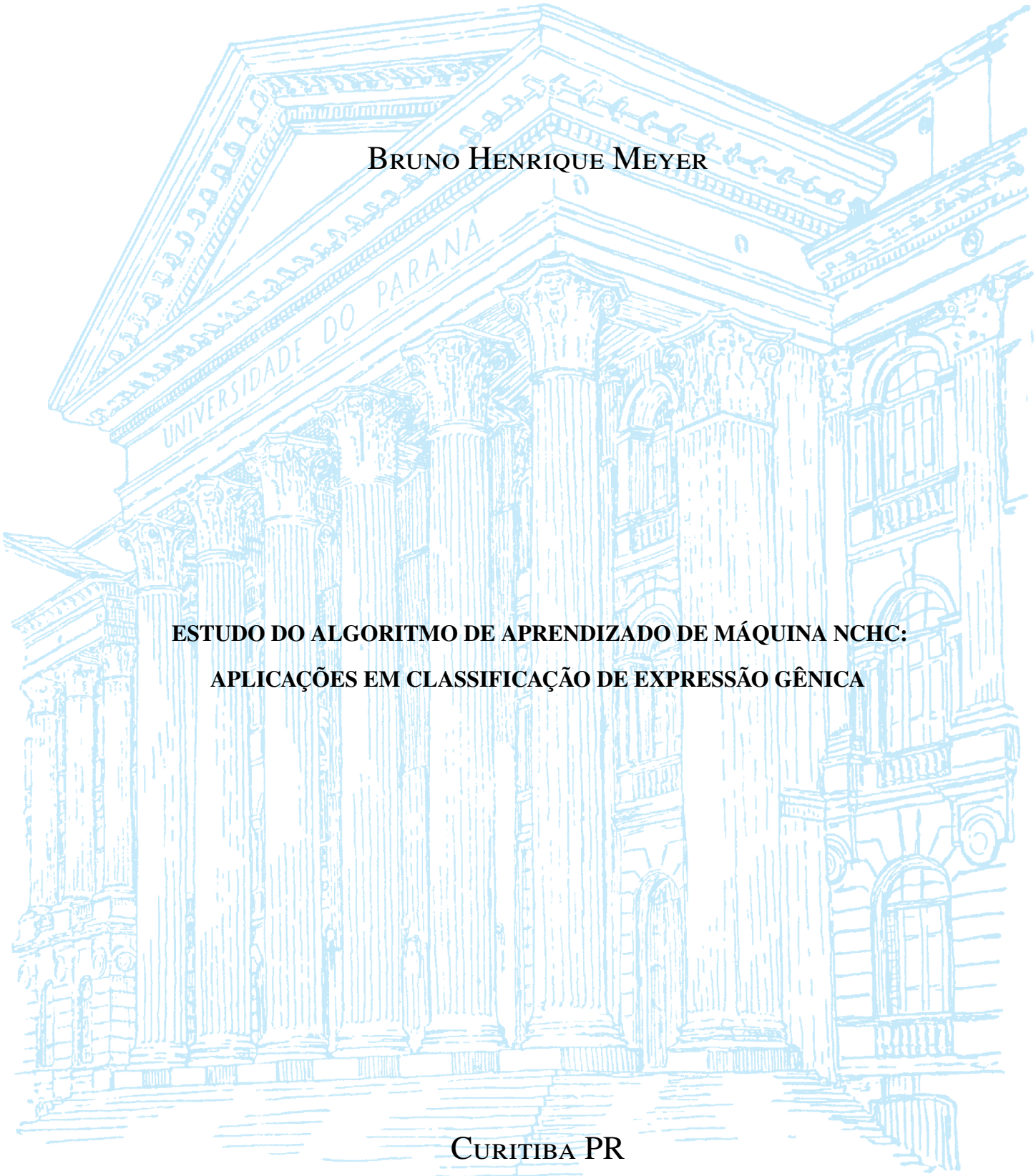
Universidade Federal do Paraná

Bruno Henrique Meyer



ESTUDO DO ALGORITMO DE APRENDIZADO DE MÁQUINA NCHC: APLICAÇÕES EM CLASSIFICAÇÃO DE EXPRESSÃO GÊNICA

Curitiba PR
2018

Bruno Henrique Meyer

ESTUDO DO ALGORITMO DE APRENDIZADO DE MÁQUINA NCHC: APLICAÇÕES EM CLASSIFICAÇÃO DE EXPRESSÃO GÊNICA

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Informática Biomédica, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Informática Biomédica*.

Orientador: Prof. Dr. André Luiz Pires Guedes.

Curitiba PR
2018

Agradecimentos

Agradeço à Universidade Federal do Paraná pela oportunidade e infraestrutura disponibilizada para a realização do meu curso de graduação. Também agradeço aos membros e colegas do Departamento de Informática, docentes e discentes, em especial ao professor Dr. André Luiz Pires Guedes por sua orientação e paciência e também ao grupo PET Computação por possibilitar o amadurecimento das melhores qualidades que despertei durante a graduação, necessárias para realizar o presente trabalho.

Resumo

O problema de análise e predição de expressão gênica pode envolver o processamento de vários dados biológicos. Algoritmos da área de aprendizado de máquina supervisionado são frequentemente utilizados para resolver tal problema. Neste trabalho foi estudado o desempenho do algoritmo *Nearest Convex Hull Classification* (NCHC) comparado a outros algoritmos de aprendizado de máquina supervisionado, quando aplicados à quatro bases de dados diferentes, sendo duas delas relacionadas à expressão gênica. O NCHC, com o auxilio da técnica de redução de dimensionalidade SVM-RFE, apresentou resultados significativos para bases de dados com poucas dimensões.

Palavras-chave: Geometria computacional, Aprendizado de máquina, Análise de expressão gênica.

Abstract

The problem of analysis and prediction of gene expression may involve the processing of various biological data. Algorithms of the supervised machine learning area are often used to solve such a problem. In this work, the performance of the Nearest Convex Hull Classification (NCHC) algorithm was compared to other supervised machine learning algorithms when applied to four different databases, two of them related to gene expression. The NCHC, with the aid of the dimensional reduction technique SVM-RFE, presented significant results for databases with few dimensions.

Keywords: Computational geometry, Machine learning, Gene expression analysis.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

KNN	*K-Nearest Neighbors*
MLP	*Multilayer Perceptron*
NCHC	*Nearest Convex Hull Classification*
QDA	*Quadratic Discriminant Analysis*
RDF	*Random Forest*
RBF	*Radial Basis Function*
SVM	*Support Vector Machine*
SVM-RFE	*SVM Recursive Feature Elimination*

Sumário

1 Introdução

Na área de aprendizado de máquina supervisionado, a análise de dados e o estudo da correlação entre característica são aspectos comuns em diversos contextos. Sistemas de predições e classificações são tópicos amplamente abordados e utilizados para a solução de diversos problemas, como a análise de expressão gênica e o estudo da longevidade em seres vivos, o que é descrito por Fabris et al. (2017).

Dentre os problemas citados, existe o da diferenciação de três tipos de flores da espécie Iris. As figuras 1.1(a), 1.1(b), 1.1(c) e 1.1(d) ilustram uma representação geométrica dos dados referentes a 4 atributos, como o tamanho das pétalas, de um conjunto de 50 flores que já estão rotuladas em 3 tipos: *Iris setosa*, *Iris virginica* e *Iris versicolor*. Cada gráfico contém uma combinação entre os atributos mencionados para cada eixo.

Por conveniência, foram usados diferentes gráficos representando o $\mathbb{R}^3$ devido à dificuldade de representar os pontos no $\mathbb{R}^4$. Cada ponto representa uma flor e os valores dos atributos relacionados a ela.

Nesse exemplo, fica explicitado o motivo de classificadores que se baseiam na análise geométrica dos atributos funcionarem em diversos casos, e além disso percebe-se a formação de politopos, convexos ou não, que representam as diferentes classes em um sistema (no caso para cada tipo de flor).

Diversos algoritmos de classificação utilizam a ideia da representação geométrica dos dados para criar um hiperplano que representa um modelo que pode ser usado de forma efetiva na etapa de classificação. O algoritmo KNN também utiliza a distância euclidiana entre os pontos, porém essa característica acaba considerando uma delimitação hiperesférica das classes, o que não é observado no exemplo da Figura 1.1, além de desconsiderar a correlação entre os pontos de um mesmo conjunto.

Em um caso genérico de classificação, se deseja identificar qual a classe mais provável que um ponto qualquer no $\mathbb{R}^N$, considerando outros diversos pontos já rotulados. Será que existe um algoritmo, que tenha custo computacional viável, capaz de considerar a construção dos politopos gerados por cada classe e com isso, calcular a distância do ponto a cada politopo? Tal algoritmo teria bons resultados em relação aos algoritmos de classificação mais comuns observados atualmente na literatura?

Nalbantov et al. (2006) explica que o algoritmo *Nearest Convex Hull Classification* (NCHC) considera os convexos formados por diferentes classes, como ilustrado na Figura 1.2. Nesta figura, é apresentado o uso do NCHC para um problema de classificação que envolve duas classes, onde os dois fechos convexos (CH), que representam as classes "-" e "+" são utilizados para classificar o ponto "x" que não possui nenhum rótulo conhecido. Para realizar essa classificação, o algoritmo calcula a distância euclidiana do ponto "x" para todos os convexos, e o mesmo tem a sua classe estimada como a classe representada pelo convexo mais próximo de acordo com as distâncias calculadas. Alguns trabalhos, como o de Chen et al. (2014), demonstram bons resultados do NCHC em comparação a outros algoritmos como o SVM em algumas situações.

(a) Representação 1

(b) Representação 2

(c) Representação 3

(d) Representação 4

Figura 1.1: Representação da correlação entre os quatro atributos do *dataset* Iris. Cada cor representa uma das classes (tipo da Iris). Os eixos representam os atributos. Fonte: Elaborado pelo autor.

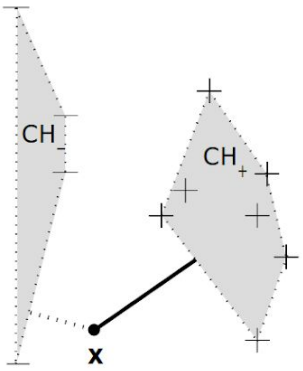


Figura 1.2: Exemplificação do algoritmo NCHC.
Fonte: Nalbantov et al. (2006).

Em alguns casos, pode haver diferença em se considerar um conjunto de pontos como um convexo e um não convexo. A Figura 1.3 ilustra tal característica:

(a) Representação dos conjuntos por politopos convexos

(b) Representação dos conjuntos por politopos não convexos

Figura 1.3: Exemplo de classificações baseadas em polígonos convexos e não convexos. Deseja-se predizer se o "?" é membro do grupo "X" ou "O". Fonte: Elaborado pelo autor.

Com isso, podemos considerar que em algumas situações o algoritmo NCHC pode errar a classificação devido a essa pequena diferença. Percebe-se que na Figura 1.3, a distância entre ponto e conjuntos pode ser divido em três etapas:

- Encontrar as faces formadas por cada conjunto que são as mais próximas do ponto que se deseja calcular a distância;
- Calcular a distância entre o ponto e cada face encontrada;
- Identificar o grupo mais próximo do ponto e realizar a predição considerando as distâncias às faces.

Vemos que na representação apresentada na Imagem 1.3(a), utilizada no algoritmo NCHC, pode generalizar situações onde faces relativamente grandes são criadas. Estas faces representam uma interpolação entre os pontos que a formam, que pode acontecer em um número qualquer de dimensões. Percebe-se que a distribuição e quantidade de pontos que são utilizados para formar os fechos convexos são critérios que influenciam diretamente no tamanho das faces geradas. Na Imagem 1.3(b) vemos uma outra representação, que não acontece no algoritmo NCHC, onde uma das faces representadas na Imagem 1.3(a) é estruturada como duas faces diferentes. Repare que o ponto "?" apresentado na imagem tem suas classificações diferentes para cada representação, de acordo com a distância calculada para os polígonos.

Diversos problemas podem ser encontrados em ambas representações. Um desses problemas é a necessidade de número de representantes por classe que cresce de acordo com o número de dimensões do problema.

1.1 Motivação

Diversos estudos que utilizam algoritmos de aprendizagem de máquina supervisionado para a classificação e análise de dados biológicos como as de análise de expressão gênica comumente utilizam o algoritmo *Support Vector Machine* (SVM) em seus estudos devido à sua consolidação no estado da arte da área, como vemos nos trabalhos de Ramaswamy et al. (2001), Yeoh et al. (2002) e Brown et al. (2000).

Também, vemos que as descrições dos algoritmos de aprendizado de máquina supervisionado muitas vezes exploram a interpretação geométrica dos dados para formalizar suas definições. Neste trabalho, buscou-se explorar o desempenho do algoritmo NCHC em diferentes bases de dados, incluindo a de expressão gênica, seguindo as metodologias indicadas por diferentes trabalhos mencionados no capítulo de revisão bibliográfica. Dentre os objetivos apresentados, está a análise do NCHC quando aplicado junto a técnicas de redução de dimensionalidade como o SVM-RFE e a transformação de *kernel Radial Basis Function* (RBF). Também, o trabalho se propõe a observar e discutir os motivos que influenciam os erros de classificação pelo algoritmo NCHC.

Com isso, espera-se observar as diferenças entre o uso de diferentes classificadores como o NCHC e SVM quando aplicados a diferentes conjuntos de dados. Além disso, diversas ferramentas que existem hoje como a biblioteca *scikit-learning* da linguagem de programação Python podem facilitar o estudo dos algoritmos de aprendizado de máquina, sendo que estes existiam nos tempos em que alguns dos classificadores como o NCHC foram propostos.

1.2 Conteúdo do trabalho

No Capítulo 2 deste trabalho serão apresentados os fundamentos e definições necessárias para o entendimento dos trabalhos relacionados e dos algoritmos de aprendizado de máquina estudados. Tais conceitos serão apresentados posteriormente no Capítulo 3 que elucidará os trabalhos relacionados e a revisão de literatura do problema, e abordará alguns trabalhos que envolvem análise de expressão gênica com aprendizado de máquina supervisionado. Após isso, o Capítulo 4 apresentará a proposta, objetivos e metodologia dos experimentos realizados neste trabalho, que envolvem a implementação do algoritmo NCHC e a descrição da forma como o mesmo será comparado com outros classificadores. Com isso, os resultados, presentes no Capítulo 5 serão demonstrados e discutidos e por fim, serão apresentadas as conclusões obtidas junto às considerações finais no Capítulo 6.

2 Fundamentação Teórica

Os algoritmos de aprendizado de máquina possuem, em sua maioria, algumas definições formais. Neste capítulo, serão apresentados os fundamentos necessários para a interpretação do funcionamento desses algoritmos, além de definições em relação a alguns dos fundamentos utilizados na definição do algoritmo NCHC.

2.1 Convexos

Um convexo pode ser definido como um conjunto C tal que $C \subseteq R^d$ e para todo x_i e x_j pertencentes a C, o segmento de reta entre x_i e x_j deve também ser contido em C.

Pode-se representar um convexo com um conjunto de equações que representam hiperplanos tangentes à cada uma das faces do convexo, no formato:

$$w_i : \lambda_{i0}x_0 + \lambda_{i1}x_1 + ... + \lambda_{i(d-1)}x_{d-1} = b_i \tag{2.1}$$

Onde w_i representa a i-ésima face do convexo C, e d a dimensão do problema. Para um ponto x qualquer, x pertence ao convexo C se e somente se o mesmo satisfaz todas as equações $\lambda_{i0}x_0 + \lambda_{i1}x_1 + ... + \lambda_{i(d-1)}x_{d-1} - b_i \leq 0$.

2.2 *Convex Hull*

A determinação do *Convex Hull* ou fecho convexo é um problema que busca determinar as faces de um politopo de menor área possível que possui em seu interior um conjunto de pontos especificados. Para isso, é necessário identificar quais são os pontos que fazem parte dessas faces e representá-las apropriadamente. Genericamente, podemos definir o problema de encontrar o *Convex Hull* de um conjunto finito de pontos S como:

$$ConvexHull(S) = \left\{ \sum_{i=1}^{|S|} \alpha_i x_i \middle| \forall i : \alpha_i \geq 0 \; e \; \sum_{i=1}^{|S|} \alpha_i = 1 \right\} \tag{2.2}$$

Onde α_i representa um peso atribuído ao i-ésimo ponto do conjunto, de forma que os pesos de cada ponto são utilizados como combinação para obter-se qualquer ponto dentro do convexo. Uma analogia utilizada frequentemente para a explicação da determinação do *Convex Hull* de um conjunto finito de pontos é a de se esticar um elástico em volta de diversos pregos e tentar descobrir a conformação final do elástico após solta-lo. A Figura 2.1 ilustra a ideia básica do problema no $\mathbb{R}^2$. É importante ressaltar que os convexos podem ser definidos em um número qualquer de dimensões, onde as faces serão formadas sempre por N pontos em convexos construídos a partir de pelo menos $N + 1$ pontos no $\mathbb{R}^N$.

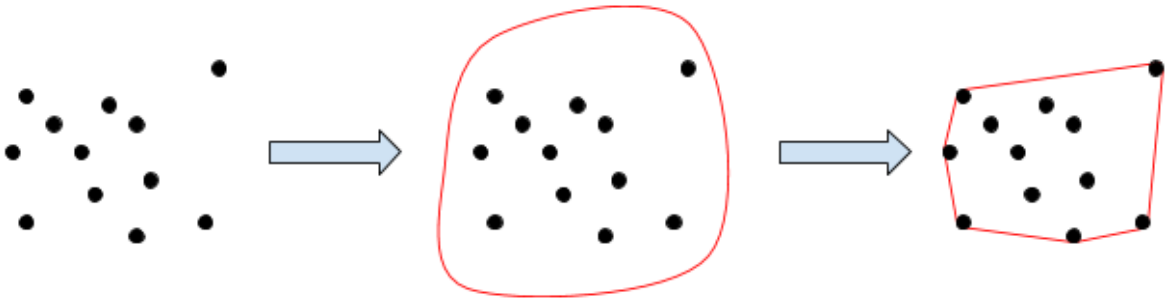


Figura 2.1: Representação da identificação do fecho convexo.
Fonte: Elaborado pelo autor.

2.3 Classificação

Em aprendizado de máquina supervisionado, classificar significa atribuir um rótulo a uma determina instância (inicialmente não rotulada) considerando uma base de dados com diversas outras instâncias já rotuladas. As instâncias podem ser representadas por pontos que podem estar dispostos em um número variado de dimensões, que representam diferentes atributos. Considere dois ou mais conjuntos de pontos (em uma dimensão finita) C1, C2, . . . , CN, que representam diferentes classes. Um ponto P qualquer no $\mathbb{R}^N$, não pertencente a nenhum dos conjuntos C1, C2, . . . CN pode possuir diferentes tipos de distâncias a cada um destes conjuntos, como a menor distância dentre o ponto P e cada ponto de um conjunto. Tais distâncias podem ser utilizadas para estimar a provável classe do ponto P, entretanto não haverá nenhuma garantia que o mesmo pertença à qualquer uma das classes envolvidas no problema. De forma genérica, os conjuntos de dados que são utilizados pelos algoritmos de classificação podem ser representados pela expressão $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$ onde y_i pertence ao domínio dos números inteiros e representa a classe de um ponto qualquer, e $\vec{x}_i$ é um vetor real de dimensão finita (comum a todos os pontos), que representa as características de cada instância.

Um exemplo de um sistema de classificação é definido por dois conjuntos H e M, que representam homens e mulheres, e que contém diversos pontos no $\mathbb{R}^2$ onde um dos eixos representa a altura e o outro eixo representa o peso de um determinado indivíduo. Nesse exemplo, haverá 9 pontos que representam 9 homens e 10 pontos que representam 10 mulheres. A Figura 2.2 ilustra uma possível distribuição dos valores de cada ponto. Neste exemplo, vemos que todos os homens possuem altura superior a 1 (um) metro e 80 (oitenta) centímetros.

No exemplo, mostra-se um problema de classificação binário, onde existem apenas duas classes. Entretanto, é comum a existência de problemas de classificação onde há mais que duas classes. Parte dos classificadores conhecidos funcionam apenas para classificação binária, mas como explicitado no trabalho de Hsu e Lin (2002), existem diversas abordagens para resolver problemas de classificação multi-classes com classificadores binários.

É comum que os sistemas de classificações dividam as instâncias do problema em três conjuntos: treino, validação e teste. O conjunto de treino é o conjunto que se utiliza para a construção dos modelos dos classificadores, o conjunto de validação são as instâncias utilizadas para avaliar o desempenho dos classificadores e alterar possíveis parâmetros dos mesmos, e o conjunto de teste é utilizado para avaliar o desempenho final do classificador após aos ajustes feitos de acordo com a base de teste. Os parâmetros mencionados anteriormente também podem ser chamados de hiper-parâmetros que podem ser usados para alterar e flexibilizar os modelos gerados pelos classificadores de forma que tenham um melhor desempenho nas bases de dados.

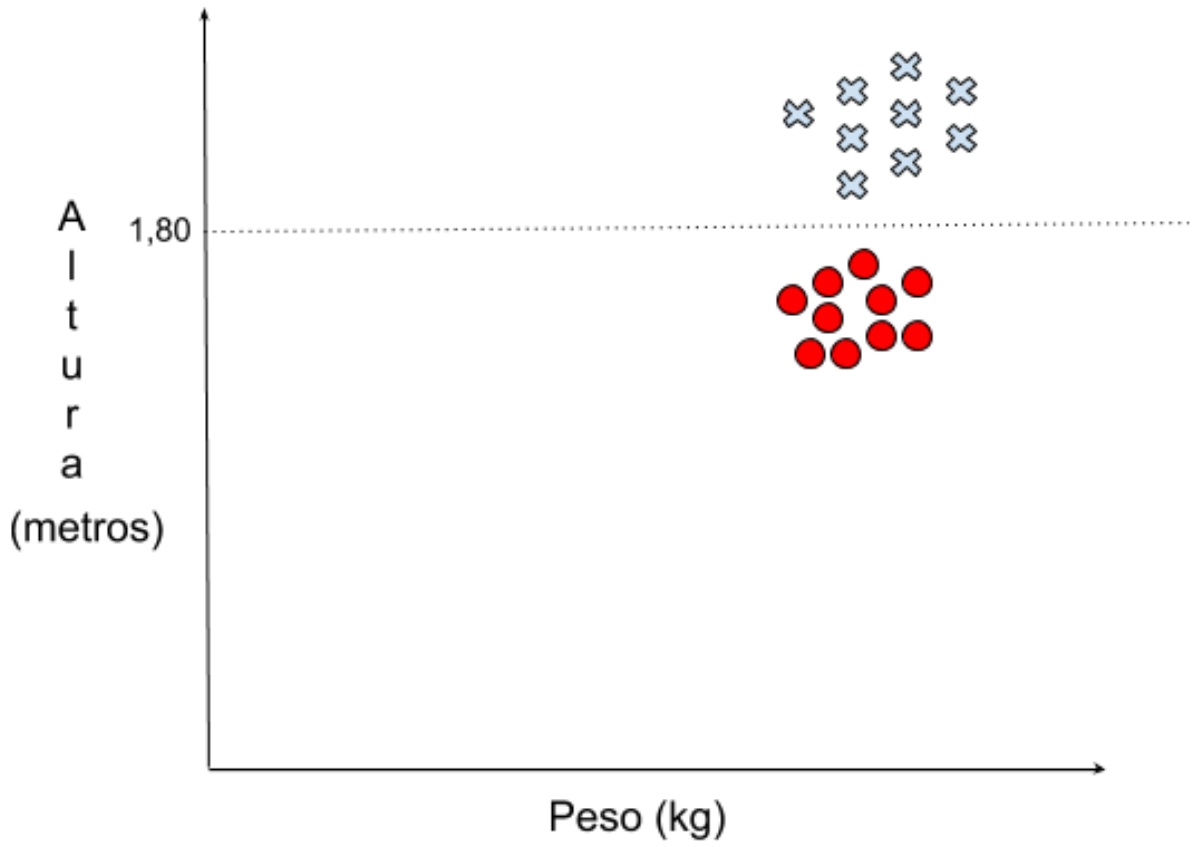


Figura 2.2: Exemplo de um problema de classificação.
A Figura ilustra a representação de 19 indivíduos representando homens (círculos em azul) e mulheres (símbolo "X"em vermelho) de acordo com suas alturas e pesos. Fonte: Elaborado pelo autor.

O processo de encontrar os melhores hiper-parâmetros é comumente chamado de *tunning* de parâmetros.

2.3.1 Support Vector Machines (SVM)

O algoritmo *Linear* SVM (neste trabalho simplificaremos como SVM), cuja ideia principal é ilustrada na Figura 2.3 pode ser utilizado para diversos problemas de regressão e classificação. Esse classificador é binário e busca encontrar um hiperplano que separa as duas classes do problema e que também maximize a distância entre os pontos mais próximos da fronteira de decisão gerada. Os pontos mais próximos da fronteira de decisão são denominados vetores de suporte, e em casos onde as classes não podem ser separadas linearmente, o algoritmo utiliza um parâmetro fornecido pelo usuário que funciona como variável de folga para permitir a construção do hiperplano que separa a maior parte dos pontos. Existem casos onde instâncias de classes fogem da distribuição da maior parte dos membros da classe ao qual pertence, o que pode causar situações onde as classes não podem ser separadas linearmente. Essas instâncias são chamadas de *outliers*, e a Figura 2.4 ilustra um exemplo desses casos.

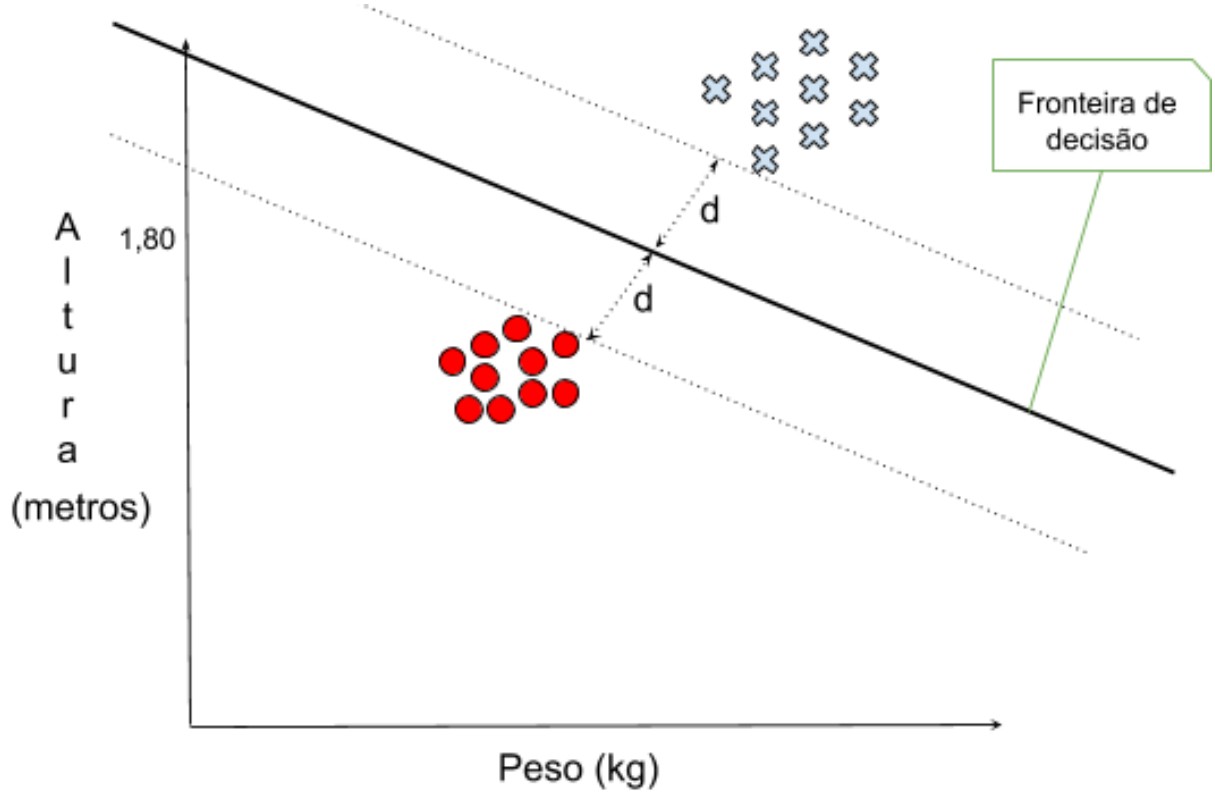


Figura 2.3: Exemplo da estratégia utilizada pelo algoritmo SVM.
Fonte: Elaborado pelo autor.

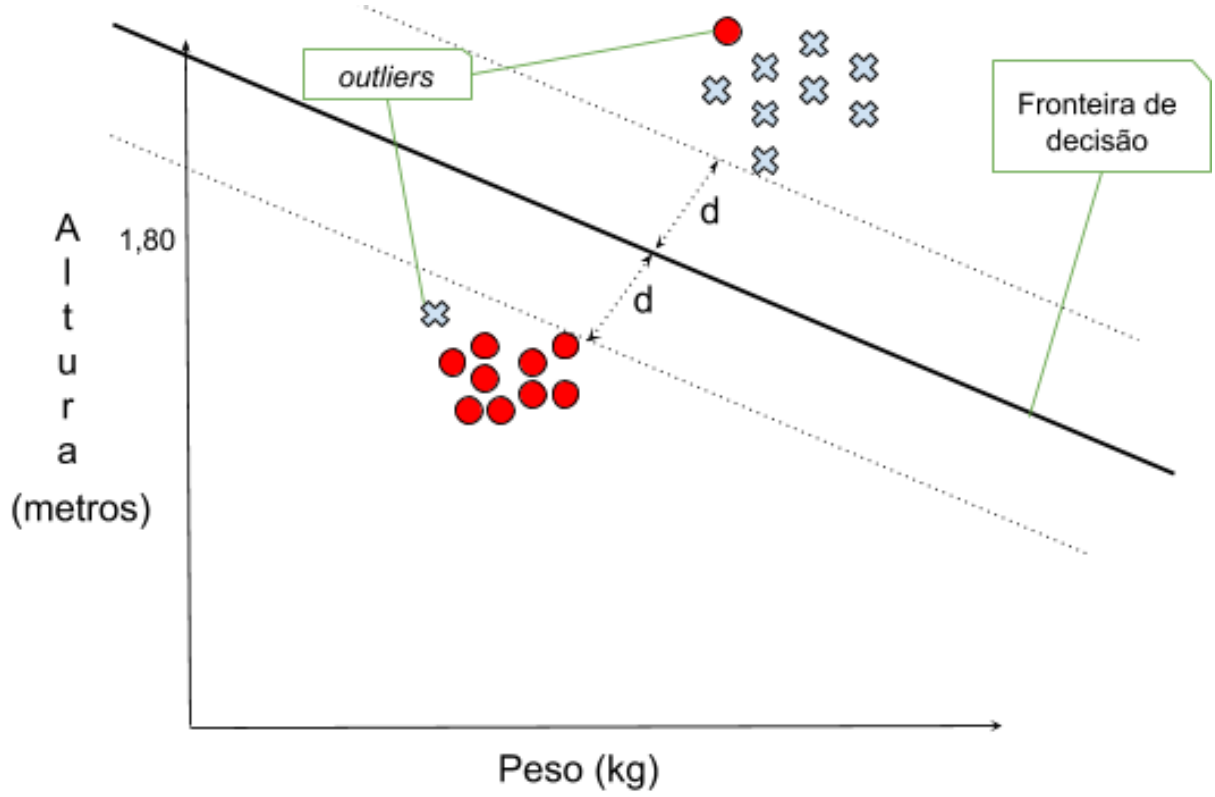


Figura 2.4: Exemplo da estratégia utilizada pelo algoritmo SVM com a ocorrência de *outliers*.
Fonte: Elaborado pelo autor.

2.3.2 *Árvores de Decisão e Florestas Randômicas*

As árvores de decisões, no contexto de aprendizado de máquina, são estruturas onde uma árvore de busca é construída de forma que os nós folhas representam prováveis classes do problema, e os demais nós representam valores relacionados às características do problema que são utilizados para a comparação. A partir da raiz, os atributos da instância que se deseja classificar são comparados com os valores dos nós não folhas (relacionados a uma característica), e de acordo com o resultado da comparação o fluxo da busca é alterado para algum dos ramos do nó.

A etapa de modelagem das árvores de decisão a partir de um conjunto de dados de treinamento é a mais complicada da técnica, que possui diversos tipos de implementação na literatura, e que em sua maioria consideram a distribuição dos dados como a entropia observada em cada característica das instâncias conhecidas.

Atualmente, a técnica de *ensemble*, que consiste na união de dois ou mais classificadores, é utilizada em muitos casos. A técnica considera conjuntos classificadores diferentes, ou várias instâncias de um mesmo classificador com parâmetros diferentes, como o *Random Forest* (Floresta Randômica). O *Random Forest* consiste na elaboração de várias árvores de decisão diferentes, geradas a partir da divisão do conjunto de dados em diferentes subconjuntos que contêm quantidades de características e instâncias da base de treino variadas.

O *Random Forest* tem como um de seus principais parâmetros o número de árvores que serão criadas. Apesar do processo de estimar o número de árvores não ser trivial, o algoritmo é usado amplamente na literatura por ter um bom desempenho ao tentar-se distinguir classes com fronteiras complexas como casos onde algumas das classes do problema precisam ser definidas por funções contínuas por partes.

2.3.3 *K-Nearest Neighbors* (KNN)

O algoritmo KNN é um algoritmo de aprendizado supervisionado, que é frequentemente utilizado para classificação ou para comparação com outras classificadores. Sua ideia central está na comparação da distância euclidiana entre as instâncias das classes. Apesar de simples, o algoritmo apresenta grandes taxas de acerto nas classificações de várias bases de dados.

Para classificar uma instância desconhecida, ainda não rotulada, o algoritmo realiza a distância euclidiana entre a instância desconhecida e todas as que já possuem uma classe conhecida. Após isso, o algoritmo seleciona os K pontos com menor distância, onde K é um parâmetro do algoritmo, e dentre estes pontos verifica-se a classe mais presente. Com isso, pode-se estimar uma classe à instância desconhecida e também uma probabilidade relacionada à classificação, onde esta probabilidade está relacionada à quantidade de vezes que a classe escolhida foi identificada nos K pontos escolhidos.

2.3.4 *Multilayer Perceptron* (MLP)

Multilayer Perceptron (MLP) é um algoritmo que pode ser usado no problema de classificação. A Figura 2.5 ilustra a estrutura de seu funcionamento, onde o algoritmo busca otimizar os pesos $w_0, w_1, ...w_N$ de cada neurônio (ou perceptron) de uma rede, de forma que as instâncias dos conjuntos de treinamento que possuem N dimensões tenham o erro de classificação minimizado. O cálculo do erro consiste na diferença entre os valores de saída das redes e os rótulos corretos de cada instância da base de treinamento.

O algoritmo executa de forma iterativa, dependendo da convergência em suas etapas de treinamento. Além disso, outros fatores fundamentais para o bom funcionamento deste algoritmo consiste na escolha da função σ presente na imagem mencionada, que é chamada de função de ativação dos neurônios. Também, o número de camadas e neurônios por camadas são parâmetros importantes para a convergência do classificador. Essas características permitem que o MLP aprenda fronteiras de decisões complexas entre diferentes classes de um problema de classificação.

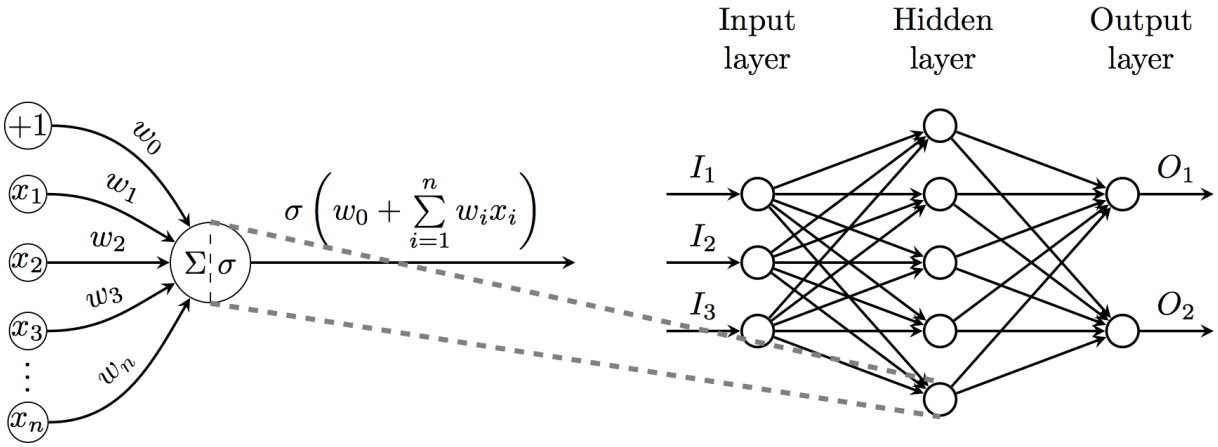


Figura 2.5: Ilustração da estrutura do algoritmo MLP.
Fonte: Retirado de
`https://github.com/PetarV-/TikZ/tree/master/Multilayer%20perceptron`.

2.3.5 *Naive Bayes Classifier*

O algoritmo de aprendizado de máquina *Naive Bayes* utiliza o conceito de inferência bayesiana para realizar as classificações. A Equação 2.3 apresenta o cálculo necessário para estimar a probabilidade de uma instância com valores $x1, \ldots, x_n$ pertencer a uma classe C_k.

Apesar de simples, o algoritmo funciona bem em diversas situações, porém, é considerado 'ingênuo" por assumir que as características do problema são independentes, característica necessária nas etapas em que os conceitos de inferência bayesiana são utilizados.

$$p(C_k \mid x_1, \ldots, x_n) = p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k), \tag{2.3}$$

2.3.6 *Quadratic Discriminant Analysis* (QDA)

O classificador QDA é um classificador que busca encontrar uma fronteira de decisão quadrática. A Figura 2.6 ilustra uma situação onde uma fronteira de decisão definida por uma função quadrática pode ter vantagens ao tentar diferenciar duas classes em um problema de classificação em relação a um modelo de classificação linear simples.

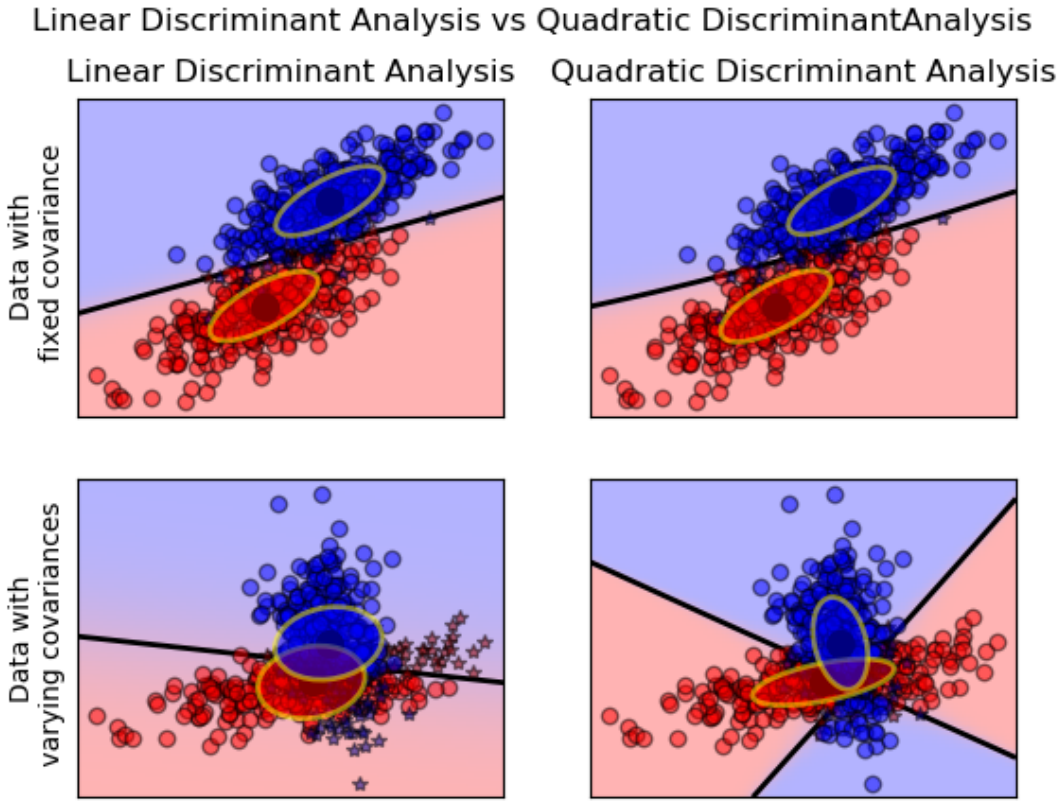


Figura 2.6: Ilustração da estrutura do algoritmo QDA.
Fonte: Retirado de
`https://scikit-learn.org/stable/modules/lda_qda.html#lda-qda`.

2.4 Redução de características

É comum que problemas de classificações apresentem diversas características que são desnecessárias ou que serão desconsideradas pelos classificadores, podendo prejudicar o modelo de classificação criados pelos mesmos. Com isso, diversas técnicas de seleção e transformação de características foram propostas de forma que possam ser utilizadas como pré-processamento dos dados com o objetivo de melhorar os desempenhos dos classificadores.

2.4.1 *Kernel Trick*

As transformações de *kernel* consistem em mapear as dimensões de um espaço em outro, e em aprendizado de máquina é comum o seu uso em casos onde os dados estão dispostos de forma que as classes do problema não são linearmente separáveis, prejudicando o uso de algoritmos como o SVM.

Nesses casos, a técnica é também chamada de *kernel trick*, e pode mapear as instâncias do problema de classificação de um espaço para outro que não terá necessariamente o mesmo número de características. Entretanto, encontrar um *kernel* de transformação de um espaço em que os dados não são linearmente separáveis para um espaço em que as instâncias podem ser separadas linearmente pode se tornar um problema de otimização, sendo assim frequente a implementação de algoritmos de aproximação ou heurísticas para resolver tal problema.

2.4.2 *Radial Basis Function*

Um dos tipos de *kernels* frequentemente utilizados é o kernel RBF (*Radial Basis Function*). Nele, as características são mapeadas de acordo com uma função gaussiana, explorando a relação entre as características. A Figura 2.7 ilustra um caso de transformação de um espaço do $\mathbb{R}^2$ onde duas classes que não podem ser separadas linearmente é transformado para um espaço do $\mathbb{R}^3$ por meio de uma transformação gaussiana onde os dados podem ser separados por um plano. Um parâmetro importante para esta transformação é a escolha do valor γ, utilizado para construir a distribuição gaussiana.

2.4.3 SVM-RFE (*SVM Recursive Feature Elimination*)

O SVM-RFE é uma técnica de seleção de características onde o algoritmo SVM é utilizado várias vezes no conjunto de dados (ou parte dele) recursivamente da seguinte forma: cria-se o hiperplano para separar as classes do problema; computa-se a ordenação dos coeficientes do hiperplano considerando as dimensões mais relevantes para o processo de separação; elimina-se as dimensões menos relevantes; inicia-se o próximo passo da recursão com uma dimensão a menos.

O algoritmo pode exigir um grande custo computacional em alguns casos, podendo ser flexibilizado ao modificar o tamanho do passo de cada recursão, onde o tamanho do passo representa o números de características eliminadas a cada etapa. É comum observar o uso desse algoritmo em problemas de classificação de expressões gênicas e seleções de genes, o que pode ser explicado pelo fato de que esses problemas envolvem muitas vezes uma quantidade de genes maior que um humano poderia analisar sem nenhum tipo de filtragem.

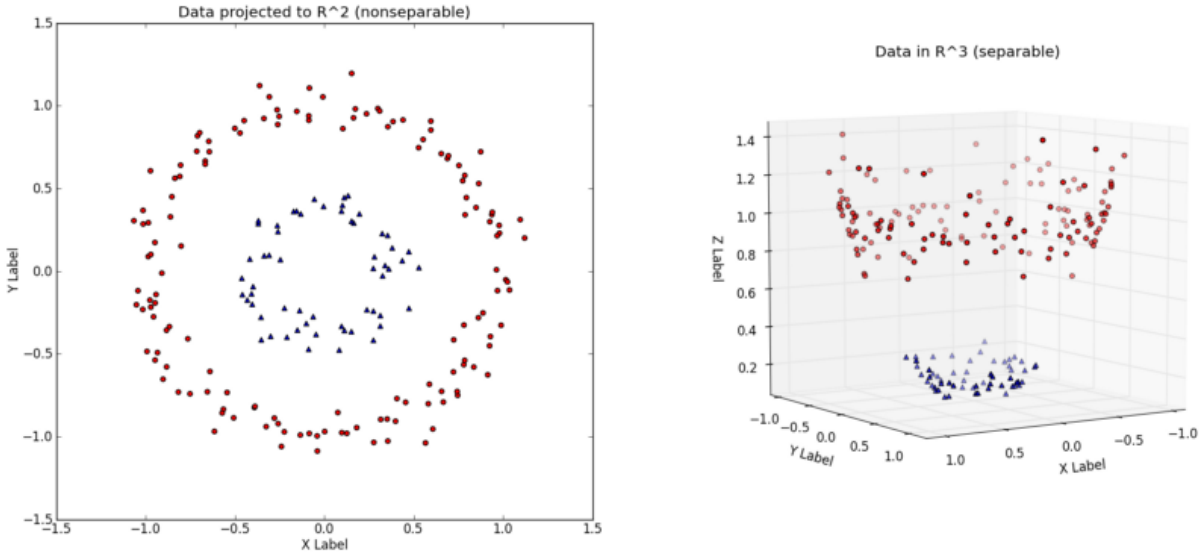


Figura 2.7: Exemplo da transformada de *kernel* RBF.
Fonte: Retirado de
`https://medium.com/@vivek.yadav/how-neural-networks-learn-nonlinear-functions-and-classify-linearly-non-separable-data-22328e7e5be1`.

3 Revisão Bibliográfica

3.1 Aprendizado de Máquina e Bioinformática

Predição ou classificação é um problema dentro área de aprendizagem de máquina que é explorado atualmente em diversas aplicações e situações. Pode-se utilizar ferramentas computacionais para a predição de câncer, estimar genes relacionados a determinadas doenças, entre outros assim como mostrado por Baldi e Brunak (2001) e Larranaga et al. (2006). Para que essas aplicações possuam um bom resultado, é necessário um conjunto relativamente grande de base de dados.

Segundo Touw et al. (2012), na área da bioinformática a ciência dos "omics" é uma fonte considerável de geração de dados, assim como pode ser visto em *genomics*, *proteomics*, entre outras. A análise desses dados pode muitas vezes gerar informações e conhecimentos ou confirmar teorias e sensos comuns (ainda não provados ou confirmados estatisticamente) como a predição da relevância da ontologia de genes e interações de componentes químicos na predição da expectativa de vida como mostrado por Barardo et al. (2017).

O uso de algoritmos de aprendizado de máquina supervisionados podem servir de apoio a suposições devido aos seus embasamentos estatísticos e ao fato de que os mesmos possam provir um modelo interpretável para a identificação das principais características em um problema de classificação. Entretanto, a escolha dos genes que devem ser utilizados para caracterizar e classificar amostras de expressões gênicas pode não ser uma tarefa trivial e que causa grande influência nos algoritmos de classificação, como mostrado no trabalho de Statnikov et al. (2004), onde são apresentados alguns métodos de seleção de características em problemas de classificação de expressão gênica, que são amplamente utilizados para melhorar os resultados e evitar *overfitting* (especifidade da solução para apenas os dados utilizados) em alguns casos.

Como ilustra a Figura 3.1 a seguir, a expressão de genes em um estudo pode ser representada em um formato de matriz. Ao analisar essas matrizes, podemos observar correlações entre expressões gênicas devido ao processo natural de casos onde genes fazem parte de uma mesma via metabólica e/ou fazem parte de um mecanismo de regulação ou expressão gênica. Há casos onde se considera até 15000 genes em um único estudo como mostrado por Statnikov et al. (2004), o que pode acabar influenciando na etapa de classificação e assim, fazendo necessário o processo de seleção de genes.

Samples-->	T1	T2	T3	T4	T5	T6	T7	N1	N2	N3	N4	*P* value
Genes	Expression level relative to non-tumor pool											
Gene 1	2.4		2	2.5	1.5	2	2.3	1.7	1	1	0.81	9.5E-06
Gene 2	2.9	3.2	1.4	1.7	2.6	3.7	2.5	1	0.9	0.7	0.7	8.0E-05
Gene 3	2	3.5	1.4	1.8	2	2.5	2.2	0.7	1	0.7	0.6	8.6E-05
Gene 4	2.2	2.2	2.7	1.3	2.3	3.7	1.7	0.7	0.9	0.9	0.9	0.00012
Gene 5	5.2	2	3.7	2	5.8	3.2	1.6	0.7	0.7	0.7	0.6	0.00014
Gene 6	6.9	15	18	5.8	12	21	2.3	0.9	1.6	0.7	1.2	0.00015
Gene 7	5.4	2.1	3	2.2	3.5	2.8	1.5	0.8	1.2	0.9	0.8	0.00022
Gene 8	3.1	2.3	1.8	1.7	2.9	1.5	1.2	0.7	0.7	1	0.7	0.00023
Gene 9	9.7	25	23	6.1	9.5	23	2.4	1.2	1.6	0.8	1.1	0.00024
Gene 10	7.6	14	13	4.7	8.2	24	2.1	0.9	1	0.8	1.1	0.00025
Gene 11	4.8	7.7	2.1	2.3	6.6	3.7	7.4	1.1	1.2	0.6	1.2	0.00028
Gene 12	3.6	5.7	3.8	3.3	4.7	6.6	1.8	1.7	0.9	1.1	1	0.00029
Gene 13	5.7	9.8	12	4.5	6	17	1.7	1	1.3	0.8	0.8	0.00031
Gene 14	1.5	2.1	1	1.1	1.2	1.2	1.4	0.6	0.8	0.6	0.8	0.00031
Gene 15	2.5	2.9	1.9	1.8	5.5	2	1.3	1	0.8	0.7	0.7	0.0004
Gene 16	2.2	1.5	1.3	1.2	1.4	1.8	1.1	0.8	0.8	0.8	0.8	0.00042
Gene 17	5.9	2	3.4	2.5	4.3	3.1	2.1	1.2	1.3	1.5	1	0.00048
Gene 18	4	1.6	2.8	1.4	2.9	2.2	1.6	0.9	0.9	1	0.8	0.00052
Gene 19	1.6	1.5	2.3	1.4	1.4	1.8	1.6	1.1	1.2	1	1.1	0.00059
Gene 20	3.9	6.7	6.6	2.3	4	11	1.5	0.8	1.1	0.8	0.8	0.00059
Gene 21	5.3	1.8	2.6	1.4	3.4	2.2	1.6	0.7	0.8	0.8	0.8	0.0006
Gene 22	4.2	1.9	1	2	4.2	4.3	7.9	16	1.1	1.3	10.9	0.00061
Gene 23	2.3	1.3	2.5	1.8	5.7	2.2	1.8	1.1	0.7	0.6	0.7	0.00066
Gene 24	2.9	1	2.9	2.2	3.8	1.9	2.3	0.9	0.8	0.9	0.8	0.00071
Gene 25	2.6	1.4	1.7	1.4	2.4	1.7	1.3	0.9	0.9	0.9	0.9	0.00079
Gene 26	5.8	2.3	3.4	2.1	5	4.7	1.7	1.2	0.9	1.2	0.6	0.0009
Gene 27	5.7	2	4	2.4	5	3.5	1.5	0.8	1.2	1.3	1.1	0.00093
Gene 28	1.6	2.8	1.7	1.7	1.5	2.8	1.9	1.2	1.1	0.8	1.1	0.00094

trends in Biotechnology

Figura 3.1: Exemplo de resultados de matrizes de expressão gênica.
Fonte: Retirado de (Zweiger, 1999, Figure 3).

3.2 Análise de expressão gênica

A análise de expressão gênica por meio do processamento de dados provindos de *microarrays* DNA é um problema abordado por diversos trabalhos, onde frequentemente se utiliza os algoritmos SVM e *Random Forest* (Ramaswamy et al. (2001), Yeoh et al. (2002), Brown et al. (2000), Statnikov et al. (2008)). Esses estudos em sua maioria enfrentam o problema de identificar a correlação da expressão de diferentes genes de amostras coletadas de diversas situações, como a de diferentes tipos de cânceres.

Nesse tipo de problema, é comum encontrar situações onde deseja-se aplicar algoritmos de aprendizagem de máquina supervisionado para classificar diversas caracterizações de expressões. Devido à natureza da dificuldade da coleta desses tipos dados é comum que tais trabalhos possuam cenários onde o número de instâncias é consideravelmente menor que a dimensionalidade do problema, ou seja, o número de características (genes) é muito maior que o número de instâncias (amostras).

3.3 *Deep Learning* e classificação de expressão gênica.

Recentes trabalhos como o de Singh et al. (2016) utilizam técnicas de *Deep Learning* (ou aprendizagem profunda), para resolver problemas de inferência e predição relacionados à analise de expressão gênica. *Deep Learning* é uma técnica que estende o conceito de redes neurais, apresentado na Subseção 2.3.4, podendo agregar diferentes conceitos como a operação de convolução, formando as CNN (*Convolutional Neural Networks*), apresentadas no trabalho

citado. Apesar do autor desse trabalho apresentar e discutir bons resultados utilizando esta técnica para resolver o problema de predição de expressão gênica em um contexto específico, a técnica necessita de uma quantidade significativa de dados além dos dados presentes nas matrizes de expressão gênica. No caso do trabalho de Singh et al. (2016), foram utilizadas informações relacionadas a códigos genéticos de DNA para realizar as etapas de predição.

Além da necessidade de uma quantidade grande e variada de dados, outra limitação das técnicas baseadas em *Deep Learning* é a dificuldade da interpretação dos modelos construídos pelas mesmas, como apresentado no trabalho de Chen et al. (2016), que pode ser um aspecto necessário no estudo de alguns casos. Contudo, até o conhecimento do autor deste trabalho, a aprendizagem profunda é uma técnica relativamente recente quando utilizada para resolver problemas de análise de expressão gênica, e que apresentam resultados interessantes nos estudos encontrados. Também, é importante considerar a capacidade das topologias das redes neurais envolvidas nessas redes poderem correlacionar as diferentes dimensões (genes) nos problemas de análise de classificação.

3.4 *Nearest Convex Hull Classification*

O uso do classificador NCHC (Nearest Convex Hull Classification), proposto por Nalbantov et al. (2006), se baseia na ideia de encontrar um convexo para cada classe de um conjunto de dados, assumindo que as mesmas são separáveis linearmente. A figura 1 apresentada no Capítulo 1 ilustra o funcionamento do algoritmo, que inclui a computação dos fechos convexos dos pontos de cada classe, que pode ser realizada por diferentes algoritmos como o QuickHull (Barber et al. (1996)). Assim como também explicado no Capitulo 1 na Figura 1.3, o convexo gerado por esta etapa pode influenciar no desempenho do algoritmo NCHC, uma vez que as faces geradas podem ser diferentes caso o algoritmo utilizado para computar o fecho convexo tenha sua solução aproximada, o que pode ser necessário em situações onde o problema possui uma alta dimensionalidade, considerando que o custo computacional para computar o fecho convexo de um conjunto de pontos é exponencial em relação ao número de dimensões.

Uma vez que as faces dos convexos são computadas, é possível calcular a distância de um ponto qualquer a esses convexos. Essa distância é calculada a partir da computação da distância do ponto para cada uma das faces dos convexos, que pode ser feita a partir do produto vetorial dos valores do ponto e dos vetores normais dos hiperplanos que representam as faces. Com isso, a menor distância dentre as distâncias das faces ao ponto em questão será a distância entre o ponto e o convexo. Assim que o algoritmo tem a distância entre o ponto que está sendo classificado e os convexos que representam as diferentes classes, estima-se como a provável classe do ponto a classe do convexo que o mesmo é mais próximo.

No trabalho referido, o NCHC foi comparado ao SVM considerando duas situações possíveis em relação à base de dados: casos em que as classes são separáveis e casos onde as classes não são separáveis.

Os casos separáveis são considerados aqueles onde os convexos gerados para cada classe não possuem interseção. Nesses casos, Nalbantov et al. (2006) conclui que o SVM padrão poderia resolver da mesma maneira que o NCHC. Porém, em casos onde existe uma sobreposição entre os convexos de duas ou mais classes, o autor conclui que a formulação do SVM não pode ser aplicada automaticamente para resolver o problema de encontrar convexo mais próximo.

Para casos não separáveis, diversas técnicas já são encontradas na literatura para a solução do problema. Uma das soluções consiste em encontrar um *kernel* tal que seja possível mapear as características (ou dimensões) em um novo espaço (não necessariamente com o mesmo número de dimensões). No trabalho de Nalbantov et al. (2006) foram utilizados três tipos de

transformações na base de dados para o SVM e o NCHC em 6 bases de dados diferentes e os resultados demonstraram um aumento na taxa de predição correta ao se utilizar o *kernel* RBF.

No trabalho de Zhou e Shi (2009), vemos o uso de um algoritmo semelhante ao NCHC apresentado por Nalbantov et al. (2006) para a classificação de imagens, onde o tal algoritmo foi comparado com o KNN, com $K = 1$, demonstrando algumas características em comum entre os dois algoritmos como o uso da distância euclidiana entre o ponto que se deseja classificar e alguma outra estrutura que representa as classes no espaço do problema. Trabalhos recentes como o de Chen et al. (2014) também exploram a combinação de diferentes técnicas, como a redes neurais recorrentes, junto ao NCHC, demonstrando bons resultados quando submetidos a algumas situações específicas.

É possível perceber a semelhança entre o NCHC e o SVM, ao notar-se que ambos algoritmos buscam estabelecer hiperplanos que modelam as fronteiras de cada classe envolvida no problema. Porém, percebe-se que o SVM busca construir apenas um hiperplano, que se localiza entre as diferentes classes, e o NCHC busca construir diversos hiperplanos, que representam as faces dos convexos, que limitam as classes de forma mais específica.

Um dos aspectos em comum entre o NCHC e o KNN, é a característica do KNN buscar encontrar os pontos rotulados mais próximos das classes próximas aos pontos que são classificados. Essa característica também acontece no NCHC, uma vez que as faces próximas dos pontos classificados representam os pontos das classes próximas. Porém, em situações onde existe grande distâncias entre os pontos de uma classe, que pode acontecer em situações que essas classes possuem poucas instâncias, pode haver diferenças entre o NCHC e KNN, uma vez que o NCHC possui capacidade de interpolar e considerar o espaço entre os pontos das classes, enquanto o KNN trata cada ponto rotulado de maneira individual, sem buscar estabelecer nenhum tipo de correlação entre esses em relação ao hiperespaço considerado.

4 Proposta

4.1 Objetivos

Como mencionado nos capítulos anteriores, diversos parâmetros são necessários para a utilização de vários algoritmos de classificação. Este trabalho se propõe a analisar o impacto da modificação de parâmetros na taxa de acerto das predições de diferentes classificadores, incluindo o NCHC, aplicados a diferentes bases de dados. Também, espera-se identificar o comportamento do algoritmo NCHC quando utilizado em diferentes situações, como diferentes bases de dados com dados pré-tratados de diferentes formas.

4.1.1 Objetivos Específicos

Dentre os objetivos específicos, deseja-se analisar o impacto do classificador NCHC e sua comparação com outros algoritmos de classificação, em principal o SVM devido ao seu amplo uso observado na literatura em diferentes situações. Os seguintes cenários serão considerados:

- Diferentes formas de transformação dos dados anteriormente ao uso dos algoritmos de classificação.
- Uso da transformação de *kernel* RBF, utilizando diferentes variações do parâmetro γ.
- Aplicação dos algoritmos em diferentes bases de dados, com natureza, número de instâncias, número de classes e dimensionalidades diferentes.

4.2 Materiais e métodos

Este trabalho busca avaliar o desempenho do classificador NCHC em diferentes base de dados. Também busca-se identificar os motivos das diferenças identificadas durante a execução do algoritmo para os diferentes conjuntos de dados adquiridos por meio da biblioteca *scikit-learn* (Pedregosa et al. (2011)) e também duas das bases de dados utilizadas por Statnikov et al. (2008).

4.2.1 Bases de dados

As bases de dados utilizadas são descritas a seguir:

- **Iris**: Conjunto de dados que representa o tamanho das pétalas e sépalas de diferentes amostras separadas em três tipos de flores do gênero Iris. Cada classe possui 50 instâncias descritas por quatro características.

- ***Breast cancer*** **(Câncer de mama)**: Conjunto de dados que representa imagens de células que pertencem a uma das duas classes do conjunto: maligno e benigno. A base contém 357 instâncias da classe "maligno"e 212 instâncias da classe "benigno", onde cada uma é representada por 30 características que descrevem informações já processadas em relação ao núcleo das células presentes nas imagens.

- ***Lung cancer*** **(Câncer de pulmão)**: Conjunto de dados que representa a expressão de 12600 genes de 203 amostras separada em cinco classes: 139 amostras normais e quatro tipos de câncer, os quais possuem 17, 21, 20 e 6 amostras dentre as classes.

- ***Prostate tumor*** **(Câncer de próstata)**: Conjunto de dados que representa a expressão de 10509 genes de 102 amostras separada em duas classes: 52 instâncias representando amostras normais e 50 amostras que representam amostras cancerígenas.

4.2.2 Implementação do classificador NCHC

A biblioteca Scipy (Jones et al. (2014)) da linguagem de programação Python contém diversos implementações eficientes de métodos de computação científica e algoritmos como o QuickHull (Barber et al. (1996)) que encontra o convexo a partir de um conjunto de pontos em um espaço de dimensões variadas. Utilizando-se a implementação desta biblioteca, uma versão do NCHC foi implementada da seguinte forma:

Algoritmo 1 Nearest Convex Hull Classification

Entrada:
C : Número total de classes do problema $D_{i,j}$: Matriz $M \times N$ que representa as instâncias do problema ,
Y_i : Vetor de tamanho M que contém as classes de cada instância tal que $Y_i \geq 0$ e $Y_i < C$,
x_j : Vetor de tamanho N que representa uma instância sem classe

Saída: Probabilidade do ponto sem classe pertencer a cada classe do problema

```
1: for i = 0 to N do
2:     hull(i) ← ConvexHull(D_i)   // Cria um convexo para cada classe
3:     p(i) ← 0
4:     candidatos(i) ← false
5: end for
6: for i = 0 to N do
7:     if dentro(hull(i), x) then
8:         candidatos(i) ← true
9:     end if
10: end for
11: if conta(candidatos) = 1 then
12:     i ← índice da classe candidata única(candidatos)
13:     p(i) ← 1
14: else
15:     for i = 0 to conta(candidatos) do
16:         d ← distância(x, hull(i))
17:         p(i) ← 1/d
18:     end for
19:     p ← normaliza(p)
20: end if
21:
22: return p
```

Dentre os trabalhos apresentados na seção 3.4, não foi possível encontrar detalhes das implementações do NCHC. Portanto, as comparações com outras produções foram dificultadas na realização do presente trabalho.

4.2.3 Experimentos

Considerando as diferenças entre as bases de dados, diferentes comparações foram feitas em relação os desempenhos dos classificadores. As bases de dados foram divididas entre treinamento e validação de acordo com o método *Cross Validation* (validação cruzada) onde a base é dividida em K *folds*, de maneira que cada *fold* contém um subconjunto das instâncias, com quantidades de classes proporcionais ao conjunto total e os *folds* possuem tamanhos equivalentes entre si.

Com isso, os classificadores têm suas acurácias (taxas de acerto) quantificadas de acordo com a média da taxa de acerto para o resultado da taxa de acerto individual ao se utilizar um *fold* como treinamento e os seus complementares como validação. A Figura 4.1 ilustra um caso onde uma base de dados é separada em 5 *folds* e 5 experimentos, e para cada experimento utiliza um dos *folds* como base de validação e os demais como base de treino. Ao final, os experimentos

tem seus resultados agregados a partir da média do desempenho dos classificadores utilizando as bases de treino e validação construídas. Neste trabalho, valor de K foi adotado como 6 devido ao fato de ser o maior valor possível para a divisão da base *Lung Cancer* onde uma das classes possui apenas 6 instâncias.

Anteriormente às etapas de classificações realizadas nos experimentos, as características de cada problema foram normalizadas de forma que os valores das características de qualquer instância da base pertencesse ao intervalo $[-1, 1]$.

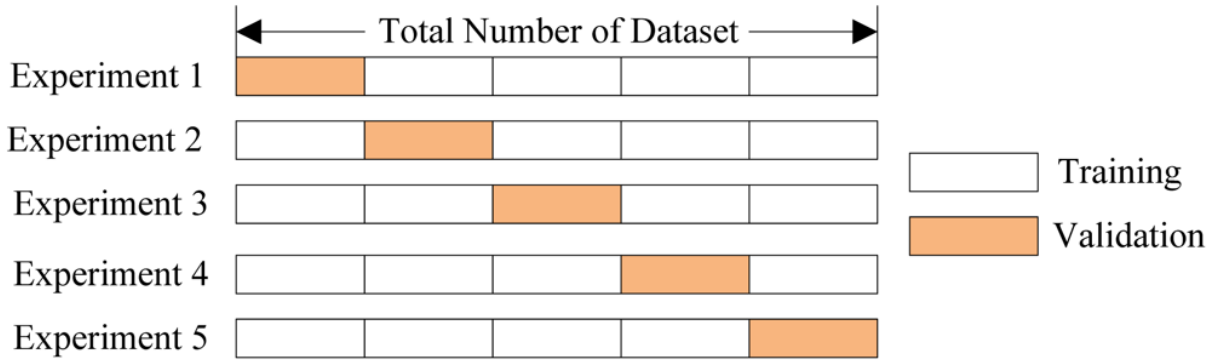


Figura 4.1: Ilustração do funcionamento da técnica *Cross Validation*.
Fonte: Retirado de `https://www.kaggle.com/dansbecker/cross-validation`.

Comparação de classificadores

Durante os experimentos realizados, o classificador NCHC implementado foi comparado com outros classificadores. Dentre os classificadores utilizados como comparação foram escolhidos:

- *Linear* SVM (*Support Vector Machine*)
- KNN (*K-Nearest Neighbors*), com valor de $K = 1$
- RDF (*Random Forest*
- Árvore de decisão
- MLP (*Multilayer Perceptron*)
- AdaBoost
- *Naive Bayes*
- QDA

O classificador AdaBoost consiste em uma técnica de *ensemble* onde vários classificadores fracos são utilizados para realizar as classificações. Na construção do modelo deste algoritmo, os classificadores fracos são adicionados um a um, de forma incremental por uma quantidade finita de iterações, de maneira que um conjunto de pesos são estabelecidos para minimizar um erro calculado para a iteração anterior. Neste trabalho, foi utilizado o classificador implementado na biblioteca Scikit-Learn para compor o AdaBoost, que consiste em árvores de decisão de altura 1. Portanto, a implementação utilizada é similar ao *Random Forest*, mas com algumas particularidades.

Dentre os classificadores estudados, nenhuma técnica de *Deep Learning* foi abordada. Como apresentado no Capítulo 3, existem restrições para o uso de técnicas baseadas em *Deep Learning*. Até o melhor conhecimento do autor deste trabalho, as bases de dados utilizadas, que são constituídas de pontos no hiperespaço, não são suficientes para o uso das técnicas observadas nos trabalhos citados. Com isso, na perspectiva do autor do presente trabalho, apenas o algoritmo MLP que inclui o conceito de redes neurais pode ser utilizado para realizar predições a partir das bases de dados abordadas.

As implementações presentes na biblioteca Scikit-Learn foram utilizadas para adquirir os resultados das comparações, não especificando quaisquer tipos de hiper-parâmetros iniciais, que serão atribuídos com os valores padrões da biblioteca. Alguns dos algoritmos como o MLP e o *Random Forest* dependem de critérios como *seeds* para gerar números aleatórios e a limitação de suas iterações, o que pode causar variações nos resultados das execuções de alguns experimentos.

Avaliação do parâmetro gamma para a função de kernel RBF

Assim como mostrado por Nalbantov et al. (2006), o uso do da transformada *kernel* RBF pode melhorar significativamente o resultado da acurácia do NCHC. Entretanto, a função depende do parâmetro γ que não foi especificado pelo autor do trabalho citado, o que foi explorado em parte dos experimentos desse trabalho. Os valores de γ utilizados foram do valor 0 a 10, com intervalo de 0.5, totalizando 20 execuções de cada classificador em cada *fold* da validação cruzada. Foi utilizado uma função disponibilizada na biblioteca Scikit-Learn que implementa uma aproximação da função de *kernel* RBF, que exige como parâmetro o número de dimensões que se deseja mapear, onde foi escolhido um espaço de 7 dimensões devido às limitações de recurso e tempo para os experimentos.

Seleção de atributos

O algoritmo SVM-RFE foi utilizado de forma que algumas das dimensões dos problemas de classificações fossem descartadas. Diferentes números de atributos selecionados foram utilizados, de forma que os resultados das classificações não sejam necessariamente os mesmos, o que será apresentado no Capítulo 5 deste trabalho. Para cada base de dados foi escolhido um intervalo de características selecionadas pelo SVM-RFE devido às distribuições de instâncias por classes em cada base.

Devido à limitação de tempo para elaboração deste trabalho, somente a técnica SVM-RFE foi abordada. A mesma foi escolhida devido ao seu amplo uso e resultados significativos observados nos trabalhos apresentados no Capítulo 3.

5 Resultados e Discussão

Neste capítulo serão discutido os resultados obtidos a partir das propostas mencionadas no Capítulo 4. As figuras 5.1 e 5.5 apresentam comparações entre as acurácias dos classificadores mencionados nesse trabalho que obtiveram o melhor desempenho dentre os experimentos realizados. As figuras 5.2, 5.3 e 5.4 apresentam resultados de algumas das transformações obtidas pelo *kernel* RBF em algumas das bases de dados utilizadas. As figuras 5.6, 5.7, 5.9 e 5.8 ilustram as transformações feitas pelo SVM-RFE nas bases de dados ao selecionar três dimensões.

5.1 Desempenho de outros classificadores

A Tabela 5.1 demonstra as acurácias de alguns dos classificadores implementados na biblioteca Scikit-Learn quando utilizados nas bases de dados sem nenhum tipo de transformação ou redução de características. Observa-se que as bases *Lung cancer* e *Prostate tumor* apresentaram uma pior taxa de reconhecimento de maneira geral, o que pode ser explicado pelas quantidades de atributos presentes nessas bases, que são consideravelmente maior que nas demais bases. Outro detalhe interessante é que apesar da base *Lung cancer* apresentar mais classes que a *Prostate tumor*, a mesma teve taxas de acerto maiores.

Tabela 5.1: Acurácias de diferentes classificadores implementados pela biblioteca scikit-learn em quatro bases de dados distintas

	Iris	*Breast cancer*	*Lung cancer*	*Prostate tumor*
SVM	0.9660	**0.9718**	**0.9405**	**0.9133**
KNN	0.9591	0.9612	0.8719	0.82543
Random Forest	0.9591	0.9561	0.8808	0.8231
Árvore de decisão	0.9591	0.9228	0.8703	0.8416
MLP	**0.9722**	0.9631	0.9224	0.7864
AdaBoost	0.9591	0.9630	0.6230	0.9028
Naive Bayes	0.9529	0.9298	0.8953	0.6156
QDA	0.9652	0.9542	0.3998	0.4780

5.2 Implementação do NCHC

O algoritmo implementado é dependente das funções presentes da biblioteca utilizada para encontrar-se o *convex hull* dos conjuntos. Durante a implementação, foi verificado uma grande variedade de parâmetros que podem ser utilizados de forma que o convexo gerado seja modificado.

Uma limitação da implementação elaborada é o fato de que é necessário um conjunto de pontos maior que $N - 1$, sendo N o número de características, para a construção de qualquer convexo. Tal limitação influencia diretamente no uso do algoritmo em casos como os das bases de expressão gênica, que possuem grande números de características em comparação com as quantidades de instâncias do problema, criando-se a necessidade da redução de atributos. Outra limitação observada foi o tempo de execução do algoritmo para testar a distância de pontos aos convexos gerados, que é relacionado ao total de faces em cada convexo.

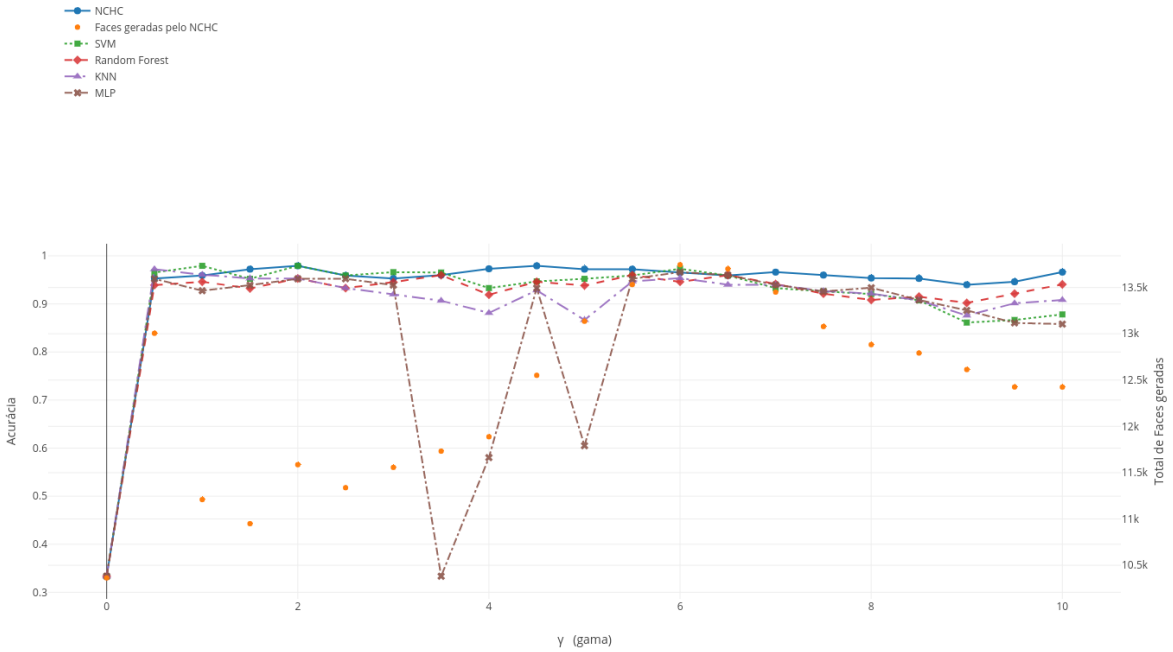
5.3 Kernel RBF

A função de *kernel* RBF apresentou resultados variados quando utilizada nas diferentes bases de dados. A Figura 5.1 ilustra os resultados da acurácia, calculada a partir da validação cruzada, dos classificadores utilizando diferentes valores de γ como parâmetro da transformação RBF. Os resultados dos classificadores na base de dados Iris mostrados na Figura 5.1(a) mostram um bom desempenho do classificador NCHC, atingindo a melhor taxa de acerto dentre as execuções em dois casos, tendo um desempenho geral melhor que os outros classificadores nos experimentos mostrados na imagem. Entretanto, os valores de γ testados e o número de dimensões utilizado pelo algoritmo de transformação foram escolhidos arbitrariamente sem o intuito de maximizar o desempenho dos classificadores, e portanto, um estudo mais especificado poderia ser feito para compreender melhor os resultados.

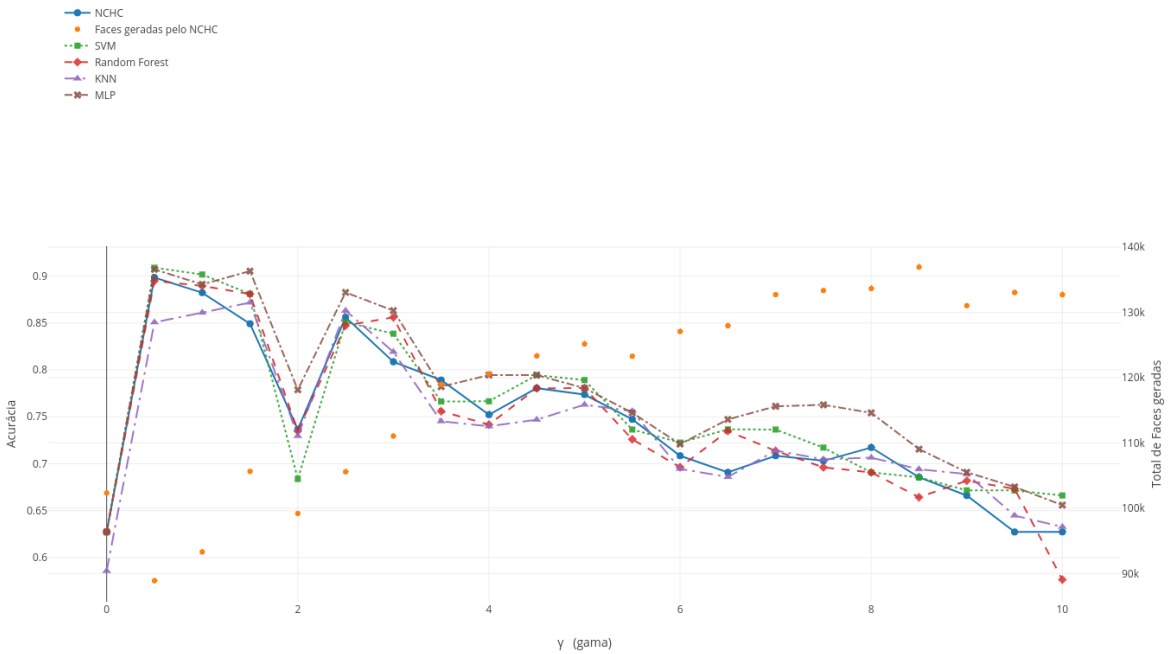
A transformação do espaço de características pelo *kernel* RBF apresentou a redução da acurácia dos classificadores quando utilizada nas bases de expressão gênica, o que pode ser visualizado na comparação dos resultados das figuras 5.1(c) e 5.1(d) e a Tabela 5.1. Tais resultados podem ser explicados devido ao fato de que a transformação pelo *kernel* RBF está sendo utilizada de forma que as informações de algumas dimensões são descartadas ao reduzir mais de 10000 dimensões das bases para apenas 7 dimensões, o que pode não ser adequado nesse problema. A Figura 5.1(c), que representa os resultados das taxas de acertos dos classificadores na base de dados *Prostate Tumor* transformada pela função de *kernel* RBF, apresenta uma situação interessante, onde a transformação realizada impactou os resultados dos classificadores de forma que a taxa de acerto foi próxima de 0.5. É importante considerar que a base de dados *Prostate Tumor* é constituída por apenas duas classes com quantidades de representantes igual a 52 e 50, sendo esperado que a acurácia seja próxima a 0.5 ao escolher aleatoriamente um dois dois possíveis rótulos para realizar as classificações.

As figuras 5.2, 5.3 e 5.4 ilustram a influência do valor γ na transformação de *kernel* RBF nas bases Iris, *Breast cancer* e *Lung cancer* para um espaço de três dimensões. Nas imagens, fica explícito um dos problemas que redução de dimensões pode causar nos conjuntos de dados, onde as interseções entre as classes aumentam. Percebe-se que a base de dados Iris, que possui apenas 4 características, é a menos afetada pela transformação, apesar de ter uma grande variação em relação ao valor γ.

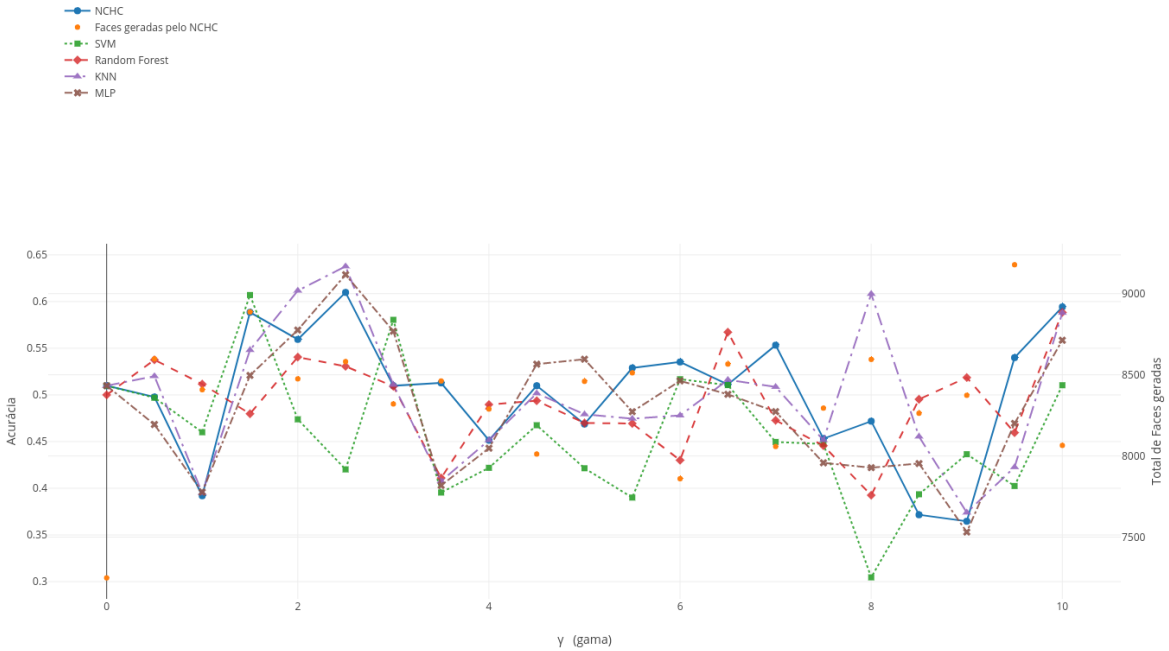
A Figura 5.1(a) também ilustra o número de faces geradas pelos convexos em cada execução. Percebe-se uma grande variação na quantidade de faces geradas, que influencia no tempo de execução do algoritmo, mas não identifica-se uma relação direta com a acurácia dos classificadores. Tal característica pode ser explicada pela hipótese de que a maior parte das faces geradas não são utilizadas ou não causam diferença no resultado da classificação. Ao considerar que o SVM consiste na construção de um único hiperplano para a classificação das instâncias, tal hipótese tem mais fundamento, o que pode ser estudado em futuros trabalhos.



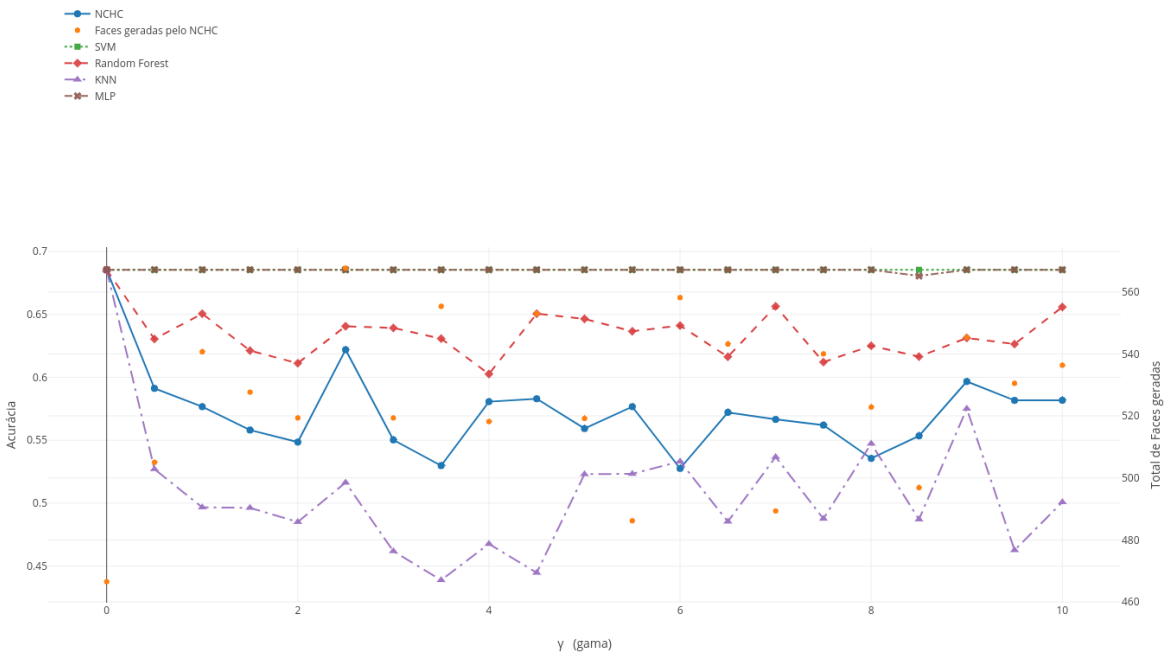
(a) Desempenho dos classificadores na base de dados Iris utilizando utilizando a função de kernel RBF para a transformação em 7 caracacterísticas.



(b) Desempenho dos classificadores na base de dados *Breast cancer* utilizando utilizando a função de kernel RBF para a transformação em 7 caracacterísticas.

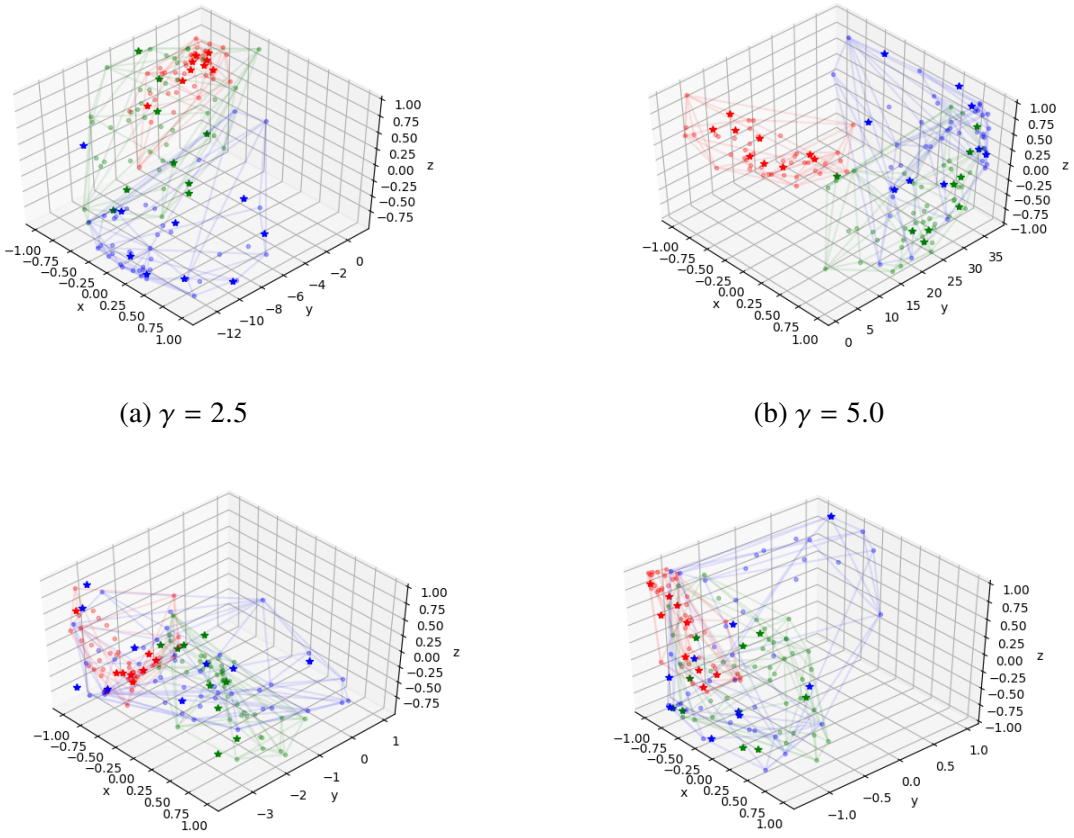


(c) Desempenho dos classificadores na base de dados *Prostate tumor* utilizando utilizando a função de kernel RBF para a transformação em 7 caracacterísticas.



(d) Desempenho dos classificadores na base de dados *Lung cancer* utilizando utilizando a função de kernel RBF para a transformação em 4 caracacterísticas.

Figura 5.1: Resultados de diferentes classificadores aplicados a diferentes bases de dados transformadas pela função de kernel RBF.
O eixo das abcissas representa as transformações de kernel RBF utilizando diferentes valores de γ. O eixo das ordenadas representa a acurácia dos classificadores e a média das faces para cada convexo gerado pelo NCHC. Fonte: Elaborado pelo autor.



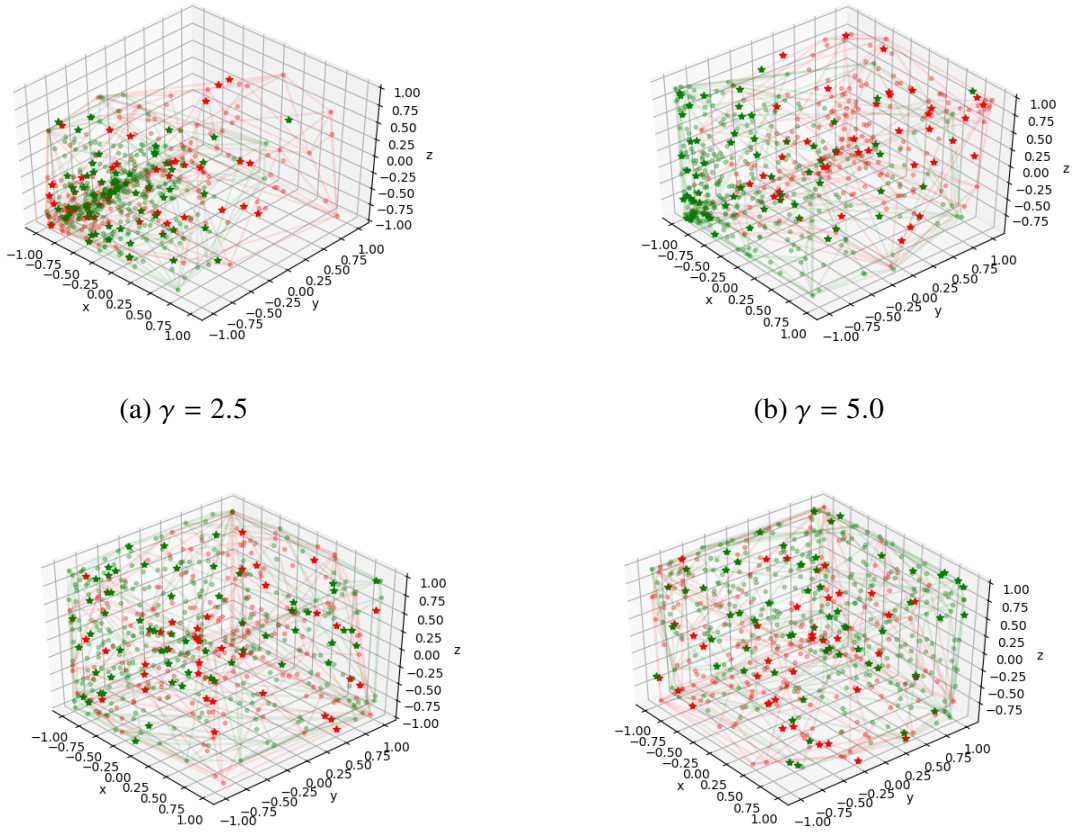
(a) $\gamma = 2.5$

(b) $\gamma = 5.0$

(c) $\gamma = 7.5$

(d) $\gamma = 10.0$

Figura 5.2: Influência do valor γ na função de transformação de *kernel* RBF na base Iris. Cada cor representa uma classe do problema. Os segmentos de retas fazem parte do convexo gerado pelo NCHC. Os pontos representam as instâncias da base de treino e as estrelas representam as instâncias de um *fold* que está sendo classificado. Fonte: Elaborado pelo autor.



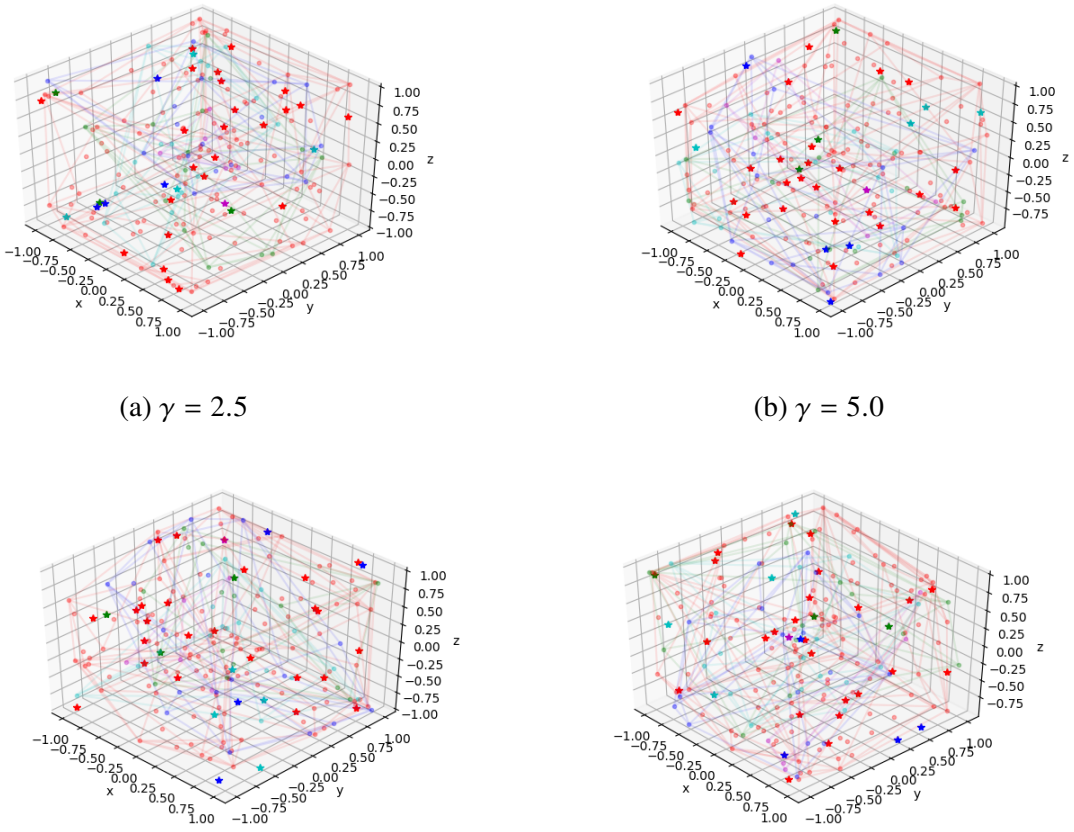
(a) $\gamma = 2.5$

(b) $\gamma = 5.0$

(c) $\gamma = 7.5$

(d) $\gamma = 10.0$

Figura 5.3: Influência do valor γ na função de transformação de kernel RBF na base *Breast cancer*. Cada cor representa uma classe do problema. Os segmentos de retas fazem parte do convexo gerado pelo NCHC. Os pontos representam as instâncias da base de treino e as estrelas representam as instâncias de um *fold* que está sendo classificado. Fonte: Elaborado pelo autor.



(a) $\gamma = 2.5$

(b) $\gamma = 5.0$

(c) $\gamma = 7.5$

(d) $\gamma = 10.0$

Figura 5.4: Influência do valor γ na função de transformação de *kernel* RBF na base *Lung cancer*. Cada cor representa uma classe do problema. Os segmentos de retas fazem parte do convexo gerado pelo NCHC. Os pontos representam as instâncias da base de treino e as estrelas representam as instâncias de um *fold* que está sendo classificado. Fonte: Elaborado pelo autor.

5.4 SVM-RFE

Como apresentado na Subseção 2.4.3, o algoritmo SVM-RFE seleciona um subconjunto de características, de forma que as características escolhidas sejam estimadas como as que possuem maior impacto para classificação. A escolha acontece eliminando-se recursivamente a característica estimada como menos significativa até que reste um subconjunto de características de tamanho igual ao especificado.

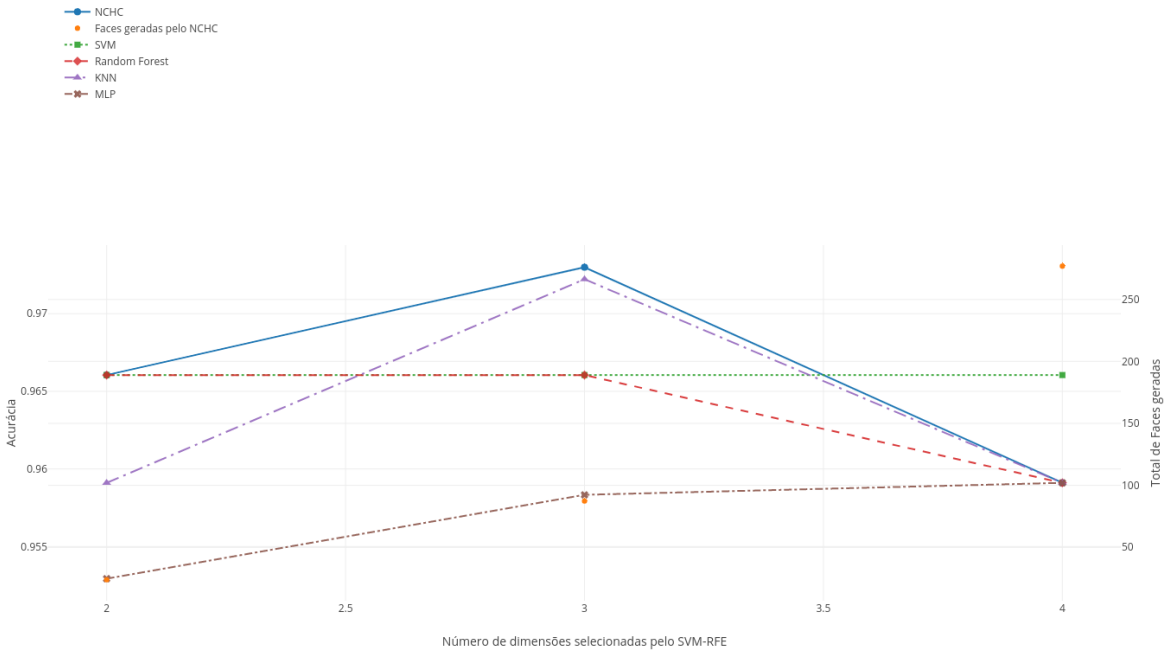
A Figura 5.5 ilustra a comparação das implementações de alguns classificadores da biblioteca Scikit-Learn com a implementação do classificador NCHC deste trabalho por meio de execuções dos mesmos nas bases de dados descritas anteriormente utilizando o algoritmo SVM-RFE como pré-processamento com variações no número de características selecionadas.

Na Imagem 5.5(a) observa-se o destaque do classificador NCHC na base de dados Iris em relação aos outros classificadores, e seu modelo também é ilustrado na Figura 5.6. A base contém apenas quatro características, e ao selecionar duas e três delas para a classificação, foi possível observar um comportamento semelhante entre o KNN e o NCHC. É importante ressaltar que os resultados foram obtidos por meio do uso das implementações da biblioteca Scikit-Learn sem a especificação de quaisquer parâmetros, com exceção do KNN que teve valor de K=1, que poderiam influenciar consideravelmente o desempenho dos classificadores.

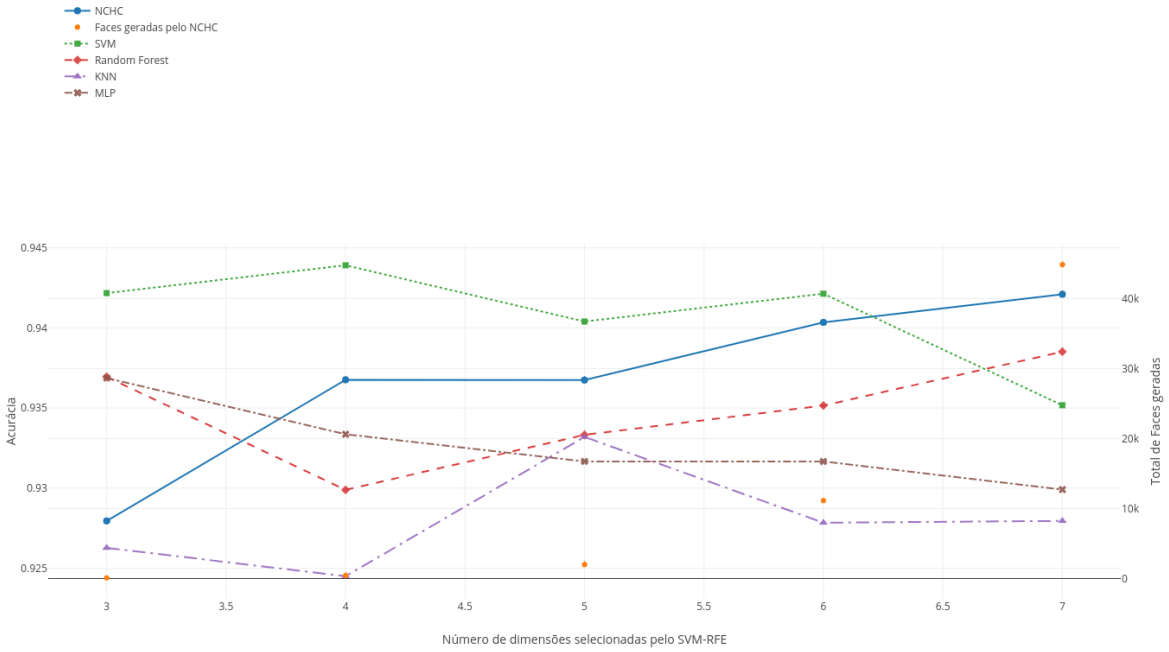
Por outro lado, vemos na Imagem 5.5(d) que o NCHC possui uma baixa taxa de acerto comparado aos outros classificadores em um problema com várias classes e poucos representantes por classe, como a base de dados *Lung cancer*. A Figura 5.8 mostra exemplos onde há uma grande interseção entre três das cinco classes do problema, que poderia ser eliminada ou diminuída com a adição de novas dimensões ou escolha de outras características para formar os convexos. Porém, tal problema pode não ter solução trivial dado que uma das dimensões escolhidas é essencial para diferenciar as três classes mencionadas de outras duas (em rosa e azul), e também há uma limitação da quantidade de dimensões selecionadas devido ao fato de que algumas das classes não possuem pontos suficientes para formar convexos em dimensões maiores que 5.

Os resultados obtidos também confirmam os resultados de diversos trabalhos que utilizam o SVM-RFE para a seleção de atributos em bases de seleção de expressão gênica com o objetivo de aumentar a taxa de acerto dos classificadores. Além disso, é comum que os trabalhos utilizem o SVM como classificador para predições de expressões gênicas devido ao seu bom desempenho em vários problemas de classificações sem ter grande influência do número de dimensões, o que pode ser também visualizado na Figura 5.5 e em específico na base *Prostate tumor*, que teve taxa de acerto máxima em alguns dos experimentos como mostrado na Figura 5.5(c), superando os resultados da Tabela 5.1 onde não houve o uso do SVM-RFE.

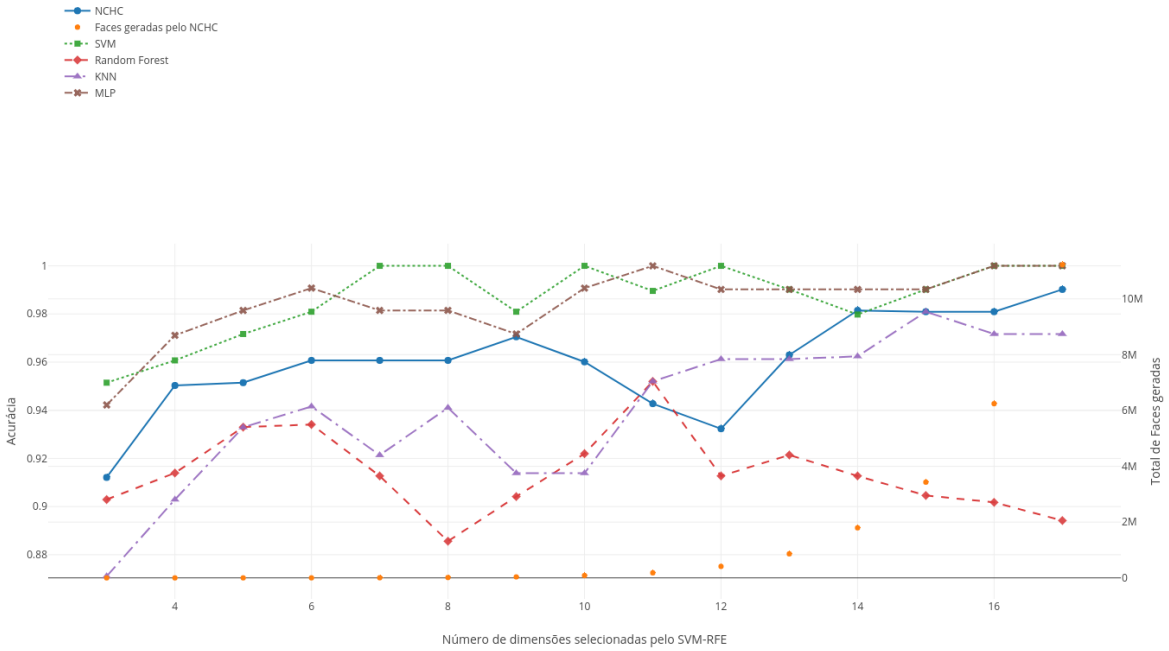
A imagem comentada anteriormente também apresenta uma característica relevante, que é o fato da quantidade de faces geradas pelos convexos apresentar um crescimento não linear em relação ao total de dimensões. A quantidade de faces dos fechos convexos que representam as instâncias de cada classe estão relacionadas ao tempo de execução da etapa de treinamento do algoritmo e também à etapa de teste onde as distâncias entre os convexos e outras instâncias devem ser calculadas. Portanto, a dificuldade do aumento de características selecionadas pelo SVM-RFE pode acabar sendo uma limitação do algoritmo NCHC que tem sua complexidade computacional relacionada à dimensionalidade do problema. Implementações que diminuam o custo computacional do NCHC podem auxiliar a eficiência de seu uso ao permitir que o mesmo seja comportado em problemas que necessitem de um grande número de dimensões.



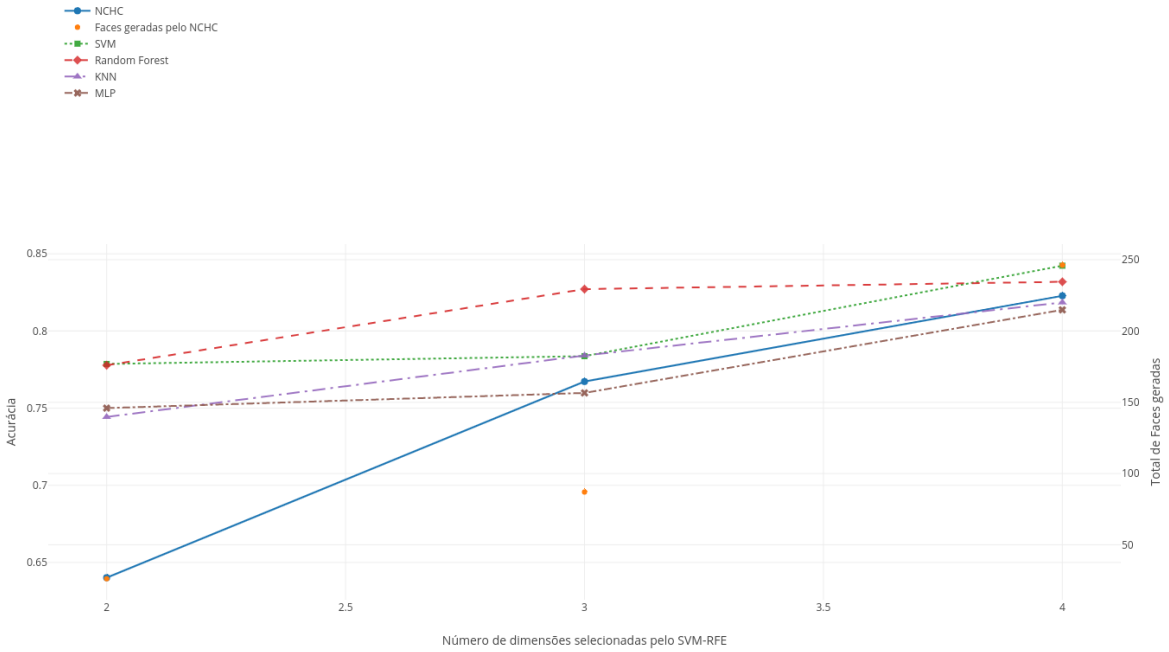
(a) Desempenho dos classificadores na base de dados Iris utilizando o SVM-RFE com diferentes números de características selecionadas.



(b) Desempenho dos classificadores na base de dados *Breast cancer* utilizando o SVM-RFE com diferentes números de características selecionadas.



(c) Desempenho dos classificadores na base de dados *Prostate tumor* utilizando o SVM-RFE com diferentes números de características selecionadas.



(d) Desempenho dos classificadores na base de dados *Lung cancer* utilizando o SVM-RFE com diferentes números de características selecionadas.

Figura 5.5: Resultados de diferentes classificadores aplicados a diferentes bases de dados transformadas pelo SVM-RFE.
Não foram especificados quaisquer parâmetros para os classificadores comparados. O eixo das abcissas representa a quantidade de características selecionadas pelo SVM-RFE. O eixo das ordenadas representa a acurácia dos classificadores e a média das faces para cada convexo gerado pelo NCHC. Fonte: Elaborado pelo autor.

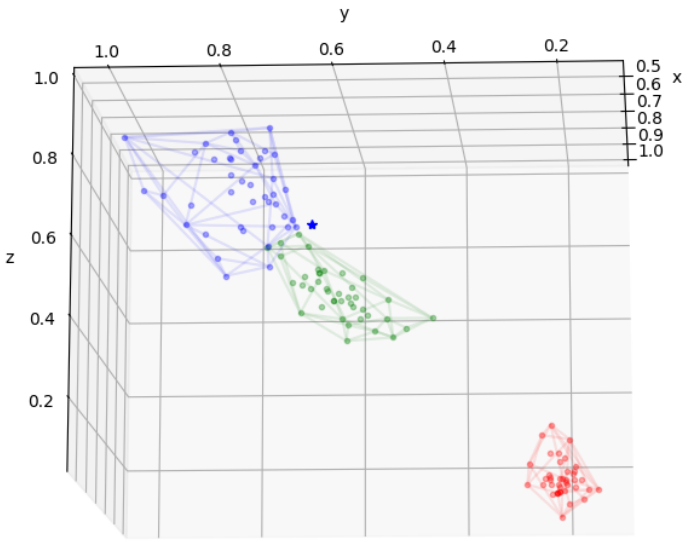
5.5 Interpretação geométrica

Nesta seção, serão apresentados os resultados das aplicações do classificador NCHC nas quatro bases de dados estudadas neste trabalho após a redução do espaço de características de cada uma, por meio do algoritmo SVM-RFE, para apenas três dimensões. Ou seja, escolheu-se três dimensões de cada base de forma que as mesmas possam ser bem utilizadas em uma etapa de classificação e visualizadas em um gráfico tridimensional posteriormente.

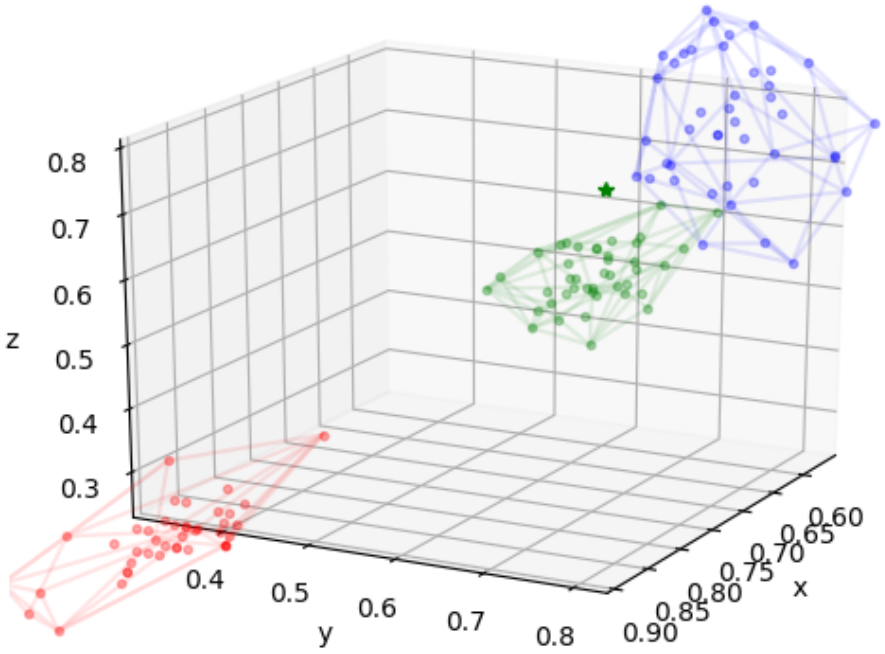
5.5.1 Iris

A base de dados Iris, composta por 3 classes, possui a característica de ter a mesma quantidade de representantes por cada classe. A Figura 5.6 ilustra o resultado da execução do algoritmo NCHC. As imagens da figura também apresentam um cenário onde fica explicito a diferença entre uma das classes (em vermelho) e as demais classes.

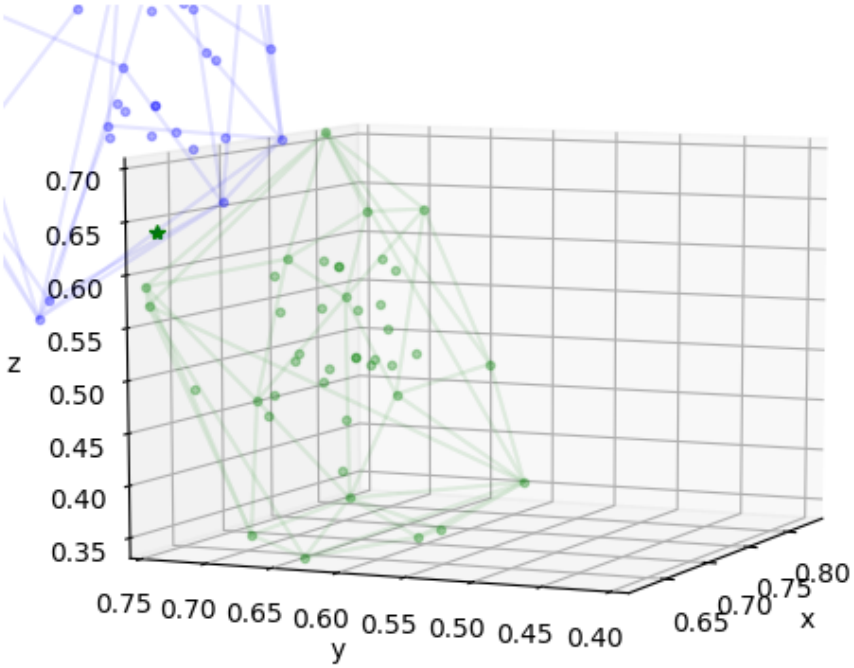
Também, observa-se que os erros estão em geral relacionados à região de interseção entre os convexos.



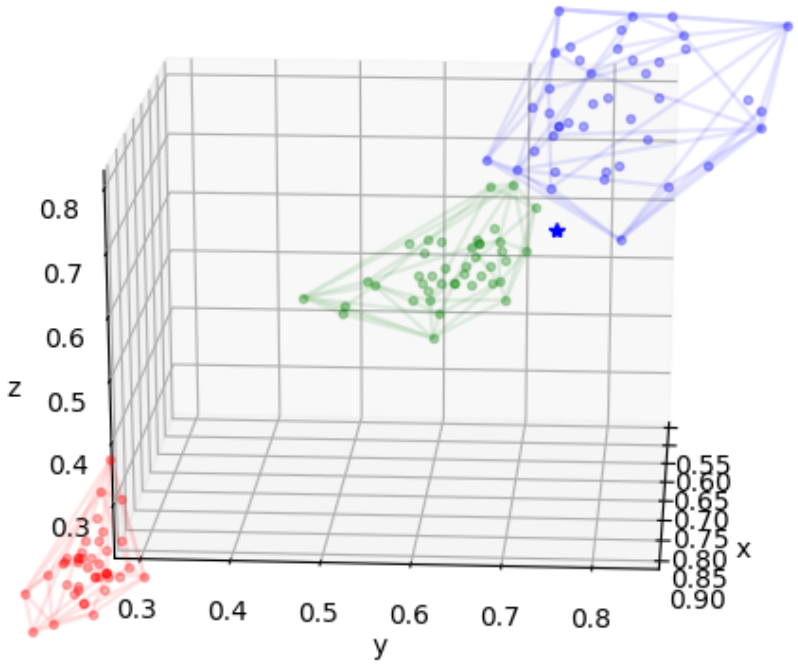
(a)



(b)


0.70
0.65
0.60
0.55
z
0.50
0.45
0.40
0.35
0.75 0.70 0.65 0.60 0.55 0.50 0.45 0.40
y
0.65 0.70 0.75 0.80
x

(c)



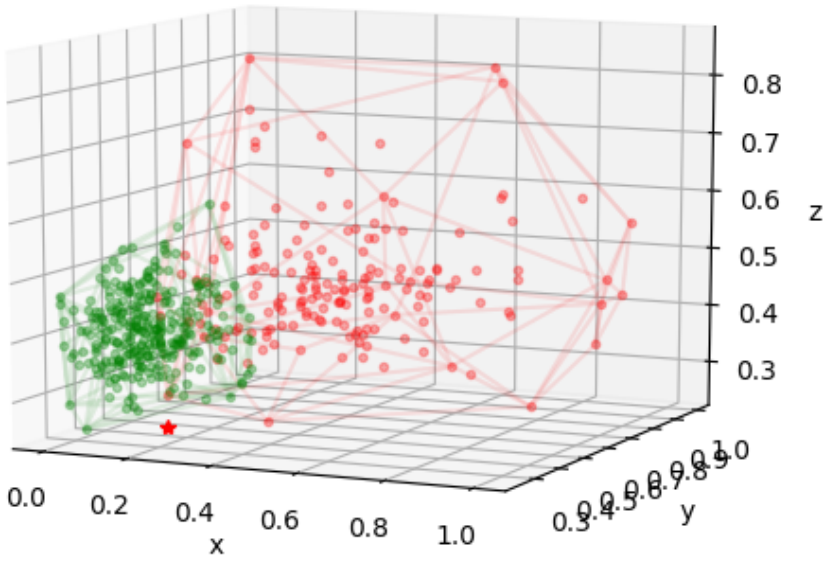
(d)

Figura 5.6: Ilustração dos resultados das aplicações do algoritmo NCHC na base Iris nos casos onde ocorreram classificações incorretas.
Cada cor representa uma classe do problema, onde os pontos são parte do conjunto de treinamento da execução do algoritmo. Os segmentos de reta representam as faces dos convexos gerados para cada classe. A estrela representa a instância que teve a classificação diferente do seu rótulo real, sendo que sua coloração representa a classe correta. Fonte: Elaborado pelo autor.

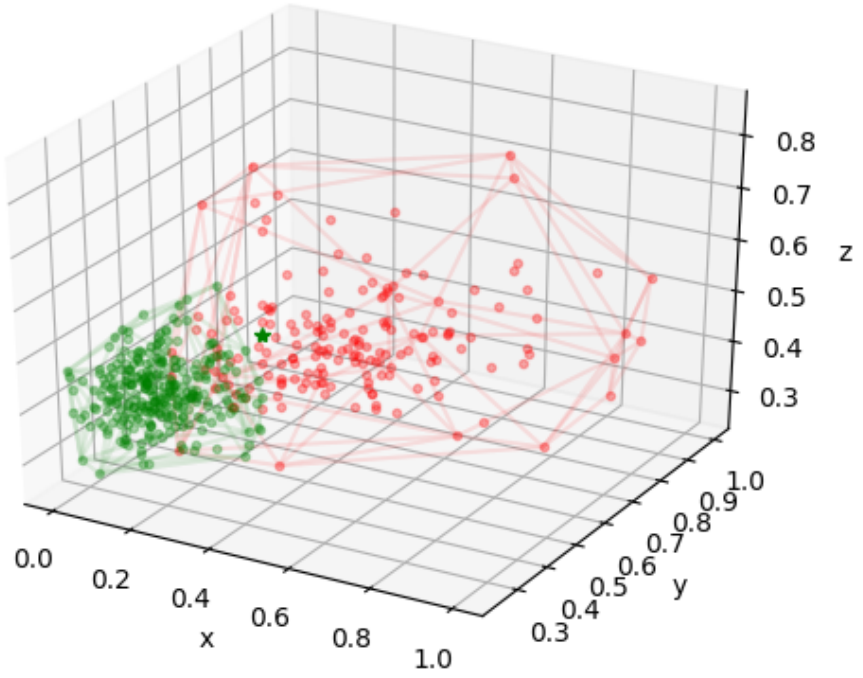
5.5.2 *Breast cancer*

No conjunto de dados *Breast cancer*, existem somente duas classes, e na Figura 5.7 pode-se visualizar uma maior densidade das instâncias da classe de câncer benigno (em verde) em comparação às instâncias da classe de câncer maligno (em vermelho) em relação às expressões dos três genes selecionados pelo algoritmo SVM-RFE. Tal característica pode ser explicada pelo fato de que a célula cancerígena pode ter sua regulação de expressão gênica comprometida de acordo com as possíveis mutações, causando anomalias que podem não possuir padrões definidos nas estruturas das células e portanto, modificar as métricas coletadas das imagens das amostras da base de dados utilizada.

Nessa base, podemos observar uma interseção de tamanho considerável entre os dois convexos, causando grande dificuldade em diferenciar as instâncias sem rótulos próximas à interseção dos convexos gerados que representam as duas classe. É importante notar que essa característica pode continuar existindo em espaços de dimensões maiores que 3.



(a)



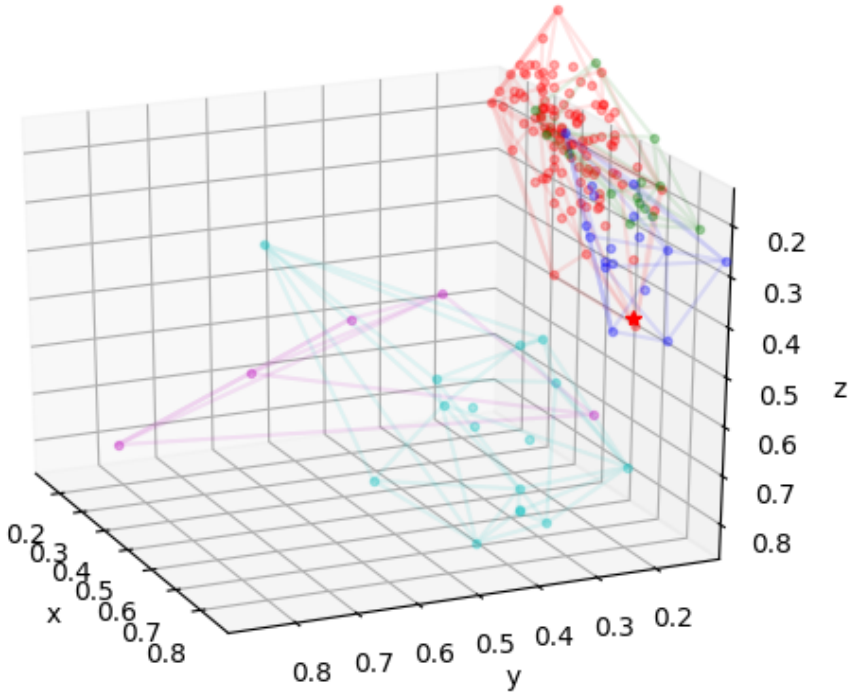
(b)

Figura 5.7: Ilustração dos resultados das aplicações do algoritmo NCHC na base *Breast cancer* nos casos onde ocorreram classificações incorretas.
A cor vermelha representa a classe "maligno"e cor verde representa a classe "benigno", onde os pontos são parte do conjunto de treinamento da execução do algoritmo. Os segmentos de reta representam as faces dos convexos gerados para cada classe. A estrela representa a instância que teve a classificação diferente do seu rótulo real, sendo que sua coloração representa a classe correta. Fonte: Elaborado pelo autor.

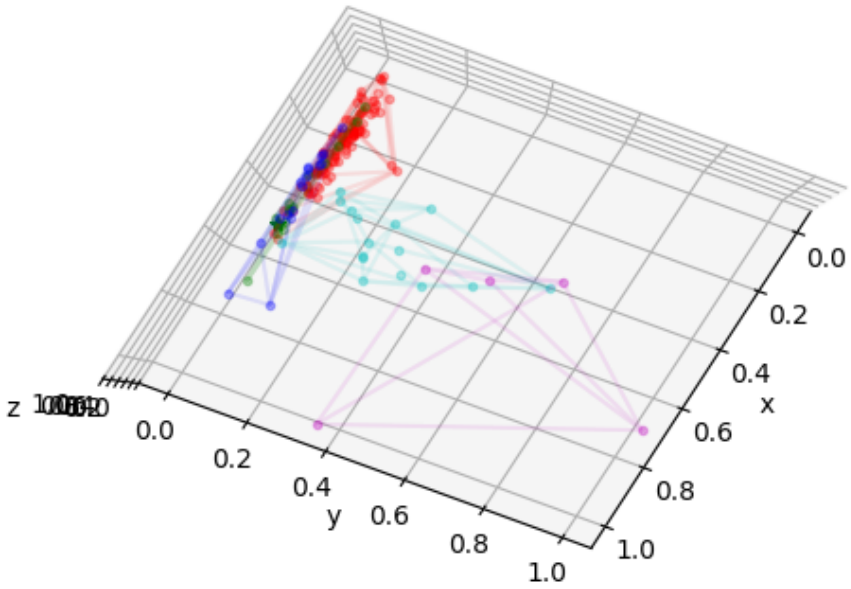
5.5.3 *Lung cancer*

A base de dados *Lung cancer* é composta por 5 classes, e 12600 características (genes). A redução do espaço de dimensões para a classificação pode ser um problema muito complexo, mas para simplificar a visualização dos resultados somente foram escolhidas três dimensões como mostrado na Figura 5.8. Na figura, percebe-se que duas das classes (uma delas é a que possui menos instâncias na base de dados) são linearmente separáveis das demais classes. Observa-se também que uma das dimensões escolhidas pelo SVM-RFE permite a divisão comentada anteriormente.

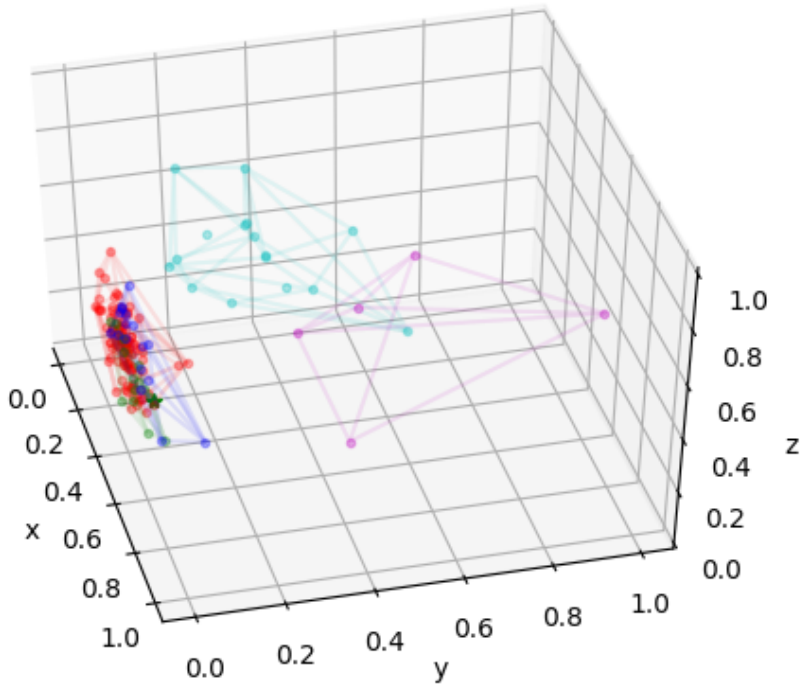
Nesta base, cujos dados têm origem de análises de expressões gênicas, é explicito a falta de representatividade por algumas das classes é uma característica que prejudica a distinção entre duas das classes do problema (em azul claro e rosa), como mostrado na Imagem 5.8. Em tal situação, a adição de uma das dimensões não escolhidas poderia auxiliar na divisão dos convexos, o que pode ser explorado em problemas de classificação que envolvem várias classes como o desta base.



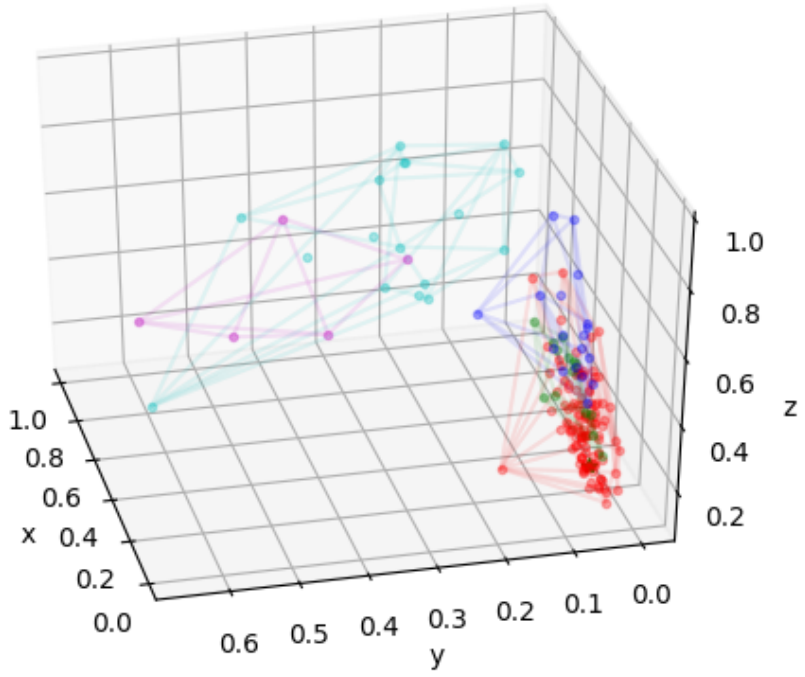
(a)



(b)



(c)



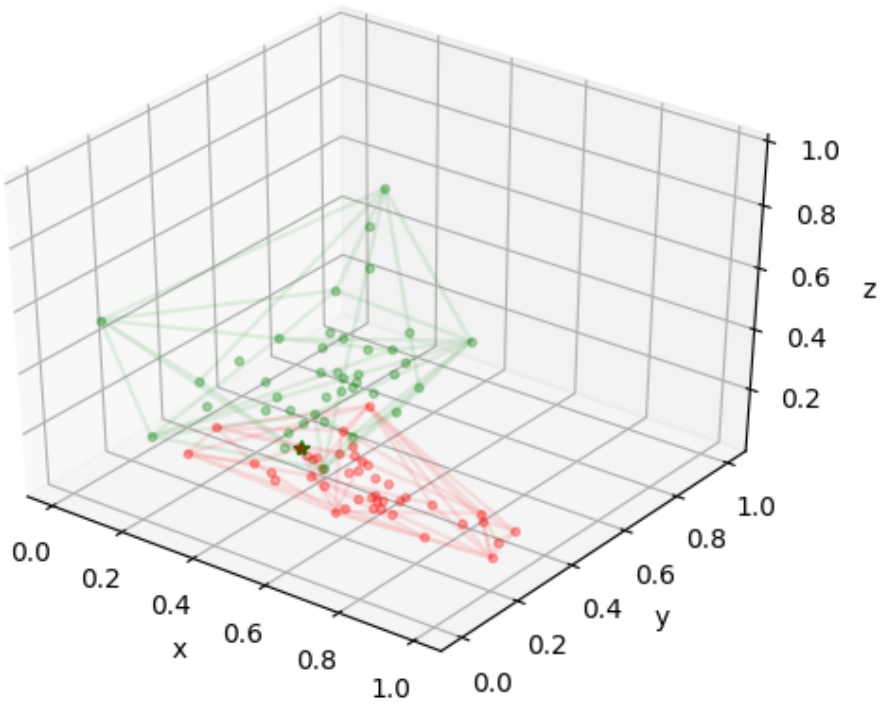
(d)

Figura 5.8: Ilustração dos resultados das aplicações do algoritmo NCHC na base *Lung cancer* nos casos onde ocorreram classificações incorretas.
A cor vermelha representa a classe de expressões gênicas de células normais e as demais cores representam diferentes tipos de câncer, onde os pontos são parte do conjunto de treinamento da execução do algoritmo. Os segmentos de reta representam as faces dos convexos gerados para cada classe. A estrela representa a instância que teve a classificação diferente do seu rótulo real, sendo que sua coloração representa a classe correta. Fonte: Elaborado pelo autor.

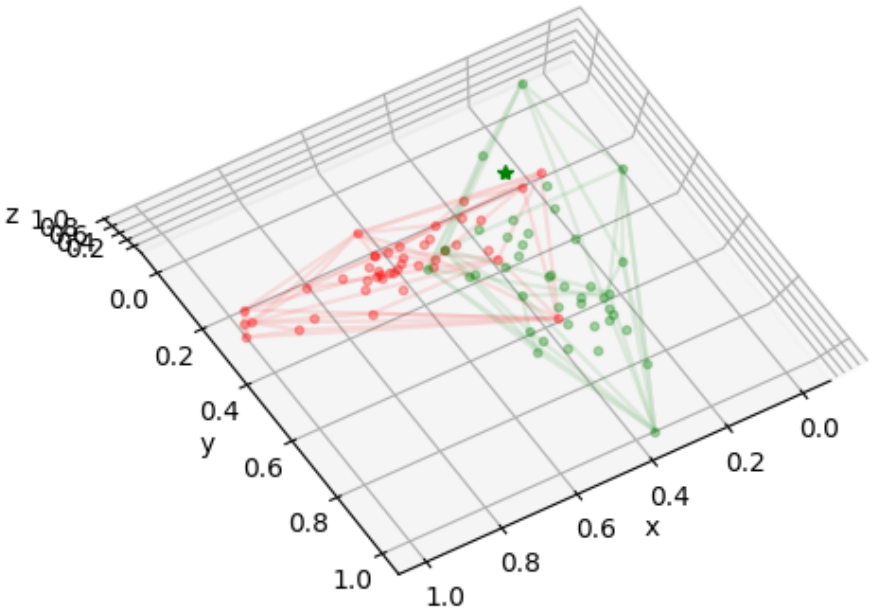
5.5.4 *Prostate tumor*

A base de dados *Prostate tumor*, possui natureza das coletas dos dados semelhante à base *Lung cancer*, onde há uma grande quantidade de características em comparação ao total de instâncias. Entretanto, essa base possui apenas duas classes, e não sofre com a falta de representatividade por classe como a base *Lung cancer*.

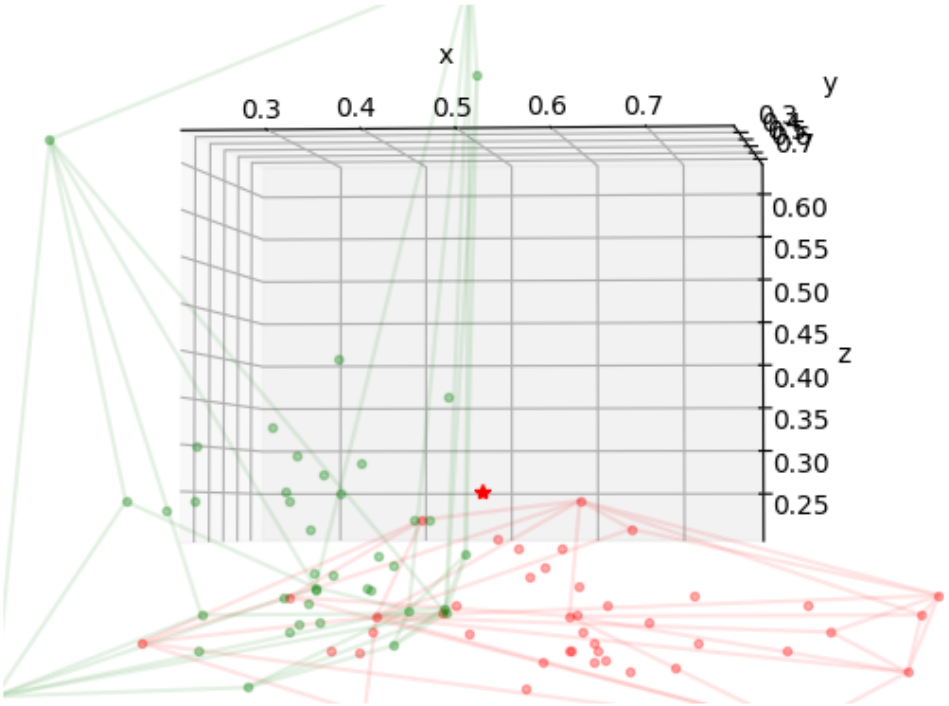
Assim como a base *Breast cancer*, a base *Prostate tumor* apresenta uma diferença entre a densidade de pontos entre os convexos de cada classe, como ilustra a Figura 5.9. A Imagem *b* da figura exemplifica o impacto de *outliers* no modelo gerado pelo NCHC, gerando faces relativamente grandes que podem prejudicar a classificação em algumas situações, como mostra a Imagem 5.9*c* onde a instância representada pela estrela vermelha foi classificada como parte da classe relacionada ao convexo em verde. Repare que a face mais próxima da instância classificada de forma errada possui seus pontos distantes entre si, gerando uma face relativamente grande em comparação às outras.



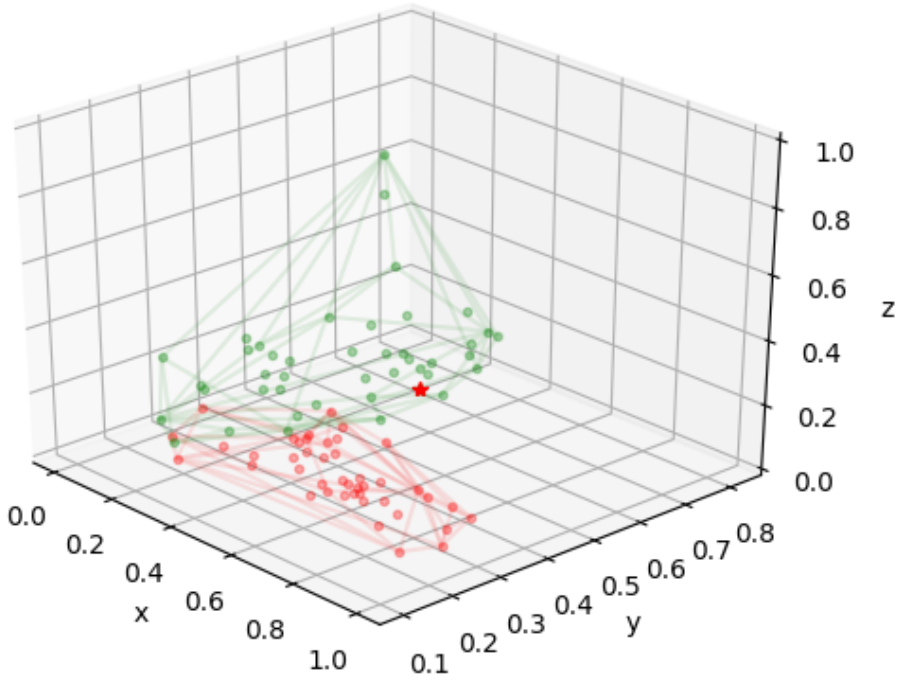
(a)



(b)



(c)



(d)

Figura 5.9: Ilustração dos resultados das aplicações do algoritmo NCHC na base *Prostate tumor* nos casos onde ocorreram classificações incorretas.
As cores dos convexos representam as classes relacionadas às expressões normais e cancerígenas. Os segmentos de reta representam as faces dos convexos gerados para cada classe. A estrela representa a instância que teve a classificação diferente do seu rótulo real, sendo que sua coloração representa a classe correta. Fonte: Elaborado pelo autor.

6 Conclusões

Este trabalho teve por objetivo identificar o comportamento do algoritmo de aprendizado de máquina supervisionado *Nearest Convex Hull Classification* (NCHC) quando aplicado a diferentes bases de dados, em específico às de expressões gênicas, em relação à diferentes técnicas e algoritmos que auxiliam as soluções de problemas de classificação.

Durante a implementação do algoritmo NCHC feita neste trabalho, foi possível perceber algumas de suas limitações. Uma delas é a necessidade de um conjunto mínimo de pontos para a construção dos convexos que representam as classes dos problemas, que é equivalente ao número de dimensões. Outra limitação relacionada à dimensionalidade, é que o número de dimensões possui impacto significativo no tempo de execução dos algoritmos que computam os fechos convexos e também dos cálculos de distâncias entre pontos e convexos construídos.

Apesar das limitações mencionadas, o classificador investigado parece ter bom desempenho quando comparado aos outros classificadores testados nesse trabalho. Na base de dados Iris, com a menor quantidade de dimensões, o NCHC teve desempenho muito similar ao classificador KNN (com valor $K = 1$) o que pode ser explicado pelo fato de que ambos classificadores compartilharem a estratégia de analisar as distâncias dentro do espaço de características. Além disso, foi possível também observar que o classificador implementado no trabalho teve a taxa de acerto maior que o KNN em alguns casos, o que pode ser causado pela diferença entre ambos em relação ao fato de que o NCHC inclui a construção de vários hiperplanos, que representam as faces dos convexos, que é uma estratégia presente no algoritmo SVM, que busca definir um hiperplano para diferenciar as classes do problema. Todavia, o SVM teve as melhores taxas de acerto na maior parte dos experimentos realizados, o que pode ser explicado pela característica desse classificador estabelecer uma fronteira no hiperespaço que separa todas (ou maior parte) das instâncias. Quando as classes possuem interseção, a característica mencionada pode ser essencial para criar generalizações adequadas, uma vez que os hiperplanos serão gerados de forma que o espaço de interseção será "cortado" em segmentos de tamanho equivalentes.

Em relação às técnicas de redução dimensionalidade por meio de seleção de característica, observou-se um grande desafio na atribuição do parâmetro γ utilizado pela técnica de transformação de espaço pelo *kernel* RBF. Foi observado que o valor de γ causa também grande variação na acurácia dos classificadores testados nos experimentos. Também, nas bases de dados de expressões gênicas, que possuem alta dimensionalidade em relação à quantidade de instâncias, foi observado que os classificadores têm seus desempenhos comprometidos quando utiliza-se a técnica de redução de características por meio da transformada de *kernel* RBF. Entretanto, as acurácias dos classificadores, incluindo o NCHC, foram aumentadas quando a técnica SVM-RFE foi utilizada para reduzir a dimensionalidade do problema.

Nas interpretações geométricas dos resultados do NCHC, em espaços reduzidos para três dimensões, foi possível notar que uma das principais causas de erros na classificação é a presença de interseções entre os convexos gerados, ou a alta proximidade entre eles. As faces dos convexos gerados criam uma generalização onde se considera os espaços próximos às faces

como pertencentes às classes, o que pode ser um equívoco quando os pontos relacionados às faces são muito distantes entre si, causando o erro das classificações.

Uma vez que as limitações e problemas do NCHC comentadas anteriormente são superadas, o classificador pode apresentar resultados interessantes, podendo ser equiparado aos outros algoritmos utilizados na área de aprendizado de máquina.

6.1 Trabalhos futuros

O algoritmo de classificação NCHC apresentou diversas limitações nos experimentos realizados neste trabalho. As faces com tamanhos relativamente grandes podem indicar problemas no modelo generalizado pelo algoritmo. A existência de *outilers* é um dos motivos da construção dessas faces "grandes" e também pode ser o motivo para criar interseções entre as classes dos problemas de classificação. Variações do NCHC poderiam ser propostas de forma que os *outliers* fossem detectados e ignorados. Também, percebe-se que a distância entre os pontos e os convexos pode não ser melhor métrica para inferir as classes das instâncias não rotuladas, o que pode ser alterado em outras variações do NCHC que consideram os tamanhos das faces dos convexos além das distâncias calculadas.

Uma das vantagens do modelo gerado pelo algoritmo NCHC é a simplicidade de sua interpretação, que pode ser facilmente visualizada em problemas com tamanhos de dimensões até 3. Além disso, a maior parte das interpretações dos convexos são as mesmas em qualquer quantidade de dimensões. A interseção entre os convexos gerados para cada classe pelo algoritmo representa uma região e um conjunto de instâncias que têm uma grande semelhança entre si. O cálculo do tamanho dessa interseção em relação ao tamanho dos convexos de cada classe pode ser uma métrica interessante em algumas situações, representando a similaridade entre as classes em relação às características representadas pelas dimensões do problema. Entretanto, calcular tal métrica pode não ser um processo trivial, podendo ser necessário utilizar várias ferramentas e teorias complexas da área de geometria computacional.

Além disso, percebe-se que quando não há interseção entre os convexos de cada classe, o problema passa a ser linearmente separável. Também, espera-se que quanto mais distante os convexos são entre si, menor a probabilidade de classificar uma instância de forma incorreta. Quando as dimensões do problema de classificação são mapeadas para um espaço onde a distância entre os convexos gerados pelas instâncias de cada classe são maximizadas, o processo de classificação pode se tornar uma etapa trivial. Entretanto, encontrar tal função é um problema altamente complexo, que pode ser explorado em futuros trabalhos.

Referências

Baldi, P. e Brunak, S. (2001). *Bioinformatics: the machine learning approach*. MIT press.

Barardo, D. G., Newby, D., Thornton, D., Ghafourian, T., de Magalhães, J. P. e Freitas, A. A. (2017). Machine learning for predicting lifespan-extending chemical compounds. *Aging (Albany NY)*, 9(7):1721.

Barber, C. B., Dobkin, D. P. e Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483.

Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M. e Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences*, 97(1):262–267.

Chen, Y., Li, Y., Narayan, R., Subramanian, A. e Xie, X. (2016). Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839.

Chen, Y., Zhang, L. e Yi, Z. (2014). Nearest convex hull classification by using lotka–volterra recurrent neural networks. *Neurocomputing*, 138:157–166.

Fabris, F., De Magalhães, J. P. e Freitas, A. A. (2017). A review of supervised machine learning applied to ageing research. *Biogerontology*, 18(2):171–188.

Hsu, C.-W. e Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425.

Jones, E., Oliphant, T. e Peterson, P. (2014). {SciPy}: open source scientific tools for {Python}.

Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Pérez, A. et al. (2006). Machine learning in bioinformatics. *Briefings in bioinformatics*, 7(1):86–112.

Nalbantov, G., Groenen, P. e Bioch, C. (2006). Nearest convex hull classification. Relatório técnico.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P. et al. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, 98(26):15149–15154.

Singh, R., Lanchantin, J., Robins, G. e Qi, Y. (2016). Deepchrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, 32(17):i639–i648.

Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D. e Levy, S. (2004). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631–643.

Statnikov, A., Wang, L. e Aliferis, C. F. (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC bioinformatics*, 9(1):319.

Touw, W. G., Bayjanov, J. R., Overmars, L., Backus, L., Boekhorst, J., Wels, M. e van Hijum, S. A. (2012). Data mining in the life sciences with random forest: a walk in the park or lost in the jungle? *Briefings in bioinformatics*, 14(3):315–326.

Yeoh, E.-J., Ross, M. E., Shurtleff, S. A., Williams, W. K., Patel, D., Mahfouz, R., Behm, F. G., Raimondi, S. C., Relling, M. V., Patel, A. et al. (2002). Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer cell*, 1(2):133–143.

Zhou, X. e Shi, Y. (2009). Nearest neighbor convex hull classification method for face recognition. Em *International Conference on Computational Science*, páginas 570–577. Springer.

Zweiger, G. (1999). Knowledge discovery in gene-expression-microarray data: mining the information output of the genome. *Trends in biotechnology*, 17(11):429–436.