Departamento de Computação e Matemática
Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto
Universidade de São Paulo (USP)

Relatório Final (01/08/2022 – 15/09/2022) – Programa Institucional de Bolsas de Iniciação Científica (PIBIC)

Projeto 121814/2021-1

Vigência: 01/09/2021 – 31/08/2022

Classificação Baseada em K-Vizinhos Mais Próximos e no Grafo de Interações Nk

Gustavo Fernandes Carneiro de Castro
(Aluno)

Prof. Dr. Renato Tinós
(Orientador)

Ribeirão Preto, Março de 2021

O relatório está dividido da seguinte forma:

- Seção A: Resumo do Projeto;
- Seção B: Introdução;
- Seção C: Objetivos;
- Seção D: Materiais e Métodos;
- Seção E: Resultados e Discussão;
- Seção F: Conclusões;
- Seção G: Referências;

SEÇÃO A.
RESUMO DO PROJETO

K-Vizinhos mais Próximos (K-Nearest Neighbors – KNN) é um algoritmo simples e intuitivo de classificação não-paramétrica. No KNN, os *K* vizinhos mais próximos são determinados de acordo com a distância ao exemplo a ser classificado. Geralmente é usada a distância Euclidiana, que acaba por facilitar a formação de agrupamentos hiper–episódicos. Neste projeto, propõe-se o uso do grafo de interações Nk para retornar os *K* vizinhos mais próximos no algoritmo KNN. O grafo de interações Nk, originalmente empregado em clusterização, é construído com base na distância e densidade espacial de objetos em pequenos grupos formados por *k* objetos. Ao usar a distância aliada à densidade espacial, possibilita-se a formação de aglomerados com formatos arbitrários, diferentes dos aglomerados hiper-elipsóides gerados pelo KNN original, devido ao uso de uma única métrica espacial, a distância euclidiana. Duas variações do método serão investigadas; os algoritmos diferem na forma em que os vértices associados aos *N* objetos da base de treinamento são visitados. Os $K = k$ objetos relacionados aos vértices visitados são retornados como vizinhos mais próximos. As variações propostas serão comparadas entre si e com o KNN original em experimentos com bases de dados com propriedades diversas para se observar a eficiência deste método de classificação original, apelidado de KNN modificado.

SEÇÃO B.
INTRODUÇÃO

Este é o relatório final do projeto de iniciação científica PIBIC de número 121814/2021-1 em desenvolvimento pelo bolsista Gustavo Fernandes Carneiro de Castro.

Aprendizado de máquina é frequentemente aplicado em problemas cujo principal objetivo é descobrir a relação entre os múltiplos exemplos de uma base de dados. Caso estes exemplos possuam uma classe, predeterminando os grupos que estas se encontram, o aprendizado é considerado supervisionado [KOTSIANTIS et al, 2007]. Um dos mais importantes problemas de aprendizado supervisionado é classificação, um pré-requisito para diversas tecnologias presentes no nosso dia-a-dia, como reconhecimento de fala, identificação biométrica, visão computacional, sistemas de recomendação, entre outros.

Os algoritmos utilizados para classificação são normalmente empregados utilizando métricas de distância, especialmente a Euclidiana. Essa característica comum entre a maioria dos algoritmos facilita a formação de aglomerados hiper-elipsóides de objetos com uma mesma classe. Com isso, objetos pertencentes a aglomerados arbitrários, que não seguem este padrão de organização, definidos tanto pela distância como pela densidade espacial [RODRIGUEZ & LAIO, 2014], [ESTER et al., 1996], acabam por não serem classificados corretamente por estes algoritmos.

O algoritmo de classificação dos K-vizinhos mais próximos (K-Nearest Neighbors Algorithm - KNN) é um destes algoritmos de base Euclidiana: um novo objeto é classificado com o rótulo da classe majoritária entre os *K* vizinhos mais próximos ao objeto, determinados de acordo com a distância até o novo objeto. O algoritmo KNN é simples, intuitivo e um dos primeiros métodos eficientes de classificação não-paramétrica [FIX, 1985], o que o torna uma ótima base para diversos outros algoritmos de classificação, regressão e, com algumas modificações, para outros problemas de aprendizado de máquina.

Aproveitando das vantagens do KNN, mas tentando solucionar os problemas advindos do uso de apenas uma única métrica (distância) para definição dos vizinhos próximos, propôs-se aqui o KNN modificado. Também são propostas duas variações do método, que se diferem na forma cujos vértices associados com os exemplos da base de dados são visitados. Este é um algoritmo baseado no grafo de interações Nk, que é formado a partir de duas métricas: a

distância Euclidiana e a densidade espacial. O grafo de interações Nk consiste de *N* vértices, cada um para um dos *N* objetos na base de dados. Cada vértice é ligado a *k* objetos por meio de arestas definidas por densidade espacial e distância Euclidiana. Este grafo foi proposto inicialmente no NK Hybrid Genetic Algorithm - NKGA [TINÓS et al., 2018], utilizado para problema de *clustering*. Em [MORAES & TINÓS, 2020], o grafo de interações Nk foi utilizado para o problema de busca por similaridade. Basicamente, o método proposto em [MORAES & TINÓS, 2020] retorna K objetos similares ao objeto consultado visitando $k = K$ vértices do grafo de interações Nk, vizinhos ao objeto a ser classificado. O método proposto se mostrou interessante para a consulta de objetos em bases de dados com aglomerados de formato arbitrário.

Nos métodos previamente propostos para clusterização [TINÓS et al., 2018] e busca por similaridade [MORAES & TINÓS, 2020], para $k \geq 1$, apenas uma aresta de cada vértice do grafo de interações Nk era conectada baseada na densidade espacial de se seu respectivo exemplo na base de treinamento. Todas as outras eram baseadas unicamente na distância entre eles. Nesta etapa do projeto, para o KNN modificado, foi proposta a mudança da razão α entre o número de arestas definidas por densidade espacial e distancia durante a criação do grafo de interações Nk.

SEÇÃO C.
OBJETIVOS

Na Seção B, cita-se três observações relevantes: I) a existência de agrupamentos de dados em formatos diferentes de hiper-elipsóides, definidos não somente pela distância mas também pela densidade espacial; II) a simplicidade e eficiência do método KNN para o problema de classificação; III) e o processo de construção do grafo Nk ser intuitivo, de fácil alteração e o grafo ser fácil de se observar, ler e ser construído a partir da distância e densidade.

Dadas estas observações, pode-se então definir o objetivo principal do projeto: Investigar estratégias eficientes para a utilização do grafo de interações Nk para retornar os K objetos da base de treinamento que serão utilizados para a rotulação do novo objeto em KNN. Outros objetivos secundários são: Explorar as possibilidades oferecidas pela fusão de KNN e das propriedades do grafo de interações Nk; Modificar a maneira em que o grafo de interações Nk é construído; Comparar variantes do KNN modificado com o KNN original em bases de dados, com propriedades distintas, de repositórios públicos para diferentes valores de K e α; Comparar as diferenças entre os algoritmos ao variar separadamente os parâmetros que definem o número de arestas K a razão α entre as arestas definidas por densidade e por distância; Comparação entre a densidade e a distância perante a precisão na classificação das diferentes bases de dados; e a capacitação do aluno nas áreas de Inteligência Artificial, Aprendizado de Máquina, Teoria dos Grafos e Classificação.

SEÇÃO D.
MATERIAIS E MÉTODOS

Essa seção está dividida no seguinte modo:

D.1. K-Vizinhos Mais Próximos (KNN)

D.2. Grafo de Interações Nk;

D.3. KNN baseado no Grafo de Interações Nk;

D.4. Bases de dados;

D.5. Implementação;

D.1. K-Vizinhos Mais Próximos (KNN)

O KNN é um algoritmo de classificação no qual um novo objeto *x* é rotulado examinando-se a classe majoritária entre os *K* objetos da base de treinamento mais próximos a x [AHA et al., 1991]. Estes *K* vizinhos são selecionados a partir da distância, geralmente Euclidiana, do objeto *x* a ser classificado. O KNN pode ser modificado para regressão; também é possível atribuir pesos aos *K* vizinhos analisados [ALTMAN, 1992]; e também existe uma abordagem de área, advinda da análise de uma hiper-esfera onde o parâmetro *K* é seu raio. Para qualquer abordagem ou modificação, o parâmetro *K* possui um impacto significativo no resultado, desempenho e na definição das regiões de decisão do classificador.

D.2. Grafo de Interações Nk

O grafo de interações Nk é gerado para armazenar as informações dos grupos de objetos gerados pelo algoritmo NKGA [TINÓS et al., 2018]. Este algoritmo utiliza tanto a distância entre objetos como a densidade espacial para criar *N* pequenos grupos, cada um com *k* objetos, com o objetivo de agrupar exemplos de uma base de dados de tamanho *N*.

O grafo de interações Nk é um grafo direcionado com *N* vértices, cada um com grau de saída *k*. Cada vértice é associado a um exemplo do conjunto de treinamento.

Originalmente, cada vértice do grafo tem um auto-loop, uma aresta de saída definida por densidade espacial e *k*-2 arestas de saída definidas pela distancia Euclidiana entre os exemplos da base de treinamento. Nesta etapa do projeto, foi proposta a utilização de um parâmetro α que especifica a razão das arestas definidas por densidade. Sabe-se que $A=\lceil \alpha k \rceil$, onde *A* é um número inteiro que representa a quantidade de arestas de saída definidas por densidade espacial, para cada vértice. Como dito previamente, no grafo Nk original, α sempre leva a $A=1$ e, portanto, sempre haverá um auto-loop, uma aresta definida por densidade e *k*-2 arestas definidas por distância, para $k \geq 2$. Para $k = 1$, sempre haverá apenas o auto-loop e, portanto, $A=0$. Aqui é proposto usar α tal que $A \neq 1$, mudando o número de arestas definidas por ambas densidade espacial e distância Euclidiana. Cada aresta (v_j, v_i) do grafo indica que o *j*-ésimo objeto é relacionado com o *i*-ésimo objeto. A densidade espacial ρ_i para o *i*-ésimo exemplo (y_i) da base de dados com *N* exemplos é dada por:

$$\rho_i = \sum_{j=i}^{N} \mathbf{K}(y_i - y_j) \tag{1}$$

onde **K** é a função kernel, aqui definida por:

$$\mathbf{K}(y_i - y_j) = e^{\frac{-\left\|y_i - y_j\right\|^2}{2\epsilon^2}} \tag{2}$$

Onde ϵ é o parâmetro que define a distância de corte. Neste trabalho, esse parâmetro é igual a 2%, como sugerido em [RODRIGUEZ & LAIO, 2014].

Para a construção do grafo de interações Nk, dada uma base de dados com *N* exemplos, primeiro um vértice v_i com um auto-loop é adicionado para cada exemplo y_i da base de dados. Segundo, as *k*-1 arestas remanescentes são definidas tanto por densidade espacial quanto por distância Euclidiana, enquanto sua razão é definida por um parâmetro α. Esse parâmetro define a porcentagem de arestas definidas por densidade espacial, conectando-as aos exemplos mais próximos com a densidade maior que y_i. O número de arestas conectadas dessa for é igual a *A*, sendo $A=\lceil \alpha k \rceil$, e $A < k$ (uma das arestas deve ser o auto-loop). Por último, as arestas $k - A - 1$ restantes são conectadas por distância, ao vértice dos exemplos mais próximos a y_i. A Figura D.2.1 mostra um exemplo para a construção do grafo de interações Nk com $k = 2$, $\alpha = 1/3$ e $N = 7$.

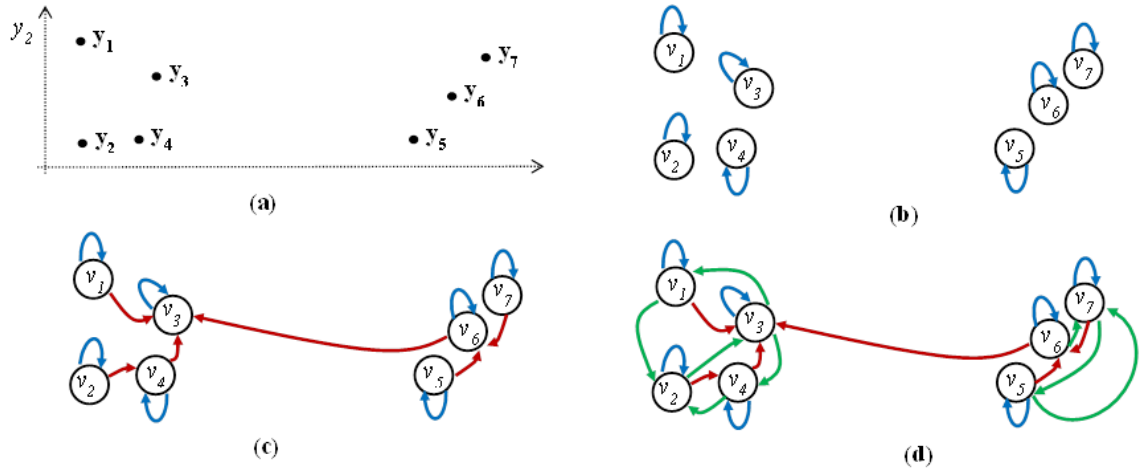

Figura D.2.1: Exemplo de construção do grafo de interações Nk com k = 3, α = 1/3 e N = 7 exemplos bidimensionais. Cada objeto da base de dados (a) é associado com um vértice com auto-loop (b). A densidade dos objetos é calculada e cada vértice é ligado a $A=\lceil \alpha k \rceil$ objetos mais próximos com maior densidade que si (c). Então, os vértices restantes são conectados aos vértices mais próximos, resultando no grafo de interações (d) com N = 7 vértices e Nk arestas.

D.3. KNN baseado no Grafo de Interações Nk

O KNN modificado utiliza do grafo de interações Nk para retornar os K vizinhos mais próximos a um novo exemplo x (a ser classificado). São utilizadas duas variações do método, que se diferem na forma em que os vértices associados aos N objetos da base de treinamento são visitados. Em ambas as variações: dado N exemplos da base de treinamento e os parâmetros K e α, o grafo de interações é montado (conforme a seção D.2.); o parâmetro k é igual a K, ou seja, o grau de saída para cada vértice (k) é igual ao número de vizinhos mais próximos (K) do KNN; para cada novo exemplo (a ser classificado) x, a densidade espacial de x (considerando todos os exemplos da base de treinamento) e distâncias de x a todos os exemplos da base de treinamento são computados; dado um exemplo x, os K vértices visitados (como explicado no próximo parágrafo) definem os K vizinhos mais próximos de x; dado os K vizinhos mais próximos de x (definidos utilizando o grafo de interações Nk), a classificação é realizada como no KNN original, ou seja, o KNN baseado no grafo de interações Nk difere do KNN original apenas na forma na qual os K vizinhos mais próximos são definidos.

Para classificar um novo exemplo x, o vértice v_x relacionado ao exemplo (de acordo com a distância Euclidiana) da base de treinamento mais próximo a x é escolhido dessa base para representar x no grafo de interações Nk. Na primeira variação do algoritmo, chamada de *KNN modificado tipo A*, a lista de adjacência de v_x é obtida ao utilizar a densidade espacial e

distancia de x aos exemplos da base de dados. Então os k=K exemplos associados aos vértices na lista de adjacências de v_x são tidos como os vizinhos mais próximos e, portanto, utilizados para classificá-lo.

A segunda variação, chamada de *KNN modificado tipo B*, consiste e, encontrar e salvar o vértice v_j, da lista de adjacência de v_x, cujo exemplo y_i é o mais próximo (de acordo com a distância Euclidiana) a x, ignorando o auto-loop. Então a operação $v_x = v_j$ é realizada, e esse mesmo passo é repetido, totalizando k vezes. A lista de vértices v_j gerada por este processo é tida como os vizinhos mais próximos de x e, portanto, utilizada para classificá-lo. A Figura D.3.1 exemplifica a diferença dos *KNN modificados tipos A* e *B*.

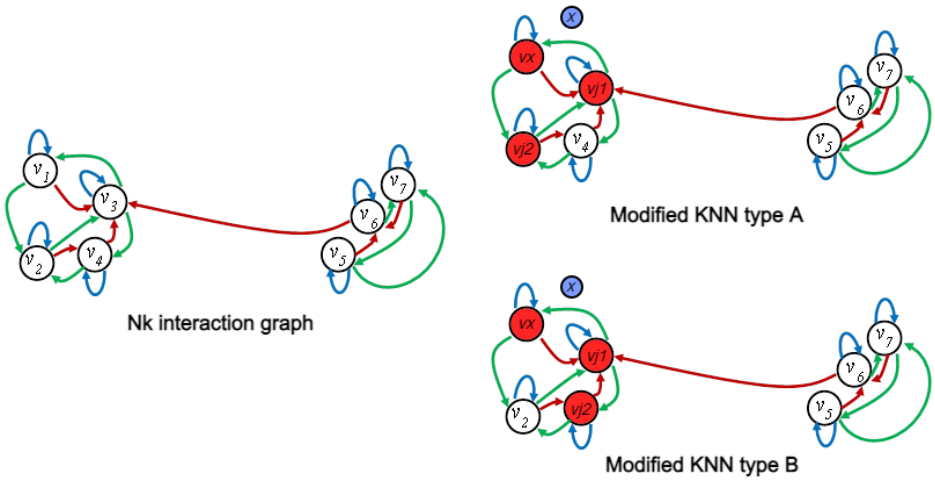

Figura D.3.1: Exemplos de classificação utilizando os KNN modificados tipo A e B, ao classificar um exemplo x. No KNN modificado tipo A, os vizinhos são definidos pela lista de adjacência de v_x. No KNN modificado tipo B, os vizinhos são definidos pelo vértice adjacente a v_x relacionado ao exemplo mais próximo a x, seguido por $v_j = v_x$ e iterativamente repetindo o processo.

D.4. Bases de dados

Para o primeiro experimento, realizado antes do relatório parcial, foram utilizadas um conjunto de 11 bases de dados advindas dos repositórios públicos UCI [DUA & GRAFF, 2019] e Shape [FRÄNTI & SIERANOJA, 2018], com características distintas que permitem analisar as propriedades do KNN modificado em diversas situações de aglomeração e disposição dos dados. São estas bases *Aggregation*, *Compound*, *D31*, *ecoli*, *flame*, *ionosphere*, *iris*, *jain*, *pathbased*, *R15* e *spiral*. Algumas destas bases foram utilizadas no projeto de mestrado de José Carlos Bueno de Moraes no Programa de Computação Aplicada [MORAES, 2020], mestrado no qual este projeto é derivado.

Para o segundo experimento, realizado após o relatório parcial, foram utilizadas mais 4 bases de dados S-sets *s1*, *s2*, *s3* e *s4* [FRÄNTI & SIERANOJA, 2018], para complementar os testes realizados para analisar as variações no parâmetro α.

D.5. Implementação

Os programas para a implementação das duas variações do KNN modificado e a adição do parâmetro α foram implementadas em Python. Diversas bibliotecas estão sendo utilizadas para auxiliar na manipulação dos dados, criação do grafo Nk, na implementação dos algoritmos, na classificação e nos testes.

Estre essas bibliotecas, podemos citar: *Pandas*, focado na manipulação de dados e criação de um tipo de estrutura de dado chamada *Data Frame*; *Scikit-learn*, uma biblioteca de aprendizado de máquina, com ferramentas para a aplicação de algoritmos, testes de eficácia, validação cruzada, entre outros; *SciPy* e *Numpy*, bibliotecas com diversos algoritmos e estruturas próprias para computação científica; e *NetworkX*, focada na criação, construção e manipulação de grafos.

Todas estas foram utilizadas como ferramentas para auxiliar na construção do algoritmo, desde a criação do grafo, até a criação do KNN e dos KNN modificados (do tipo A e B), até a aplicação dos métodos de verificação dos resultados, eficiência e eficácia.

SEÇÃO E.
RESULTADOS E DISCUSSÃO

As atividades realizadas estão divididas no seguinte modo:

E.1. Leitura e manipulação dos dados e bases de dados;

E.2. Criação do grafo de interações Nk;

E.3. Implementação dos algoritmos KNN e KNN modificado tipo A e tipo B;

E.4. Implementação da validação cruzada para os algoritmos de classificação;

E.5. Automação dos testes e impressão dos resultados;

E.6. Adição do parâmetro α aos KNN modificados;

E.7. Análise de resultados;

De acordo com o cronograma proposto, houve levantamento bibliográfico, estudo das referências relacionadas ao tema (grafos, grafo de interações Nk, algoritmos de classificação, KNN, validação cruzada).

E.1. Leitura e manipulação dos dados e bases de dados

As bases de dados utilizadas nas análises foram lidas e armazenadas em estruturas de dados chamadas *Data Frames*, disponibilizadas pela biblioteca *Pandas*, utilizando um método desenvolvido em Python. Após a leitura e armazenamento, as manipulações realizadas no decorrer do código foram feitas utilizando dos *Data Frames* e das bibliotecas *Numpy* e *SciPy*, utilizando estruturas e algoritmos disponibilizados por estas.

E.2. Criação do grafo de interações Nk

Após a leitura, foi desenvolvido um método para a criação do grafo de interações Nk, que recebe o *Data Frame* com os dados; a distância dentre todos os objetos da base de dados; a densidade dos objetos; e o valor de *k* que representa a quantidade de conexões do grafo de interações. Para a aquisição da distância e da densidade, foram desenvolvidos rotinas e métodos próprios para tal, com auxílio das bibliotecas de computação científica. Para o armazenamento do grafo, e futuras manipulações nele, foi utilizada a biblioteca *NetworkX*.

A criação do grafo de interações segue a lógica explicada previamente em **D.2**. Um código em *C++*, utilizado para a criação do grafo em [TINÓS et al., 2018], também foi usado de base para a construção deste.

E.3. Implementação dos algoritmos KNN e KNN modificado tipo A e tipo B

Depois, foram implementados os algoritmos de classificação KNN (seção **D.1.**) e KNN modificado dos tipos A e B (seção **D.3.**). Ambos recebem um novo objeto, e a partir da distância, para o KNN, e do grafo, para o KNN modificado, o classificam conforme as classes dadas pela base de dados utilizada no treinamento do algoritmo.

Os métodos possuem parâmetros parecidos: o objeto *x* a ser classificado, o valor de *K* (ou *k*) para a quantidade de vizinhos ou conexões utilizadas na classificação, o *Data Frame* com os dados, e um *Data Frame* com a classificação dos objetos da base, além de dois parâmetros utilizados para ajustar a impressão dos dados. A diferença consiste no parâmetro da distância entre os objetos da base de dados analisada, para o KNN, e no parâmetro do grafo de interações Nk da base analisada, para os KNN modificados. Todos os métodos retornam a classificação de *x*.

E.4. Implementação da validação cruzada para os algoritmos de classificação;

Para confirmar a eficiência e eficácia dos algoritmos, é necessário aplicar um método de verificação. O método escolhido foi a validação cruzada, um método que consiste em: dividir a base de dados em *p* partes, pegar uma destas e utilizar de teste para as outras, utilizadas como treino para o algoritmo escolhido; realizar esta etapa *p* vezes, até todas as partes serem utilizadas de teste para o algoritmo.

Dois métodos foram criados para aplicar a validação cruzada nos algoritmos. Um para o KNN, e outro para o KNN modificado. Os parâmetros de ambos são parecidos: *K* (ou *k*) como o número de vizinhos ou conexões que serão utilizadas na classificação; *p* como o número de partes para a validação; e dois parâmetros utilizados para ajustar a impressão dos dados. Porém o método para os KNN modificados possui um parâmetro utilizado para escolher qual tipo de algoritmo será utilizado: o A ou o B. Ambos os métodos retornam a matriz de confusão dos resultados obtidos.

A matriz de confusão é uma estrutura que permite a visualização do desempenho do algoritmo. Ela mostra a taxa de acertos e erros para cada uma das classes, além da proporção de cada tipo de erro.

E.5. Automação dos testes e impressão dos resultados;

Para realizar os testes, um script Python foi desenvolvido, que itera sobre todas as bases de dados, realiza a execução de todos os métodos de validação cruzada e armazena todos os resultados para um intervalo de valores para *K*. Este script utiliza dos métodos citados previamente, pertencentes a objetos do tipo NKGraph. Estes objetos representam uma base de dados, e possuem os *Data Frames* com os dados, as classes, e possuem as distâncias e densidades armazenadas.

E.6. Adição do parâmetro α aos KNN modificados;

Posteriormente o parâmetro α foi adicionado. Este parâmetro, presente em ambos KNN modificados, recebe uma porcentagem que define a razão de arestas definidas apenas por densidade espacial e, portanto, define o número *A* de arestas definidas por densidade. Sabe-se que $A=\lceil \alpha k \rceil$, e $A < k$, por conta do auto-loop.

Para tal adição, o método de criação do grafo de interações sofreu pequenas modificações, seguido dos métodos e scripts dos KNN modificados, de validação cruzada, de automação dos testes, etc., permitindo-os receber tal parâmetro, ajeitar a criação do grafo, impressão correta dos dados, etc.

E.7. Análise de resultados;

No primeiro experimento, o desempenho dos algoritmos foi testado para variações no parâmetro *K* no intervalo [1, 10]. Nas variações propostas do KNN, α foi definido para manter $A=1$, ou seja, apenas uma aresta de saída para cada vértice é definida por densidade espacial. No segundo experimento, foi testado o impacto da variação do parâmetro α. Neste caso, os resultados foram gerados para dois valores do parâmetro *K* (5 e 10). Para $K = 5$, o parâmetro α varia resultando em *A* no intervalo [0, 4]; para $K = 10$, o parâmetro α varia resultando em *A* no intervalo [0, 9]. No segundo experimento, o KNN original é testado apenas uma única vez para cada base de dados e *K* respectivo, por não possuir o parâmetro α.

Nas Tabelas, os resultados destacados com um fundo cinza escuro são os melhores resultados gerais da respectiva base de dados, enquanto os resultados destacados com um fundo cinza claro representam os melhores resultados de sua respectiva linha (para um dado valor *K* ou α, dependendo do experimento)

A Tabela **E.7.1.** mostra os resultados do primeiro experimento, desenvolvido para testar os impactos de mudança do parâmetro *K*. Os resultados indicam que, num geral, melhor desempenho é obtido pelo KNN modificado tipo B nas bases de dados do UCI Machine Learning Repository (*iris, ecoli* e *ionosphere*). Estes são as únicas bases de dados com dimensões maiores que dois. Também é possível de se observar que o KNN tipo A e o KNN original apresentam resultados similares para valores maiores de *K*. Isso é explicado pelo pequeno número de vizinhos definidos pela densidade (*A*=1). Também é possível de se observar que, na maior parte das bases de dados com agrupamentos hiper-elipsoidais, o KNN modificado tipo A e o KNN original apresentaram melhor precisão; a exceção é para as bases com muitos agrupamentos com sobreposição (D31), onde os melhores resultados foram obtidos pelo KNN tipo B.

As Tabelas **E.7.2.** e **E.7.3.** mostram os resultados do segundo experimento, desenvolvido para testar o impacto de mudança do parâmetro α. A Tabela **E.7.2.** mostra os resultados para *K*=5, enquanto a Tabela **E.7.3.** mostra os resultados para *K*=10. É possível de se observar que o KNN modificado tipo B obteve os melhores resultados gerais para as bases de dados *D31, ionosphere* e *flame* na Tabela **E.7.2.**, e nas bases de dados *D31, s3* e *s4* na Tabela **E.7.3.** Os resultados dos experimentos indicam que o pior desempenho foi obtido, na maioria das bases, para maiores valores de α, ou seja, quando todos ou a maioria dos *K* vizinhos são definidos por densidade espacial. Também é possível de se observar que os melhores resultados foram obtidos para maiores valores de α em bases com agrupamentos em sobreposição (*D31, s3* e *s4*). O grau de sobreposição aumenta gradativamente do *s1* até o *s4* nas bases S-set, sendo possível de se observar que melhores resultados são obtidos para o *s3* e *s4* para maiores valores de α. Ou seja, escolher mais vizinhos por densidade espacial no KNN modificado impacta positivamente o desempenho ao se aumentar o grau de sobreposição de agrupamentos. Finalmente, é possível de se observar que o KNN modificado tipo A se comporta como o KNN original para $A = 0$.

Tabela E.7.1. Resultados do Experimento 1.

Aggregation							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	787	0.999	787	0.999	787	0.999	788
2	783	0.994	783	0.994	**786**	**0.997**	
3	**786**	**0.997**	785	0.996	**786**	**0.997**	
4	**784**	**0.995**	**784**	**0.995**	**784**	**0.995**	
5	**786**	**0.997**	783	0.994	**786**	**0.997**	
6	**786**	**0.997**	784	0.995	**786**	**0.997**	
7	**786**	**0.997**	784	0.995	**786**	**0.997**	
8	787	0.999	784	0.995	787	0.999	
9	**786**	**0.997**	**786**	**0.997**	**786**	**0.997**	
10	**786**	**0.997**	784	0.995	**786**	**0.997**	

Compound							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	389	0.975	389	0.975	389	0.975	399
2	384	0.962	384	0.962	**389**	**0.975**	
3	**382**	**0.957**	381	0.955	**382**	**0.957**	
4	**387**	**0.970**	381	0.955	**387**	**0.970**	
5	**381**	**0.955**	373	0.935	**381**	**0.955**	
6	**383**	**0.960**	372	0.932	**383**	**0.960**	
7	**379**	**0.950**	372	0.932	**379**	**0.950**	
8	**379**	**0.950**	372	0.932	**379**	**0.950**	
9	**377**	**0.945**	369	0.925	**377**	**0.945**	
10	**377**	**0.945**	369	0.925	**377**	**0.945**	

D31							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	**2981**	**0.962**	**2981**	**0.962**	**2981**	**0.962**	3100
2	2965	0.956	2965	0.956	**2991**	**0.965**	
3	2992	0.965	**2997**	**0.967**	2989	0.964	
4	2991	0.965	**2998**	**0.967**	2989	0.964	
5	3001	0.968	**3006**	**0.970**	3000	0.968	
6	**3004**	**0.969**	3001	0.968	**3004**	**0.969**	
7	**3000**	**0.968**	2997	0.967	**3000**	**0.968**	
8	**3000**	**0.968**	2997	0.967	**3000**	**0.968**	
9	2997	0.967	**3003**	**0.969**	2997	0.967	
10	2999	0.967	**3007**	**0.970**	2999	0.967	

ecoli							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	**274**	**0.815**	**274**	**0.815**	**274**	**0.815**	336
2	265	0.789	265	0.789	**274**	**0.815**	
3	285	0.848	**288**	**0.857**	285	0.848	
4	285	0.848	**287**	**0.854**	286	0.851	
5	285	0.848	283	0.842	**286**	**0.851**	
6	**286**	**0.851**	282	0.839	**286**	**0.851**	
7	**287**	**0.854**	286	0.851	**287**	**0.854**	
8	291	0.866	285	0.848	290	0.863	
9	287	0.854	287	0.854	**288**	**0.857**	
10	288	0.857	278	0.827	**289**	**0.860**	

flame							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	240	1	240	1	240	1	240
2	**237**	**0.988**	**237**	**0.988**	**238**	**0.992**	
3	**239**	**0.996**	238	0.992	**239**	**0.996**	
4	238	0.992	240	1.000	238	0.992	
5	239	0.996	240	1.000	239	0.996	
6	**239**	**0.996**	237	0.988	**239**	**0.996**	
7	**238**	**0.992**	236	0.983	**238**	**0.992**	
8	**238**	**0.992**	236	0.983	**238**	**0.992**	
9	**238**	**0.992**	237	0.988	**238**	**0.992**	
10	**238**	**0.992**	236	0.983	**238**	**0.992**	

ionosphere							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	306	0.872	306	0.872	306	0.872	351
2	310	0.883	310	0.883	307	0.875	
3	291	0.829	**295**	**0.840**	291	0.829	
4	298	0.849	**300**	**0.855**	298	0.849	
5	293	0.835	**296**	**0.843**	293	0.835	
6	**297**	**0.846**	296	0.843	**297**	**0.846**	
7	293	0.835	**298**	**0.849**	293	0.835	
8	**295**	**0.840**	292	0.832	**295**	**0.840**	
9	**293**	**0.835**	290	0.826	**293**	**0.835**	
10	**293**	**0.835**	291	0.829	**293**	**0.835**	

iris							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	**144**	**0.960**	**144**	**0.960**	**144**	**0.960**	150
2	139	0.927	139	0.927	**144**	**0.960**	
3	143	0.953	**144**	**0.960**	**144**	**0.960**	
4	**142**	**0.947**	**142**	**0.947**	**142**	**0.947**	
5	**143**	**0.953**	**143**	**0.953**	**143**	**0.953**	
6	142	0.947	142	0.947	**143**	**0.953**	
7	143	0.953	**144**	**0.960**	143	0.953	
8	140	0.933	**145**	**0.967**	141	0.940	
9	141	0.940	146	0.973	141	0.940	
10	142	0.947	**143**	**0.953**	**143**	**0.953**	

jain							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	373	1.000	373	1.000	373	1.000	373
2	370	0.992	370	0.992	373	1.000	
3	373	1.000	373	1.000	373	1.000	
4	373	1.000	373	1.000	373	1.000	
5	373	1.000	373	1.000	373	1.000	
6	373	1.000	373	1.000	373	1.000	
7	373	1.000	373	1.000	373	1.000	
8	373	1.000	373	1.000	373	1.000	
9	373	1.000	373	1.000	373	1.000	
10	373	1.000	373	1.000	373	1.000	

pathbased							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	300	1.000	300	1.000	300	1.000	300
2	299	0.997	299	0.997	300	1.000	
3	**298**	**0.993**	**298**	**0.993**	**298**	**0.993**	
4	**297**	**0.990**	296	0.987	**297**	**0.990**	
5	**298**	**0.993**	297	0.990	**298**	**0.993**	
6	**297**	**0.990**	**297**	**0.990**	**297**	**0.990**	
7	**297**	**0.990**	295	0.983	**297**	**0.990**	
8	296	0.987	**297**	**0.990**	296	0.987	
9	**297**	**0.990**	296	0.987	**297**	**0.990**	
10	**296**	**0.987**	295	0.983	**296**	**0.987**	

R15							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	**597**	**0.995**	**597**	**0.995**	**597**	**0.995**	600
2	589	0.982	589	0.982	**597**	**0.995**	
3	598	0.997	598	0.997	598	0.997	
4	**597**	**0.995**	596	0.993	**597**	**0.995**	
5	598	0.997	597	0.995	598	0.997	
6	598	0.997	596	0.993	598	0.997	
7	598	0.997	596	0.993	598	0.997	
8	598	0.997	596	0.993	598	0.997	
9	598	0.997	596	0.993	598	0.997	
10	598	0.997	596	0.993	598	0.997	

spiral							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	312	1.000	312	1.000	312	1.000	312
2	312	1.000	312	1.000	312	1.000	
3	312	1.000	312	1.000	312	1.000	
4	312	1.000	312	1.000	312	1.000	
5	312	1.000	312	1.000	312	1.000	
6	312	1.000	311	0.997	312	1.000	
7	312	1.000	311	0.997	312	1.000	
8	**311**	**0.997**	307	0.984	**311**	**0.997**	
9	**311**	**0.997**	304	0.974	**311**	**0.997**	
10	**308**	**0.987**	302	0.968	**308**	**0.987**	

Tabela E.7.2. Resultados do Experimento 2 para *K*=5.

Aggregation -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	786	0.997	783	0.994	786	0.997	788
0.2 | 1	786	0.997	783	0.994			
0.4 | 2	786	0.997	783	0.994			
0.6 | 3	778	0.987	781	0.991			
0.8 | 4	773	0.981	763	0.968			

Compound -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	381	0.955	372	0.932	381	0.955	399
0.2 | 1	381	0.955	372	0.932			
0.4 | 2	381	0.955	372	0.932			
0.6 | 3	372	0.932	368	0.922			
0.8 | 4	366	0.917	344	0.862			

D31 -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	3000	0.968	**3006**	**0.970**	3000	0.968	3100
0.2 | 1	3000	0.968	**3006**	**0.970**			
0.4 | 2	3002	0.968	**3007**	**0.970**			
0.6 | 3	2960	0.955	2987	0.964			
0.8 | 4	2932	0.946	2863	0.924			

ecoli -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	286	0.851	284	0.845	286	0.851	336
0.2 | 1	286	0.851	284	0.845			
0.4 | 2	285	0.848	285	0.848			
0.6 | 3	284	0.845	281	0.836			
0.8 | 4	279	0.830	254	0.756			

flame -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	239	0.996	240	1.000	239	0.996	240
0.2 | 1	239	0.996	240	1.000			
0.4 | 2	239	0.996	240	1.000			
0.6 | 3	239	0.996	240	1.000			
0.8 | 4	238	0.992	238	0.992			

ionosphere -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	293	0.835	**296**	**0.843**	293	0.835	351
0.2 | 1	293	0.835	**296**	**0.843**			
0.4 | 2	293	0.835	297	0.846			
0.6 | 3	290	0.826	289	0.823			
0.8 | 4	281	0.801	277	0.789			

iris -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	143	0.953	143	0.953	143	0.953	150
0.2 | 1	143	0.953	143	0.953			
0.4 | 2	143	0.953	143	0.953			
0.6 | 3	141	0.940	143	0.953			
0.8 | 4	139	0.927	136	0.907			

jain -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	373	1.000	373	1.000	373	1.000	373
0.2 | 1	373	1.000	373	1.000			
0.4 | 2	373	1.000	373	1.000			
0.6 | 3	372	0.997	371	0.995			
0.8 | 4	371	0.995	365	0.979			

pathbased -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	298	0.993	297	0.990	298	0.993	300
0.2 | 1	298	0.993	297	0.990			
0.4 | 2	298	0.993	296	0.987			
0.6 | 3	296	0.987	295	0.983			
0.8 | 4	293	0.977	287	0.957			

R15 -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	598	0.997	597	0.995	598	0.997	600
0.2 | 1	598	0.997	597	0.995			
0.4 | 2	598	0.997	597	0.995			
0.6 | 3	586	0.977	588	0.980			
0.8 | 4	567	0.945	532	0.887			

spiral -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	312	1.000	312	1.000	312	1.000	312
0.2 | 1	312	1.000	312	1.000			
0.4 | 2	312	1.000	312	1.000			
0.6 | 3	307	0.984	311	0.997			
0.8 | 4	304	0.974	300	0.962			

Tabela E.7.3. Resultados do Experimento 2 para *K*=10.

Aggregation -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	786	0.997	784	0.995	786	0.997	**788**
0.1 | 1	786	0.997	784	0.995			
0.2 | 2	786	0.997	784	0.995			
0.3 | 3	786	0.997	784	0.995			
0.4 | 4	786	0.997	784	0.995			
0.5 | 5	781	0.991	784	0.995			
0.6 | 6	775	0.984	785	0.996			
0.7 | 7	775	0.984	785	0.996			
0.8 | 8	767	0.973	774	0.982			
0.9 | 9	760	0.964	698	0.886			

Compound -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	377	0.945	369	0.925	377	0.945	**399**
0.1 | 1	377	0.945	369	0.925			
0.2 | 2	377	0.945	369	0.925			
0.3 | 3	377	0.945	369	0.925			
0.4 | 4	377	0.945	369	0.925			
0.5 | 5	374	0.937	369	0.925			
0.6 | 6	369	0.925	367	0.920			
0.7 | 7	369	0.925	367	0.920			
0.8 | 8	350	0.877	341	0.855			
0.9 | 9	342	0.857	277	0.694			

D31 -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	2999	0.967	**3007**	**0.970**	2999	0.967	**3100**
0.1 | 1	2999	0.967	**3007**	**0.970**			
0.2 | 2	2999	0.967	**3007**	**0.970**			
0.3 | 3	2999	0.967	**3007**	**0.970**			
0.4 | 4	2999	0.967	**3009**	**0.971**			
0.5 | 5	2982	0.962	**3012**	**0.972**			
0.6 | 6	2952	0.952	3014	0.972			
0.7 | 7	2952	0.952	3014	0.972			
0.8 | 8	2876	0.928	**2923**	**0.943**			
0.9 | 9	2842	0.917	2492	0.804			

ecoli -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	289	0.860	278	0.827	289	0.860	**336**
0.1 | 1	289	0.860	278	0.827			
0.2 | 2	289	0.860	278	0.827			
0.3 | 3	286	0.851	278	0.827			
0.4 | 4	285	0.848	278	0.827			
0.5 | 5	281	0.836	279	0.830			
0.6 | 6	282	0.839	275	0.818			
0.7 | 7	282	0.839	275	0.818			
0.8 | 8	270	0.804	264	0.786			
0.9 | 9	265	0.789	174	0.518			

Tabela 3. (continuação)

flame -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	238	0.992	236	0.983	238	0.992	**240**
0.1 | 1	238	0.992	236	0.983			
0.2 | 2	238	0.992	236	0.983			
0.3 | 3	238	0.992	236	0.983			
0.4 | 4	238	0.992	236	0.983			
0.5 | 5	238	0.992	236	0.983			
0.6 | 6	238	0.992	236	0.983			
0.7 | 7	238	0.992	236	0.983			
0.8 | 8	234	0.975	233	0.971			
0.9 | 9	234	0.975	217	0.904			

ionosphere -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	293	0.835	291	0.829	293	0.835	**351**
0.1 | 1	293	0.835	291	0.829			
0.2 | 2	293	0.835	291	0.829			
0.3 | 3	293	0.835	291	0.829			
0.4 | 4	293	0.835	291	0.829			
0.5 | 5	290	0.826	291	0.829			
0.6 | 6	288	0.821	290	0.826			
0.7 | 7	288	0.821	290	0.826			
0.8 | 8	281	0.801	272	0.775			
0.9 | 9	279	0.795	242	0.689			

iris -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	143	0.953	143	0.953	143	0.953	**150**
0.1 | 1	143	0.953	143	0.953			
0.2 | 2	142	0.947	142	0.947			
0.3 | 3	142	0.947	142	0.947			
0.4 | 4	141	0.940	142	0.947			
0.5 | 5	140	0.933	143	0.953			
0.6 | 6	141	0.940	142	0.947			
0.7 | 7	141	0.940	142	0.947			
0.8 | 8	136	0.907	132	0.880			
0.9 | 9	134	0.893	90	0.600			

jain -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	373	1.000	373	1.000	373	1.000	**373**
0.1 | 1	373	1.000	373	1.000			
0.2 | 2	373	1.000	373	1.000			
0.3 | 3	373	1.000	373	1.000			
0.4 | 4	373	1.000	373	1.000			
0.5 | 5	372	0.997	373	1.000			
0.6 | 6	371	0.995	372	0.997			
0.7 | 7	371	0.995	372	0.997			
0.8 | 8	365	0.979	364	0.976			
0.9 | 9	361	0.968	336	0.901			

pathbased -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	296	0.987	295	0.983	296	0.987	**300**
0.1 | 1	296	0.987	295	0.983			
0.2 | 2	296	0.987	295	0.983			
0.3 | 3	296	0.987	295	0.983			
0.4 | 4	296	0.987	295	0.983			
0.5 | 5	296	0.987	295	0.983			
0.6 | 6	295	0.983	294	0.980			
0.7 | 7	295	0.983	294	0.980			
0.8 | 8	290	0.967	280	0.933			
0.9 | 9	285	0.950	222	0.740			

R15 -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	598	0.997	596	0.993	598	0.997	**600**
0.1 | 1	598	0.997	596	0.993			
0.2 | 2	598	0.997	596	0.993			
0.3 | 3	598	0.997	596	0.993			
0.4 | 4	598	0.997	596	0.993			
0.5 | 5	593	0.988	596	0.993			
0.6 | 6	576	0.960	596	0.993			
0.7 | 7	576	0.960	596	0.993			
0.8 | 8	548	0.913	544	0.907			
0.9 | 9	531	0.885	383	0.638			

spiral -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	308	0.987	302	0.968	308	0.987	**312**
0.1 | 1	308	0.987	302	0.968			
0.2 | 2	308	0.987	302	0.968			
0.3 | 3	308	0.987	301	0.965			
0.4 | 4	308	0.987	301	0.965			
0.5 | 5	308	0.987	299	0.958			
0.6 | 6	305	0.978	297	0.952			
0.7 | 7	305	0.978	297	0.952			
0.8 | 8	296	0.949	271	0.869			
0.9 | 9	286	0.917	233	0.747			

S1 -> K = 10							
α | α*K	**A**		**B**		**KNN**		**Total**
	n	***ACC***	***n***	***ACC***	***n***	***ACC***	
0.0 | 0	**2986**	**0.597**	2973	0.595	**2986**	**0.597**	**5000**
0.1 | 1	**2986**	**0.597**	2973	0.595			
0.2 | 2	**2986**	**0.597**	2973	0.595			
0.3 | 3	**2986**	**0.597**	2973	0.595			
0.4 | 4	**2986**	**0.597**	2973	0.595			
0.5 | 5	2946	0.589	2973	0.595			
0.6 | 6	2881	0.576	2976	0.595			
0.7 | 7	2881	0.576	2976	0.595			
0.8 | 8	2823	0.565	2865	0.573			
0.9 | 9	2786	0.557	2440	0.488			

S2 -> K = 10							
α | α*K	**A**		**B**		**KNN**		**Total**
	n	***ACC***	***n***	***ACC***	***n***	***ACC***	
0.0 | 0	**2843**	**0.569**	2819	0.564	**2843**	**0.569**	**5000**
0.1 | 1	**2843**	**0.569**	2819	0.564			
0.2 | 2	**2843**	**0.569**	2819	0.564			
0.3 | 3	**2843**	**0.569**	2819	0.564			
0.4 | 4	**2843**	**0.569**	2815	0.563			
0.5 | 5	2816	0.563	2808	0.562			
0.6 | 6	2787	0.557	2808	0.562			
0.7 | 7	2787	0.557	2808	0.562			
0.8 | 8	2719	0.544	2677	0.535			
0.9 | 9	2694	0.539	2263	0.453			

S3 -> K = 10							
α | α*K	**A**		**B**		**KNN**		**Total**
	n	***ACC***	***n***	***ACC***	***n***	***ACC***	
0.0 | 0	2320	0.464	**2329**	**0.466**	2320	0.464	**5000**
0.1 | 1	2320	0.464	**2329**	**0.466**			
0.2 | 2	2318	0.464	**2327**	**0.465**			
0.3 | 3	2320	0.464	**2329**	**0.466**			
0.4 | 4	**2323**	**0.465**	2321	0.464			
0.5 | 5	2314	0.463	2318	0.464			
0.6 | 6	2302	0.460	**2328**	**0.466**			
0.7 | 7	2302	0.460	**2328**	**0.466**			
0.8 | 8	2264	0.453	2306	0.461			
0.9 | 9	2252	0.450	1968	0.394			

S4 -> K = 10							
α | α*K	**A**		**B**		**KNN**		**Total**
	n	***ACC***	***n***	***ACC***	***n***	***ACC***	
0.0 | 0	**2010**	**0.402**	2007	0.401	2010	0.402	**5000**
0.1 | 1	**2010**	**0.402**	2007	0.401			
0.2 | 2	2009	0.402	1999	0.400			
0.3 | 3	**2017**	**0.403**	1991	0.398			
0.4 | 4	**2015**	**0.403**	1985	0.397			
0.5 | 5	**2015**	**0.403**	1992	0.398			
0.6 | 6	**2019**	**0.404**	1992	0.398			
0.7 | 7	**2019**	**0.404**	1992	0.398			
0.8 | 8	1949	0.390	1954	0.391			
0.9 | 9	1939	0.388	1615	0.323			

SEÇÃO F.
CONCLUSÕES

Neste projeto, foi proposta a utilização do grafo de interações Nk para encontra os *K* vizinhos mais próximos no KNN. Também foi proposta a mudança na razão entre o número de arestas definidas por densidade espacial e definidas apenas por distancia no grafo de interações Nk. Um parâmetro α é encarregado de definir esta razão para ambos KNN modificados.

Os resultados dos experimentos indicam que o melhor desempenho é geralmente obtido pelo KNN original, ou pelo KNN proposto para um valor de α pequeno, em bases de dados bidimensionais e sem sobreposição de agrupamentos. Nestes casos, escolher os vizinhos mais próximos utilizando densidade espacial afeta o desempenho do KNN proposto neutralmente ou negativamente. Entretanto, melhores resultados são obtidos em bases com dimensões maiores e/ou sobreposição de clusters. Os resultados para os experimentos investigando os impactos de α indicam que melhores resultados são geralmente obtidos para menores valores de α. A exceção são as bases de dados com sobreposições de clusters, cujos resultados são melhores ao selecionar vizinhos baseados em densidade espacial.

A seleção automática de α no KNN proposto é um possível trabalho futuro. Além disso, outra possibilidade seria investigar seu desempenho em bases de dados de Medicina com alta dimensionalidade. Finalmente, um tópico importante de pesquisa futura seria a redução de tempo e complexidade do KNN. O uso do grafo de interações Nk pode ser investigado no futuro para reduzir o número de exemplos nas bases de treinamento analisadas pelo algoritmo ao procurar os *K* vizinhos mais próximos.

SEÇÃO G.
REFERÊNCIAS

ALTMAN, N. S. (1992). “An introduction to kernel and nearest-neighbor nonparametric regression”, The American Statistician, 46(3): 175-185.

AHA, D. W.; KIBLER, D. & ALBERT, M.K. (1991). “Instance-based learning algorithms”, Machine Learning, 6(1): 37-66.

DUA, D. & GRAFF, C. (2019). “UCI Machine Learning Repository”, [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J. & XU, X. (1996), “A density-based algorithm for discovering clusters in large spatial databases with noise”, In the Proc. of the 2nd ACM Int. Conf. Knowl. Discovery Data Min. (KDD), 226–231.

FIX, E. (1985). “Discriminatory analysis: nonparametric discrimination, consistency properties”, Technical Report, USAF School of Aviation Medicine.

FRÄNTI, P. & SIERANOJA, S. (2018). “K-means properties on six clustering benchmark datasets”, Applied Intelligence, 48 (12): 4743-4759.

KOTSIANTIS, S. B.; ZAHARAKIS, I. & PINTELAS, P. (2007). “Supervised machine learning: A review of classification techniques”, Emerging artificial intelligence applications in computer engineering, 160(1): 3-24.

MACQUEEN, J. (1967) Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1, 281-297.

MORAES, J. C. B. (2020). “Busca por similaridade utilizando grafo de interações NK”, Dissertação de Mestrado em Computação Aplicada, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo.

MORAES, J. C. B., & TINÓS, R. (2020). “Busca por Similaridade usando o Gráfico de Interação NK”, Nos Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional, 222-233.

RODRIGUEZ, A. & LAIO, A. (2014). “Clustering by fast search and find of density peaks,” Science, 344(6191): 1492–1496.

TINÓS, R.; ZHAO, L.; CHICANO, F. & WHITLEY, D. (2018). “NK hybrid genetic algorithm for clustering”, IEEE Transactions on Evolutionary Computation, 22(5): 748-761.