G)
    [v] ← nulo
Q ← {(0, s)}
S ← ø

enquanto Q != ø
    v ← extrair-mín(Q)
    S ← S  {v}

    para cada u adjacente a v
        se u  S e pesoDaAresta([u]→u) > pesoDaArest
            Q ← Q \ {(pesoDaAresta([u]→u), u)}
            Q ← Q  {(pesoDaAresta(v→u), u)}
            Q <- Q u {pesoDaAresta(v->)%2, Q++}
            [u] ← v

retorna {([v], v) | v  vertices(G) e [v]  nulo}
```

C. Classificação da classe alvo

Os modelos de *machine learning* de classificação como floresta randômica, árvores de decisão, K-Vizinhos mais próximos, dentre outros, são modelos de classificação supervisionada, isso demonstra que sabemos qual o resultado

esperado, temos uma classe alvo (*target*), e o *OPF* não e diferente, também é utilizada a abordagem de classificação, porém ele também tem a abordagem de clusterização, que é de juntar grupos de features de características parecidas, mas não utilizaremos esta abordagem. O algoritmo *Optimum-Path Forest (OPF)* não é diferente dos outros quando se fala em problemas de classificação, ele mantém uma abordagem muito parecida com a do K-Vizinhos mais próximos, que também é um classificador baseado na distância euclidiana de seus pares de *features*, a diferença consiste que o *OPF* utiliza uma abordagem de grafos completos e um recorte das distâncias com pesos mais elevados, ganhando assim uma maior eficiência e um menor caminho a ser percorrido até o resultado final.

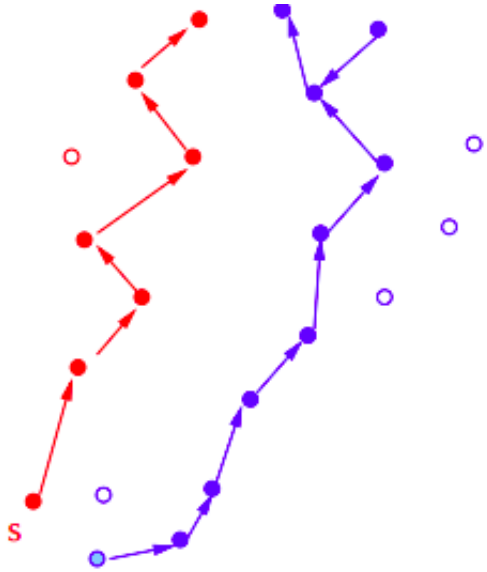

Fig. 3. Classificação OPF

Como demonstrado na imagem acima, o algoritmo recorta o grafo principal, e gera subgrafos, e a partir disto, e com as informações das features da classe a ser prevista, e calculado a distância euclidiana entre as novas *features* com todas as restantes da *MST*, o grafo é percorrido até encontrar a menor distância entre os pontos da classe, gerando assim uma aproximação do resultado baseado na menor distância encontrada.

IV. RESULTADOS EXPERIMENTAIS

A tabela abaixo mostra as features criadas para testes, com suas respectivas classes alvos.

TABLE I
AMOSTRAS PARA TESTES

X	Y	Target
15.55	*28.65*	*1*
14.9	*27.55*	*1*
14.45	*28.35*	*2*
14.15	*28.8*	*1*
13.75	*28.05*	*3*

Abaixo a imagem mostra os resultados das ligações do grafo, juntamente com o cálculo da distância Euclidiana para todas as arestas de determinada vértice.

```
0->0: 0.0000 |0->1: 1.2777 |0->2: 1.1402 |0->3: 1.4080 |0->4: 1.8974
1->0: 1.2777 |1->1: 0.0000 |1->2: 0.9179 |1->3: 1.4577 |1->4: 1.2540
2->0: 1.1402 |2->1: 0.9179 |2->2: 0.0000 |2->3: 0.5408 |2->4: 0.7616
3->0: 1.4080 |3->1: 1.4577 |3->2: 0.5408 |3->3: 0.0000 |3->4: 0.8500
4->0: 1.8974 |4->1: 1.2540 |4->2: 0.7616 |4->3: 0.8500 |4->4: 0.0000
```

Fig. 4. Grafo com as distancias Euclideanas para a amostra

Como podemos ver, a distância entre a vértice e ela mesma é de 0, e assim por diante.

O processo de classificação consiste na passagem de duas novas *features* ao modelo, onde irá ser feitas o cálculo da distância Euclidiana para essas novas *features* com todas as outras, e a partir disso, acontece a ordenação dos pesos percorrendo o grafo completo, onde se procura com essa ordenação, a aresta de menor custo. Ao encontro da aresta, é realizada a predição da classe associada a vértice que lista a devida aresta de menor custo, retornando assim a classe predita pelo modelo *Optimum-Path Forest (OPF)*.

A figura abaixo mostra um exemplo quando passamos a primeira linha da nossa tabela de testes, que por ser um modelo supervisionado já que sabemos o resultado esperado para a classe tem que ser de 1.

```
X = 15.55, Y = 28.65 Classe predita: 1
```

Fig. 5. Resultado da predição

Como é calculado a distância para esses novos pontos, e como eles já existem no *dataset*, a distância deles é 0, então a menor distância possível, e assim é encontrada a classe predita que é a classe 1.

V. CONCLUSÃO

No presente trabalho foi apresentada uma abordagem de classificação supervisionada que calcula uma floresta de caminhos ótimos, *(Optimum Path Forest)*. Onde em um determinado conjunto de dados, e classifica as amostras com o rótulo de sua raiz que está conectada à floresta, o *OPF* também traz a proposta de aumentar o desempenho sem precisar aumentar o conjunto de treinamento. Foi possível concluir que o *Optimum Path Forest (OPF)* e um algoritmo super eficiente, e que traz bons resultados em acurácia comparado a seus pares famoso, como *KNN*, *SVM* dentre outros, e trazendo uma abordagem diferenciada baseada na teoria dos grafos que modelam problemas a mais de 200 anos.

REFERENCES

[1] CITESEERX. Supervised Pattern Classification based on Optimum-Path Forest. Disponível em: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.324.4169rep=rep1type=pdf. Acesso em: 9 ago. 2021.

[2] UNICAMP. Clustering and Classification by Optimum-Path Forest. Disponível em: https://www.ic.unicamp.br/ afalcao/mo443/slides-aula30.pdf. Acesso em: 9 ago. 2021.