

---

Universidade Federal de Ouro Preto  
Departamento de Computação e Sistemas  
Colegiado de Engenharia de Computação



## **Reconhecimento de padrões em Biomecânica de Corrida usando Aprendizado de Máquina**

**Mônica de Oliveira**

### **Trabalho de Conclusão de Curso**

ORIENTAÇÃO:

Talles Henrique de Medeiros

**Julho, 2018**

**João Monlevade/MG**

---

Mônica de Oliveira

**Reconhecimento de padrões em Biomecânica de corrida usando Aprendizado  
de Máquina**

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Orientador: Tales Henrique de Medeiros

João Monlevade  
JULHO DE 2018

O482r

Oliveira, Monica.

Reconhecimento de padrões em biomecânica de corrida usando aprendizado de máquina [manuscrito] / Monica Oliveira. - 2018.

65f.: il.: color; grafs; tabs.

Orientador: Prof. MSc. Talles Henrique de Medeiros.  
Coorientador: Rafael Martins.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Inteligência artificial. 2. Aprendizado de máquina. 3. Biomecânica. I. Medeiros, Talles Henrique de. II. Martins, Rafael. III. Universidade Federal de Ouro Preto. IV. Título.

CDU: 004.8:612.76

Catalogação: [ficha@sisbin.ufop.br](mailto:ficha@sisbin.ufop.br)



**ANEXO IV – Folha de Aprovação**  
**Curso de Engenharia de Computação**

**FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA**

Reconhecimento de padrões em Biomecânica de Corrida usando Aprendizado de Máquina

**Mônica de Oliveira**

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas  
da Universidade Federal de Ouro Preto como requisito parcial da  
disciplina CSI496 – Trabalho de Conclusão de Curso II do curso de  
Bacharelado em Engenharia de Computação e aprovada pela Banca  
Examinadora abaixo assinada:

Prof. Me Ciéncia da Computação - Tallés Henrique de Medeiros  
DECSI - UFOP  
Professor Orientador

Prof. Dr. Eng. Elétrica - Rafael Frederico Alexandre  
DECSI - UFOP  
Professor Convidado

Prof. Dra Eng. Elétrica – Gilda Aparecida de Asis  
DECSI - UFOP  
Professor Convidado

João Monlevade, 17 de Julho de 2018



UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS  
COLEGIADO DE ENGENHARIA DA COMPUTAÇÃO

## ATA DE DEFESA

Aos 17 dias do mês de Julho de 2018, às 17 horas e 00 minutos, no Laboratório de Programação, foi realizada a defesa de Monografia pelo aluno **Mônica de Oliveira**, sendo a Comissão Examinadora constituída pelos professores: Prof. Prof. Me Ciência da Computação - Talles Henrique de Medeiros, Prof. Prof. Dr. Eng. Elétrica - Rafael Frederico Alexandre e Prof. Prof. Dra Eng. Elétrica – Gilda Aparecida de Asis. O candidato apresentou a monografia intitulada: "*Reconhecimento de Padrões da Biomecânica de Corrida usando Aprendizado de Máquina*". A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, concedendo-lhe o prazo de 15 dias para incorporação no texto final das alterações sugeridas. Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo formando.

João Monlevade, 17 de Julho de 2018.

Prof. Me Ciência da Computação - Talles Henrique de Medeiros

Professor Orientador/Presidente

Prof. Dr. Eng. Elétrica - Rafael Frederico Alexandre  
Professor Convidado

Prof. Dra Eng. Elétrica – Gilda Aparecida de Asis  
Professor Convidado

Mônica de Oliveira  
Mônica de Oliveira

Formando



**ANEXO III – Termo de Responsabilidade**

**TERMO DE RESPONSABILIDADE**

Eu, Mônico de Oliveira, declaro que o texto do trabalho de conclusão de curso intitulado "Reconhecimento de padrões em Biomônimos de cíndio usando Aprendizado de máquina" é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 17 de julho de 2018

Mônico de Oliveira

Assinatura do aluno

# Agradecimentos

Gostaria primeiramente de agradecer à Deus ao qual nos momentos mais difíceis não me deixou padecer e perder as esperanças, ajudando a lembrar que os planos do senhor são maiores que meus sonhos. Agradeço aos meus pais, meu herói Lázaro Luiz de Oliveira Neto e minha heroína Eliana Cardoso de Oliveira por todo apoio, carinho e paciência ao longo da minha caminhada. Agradeço à minha querida irmã Carla Milena de Oliveira pelo companheirismo, apoio e afeto durante esses anos e aos meus avós que sempre me apoiaram.

Agradecimento especial aos meus orientadores Tales Medeiros e Rafael Martins que me ajudaram a desenvolver esse trabalho e dividiram seus conhecimentos e informações valiosas comigo com total profissionalismo e amizade, desejo à eles muito sucesso. Agradeço também a UFOP, a instituição ao qual escolhi com total convicção de muito aprendizado e confiança. Obrigado a todos que fizeram parte da minha caminhada durante o curso, desde os professores que dividiram seus conhecimentos aos mais diversos funcionários como bibliotecários, técnicos e servidores. Gostaria também de agradecer aos meus amigos que juntos seguimos unidos e por um único objetivo em comum. Sou muito grata a todos!

*Todo mundo é um gênio. Mas, se você julgar um peixe por sua capacidade de subir em uma árvore, ela vai gastar toda a sua vida acreditando que ele é estúpido.*(Desconhecido)

# Resumo

*Este trabalho consiste em propor uma investigação de técnicas de aprendizado de máquina aplicadas ao problema de efeitos da velocidade de marcha com o intuito de auxiliar na prevenção de lesões musculares em atletas. O estudo propõe como objetivo investigar quais técnicas, dentre as selecionadas, apresentam melhores taxas de classificação para o problema. Antes da aplicação dos algoritmos fez-se necessário o pré-processamento dos dados brutos afim de extrair o máximo de informação para realizar a aplicação. Sendo assim, dois algoritmos de classificação foram utilizados, o kNN frequentemente aplicado em reconhecimento de padrões e o SVM capaz de separar os dados em classes distintas utilizando hiperplano de separação. Com o conjunto de dados da Biomecânica, o algoritmo kNN mostrou-se pouco eficiente ao contrário do SVM onde a taxa de classificação chegou a 77,8%.*

**Palavras-chaves:** Classificação, Algoritmo kNN, Algoritmo SVM, Biomecânica.

# Abstract

This work consists in proposing an investigation of machine learning techniques applied to the problem of gait speed effects with the intention of help in the prevention of muscular injuries in athletes. The objective of this study is to investigate which techniques, among those selected, present better classification rates for the problem. Before the application of the algorithms, it was necessary to preprocess the raw data in order to extract the maximum information to perform the application. Thus, two classification algorithms were used, the kNN often applied in pattern recognition and the SVM capable of separating the data into distinct classes using separation hyperplane. With the Biomechanics data set, the kNN algorithm showed to be inefficient unlike the SVM where the classification rate reached 77,8%

**Key-words:** Machine Learning, Classification, KNN Algorithm, SVM Algorithm, Biomechanics.

# Listas de ilustrações

Figura 1 – Hierarquia do Aprendizado de Máquina, FONTE: adaptado de Monard e Baranauskas (2003) . . . . .	18
Figura 2 – Diagrama para o aprendizado supervisionado . . . . .	19
Figura 3 – Diagrama para o aprendizado não supervisionado . . . . .	20
Figura 4 – Blocos funcionais para o reconhecimento de padrão na abordagem estatística. FONTE: JAIN, A., 2000 . . . . .	21
Figura 5 – Vizinhos no Espaço de características, com $k = 4$ . . . . .	24
Figura 6 – Escolha de um $k$ apropriado: <i>underfitting</i> e <i>overfitting</i> respectivamente . . . . .	27
Figura 7 – Representação de vários hiperplanos de separação. FONTE: adaptado de HAYKIN, 1994 . . . . .	29
Figura 8 – Princípio da minimização do risco estrutural. FONTE: SCHOLKOPF; SMOLA , 2002 . . . . .	34
Figura 9 – Representação de vários hiperplanos de separação. FONTE: adaptado de KOWALCZYK, 2014 . . . . .	35
Figura 10 – Representação do plano e margens obtidas no treinamento do SVM. FONTE: adaptado de Kavzoglu; Colkesen, 2009 . . . . .	36
Figura 11 – Marcadores reflexivos e esteira instrumentada. FONTE: (FUKUCHI; FUKUCHI; DUARTE, 2017) . . . . .	43
Figura 12 – Teste de Correlação afim de adquirir algum conhecimento sob a base de dados original . . . . .	44
Figura 13 – Dados brutos no domínio do tempo em relação aos eixos x, y e z do atributo Força de um sujeito aleatório para mostrar aperiodicidade	45
Figura 14 – Transformada de Fourier do sujeito 005, atributo "momento livre sob o eixo vertical" . . . . .	46
Figura 15 – Tempo de pisada dos atletas na plataforma sob momento livre no eixo vertical Ty nos tempos 2.5 m/s 3.5 m/s e 4.5 m/s respectivamente . . . . .	55

Figura 16 – Média do Centro de pressão nos eixos x e z nas velocidades 2.5 m/s 3.5 m/s e 4.5 m/s . . . . .	56
Figura 17 – Média da Transformada de Fourier das Forças no eixo x, 2.5 m/s 3.5 m/s e 4.5 m/s . . . . .	57
Figura 18 – Média da Transformada de Fourier das Forças no eixo y, 2.5 m/s 3.5 m/s e 4.5 m/s . . . . .	57
Figura 19 – Média da Transformada de Fourier das Forças no eixo z, 2.5 m/s 3.5 m/s e 4.5 m/s . . . . .	58
Figura 20 – Média da Transformada de Fourier do momento livre sob o eixo vertical Ty nos tempos 2.5 m/s 3.5 m/s e 4.5 m/s . . . . .	58

# **Lista de tabelas**

Tabela 1 – Algumas das funções <i>Kernels</i> mais utilizadas . . . . .	38
Tabela 2 – Exemplo matriz de confusão . . . . .	46
Tabela 3 – Matriz de confusão kNN com $k = 1$ . . . . .	48
Tabela 4 – Matriz de confusão kNN com $k = 2$ . . . . .	49
Tabela 5 – Matriz de confusão kNN com $k = 3$ . . . . .	49
Tabela 6 – Resumo dos Resultados de Classificação do kNN . . . . .	50
Tabela 7 – Matriz de confusão Kernel linear . . . . .	50
Tabela 8 – Matriz de confusão Kernel Polinomial . . . . .	51
Tabela 9 – Matriz de confusão Kernel Base Radial . . . . .	51
Tabela 10 – Matriz de confusão Kernel Sigmoide . . . . .	52
Tabela 11 – Resumo dos Resultados de Classificação . . . . .	52

# **Lista de abreviaturas e siglas**

KNN	K-nearest neighbors ou K-vizinhos mais próximos
MLP	Perceptron Multi-Camadas
RBF	Radial basis function ou rede neural com função de ativação de base radial
RNA	Redes Neurais artificiais
SVM	Support vector machine ou máquina de vetores de suporte
TAE	Teoria de aprendizado estatístico
TF	Transformada de Fourier
VC	Teoria de Vapnik-Chervonenkis

# Listas de símbolos

$\in$	Pertence
$\phi$	Letra grega phi maiúsculo
$\mu$	Letra grega mu minúsculo
$\sigma$	Letra grega sigma minúscula
$\Sigma$	Somatório
$\int$	Integral
$\subset$	Contido
$\leq$	Menor ou igual
$\geq$	Maior ou igual
$<$	Menor
$>$	Maior
$\forall$	Para todo
$\xi$	Letra grega xi minúsculo

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Problema</b>	<b>15</b>
<b>1.2</b>	<b>Objetivos</b>	<b>15</b>
<b>1.3</b>	<b>Estrutura do trabalho</b>	<b>15</b>
<b>2</b>	<b>APRENDIZADO DE MÁQUINA</b>	<b>17</b>
<b>2.1</b>	<b>Conceitos básicos</b>	<b>17</b>
<b>2.1.1</b>	Aprendizado de máquina	17
<b>2.1.1.1</b>	Aprendizado Supervisionado	18
<b>2.1.1.2</b>	Aprendizado Não Supervisionado	19
<b>2.1.2</b>	Problema da Interpolação exata	22
<b>2.2</b>	<b>KNN - k vizinhos mais próximos</b>	<b>23</b>
<b>2.2.1</b>	Introdução	23
<b>2.2.2</b>	Contexto	24
<b>2.2.3</b>	A escolha de um k apropriado	26
<b>2.3</b>	<b>Redes RBF - Função de base radial</b>	<b>27</b>
<b>2.3.1</b>	Introdução	27
<b>2.3.2</b>	Contexto	28
<b>2.3.3</b>	Treinamento - Não Supervisionado	29
<b>2.4</b>	<b>SVM - Máquinas de Vetores de Suporte</b>	<b>31</b>
<b>2.4.1</b>	Introdução	31
<b>2.4.2</b>	Contexto	32
<b>2.4.3</b>	Teoria do aprendizado estatístico - TAE	32
<b>2.4.4</b>	Hiperplano	34
<b>2.4.5</b>	Hiperplano ótimo para padrões linearmente separáveis	36
<b>2.4.6</b>	Hiperplano ótimo para padrões não-linearmente separáveis	37
<b>2.5</b>	<b>Transformada de Fourier</b>	<b>39</b>
<b>2.6</b>	<b>Considerações finais</b>	<b>40</b>

<b>3</b>	<b>MÉTODOLOGIA</b>	<b>42</b>
3.1	Introdução	42
3.2	Descrição da base de dados	42
3.3	Pré-Processamento - Features	44
3.3.1	Transformada de Fourier	44
3.4	Métodos de avaliação - Matriz de Confusão e Acurácia	46
3.5	Base de dados final	47
<b>4</b>	<b>RESULTADOS</b>	<b>48</b>
4.1	Resultados da Classificação dos Algoritmos	48
4.1.1	Algoritmo de Classificação kNN	48
4.1.2	Algoritmo de Classificação SVM	50
4.2	Características dos Atletas	52
<b>5</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>59</b>
5.1	Conclusão	59
5.2	Trabalhos Futuros	59
	<b>REFERÊNCIAS</b>	<b>61</b>

# 1 Introdução

Aprendizado de Máquina (*Machine Learning*) é uma área da Inteligência Artificial que lida com problemas de aprendizado computacional a fim de adquirir conhecimento de forma automática, explorando o estudo e construção de algoritmos que podem aprender com os erros e fazer previsões de dados. Tal sistema de aprendizado tem a função de analisar informações e gerar novos conhecimentos. ([MONARD; BARANAUSKAS, 2003](#))

A participação popular em corridas tem aumentado consideravelmente nos últimos anos e com isso vem despertando interesse tanto de empresas ligadas aos esportes quanto da medicina tanto em vista as frequentes lesões ocasionadas. A tecnologia vem sendo aliada na melhora de desempenho dos atletas uma vez que o esporte é inundado de dados (informações) que podem se tornar expressivas para alguma predição de dados. Assim, órgãos especializados, tais como ortopedistas, médicos dos esportes de grandes clubes esportivos, traumatologistas estão atentos a quaisquer ferramentas que ajudem a transformar esses dados em algo concreto que resulte em informações úteis e significativas.

Uma dessas ferramentas que ganham popularidade é o aprendizado de máquina, através da qual os computadores são capazes de aprender sem serem explicitamente programados por um operador humano. O que torna os algoritmos de aprendizado da máquina tão úteis é a capacidade de serem treinados em grandes escalas de conjuntos de dados. Esses algoritmos treinados podem ser usados para identificar padrões potencialmente complexos e significativos nos dados, o que nos permite prever ou classificar futuras instâncias ou eventos. Tais abordagens de aprendizado de máquina por muitas vezes superam as técnicas estatísticas tradicionais, que são em grande parte incapazes de explicar os padrões dinâmicos e em grande quantidade dos quase aleatórios dados obtidos no esporte. ([ROBERTSON; WOODS; GASTIN, 2015](#))

## 1.1 Problema

As lesões musculares estão entre as causas mais frequentes no atendimento ortopédico. A corrida de longa distância tornou-se uma forma muito popular de atividade física entre indivíduos que buscam um estilo de vida saudável. Tais lesões caracterizam um desafio para especialistas a julgar por uma recuperação muitas vezes lenta obtendo como consequência o afastamento dos atletas aos treinos e competições resultante das lesões durante as atividades desportivas.([HINO et al., 2009](#)) Embora haja estudos aplicados ao aprendizado de máquina em relação a classificação de marcha em atletas, eles geralmente usam poucas amostras e um conjuntos limitados de variáveis considerando apenas uma parte do ciclo de marcha atribuindo a maior parte das pesquisas as variáveis de centro de pressão.([LINDSAY; BOLLT; MCGREGOR, 2014](#))

## 1.2 Objetivos

A proposta deste estudo basicamente consiste a princípio, aplicar técnicas de algoritmos de aprendizado de máquina supervisionado na base de dados, treinando-os com o conjunto de treinamento e depois realizando classificação do conjunto de testes para verificar qual algoritmo apresenta a melhor taxa de reconhecimento. Espera-se que o modelo que atingir maior grau de classificação possa contribuir ao máximo sob predições à lesões entre os atletas e que os atributos coletados possam ser padrões para futuras análises de classificação.

## 1.3 Estrutura do trabalho

Este presente texto está dividido em cinco capítulos. No capítulo 1 introduzimos uma breve definição de aprendizado de máquina, além de discutir os problemas que podem ser solucionados com a evolução do aprendizado computacional e os objetivos a serem alcançados.

O capítulo 2 apresenta o referencial teórico onde mostra-se os conceitos básicos dos algoritmos de aprendizado de máquina, além de uma breve introdução

da Transformada de Fourier.

No capítulo 3 abordamos a metodologia utilizada para descrever a base de dados final, além do pré-processamento e os métodos de avaliação usados para avaliar o desempenho dos algoritmos de aprendizado. Sendo assim no capítulo 4, avaliamos nossa base de dados ajustando os parâmetros necessários juntamente com os *Kernels* afim de obter o melhor desempenho possível. Na última seção é introduzida uma conclusão juntamente com sugestões para trabalhos futuros.

## 2 APRENDIZADO DE MÁQUINA

### 2.1 Conceitos básicos

#### 2.1.1 Aprendizado de máquina

Aprendizado de Máquina (*Machine Learning*) é um método de análise de dados que automatiza o desenvolvimento de modelos analíticos. É uma área da Inteligência Artificial que lida com problemas de aprendizado computacional a fim de adquirir conhecimento de forma automática. Tal sistema de aprendizado tem a função de analisar informações e gerar novos conhecimentos. ([MONARD; BARANAUSKAS, 2003](#))

Tom M. Mitchell em seu livro *Machine learning* forneceu uma definição mais formal amplamente citada: "Se diz que um programa aprende pela experiência  $E$  em relação à uma classe de tarefas  $T$ , com medida de desempenho  $P$ , se seu desempenho em  $T$ , medido por  $P$ , melhora com a experiência  $E$ . "Esta definição é notável uma vez que define aprendizado de máquina como objetivo de desenvolver técnicas computacionais que permitam simular o processo de aprendizado e a construção de sistemas capazes de adquirir autoconhecimento.

Usualmente, algoritmos de aprendizado de máquina utilizam experiências anteriores denominadas casos ou exemplos, para auxiliar no processo de tomada de decisão e assim poder melhorar seu desempenho. De acordo com a característica desses casos ou exemplos tem-se o modo de aprendizado supervisionado e o não supervisionado. Tais modelos podem se distinguir pela presença ou não do atributo classe que rotula os exemplos do conjunto de dados fornecido ao algorítimo chamado conjunto de treinamento, sendo que no aprendizado supervisionado esse rotulo é conhecido enquanto no modo não supervisionado os exemplos não estão previamente rotulados. ([CHAPELLE; SCHOLKOPF; ZIEN, 2009](#)) Na Figura 1 é possível observar a hierarquia do aprendizado de Máquina.

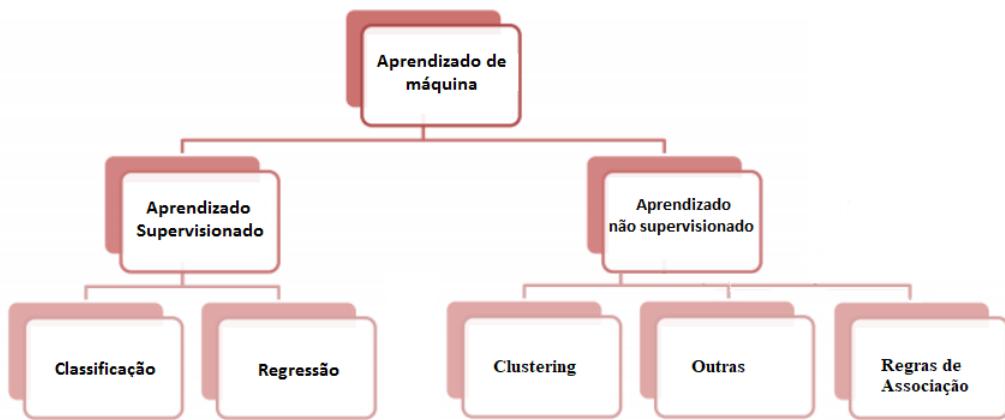


Figura 1 – Hierarquia do Aprendizado de Máquina, FONTE: adaptado de Monard e Baranauskas (2003)

#### 2.1.1.1 Aprendizado Supervisionado

Na aprendizagem supervisionada quando temos um conjunto de dados já sabemos como deve ser o nosso resultado, sendo que existe uma relação entre a entrada dos dados e o resultado final. O conjunto de treinamento para um algoritmo de aprendizado consiste basicamente de um conjunto E de N exemplos de treinamento.  $E = \{(x_1, y_1), \dots, (x_N, y_N)\}$  rotulados com valores y de uma função f desconhecida,  $y = f(x)$ , onde os valores  $x_i$  são vetores relacionados ao conjunto de atributos X. Sendo assim, o algoritmo constrói uma hipótese que deve aproximar a verdadeira função f tal que dado um novo dado X, a hipótese(x) prediz o valor de y correspondente. Sendo que a tarefa do classificador se torna determinar corretamente a classe dos novos dados recebidos que ainda não possuem rótulos. Para valores nominais dos rótulos  $y_1, y_2, \dots, y_N$ ... o processo é denominado classificação, enquanto que para valores numéricos o processo é regressão (MONARD; BARANAUSKAS, 2003). Um diagrama é apresentado na figura 2:

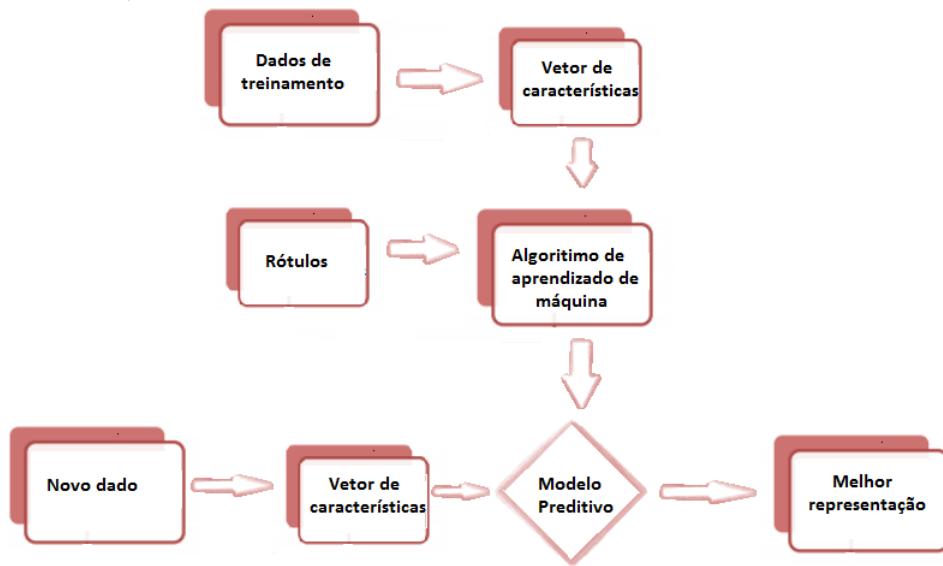


Figura 2 – Diagrama para o aprendizado supervisionado

#### 2.1.1.2 Aprendizado Não Supervisionado

Em contrapartida, o aprendizado não supervisionado, não nos permite obter ideias concisas sobre quais seriam os nossos resultados finais já que o mesmo não possui rótulos, sendo assim, o algoritmo de aprendizado só irá avaliar os dados recebidos buscando semelhanças entre eles e assim formar agrupamentos, *clusters*, com características semelhantes. O diagrama da figura 3 mostra o modelo de aprendizagem não supervisionado.

Os problemas de aprendizagem não supervisionados são categorizados em problemas de filtragem de agrupamento e colaborativo, é uma técnica definida por algumas métricas de similaridade entre membros de um mesmo *cluster* e separação entre *clusters* diferentes (MONARD; BARANAUSKAS, 2003).

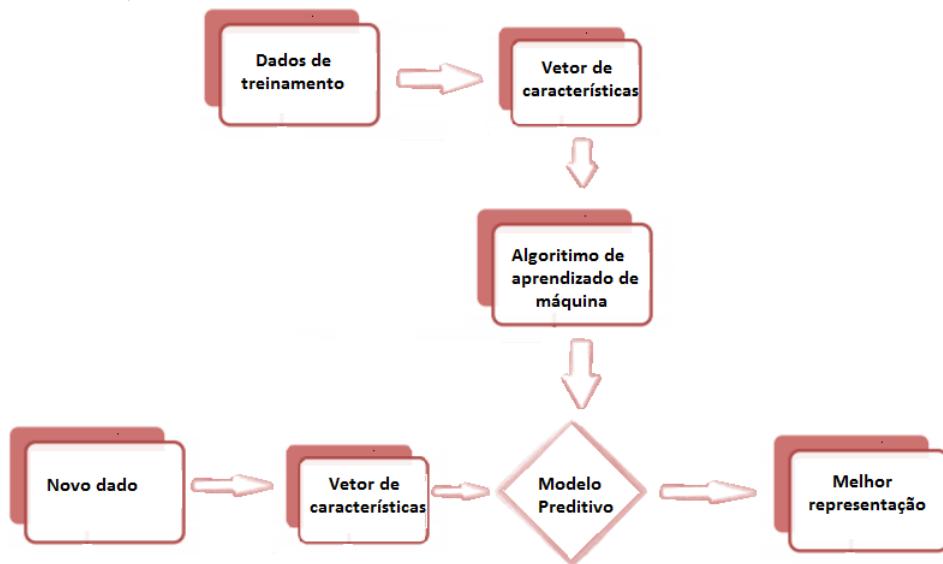


Figura 3 – Diagrama para o aprendizado não supervisionado

### Reconhecimento de padrões

O reconhecimento de padrões (*Pattern Recognition* - RP) é por vez uma disciplina científica cujo objetivo é a classificação de objetos em uma série de categorias ou classes. Por meados da década de 1960 era principalmente o resultado da pesquisa teórica na área de estatística, sendo que o advento dos computadores aumentou a demanda por aplicações práticas de reconhecimento de padrões. Tal prática faz parte integrante da maioria dos sistemas de inteligência da máquina construídos para a tomada de decisões ([THEODORIDIS; KOUTROUMBAS et al., 1999](#))

De acordo com ([BISHOP, 2006](#)), problemas de reconhecimento de padrões podem ser enquadrados em dois tipos: classificação e regressão. Em problemas de classificação as classes a qual a nova entrada será classificada já existem, e estão em número discreto. Em problemas de regressão nos permitem explorar e inferir a relação de uma variável dependente (variável de resposta) com variáveis independentes específicas (variáveis explicatórias).

Em um conceito de reconhecimento de padrões com base na estatística, um padrão pode ser representado por um conjunto de características chamado de vetor de características dimensional. Sendo que o sistema de reconhecimento é operado de

dois modos: treinamento que seria o aprendizado e a classificação que provém o teste. (JAIN; MAO; MOHIUDDIN, 1996). Na figura 4 é ilustrado os blocos funcionais do RP.

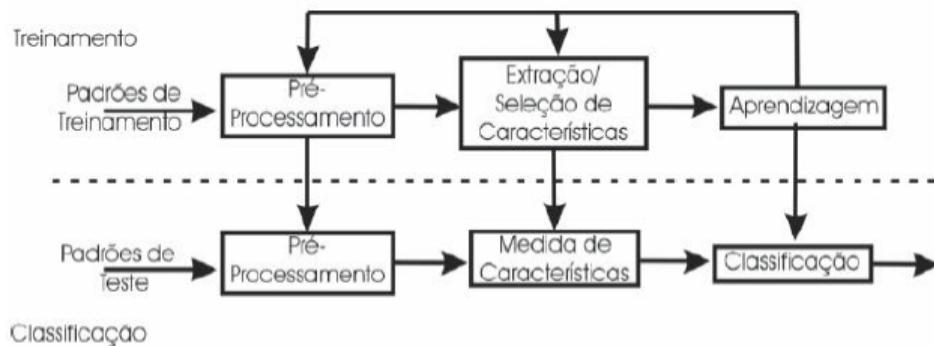


Figura 4 – Blocos funcionais para o reconhecimento de padrão na abordagem estatística. FONTE: JAIN, A., 2000

Os blocos funcionais envolvem projetar um sistema de reconhecimento de padrões, basicamente representam as etapas funcionais do sistema, sendo que inicialmente é feita a inserção de todos os padrões de treinamento a fim de descrever padrões de determinados objetos. Em seguida é realizado o pré-processamento onde é removido qualquer dado indesejado, ruídos ou dados inconsistentes, logo após os dados são enviados para extração/seleção de características que tem como objetivo representar padrões em termos de seus sub-padrões, além de permitir obter dados com características e dimensionalidades satisfatórias para realizar o reconhecimento de padrão. Na ultima etapa do processo tem-se a classificação onde são selecionadas técnicas e procedimentos que possibilitam a identificação de um objeto que o classificará para uma nova classe.

São múltiplas e importantes as relações de reconhecimento de padrões com outras disciplinas, em particular da área da Informática tais como: (JAIN; MAO; MOHIUDDIN, 1996):

- Processamento de sinal e imagem;
- Mineração de dados (*data mining and knowledge discovery*): busca por padrões significativos em espaços multi-dimensionais;

- Inteligência artificial;
- Aprendizagem automática (*machine learning*);
- Modelização neuronal (*artificial neural networks*);
- Conjuntos difusos (*fuzzy sets*);
- Busca e classificação em base de dados multimídia;
- Sistemas adaptativos;
- Reconhecimento Biométrico: identificação pessoal através da face íris ou impressão digital;
- Bioinformática: análise de sequências de proteínas ou DNA;

Reconhecer padrões nos dias atuais é parte fundamental da maioria dos sistemas de tomada de decisão, é simplesmente o ato de observar os dados brutos e tomar uma ação baseada na categoria de um padrão (DUDA; HART; STORK, 2012).

### 2.1.2 Problema da Interpolação exata

Os métodos de funções de base radiais têm origem em técnicas para realizar interpolação exata de um conjunto de dados num espaço multidimensional, essas funções em geral são não-lineares cujo valor cresce ou decresce monotonicamente à medida que a distância a um ponto central aumenta. O problema de interpolação (POWELL, 2001) pode ser formulado da seguinte maneira:

Dado um conjunto de  $p$  pontos diferentes  $\vec{x}_i \in R^n | i = 1, 2..p$  e um conjunto de escalares correspondentes  $\vec{d}_i \in R | i = 1, 2..p$ , encontre uma função  $\bar{f}: R^n \rightarrow R$  que satisfaça a condição de interpolação:

$$\bar{f}(\vec{x}_i) = \vec{d}_i, i = 1, 2..p \quad (2.1)$$

A técnica das RBF's (Rede Neural com função de ativação de base Radial) consiste em escolher uma função de aproximação  $\bar{f}$  que tem a forma:

$$\bar{f}(\vec{x}_i) = \sum_{i=0}^N w_i \cdot \phi_i \|\vec{x} - \vec{x}_i\| \quad (2.2)$$

Onde a norma representa em geral uma função euclidiana e  $\phi_i$  funções de base radial onde os centros coincidem com os pontos  $\vec{x}_i$  dados.

## 2.2 KNN - k vizinhos mais próximos

### 2.2.1 Introdução

O algoritmo *K-Nearest Neighbor* (kNN) é uma variação do algoritmo *Nearest Neighbor* (NN) (COVER; HART, 1967) o qual foi proposto por Cover e Hart em 1967. Surgiu com objetivo de melhorar o desempenho de classificação que propõe modificações durante a fase de teste/classificação do algoritmo NN original, onde se faz uso dos K-vizinhos mais próximos, ao invés de usar apenas o vizinho mais próximo, como propunha o algoritmo original.

Esse algoritmo é um tipo de aprendizagem baseada em instâncias, ou aprendizado preguiçoso (*Lazy Learning*) (WETTSCHERECK; AHA; MOHRI, 1997) e não paramétrico, isso deve ao fato de que o kNN não faz qualquer pressuposto sobre a distribuição dos dados e o termo preguiçoso refere-se ao algoritmo postergar a maior parte do processamento para o momento da predição, com isso reterá todo ou grande parte do conjunto de treino para realizar a etapa de teste. Segundo Webb “O kNN é um método simples de estimativa de densidade”, e recebe essa denominação pelo ao fato de estimar a densidade local de padrões de treinamento na vizinhança de um padrão desconhecido durante a classificação.

O kNN é uma técnica amplamente empregada para reconhecer padrões, particularmente adequado para classes multi-modal, bem como aplicações em que um objeto pode ter vários rótulos de classe . Por exemplo, para a atribuição de funções a genes baseados em perfis de expressão, alguns pesquisadores descobri-

ram que o kNN superou o SVM, que é um esquema de classificação muito mais sofisticado.(KURAMOCHI; KARYPIS, 2005)

### 2.2.2 Contexto

Basicamente, o centro do seu funcionamento está em descobrir o vizinho mais próximo em dada instância, então o kNN classifica um dado elemento com as respectivas classes dos  $k$  vizinhos mais próximos pertencentes a uma base de treinamento dado que aparece com mais frequência dentre seus  $K$ -vizinhos, o algoritmo calcula a distância do elemento dado para cada elemento da base de treinamento e então ordena os elementos da base de treinamento do mais próximo ao de maior distância. Dos elementos ordenados selecionam-se apenas os  $k$  primeiros, que servem de parâmetro para a regra de classificação. Considere o espaço de características da figura 5, as bolas azuis e as cruzes vermelhas representam classes distintas o círculo indica quais os  $k$ -vizinhos mais próximos da cruz preta. Por exemplo: 1-NN é um kNN sabendo que  $k = 1$ , ou seja, seleciona-se apenas o elemento do treinamento mais próximo da instância que se pretende classificar. 5-NN vai usar os cinco elementos mais próximos da instância baseado nas classes.

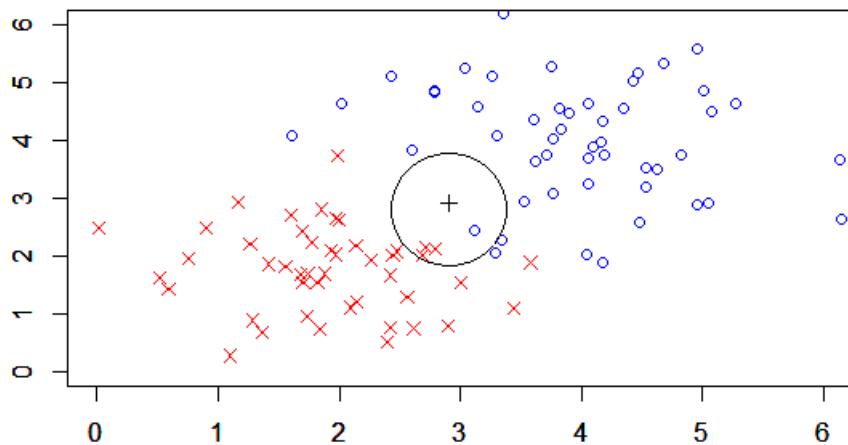


Figura 5 – Vizinhos no Espaço de características, com  $k = 4$

O algoritmo para a regra do chamado vizinho mais próximo é resumido da seguinte forma:

- Dado um vetor de característica desconhecido N (Conjunto de dados para treinamento);
- Defina uma métrica de distância;
- Defina um valor k, o qual será a quantidade de vizinhos mais próximos no espaço de características;

Dessa forma obtemos os dados a serem analisados, então o funcionamento do algoritmo pode ser descrito de forma mais clara como sendo (TAY; HYUN; OH, 2014)

- Defina um valor de k;
- Calcule uma métrica de distância;
- Fora dos N vetores de treinamento, identifique os vizinhos mais próximos, independentemente dos rótulos das classes.
- Fora dessas k amostras, conte o número de vizinhos mais próximos que pertencem a cada classe  $k_i$  com  $i = 1, 2, \dots, M$ . Assim temos:  $\sum_i k_i = k$
- Atribua o rótulo à classe mais frequente na vizinhança. Classificando o novo elemento.

A definição da similaridade entre as instâncias é dada por uma função de distância. Embora a métrica euclidiana seja a mais utilizada, a noção de uma métrica é muito mais geral e há inúmeras medidas alternativas de distância para resolver problemas-chave na classificação (THEODORIDIS; KOUTROUMBAS et al., 1999):

### 1. Distância Euclidiana

$$d(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2} \quad (2.3)$$

### 2. Distância Manhattan

$$d(x, y) = \sum_{i=1}^m |x_i - y_i| \quad (2.4)$$

3. Minkowski

$$d(x, y) = \left( \sum_{i=1}^m |x_i - y_i|^r \right)^{\frac{1}{r}} \quad (2.5)$$

4. Camberra

$$d(x, y) = \sum_{i=1}^m \frac{|x_i - y_i|}{|x_i + y_i|} \quad (2.6)$$

5. Chebychev

$$d(x, y) = \sum_{i=1}^m \max |x_i - y_i| \quad (2.7)$$

### 2.2.3 A escolha de um k apropriado

Decidir quantos vizinhos para usar para kNN determina o quanto bem o modelo vai generalizar a dados futuros. O equilíbrio entre *overfitting* e *underfitting* dos dados de treinamento é um problema conhecido como a compensação de desvio de tendência (*bias-variance tradeoff*). Fazer a escolha por um k grande reduz o impacto ou variância causado por dados ruidosos, porém na prática pode correr o risco de ignorar padrões pequenos que são importantes. Em contrapartida, o uso de um k pequeno, um vizinho mais próximo permite dados ruidosos o que pode influenciar indevidamente a classificação de exemplos. Obviamente, o melhor valor k está em algum lugar entre esses dois extremos. A figura seguinte ilustra como a decisão de um modo geral mais limite (representada por uma linha tracejada) seja maior ou menor afetadas por valores de k. Valores menores permitem que mais limites de decisão complexos cuidadosamente cabem mais que os dados de treinamento. O problema é que não sabemos que, quer o limite reta ou o limite curvo melhor representa o verdadeiro conceito subjacente a ser aprendida. (LANTZ, 2013)

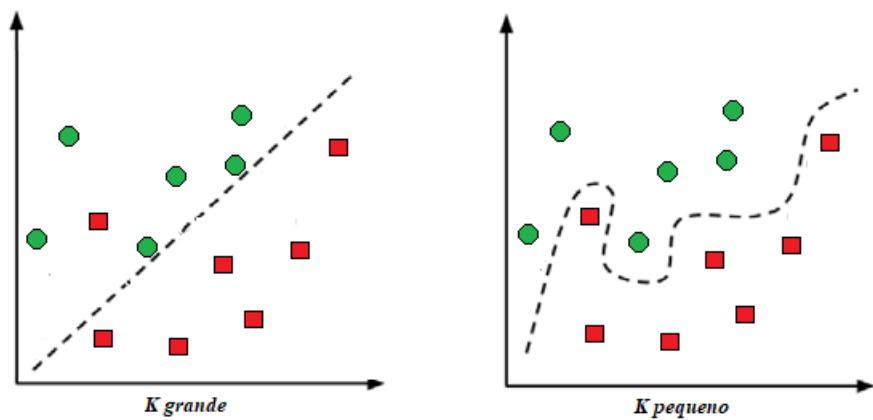


Figura 6 – Escolha de um  $k$  apropriado: *underfitting* e *overfitting* respectivamente

Na prática a escolha de  $k$  depende da dificuldade do conceito a ser aprendido e número de registros nos dados de treinamento. Basicamente segundo Lantz: "o  $k$  ótimo permanece na escala entre três à dez."

## 2.3 Redes RBF - Função de base radial

### 2.3.1 Introdução

As RNAs (Redes Neurais Artificiais) do tipo Função de base radial (RBFs) são redes supervisionadas assim como as RNAs *Multilayer Perceptron* (MLPs) porém apesar da semelhança suas arquiteturas são distintas. Em 1988, ([BROOMHEAD; LOWE, 1988](#)) descreveram um procedimento para um projeto de redes utilizando RBF aos quais oferecem alternativas aos perceptrons de múltiplas camadas, onde a saída da rede é uma combinação linear de funções base radiais das entradas e parâmetros dos neurônios. A ideia básica foi fundamentada a partir do método das funções de potencial o qual foi proposto originalmente por Bashkirov, Braverman e Muchnik (1964).

Outro embasamento teórico além do método de aproximação de funções tem como redes RBF um sistema neural de inspiração biológica e aprendizado rápido. Tal modelo baseia-se nos campos sobrepostos de recepção presentes em regiões do córtex cerebral ligadas a processamentos de informações, tais atividades locais de

seus processadores traz grande vantagem em rapidez de aprendizado (JANG, 1992).

As RBFs são redes neurais que empregam funções de base radiais nos neurônios da camada oculta. Uma rede RBF se diferencia da maioria das redes multicamadas por utilizar como argumento da função de ativação uma distância entre seus vetores de entrada e seus pesos, enquanto que MLPs (Perceptron de Múltiplas Camada) utilizam o produto escalar do vetor de entrada e do vetor peso para um nodo. Com utilização dessa função na camada intermediária as RBFs são capazes de separar os padrões de classes distintas através de hiperelipsóides, enquanto as MLPs utilizam a separação por hiperplanos. As funções de bases radiais são representantes de uma classe de funções cujo valor diminui ou aumenta em relação a um ponto central. (BRAGA; CARVALHO; LUDELMIR, 2000)

### 2.3.2 Contexto

As redes de função de base radial têm muitos usos, incluindo a aproximação de funções, previsão de séries temporais, classificação e controle de sistema. Dado um conjunto de vetores de entrada ( $x_1 \dots x_n$ ) juntamente com seus valores pesos correspondentes ( $w_1 \dots w_n$ ) tem como objetivo encontrar uma função suave  $y(x)$  que corresponde a cada valor de destino exatamente, um nó de saída associe um valor de peso a cada um dos neurônios de RBF e multiplique a ativação do neurônio por esse peso antes de adicioná-lo à resposta final.

$$y(x) = \sum_{j=0}^N w_j \cdot \phi_j \quad (2.8)$$

Onde  $\phi_j$  é a função base radial.

Normalmente as RBFs têm uma arquitetura simples com três camadas: uma camada de entrada, uma camada oculta com uma função de ativação de RBF não linear e uma camada de saída linear e seu treinamento é realizado em 2 etapas (MOCLOONE, 1998):

- Treinamento (não supervisionado): determinação dos centros ( $\mu$ ) e variância ( $\sigma^2$ ) de cada um dos K centros (camada oculta);

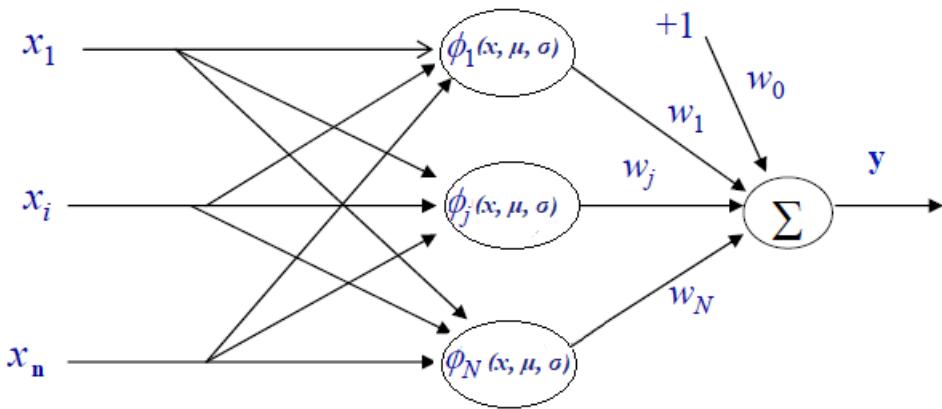


Figura 7 – Representação de vários hiperplanos de separação. FONTE: adaptado de HAYKIN, 1994

- Treinamento (supervisionado): determinação dos pesos sinápticos de saída da rede RBF (são realizados os ajustes apropriados nos pesos das conexões de modo a reduzir esta distância);

### 2.3.3 Treinamento - Não Supervisionado

Historicamente no primeiro estágio determina-se os parâmetros das funções de base radial o qual foram introduzidas com o propósito de interpolação de função exata (POWELL, 2001). O problema consiste então em localizar os centros e parâmetros da base radial e ajustar os pesos em relação ao treinamento.

Tipicamente, numa rede RBF cada neurônio da camada oculta calcula uma função base radial, onde sua característica principal é que sua resposta diminui (ou aumenta) de acordo com a distância de um ponto central. A função base radial mais comumente utilizada é a gaussiana:

$$\phi_j(x) = \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right) \quad (2.9)$$

onde  $\mu$  corresponde ao centro da função e  $\sigma$  controla a suavidade da interpolação. Pode-se encontrar inúmeras variações de funções base radial (BUHMANN, 2003), como mostrado na tabela abaixo:

## 1. Multi Quadrática

$$\phi(x) = \sqrt{(x^2 + \sigma^2)} \quad (2.10)$$

## 2. Multi Quadrática Inversa

$$\phi(x) = \frac{1}{\sqrt{(x^2 + \sigma^2)}} \quad (2.11)$$

## 3. Lâmina Spline Fina

$$\phi(x) = \frac{x}{\sigma^2} \log \frac{x}{\sigma} \quad (2.12)$$

Já a heurística para definir o raio ( $\sigma$ ) pode ser determinado pela distância Euclidiana ([HAYKIN, 2001](#)):

$$\sigma = \frac{1}{m} \sum_{i=1}^m \|k_i - k_j\| \quad (2.13)$$

onde  $k_i$  é o centro mais próximo de  $k_j$ . Atribuindo a cada unidade uma largura diferente à distância do seu centro ao da unidade mais próxima e  $1.0 \alpha 1.5$ .

$$\sigma = \alpha \|k_i - k_j\| \quad (2.14)$$

*Calculo dos pesos sinápticos ( $w_i$ )*

Com os parâmetros das RBF's definidos, basta determinar o vetor de pesos sinápticos obtido a partir de uma operação de inversão matricial aplicada as condições de interpolação. Considere a matriz de interpolação  $\phi$  formada pelos vetores  $\phi_i$ , onde  $i = 1, 2..N$ , com  $N$  sendo o número de vetores pertencentes ao conjunto de treinamento da rede neural aos quais resultam da aplicação dos vetores  $x_i \in \mathbb{R}$ , com  $i = 1, 2..N$  à entrada da rede RBF, obtendo o seguinte conjunto de equações lineares:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & .. & \phi_{n1} \\ \phi_{21} & \phi_{22} & .. & \phi_{n2} \\ : & : & : & : \\ \phi_{nn} & \phi_{nn} & .. & \phi_{nn} \end{bmatrix} \times \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \times \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix}$$

Onde  $\phi_{ji} = \phi(\vec{x}_j)$ , onde  $\phi$  é a matriz de interpolação e  $\vec{w}$  e  $\vec{d}$  correspondem ao vetor de pesos lineares e respostas desejadas respectivamente. Assumindo a não singularidade de  $\phi$  o vetor de pesos lineares  $\vec{w}$  pode ser obtido a partir da equação:

$$\vec{w} = \phi^{-1} \cdot \vec{d} \quad (2.15)$$

A não singularidade da matriz  $\phi$  pode ser provada a partir do *teorema de Micchelli*(1986), ao qual garante que para uma classe de funções de base radial a matriz  $\phi$  é não singular.(HAYKIN et al., 2009) Assim, o neurônio que compõe a camada de saída é definido como um combinador linear das funções de base radial.

## 2.4 SVM - Máquinas de Vetores de Suporte

### 2.4.1 Introdução

As Máquinas de Vetores de Suporte (*SVMs, Support Vector Machines*) são embasadas pela teoria de aprendizado estatístico (TAE), proposta por Vapnik. Tal teoria estabelece uma série de princípios que devem ser seguidos na obtenção de classificadores de padrões e regressão linear com boa generalização, definida como a sua capacidade de prever corretamente a classe de novos dados do mesmo domínio em que o aprendizado ocorreu, além de buscar minimizar o erro do conjunto de treinamento (risco empírico), juntamente com o erro do conjunto de teste.

As SVMs são classificadores discriminativos que atuam na separação de padrões por meio de hiperplanos como uma superfície de decisão que opera como um classificador de duas categorias, de modo que se maximize a margem de separação entre os exemplos pertencentes a ambas as categorias. Esta classificação linear se efetua no espaço oculto, transformado a partir do espaço de entrada, mediante um núcleo (*kernel*) não linear.(AMORIM, 2006). Em outras palavras, dado um conjunto de amostras rotuladas para treino, o algoritmo busca hiperplanos ótimos para categorizar as amostras de teste. Para isto, o algoritmo SVM baseia- se em encontrar o hiperplano que gera a maior distância entre amostras de classes diferentes para o conjunto de treino.

As Máquinas de Vetores de Suporte vem sendo utilizadas em diversas tarefas de reconhecimento de padrões, podem ser comparados à outros algoritmos de aprendizado, como as Redes Neurais Artificiais (RNA's), o qual inclui *KNN* (*K-Nearest Neighbors*) e a rede RBF (*Radial basis function*)[\(HAYKIN, 2001\)](#) obtendo em muitos casos, resultados superiores aos alcançados por essas técnicas. Algumas possíveis áreas de atuação podem ser encontradas na detecção de homologia remota de proteínas[\(RANGWALA; KARYPIS, 2005\)](#), classificação da expressão facial [\(REILLY; GHENT; MCDONALD, 2006\)](#), SVM para Ciências Geológicas e Ambientais [\(KANEVSKI; POZDNUKHOV; TIMONIN, 2008\)](#) dentre outras aplicações de sucesso.

#### 2.4.2 Contexto

Basicamente uma máquina de vetor suporte tem como objetivo encontrar o hiperplano de separação ideal o qual maximiza a margem da base de treinamento e por ser um algoritmo de classificação, obviamente é usado para predizer se algo pertence a uma classe particular.

Em 1963, Vladimir Vapnik propôs o algoritmo do hiperplano de máxima margem, para uma classificação linear. As pesquisas continuaram com a intenção de encontrar uma separação não-linear. Em 1992, Bernhard Boser, Isabelle Guyon e Vladimir Vapnik sugeriam a separação não linear, utilizando as funções *Kernel*, que transformam o espaço de entrada em um espaço de dimensão maior, onde os dados são linearmente separáveis.

#### 2.4.3 Teoria do aprendizado estatístico - TAE

A Teoria de Aprendizagem Estatística (TAE) proposta por Vapnik [\(LORENA; CARVALHO, 2007\)](#) no contexto de SVM, objetiva controlar, em termos matemáticos, a habilidade de generalização da técnica sendo que há necessidade de uma teoria que permita descrever precisamente os fatores que devem ser controlados para garantir bom desempenho de generalização com a finalidade de convergir para uma solução ótima. Dado  $f$  um classificador e  $F$  o conjunto de todos os classificadores que um dado algoritmo de aprendizado de máquina é capaz de induzir.

Um algoritmo durante o processo de aprendizado utiliza um conjunto de treinamento  $T = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  composto de  $n$  pares  $(x_i, y_i)$  onde  $x_i \in R^m$  e  $y_i \in (+1, -1)$ . Sendo que vetores  $x_i$  e  $y_i$  elementos do mundo real e possíveis classes de saída respectivamente. A solução se resume em encontrar uma função definida como  $f: R^m \rightarrow (+1, -1)$ , tal que  $f \in F$ . Para alcançar a melhor aproximação entre a saída produzida pela máquina de aprendizagem e a saída desejada, é necessário trabalhar com uma medida chamada risco (erro) mínimo.

$$R(f) = \int c(f(x), y) dP(x, y) \quad (2.16)$$

Geralmente a distribuição de probabilidade  $P(x,y)$  é desconhecida e os dados disponíveis estão relacionados somente aos conjuntos de treinamento, portanto, não é possível minimizar o risco esperado diretamente. Sendo assim, é comum utilizar o risco empírico que é definida pela medida da taxa de erro no conjunto de treinamento para um número de observações finitas e fixas. Portanto, para minimizar o erro Funcional de Risco  $R(f)$  definida em termos de uma função de distribuição desconhecida  $P(x,y)$  pode ser aplicado o princípio indutivo da Minimização do Risco Empírico. ([SMOLA; SCHÖLKOPF, 2004](#)):

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^n c(f(x_i), y_i) \quad (2.17)$$

Entretanto, apenas o cálculo risco empírico não implica em um erro de teste ou risco pequenos, pois um pequeno erro de generalização não pode ser obtido simplesmente minimizando o erro de treinamento ([SCHÖLKOPF; BURGES; SMOLA, 1999](#)). Sendo assim, o conceito de capacidade mais conhecido da Teoria VC é a dimensão VC, que é uma medida da capacidade de classificação realizadas pela máquina de aprendizagem e pode ser definida para várias classes de funções:

$$R(f) \leq E_{emp}(f) + \sqrt{\frac{h(\ln(\frac{2}{4}) + 1) - \ln(\frac{\theta}{4})}{4}} \quad (2.18)$$

Sendo  $h$  a dimensão da teoria de Vapnik e Chervonenkis (VC) da classe de funções  $F$  à qual  $f$  pertence,  $\theta$  é um valor real entre 0 e 1 e a quantidade de amostras presentes no treinamento  $T$ . Segundo Haykin o erro de treinamento, para um número

fixo de exemplos diminui quando a dimensão VC é aumentada, enquanto que o intervalo de confiança aumenta monotonicamente fazendo com que o erro convirja para o mínimo podendo tornar-se intratável. Para superar essa limitação utiliza-se a estratégia de introduzir uma "estrutura" a partir da divisão da classe de funções de entrada em subconjuntos aninhados. Então, dado o conjunto  $F_i$  pode ser definido como:  $F_0 \subset F_1 \subset F_2 \subset \dots \subset F_n \subset F$ , em que  $\text{VC}(F_0) \leq \text{VC}(F_1) \leq \text{VC}(F_2) \leq \dots \leq \text{VC}(F_n) \leq \text{VC}(F)$ . Notando -se que uma classe pequena pode reduzir o termo da raiz quadrada na equação mas o número de erros continua alto, como mostra na figura (SMOLA; SCHÖLKOPF, 2004):

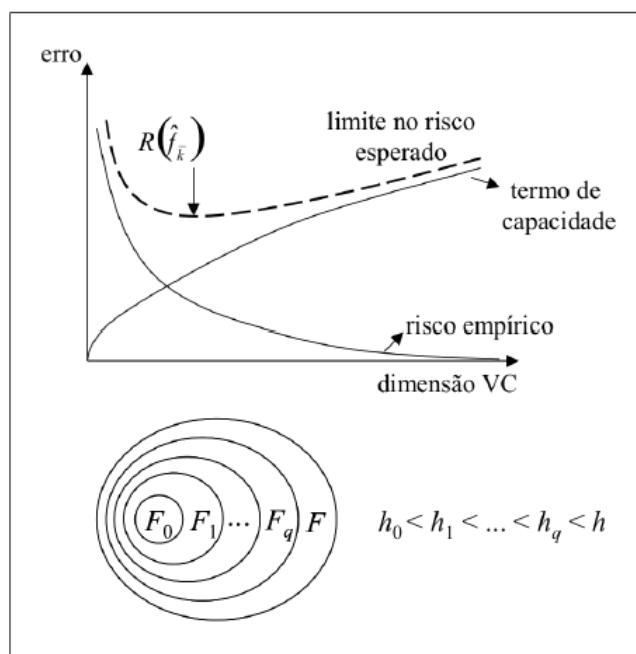


Figura 8 – Princípio da minimização do risco estrutural. FONTE: SCHOLKOPF; SMOLA , 2002

#### 2.4.4 Hiperplano

Na geometria, um hiperplano pode ser um espaço vetorial, transformação afim ou o sub-espacô de dimensão  $n-1$ . Em particular, num espaço tridimensional um hiperplano é um plano habitual. O fato é que você pode encontrar um hiperplano de separação, mas não quer dizer que é o melhor! No exemplo abaixo há vários

hiperplanos de separação. Cada um deles são válidos e separam nosso conjunto de dados em homens de um lado e mulheres de outro lado.

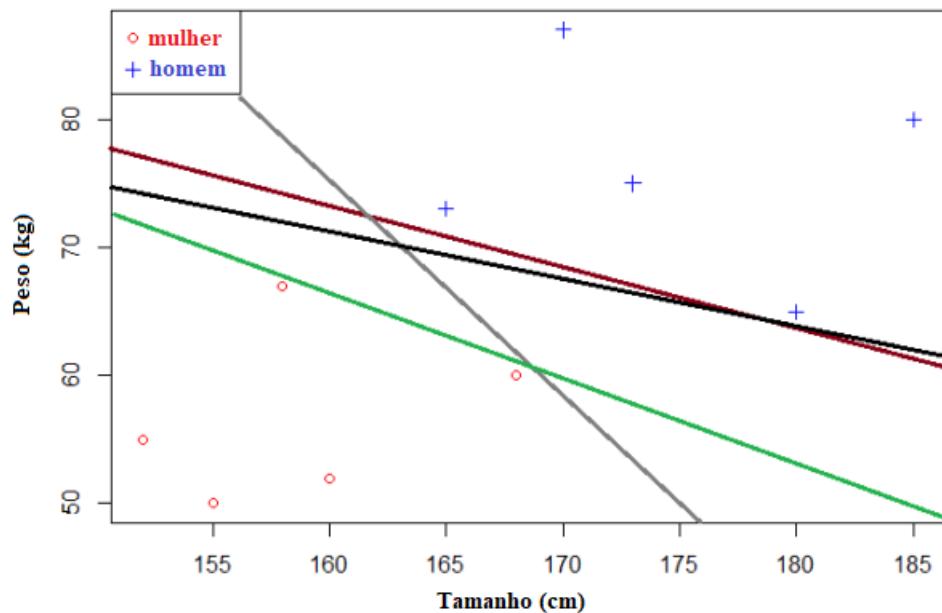


Figura 9 – Representação de vários hiperplanos de separação. FONTE: adaptado de KOWALCZYK, 2014

### 2.4.5 Hiperplano ótimo para padrões linearmente separáveis

Assumindo-se que o conjunto de treinamento é linearmente separável representado pelo subconjunto  $d_i = +1/-1$  e considerando o conjunto como:  $\{(x_i, y_i)\}_{i=1}^N$  onde  $x_i$  é o padrão de entrada para o i-ésimo exemplo e  $d_i$  é a resposta de saída desejada. o hiperplano ótimo é o hiperplano de separação com maior margem. Assim temos:

$$(w \cdot x) + b \geq 0, \text{ para } d_i = +1 \quad (2.19)$$

$$(w \cdot x) + b < 0, \text{ para } d_i = -1 \quad (2.20)$$

Sendo  $w$  e  $b$  o vetor peso ajustável e o bias respectivamente. Tais condições podem ser simplificadas como na figura abaixo:

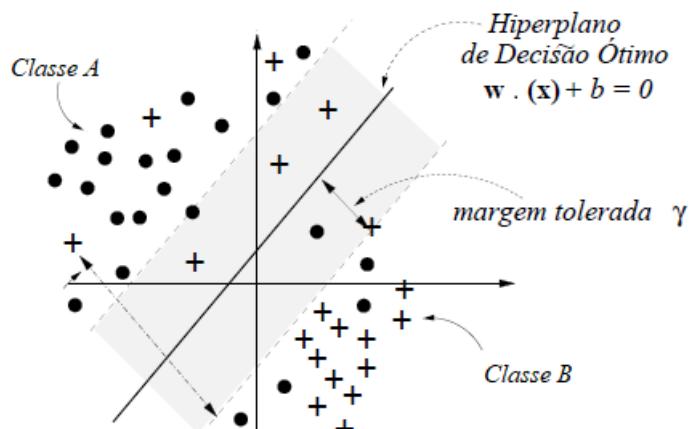


Figura 10 – Representação do plano e margens obtidas no treinamento do SVM.  
FONTE: adaptado de Kavzoglu; Colkesen, 2009

O SVM é uma metodologia que basicamente classifica os padrões em dois conjuntos, mas que pode ser ampliado para classificações em múltiplas classes. Devido à metodologia utilizar margens (vetores) de separação para os conjuntos, as quais são deslocamentos com distância igual a um, tem-se o conjunto de classe -1 e outro de classe +1. Sendo assim, a SVM produz um grupo de hiperplanos os quais os limites de dimensão VC possam ser computados e então utiliza o princípio de minimização do risco estrutural para detectar o hiperplano ótimo que potencialize a margem entre as amostras de treinamento mais próximas. Porém quando o Classificador de margem máxima é aplicado a dados não separáveis linearmente não encontra a solução desejada, para dados com ruídos, que geralmente implica em separação não linear, o mínimo para o risco esperado não pode ser calculado dessa forma, pois pode causar *overfitting*.

#### 2.4.6 Hiperplano ótimo para padrões não-linearmente separáveis

Uma aplicação pode para o problema das SVMs não lineares surge a partir do teorema de MERCER ([SMOLA; SCHÖLKOPF, 2004](#)) para que uma função *Kernel* seja encontrada. Essa estratégia permite tolerar ruídos considera mais pontos de treinamento, além dos que estão na fronteira e permite a ocorrência de erros de classificação. Portanto,  $\sum_{k=1}^N \xi$  é um limite superior para o número de erros de treinamento e suas restrições mudam em relação ao hiperplano linearmente separável:

$$(w \cdot x) + b \geq +1 - \xi, \text{ para } d_i = +1 \quad (2.21)$$

$$(w \cdot x) + b \leq -1 + \xi, \text{ para } d_i = -1 \quad (2.22)$$

$$\xi \geq 0, \forall x \quad (2.23)$$

E a função a ser minimizada passa a ser:

$$\frac{1}{2} \|B\|^2 + C(\sum_{k=1}^N \xi)^k \quad (2.24)$$

onde C representa um parâmetro a ser escolhido aleatoriamente ao qual determina a relação entre o erro empírico e o termo de confiança, k é um inteiro positivo. Assim, a única diferença do caso do hiperplano ótimo é que os  $a_i$  tem os limites superiores em C, sendo a solução:

$$w = \sum a_i y_i x_i \quad (2.25)$$

As representações *Kernel* trabalham com a projeção dos dados em um espaço de características com alta dimensão para permitir a classificação em espaços não-linearmente separáveis. Trata-se, em primeira instância, de uma estratégia de pré-processamento que envolve mudar a representação dos dados

É comum empregar a função *Kernel* sem conhecer o mapeamento que é gerado implicitamente. A utilidade dos *Kernels* está, portanto, na simplicidade de seu cálculo e em sua capacidade de representar espaços abstratos.

Tabela 1 – Algumas das funções *Kernels* mais utilizadas

<b>Tipo de Kernel</b>	<b>Função K (<math>x_i, x_j</math>)</b>	<b>Tipo do classificador</b>
Polinomial	$((x_i \cdot j_i) + 1)^2$	Máquina de aprendizagem polinomial
Gaussiano (RBF)	$\exp\left(-\frac{\ x_i - x_j\ ^2}{2\sigma^2}\right)$	Rede RBF
Linear	$x \times y$	classificador linear
Curva Sigmoide	$\tanh(\beta_0(x_i \cdot x_j)) + \beta_1$	Perceptron de duas camadas

## 2.5 Transformada de Fourier

A transformada de Fourier (TF), foi descoberta por Jean-Baptiste Joseph Fourier (1768-1830) em 1822 que descobriu que qualquer função temporal pode ser analisado por um espectro de frequência. Como se sabe há uma ampla classe de sinais que pode ser representada como uma combinação linear de exponenciais complexas, enquanto para os sinais periódicos as exponenciais complexas que o representam estão relacionadas harmonicamente, para um sinal aperiódico elas estão infinitesimalmente próximas em frequência, e a representação em termos de uma combinação linear torna a forma de uma integral em vez de uma soma. O espectro de coeficientes resultante nessa representação é chamada de TF. (OPPENHEIM, 2010)

Um sinal aperiódico pode ser visto como um sinal periódico com um período infinito. Suponha  $x(t)$  um período do sinal periódico  $\tilde{x}(t)$ . Quando o período  $T \rightarrow \infty$ ,  $x(t) \rightarrow \tilde{x}(t)$ . A representação em série de Fourier do sinal periódico  $\tilde{x}(t)$  é:

$$\tilde{x}(t) = \sum_{k=-\infty}^{\infty} a_k e^{jkw_0 t} \quad (2.26)$$

Sendo  $\tilde{x}$  :

$$a_k = \frac{1}{T} \int_T x(t) e^{-jkw_0 t} dt = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{-jkw_0 t} dt \quad (2.27)$$

Considerando:

$$\tilde{x}(t) = \begin{cases} x(t), & \text{para } |t| < T/2 \\ 0, & \text{cc} \end{cases} \quad (2.28)$$

Podemos reescrever os limites de  $a_k$ , com  $w = kw_0$  como:

$$a_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \tilde{x}(t) e^{-jkw_0 t} dt = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{-jkw_0 t} dt = \frac{1}{T} \int_{-\infty}^{\infty} x(t) e^{-jkw_0 t} dt \quad (2.29)$$

Portanto a Transformada de Fourier é definida como:

$$X(jw) = \int_{-\infty}^{\infty} x(t) e^{-jwt} dt \quad (2.30)$$

Também denominada espectro de frequências de  $x(t)$ .

## 2.6 Considerações finais

De forma geral pode-se afirmar que a classificação utilizando o algoritmo kNN é uma técnica bastante simples e de fácil implementação, isso porque é um algoritmo capaz de realizar paralelismo sendo possível executar trechos de códigos separados aumentando assim seu desempenho, além de ser uma técnica muito flexível em relação à distribuição de dados em seu espaço de características. Por outro lado o algoritmo torna-se exaustivo e sua complexidade computacional aumenta uma vez que requer um cálculo de distância para cada elemento do conjunto de treinamento, consumindo muito tempo quando se tem um conjunto de treinamento muito grande.

Já as redes neurais artificiais do tipo RBF são ferramentas que tem capacidade de aprender rapidamente padrões complexos e apresentam potencialidades de generalização local em que a interferência no aprendizado é minimizada além de se adaptar rapidamente as mudanças. Os métodos de treinamento se tornam outro grande diferencial nas RBFs uma vez que os mesmos separam as tarefas em dois estágios: as funções de base radial e os métodos lineares de convergência rápida. A vantagem dessa abordagem é a rapidez no treinamento da rede como resultado da separação das tarefas e aplicação de um método híbrido de treinamento em alternativa ao treinamento global e a otimização não-linear explícita da descida de encosta.

Outro algoritmo que se mostra bastante eficiente é o SVM o qual se dispõe de grande capacidade de generalização diante de dados com grandes dimensões sobre os quais outras técnicas de aprendizado normalmente obtém classificadores não tão satisfatórios. Tal capacidade é medida por sua eficiência na classificação de dados que não pertencem ao conjunto utilizado em seu treinamento, evitando assim, o *overfitting* (memoriza os padrões de treinamento, gravando suas peculiaridades e ruídos, ao invés de extrair as características gerais que permitirão a generalização ou reconhecimento de padrões não visto durante o treinamento). Sendo assim, pode-se dizer que é um método simples de ser utilizado, e a tarefa de classificação é bem rápida e precisa, embora o tempo de treinamento possa ser longo dependendo do número e da dimensionalidade das amostras. Uma das grandes desvantagens dos SVMs é a complexidade computacional envolvida na busca de soluções, tarefa que se

torna mais custosa para volumes grandes de dados.

## 3 MÉTODOLOGIA

### 3.1 Introdução

Para realizar a classificação dos algoritmos de aprendizado de máquina, foi selecionado uma base de dados disponibilizada na mega revista científica de acesso aberto com revisão por pares cobrindo a investigação nas ciências biológicas e médicas (PeerJ) pela Universidade Federal do ABC ([FUKUCHI; FUKUCHI; DUARTE, 2017](#)).

Os testes foram feitos em uma máquina Intel Core i5 x64 2.27GHZ, com 4GB de memória RAM e capacidade de disco de 500 GB com sistema operacional Windows 10. Os algoritmos de classificação foram implementados na linguagem MatLab (versão R2017b) e linguagem R juntamente com o Software RStudio (versão 1.0.136), que possuem alto desempenho estatístico, processamento de sinais e manipulação de matrizes. Sendo assim, para determinar os padrões descritos algumas adaptações na base de dados foram necessárias para trabalharmos com a classificação e são descritas nas subseções seguintes.

### 3.2 Descrição da base de dados

O conjunto de dados compreende-se na cinemática da marcha bruta e dos membros inferiores e os sinais cinéticos de 28 sujeitos. O estudo utilizou 48 marcadores reflexivos e uma esteira instrumentada nos atletas, e os dados foram coletados usando um sistema de captura de movimento tridimensional (3D) em uma esteira instrumentada, enquanto os sujeitos percorreram 2,5 m/s, 3,5 m/s e 4,5 m/s usando calçados neutros padrão.

Os arquivos contêm tanto as trajetórias dos marcadores brutos quanto os sinais de força, e foram divididos em arquivos separados entre "marcadores" e "sinais de força" sendo que os mesmos são amostrados a 150 Hz para cinemática e 300 Hz para dados cinéticos. Afim de realizar nossos estudos usaremos os dados cinéticos (dados de força) que são normalizados no tempo. Existem três arquivos de texto

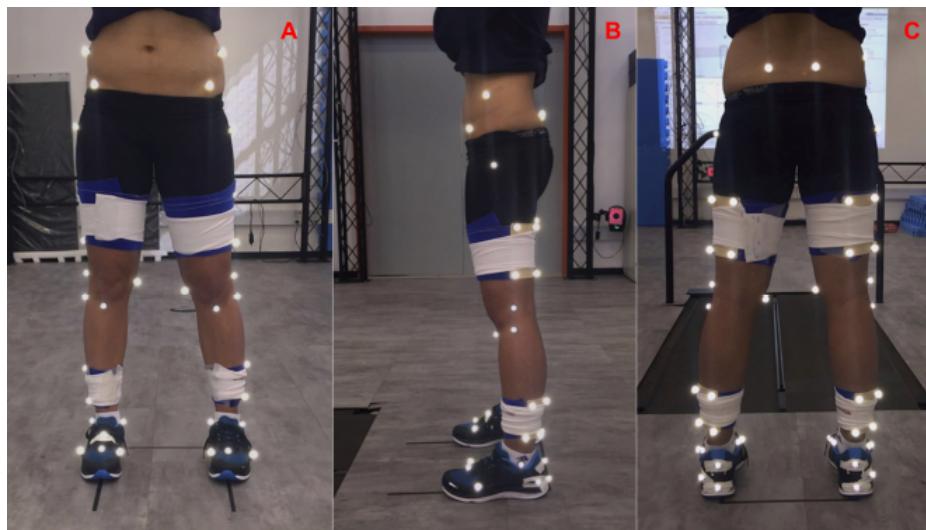


Figura 11 – Marcadores reflexivos e esteira instrumentada. FONTE: ([\(FUKUCHI; FUKUCHI; DUARTE, 2017\)](#))

por assunto, a abreviatura RBDS (*Running Biomechanics Dataset*) nos nomes dos arquivos significa "Executando Conjunto de Dados Biomecânicos" e xxx refere-se ao número de identificação do sujeito (por exemplo, o primeiro sujeito é 001 dentre os 28).

Por sua vez, as colunas dos dados das forças são compostas pelas características: tempo, forças, o centro de pressão e o momento livre sobre o eixo vertical medido pela esteira instrumentada e transformados no sistema de coordenadas laboratoriais. Sendo assim, cada arquivo de força é basicamente composto pelas forças, centro de pressão (em coordenadas x,y e z) e o momento livre sobre o eixo vertical. Segue o seguinte cabeçalho: Tempo [s], Fx [N], Fy [N], Fz [N], COPx [mm], COPY [mm], COPz [mm] e Ty [Nm], distribuídos por dados em 9.000 linhas e oito colunas com precisão numérica de seis dígitos.

Há também um arquivo chamado RBDSinfo (no formato .csv) suportado com metadados, dados demográficos, características de execução, padrões, medição muscular e medidas de flexibilidade para cada assunto, tais como a classe "lesão" que é objeto do nosso estudo, onde indica qual dentre os 28 atletas já obteve ou não, alguma lesão.

### 3.3 Pré-Processamento - Features

Antes de fazer a aplicação dos algoritmos de aprendizado de máquina se fez necessário realizar um pré-processamento e obter um autoconhecimento dos dados brutos para podermos extraír o máximo de informação afim de escolhermos nossa teia de determinantes, ou seja, os dados que iremos trabalhar. Sendo assim, observou-se alguns comportamentos como o teste da Correlação na Figura 12, ele não nos mostra exatamente e com total certeza quais dados estão correlacionados porém, em todos os testes realizados os dados nos mostram que há algumas semelhanças entre os 28 sujeitos observados.

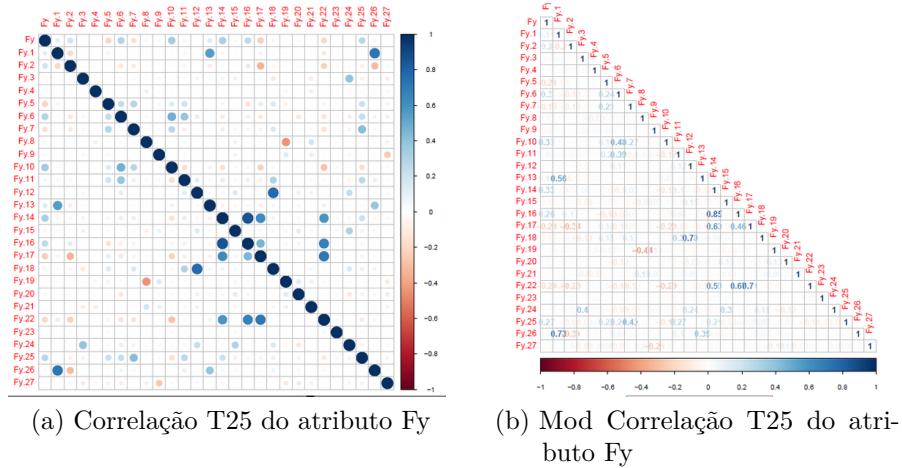


Figura 12 – Teste de Correlação afim de adquirir algum conhecimento sob a base de dados original

#### 3.3.1 Transformada de Fourier

Um atributo interessante que foi extraído são as Transformadas de Fourier (TF), tendo os sinais brutos no domínio do tempo e contínuo, observou-se um comportamento peculiar dos sinais de força nos 3 eixos x, y e z em relação ao tempo como podemos ver na Figura 13. Tal comportamento senoidal nos permite analisar de forma adequada funções aperiódicas e aplicar a TF.

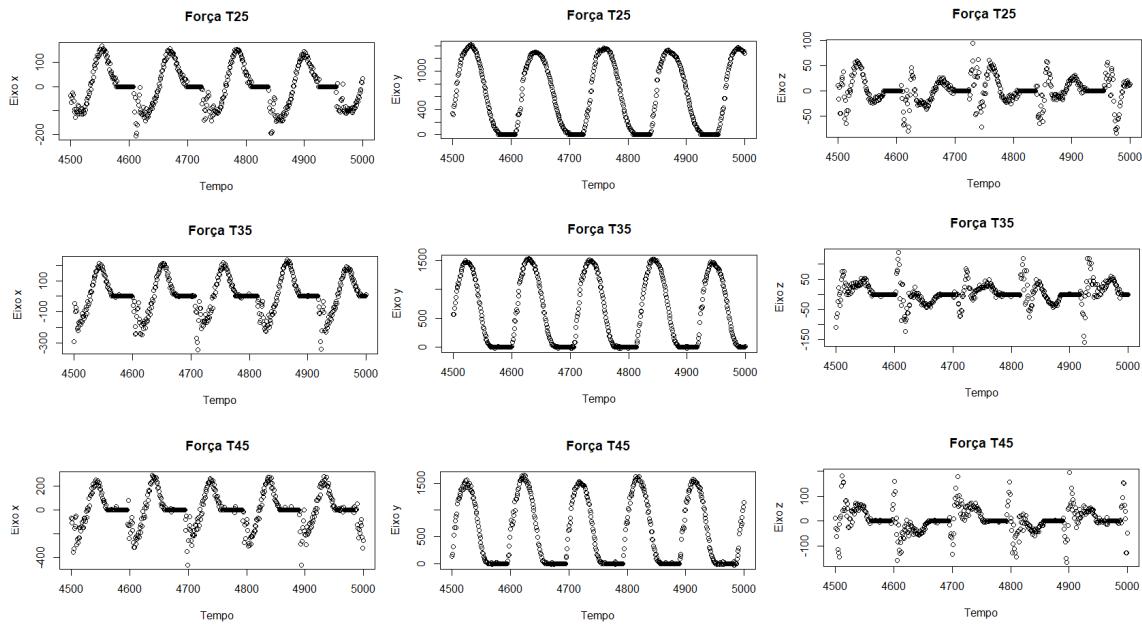


Figura 13 – Dados brutos no domínio do tempo em relação aos eixos x, y e z do atributo Força de um sujeito aleatório para mostrar aperiodicidade

Dessa forma sabemos que todo sinal pode ser escrito como uma superposição de senoides complexas, podemos escrever nossos sinais como uma soma de outros sinais. A TF nos permite analisar de forma adequada funções não periódicas. Uma vez que observamos senoides e suas respectivas frequências atribuídas à elas, mudamos claramente nossa visão periférica e perspectiva sob os sinais. Na figura 14 temos uma amostra da TF que mostra a frequência em Hz em relação à amplitude sob o atributo "Momento livre sob o eixo vertical" de um sujeito escolhido aleatoriamente.

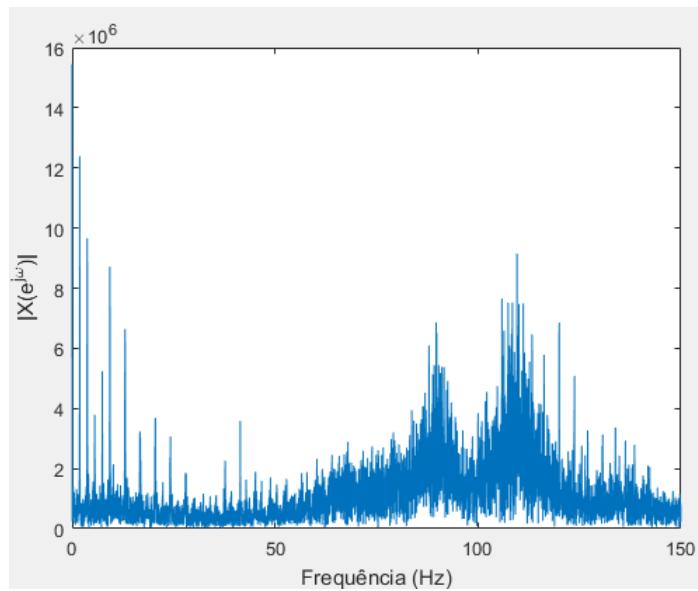


Figura 14 – Transformada de Fourier do sujeito 005, atributo "momento livre sob o eixo vertical"

### 3.4 Métodos de avaliação - Matriz de Confusão e Acurácia

Uma ferramenta padrão para avaliação de modelos estatísticos e classificação é a matriz de confusão, que nada mais é que uma matriz de valores reais e valores preditos pelo seu classificador. Um gráfico é criado quando você escolhe a opção Matriz de Classificação e compara valores reais com os previstos. As categorias usadas na análise são falsos positivos (FP), verdadeiros positivos (VP), falsos negativos (FN) e verdadeiros negativos (VN) como mostra a tabela.

Tabela 2 – Exemplo matriz de confusão

		Valor Real	
Previsto	Positivo	Negativo	
	VP	FP	
Negativo	FN	VN	

O funcionamento da matriz é simples: consideramos valores positivos que o sistema julgou positivos como verdadeiros positivos (acerto), valores positivos que o sistema julgou negativos como falso negativo (erro), valores negativos que o sistema

julgou como negativos como verdadeiros negativos (acerto), e valores negativos que o sistema julgou positivos como falso positivos (erro).

Outra medida de avaliação será a acurácia ao qual significa proporção de predições corretas, sem levar em consideração o que é positivo e o que é negativo.

### 3.5 Base de dados final

Depois de realizar o pré processamento e escolher minuciosamente as *features* extraídas da base original, ou seja, os atributos aos quais nos interessam, temos por fim um único arquivo chamado "dadoAtletas"(dadoAtletas.csv) com um cabeçalho contendo 138 colunas sendo: média, variância, valor máximo, desvio padrão, tempo (ao qual o atleta permanece sem colocar o pé na esteira), média, variância e desvio padrão das TF e as amplitudes do centro de pressão nos eixos x, z e eixo vertical, tanto no tempo 2,5 m/s quanto nos tempos 3,5 m/s e 4,5 m/s. Por último inserimos na coluna 138 a nossa predição dos 28 atletas lesionados (indicado pelo número 1) e não lesionados (indicado pelo número 0).

# 4 RESULTADOS

## 4.1 Resultados da Classificação dos Algoritmos

Os testes consistiram em 28 observações, onde a base foi dividida em 70% para treinamento e 30% para teste.

### 4.1.1 Algoritmo de Classificação kNN

Para o algoritmo de classificação kNN temos nossa primeira matriz de confusão e sempre partimos do pressuposto que o "0" é classificado como atleta não lesionado e "1" é classificado como atleta lesionado.

A matriz de confusão da tabela 3 nos mostra que o modelo classificou 4 instâncias como atletas não lesionados e que realmente não obtinham nenhuma lesão. Classificou 3 atletas como lesionados que na verdade eram atletas não lesionados, também classificou 2 atletas como não lesionados que na verdade já obtiveram lesão. Não houve casos de Verdadeiros Positivo (VP), significando que nenhum caso foi registrado, ou seja, o modelo indicou que não há atletas lesionados, mas se houver representa uma ameaça potencial já que o foco principal para aumentar a precisão do modelo é aumentar a taxa de acertos. A precisão total do modelo é de 44,4% ((VN + VP) / 28) o que mostra que pode haver chances de melhorar o desempenho do modelo alterando o valor dos  $k$  vizinhos mais próximos, podemos dizer que aqui há um caso de *overfitting* podendo permitir dados ruidosos influenciando nos resultados.

Tabela 3 – Matriz de confusão kNN com  $k = 1$

		Valor Real
Previsto	0	1
	0	2
1	3	0

A matriz de confusão da tabela 4 classificou 3 instâncias como atletas não

lesionados e que realmente não obtinham nenhuma lesão. Também classificou 3 instâncias como não lesionados que na verdade obtiveram lesões. Além disso classificou 2 instâncias como lesionados que na verdade eram atletas não lesionados e por último classificou 1 atleta como lesionado que realmente obteve lesão. A acurácia total desse modelo foi 44,4% , somando temos 33,3% para VP e 11,1% para VN,o que nos leva aumentar ainda mais o nosso valor de k.

Tabela 4 – Matriz de confusão kNN com k = 2

		Valor Real	
Previsto	0	1	
	0	3	3
1	2		1

A matriz de confusão da tabela 5 classificou 4 instâncias como atletas não lesionados e que não obtinham nenhuma lesão. Também classificou 2 instâncias como não lesionados que na verdade obtiveram lesões. Classificou 2 instâncias como lesionados que na verdade eram atletas não lesionados e por último classificou 1 atleta como lesionado que realmente obteve lesão. Portanto a acurácia foi de 44,4% de VP e 11,1% de VN o que nos leva à uma acurácia total de 55,5% de taxa de acerto. Considerando o valor de k=3 ou seja, pegando os 3 vizinhos mais próximos a taxa de classificação para o algoritmo kNN pode não ser adequado para a nossa base de dados, mesmo aumentando os vizinhos a taxa de acerto não passa de 55,5% e elevar esse número já causaria *underfitting* ignorando padrões pequenos que podem afetar no desempenho do mesmo.

Tabela 5 – Matriz de confusão kNN com k = 3

		Valor Real	
Previsto	0	1	
	0	4	2
1	2		1

A tabela 6 basicamente resume os resultados da classificação com o algoritmo kNN, o k-vizinho mais próximo e sua acurácia.

Tabela 6 – Resumo dos Resultados de Classificação do kNN

K	Acurácia
1	44,4 %
2	44,4 %
3	55,5 %

#### 4.1.2 Algoritmo de Classificação SVM

Abaixo temos o primeiro teste para o SVM, sendo assim a matriz de confusão é amostrada partindo novamente do princípio que "0" é classificado como atleta não lesionado e "1" é classificado como atleta lesionado.

A matriz da tabela 7 usa o Kernel linear com custo 1 o classificador classificou 6 instâncias como atletas não lesionados que realmente não eram lesionados. Classificou 1 instâncias como não lesionados que na verdade eram atletas lesionados. Classificou 2 atleta como lesionado que na verdade é um atleta não lesionado e por fim não classificou nenhum como verdadeiro negativo. A acurácia por sua vez ficou em torno de 60,0% Outra observação em relação ao aumento do custo o kernel linear não consegue gerar uma reta de margem máxima satisfatória.

Tabela 7 – Matriz de confusão Kernel linear

		Valor Real
Previsto	0	1
	0	1
0	6	1
1	2	0

A matriz da tabela 8 usa o Kernel Polinomial com custo 3 o classificador classificou 4 instâncias como atletas não lesionados que realmente não eram lesionados. Classificou 2 instâncias como não lesionados que na verdade eram atletas lesionados. Classificou 1 atleta como lesionado que na verdade é um atleta não lesionado e por fim classificou 2 instâncias como lesionados que realmente eram lesionados. O kernel polinomial é um kernel não estacionário sendo seus núcleos polinomiais adequados para problemas em que todos os dados de treinamento são normalizados e modelados até a ordem do polinômio, mesmo normalizando os dados e aumentando o custo a melhor performance foi de uma acurácia de 66,6%.

Tabela 8 – Matriz de confusão Kernel Polinomial

		Valor Real	
Previsto		0	1
0	0	4	2
	1	1	2

A matriz da tabela 9 usa o Kernel Base Radial com custo 5 o classificador classificou 5 instâncias como atletas não lesionados que realmente não eram lesionados. Classificou 1 instância como não lesionados que na verdade é um atletas lesionados. Classificou 1 atleta como lesionado que na verdade é um atleta não lesionado e por fim classificou 2 atletas como lesionados que realmente eram atletas que obtiveram alguma lesão. Esse kernel apresentou 77,7% de acurácia sendo o melhor resultado do SVM.

Tabela 9 – Matriz de confusão Kernel Base Radial

		Valor Real	
Previsto		0	1
0	0	5	1
	1	1	2

A matriz da tabela 10 usa o Kernel Sigmoide com custo 3 o classificador classificou 4 instâncias como atletas não lesionados que realmente não eram lesionados. Classificou 2 instâncias como não lesionados que na verdade eram atletas lesionados. Classificou 2 atleta como lesionado que na verdade é um atleta não lesionado e por fim classificou 1 atletas como lesionados que realmente era um atleta que obteve alguma lesão. O Kernel da tangente hiperbólica foi o menos eficiente mesmo com o aumento do custo a acurácia chegou apenas em 55,5% isso deve-se ao fato de ser um Kernel frequentemente usado como uma função de ativação para neurônios artificiais.

Tabela 10 – Matriz de confusão Kernel Sigmoide

		Valor Real	
Previsto		0	1
0	0	4	2
	1	2	1

A tabela 11 mostra o resumo dos resultados juntamente com o valor do custo e acurácia apresentada.

Tabela 11 – Resumo dos Resultados de Classificação

Kernel	cost	Acurácia
Linear	1	60%
Polinomial	3	66,6%
Base Radial	5	77,7%
Sigmoide	3	55,5 %

Podemos dizer que nosso classificador SVM apresentou melhores resultados do que o classificador kNN. O Kernel de base radial se mostrou eficiente sendo uma ferramenta muito interessante e poderosa, fornece uma ponte entre linearidade e não-linearidade. A escolha de um Kernel depende do problema em questão, porque depende do que estamos tentando modelar. Um núcleo polinomial, por exemplo, permite modelar conjunções de características até a ordem do polinômio. As funções de base radial permitem selecionar círculos (ou hiperesferas) em contraste com o kernel Linear, que permite apenas selecionar linhas (ou hiperplanos).

## 4.2 Características dos Atletas

Após analisar os resultados podemos obter um perfil de atleta ao qual baseia-se nas *features* extraídas da base de dados coletadas. Sendo assim, podemos dizer que cerca de 78% apresentam certos comportamentos.

A Figura 15 mostra o gráfico dos 28 atletas e quanto tempo cada um deles permaneceu fora da esteira durante os 9 segundos, sendo assim, o gráfico a) mostra que a velocidade percorrida por eles foi de 2.5 m/s, podemos ver que a média dos

atletas lesionados foi de aproximadamente 2.1 segundos fora da esteira, o mesmo ocorreu para os atletas não lesionados 2.3 segundos, a diferença efetiva ficou em relação aos milissegundos. Há uma certa dispersão entre alguns atletas podendo ser questionável uma possível lesão.

O gráfico b) tem a velocidade aumentada à 3,5 m/s nesse caso houve menos dispersão entre os atletas lesionados, apesar de aumentar a velocidade da marcha, os atletas não lesionados ficaram 2.9 segundos fora da plataforma enquanto que os lesionados 2.7 segundos.

O gráfico c) nos mostra a velocidade 4.5 m/s foi o gráfico onde a média do atleta no ar mais alta, cerca de 3.0 segundos para os atletas lesionados enquanto que para os atletas não lesionados a média foi de 3.3 segundos. Apesar da pouca dispersão entre os dados em todos os 3 casos os atletas não lesionados obtiveram certa vantagem sob os atletas lesionados.

A ideia básica da Figura 16 é mostrar a média do Centro de pressão x em relação à z, comparando e analisando o gráfico podemos perceber o comportamento dos atletas lesionados e não lesionados nas 3 velocidades apresentam uma característica um tanto quanto curiosa, podemos notar claramente que os atletas lesionados tendem a exercer um centro de pressão mais similar, os dados estão mais juntos entre a região de 2000 à 2050, enquanto que os atletas sem lesão não tem um comportamento similar, tendem a ser mais dispersos, os sinais estão mais longes um dos outros. Todas essas características nos sinais de centro de pressão tem uma frequência associada à elas e sendo assim, podemos observar claramente seu comportamento a partir da Transformada de Fourier.

Nos gráficos de 17 à 20 temos o comportamento de atletas lesionados e não lesionados a partir da Transformada de Fourier. Uma vez que observamos senoides e frequências atribuídas à elas, mudamos claramente nossa visão periférica.

Observamos a média da Força nos eixos x, y e z dos 28 atletas, observamos que para os atletas lesionados as frequências entre 0 e 50 Hz estão com maiores picos de amplitude indicando um esforço maior do atleta no começo da corrida, tal comportamento é visto principalmente na Figura 19a) onde um atleta exerce claramente maior esforço no começo. Por outro lado os atletas não lesionados se dispõe

de menos esforços, podemos notar também que há um certo esforço desses atletas nos instantes finais da corrida entre as frequências 100 à 150 Hz. Tal comportamento tem uma explicação. A velocidade escalar da corrida equivale ao produto do comprimento da passada pela frequência da passada. Durante a corrida, uma variável cinemática como o comprimento da passada não depende apenas de fatores externos como altura do corredor, mas também dos músculos, tipo de calçado, nível de fadiga, histórico prévio de lesões, etc.

Portanto, tentar modificar a relação entre o comprimento e a frequência de passada tem o efeito sob a corrida. Conforme aumentamos nosso ritmo de corrida, tanto a frequência quanto o comprimento de passadas aumentam, porém em proporções diferentes. A frequência de passadas aumenta mais lentamente, e atinge seu pico em cerca de 70-80% da velocidade máxima de corrida.

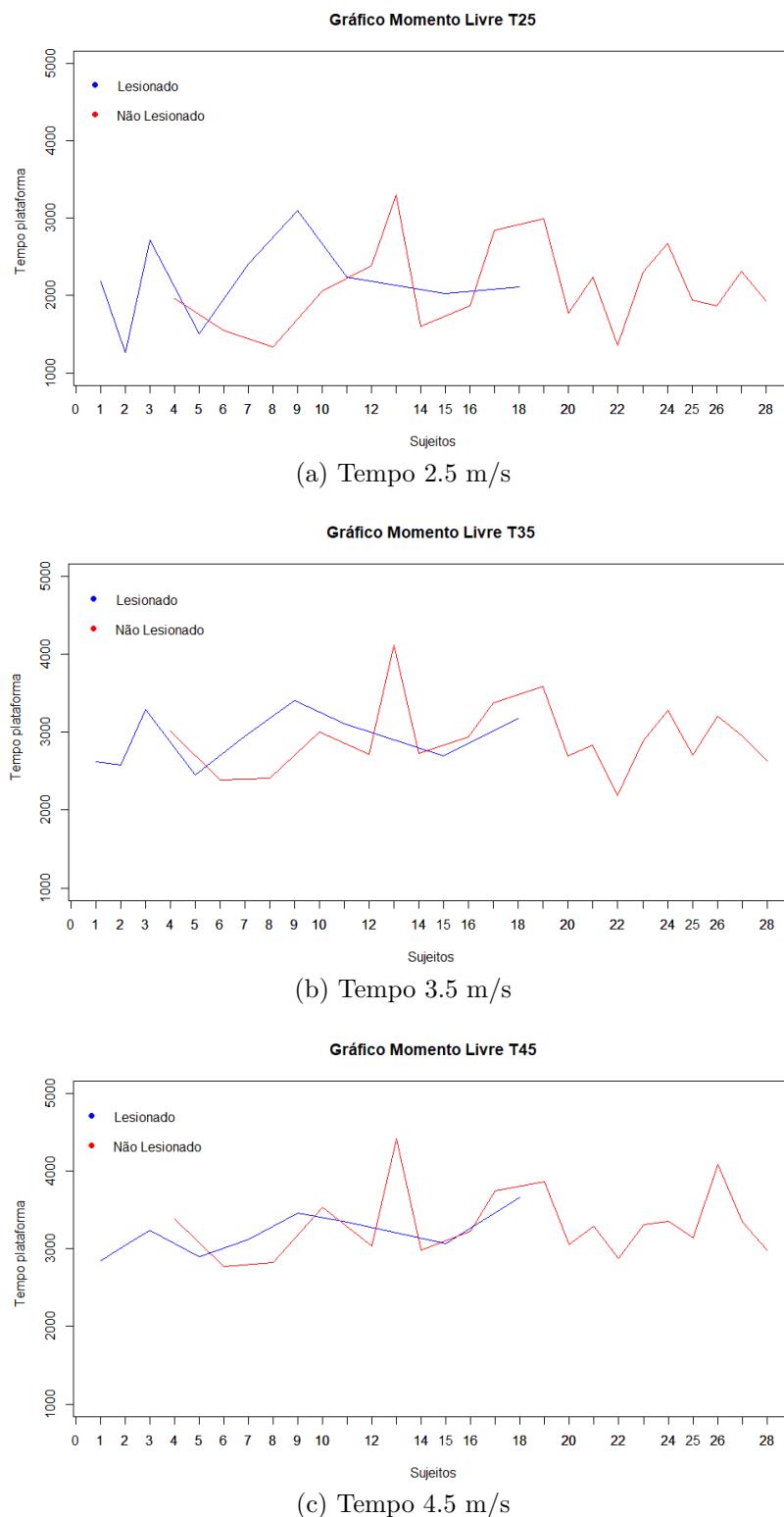


Figura 15 – Tempo de pisada dos atletas na plataforma sob momento livre no eixo vertical Ty nos tempos 2.5 m/s 3.5 m/s e 4.5 m/s respectivamente

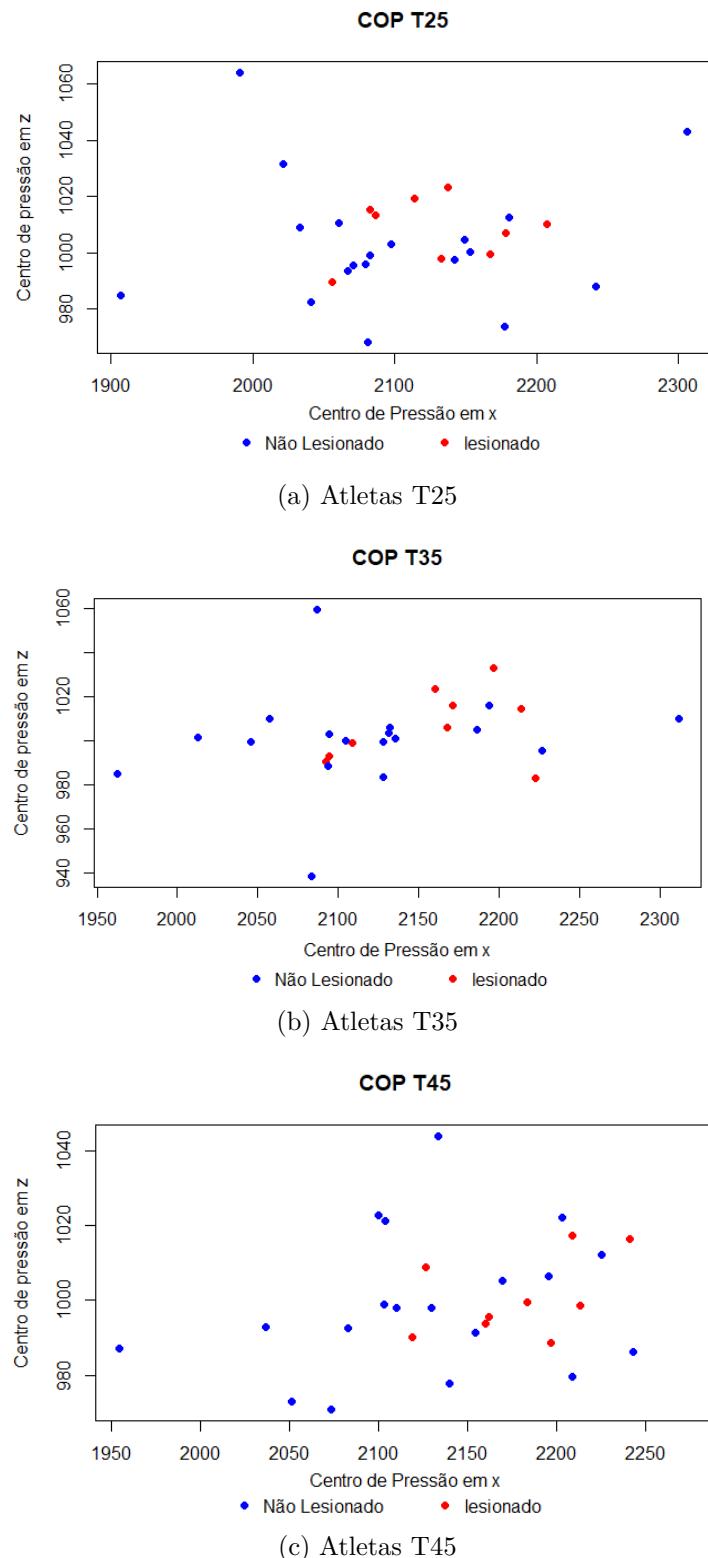


Figura 16 – Média do Centro de pressão nos eixos x e z nas velocidades 2.5 m/s 3.5 m/s e 4.5 m/s

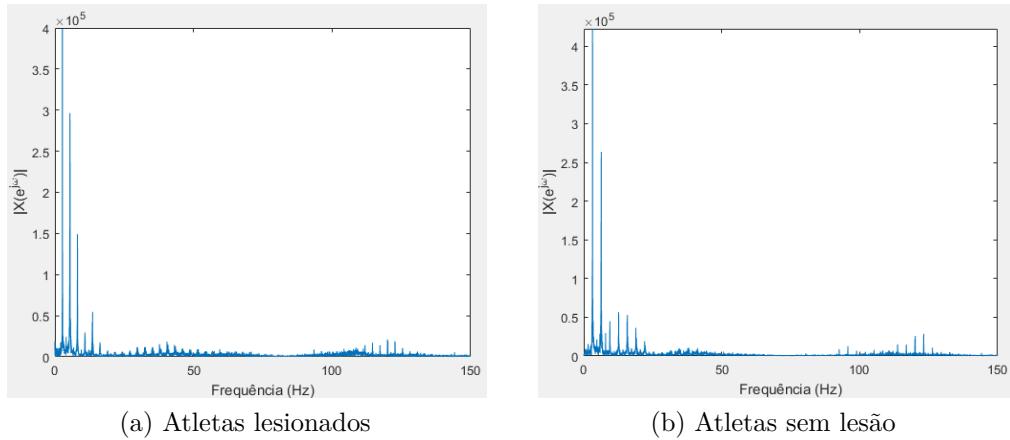


Figura 17 – Média da Transformada de Fourier das Forças no eixo x, 2.5 m/s 3.5 m/s e 4.5 m/s

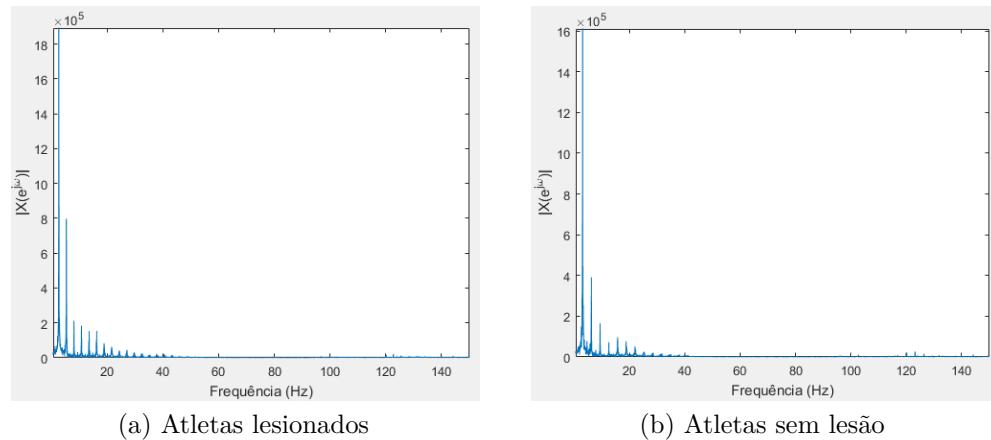


Figura 18 – Média da Transformada de Fourier das Forças no eixo y, 2.5 m/s 3.5 m/s e 4.5 m/s

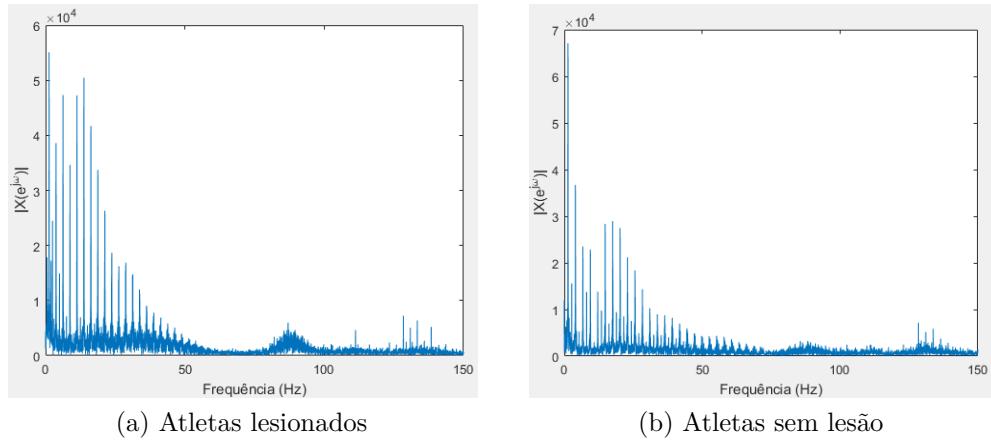


Figura 19 – Média da Transformada de Fourier das Forças no eixo z, 2.5 m/s 3.5 m/s e 4.5 m/s

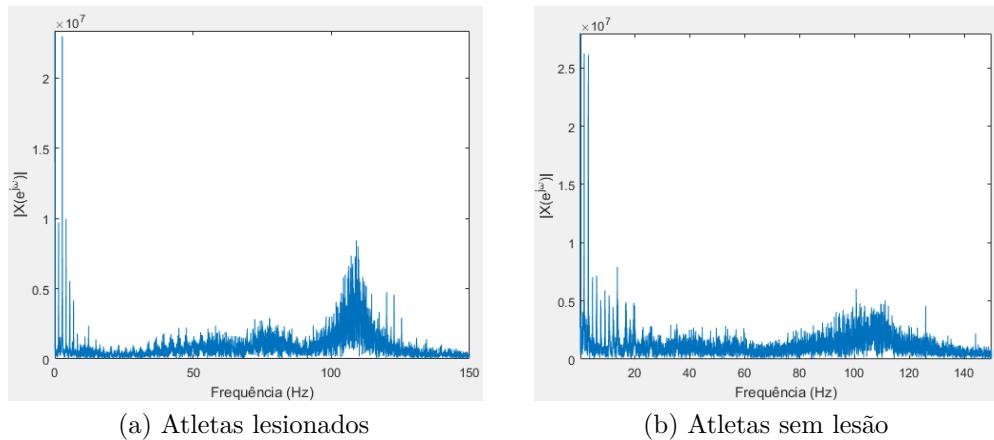


Figura 20 – Média da Transformada de Fourier do momento livre sob o eixo vertical Ty nos tempos 2.5 m/s 3.5 m/s e 4.5 m/s

# 5 Considerações Finais

## 5.1 Conclusão

O trabalho apresentou aplicação de algoritmos de aprendizado de máquina à um conjunto de dados da Biomecânica. Durante o levantamento dos dados foi observado a dificuldade em relação à escolha dos atributos, pode-se destacar o papel fundamental da Transformada de Fourier, a mesma se mostrou uma forte ferramenta de informação ajudando na escolha e visualização dos padrões das classes. Baseado nas métricas de avaliação e nos atributos coletados o algoritmo de classificação kNN desempenhou um papel pouco consistente. O processo de treinamento apesar de rápido requer tempo para ser investido na preparação de dados afim de obter um modelo robusto. Uma alternativa para melhorar sua acurácia seria inserir os dados marcadores e analisar novamente o kNN com maior precisão aumentando assim, a taxa de acerto máxima que obteve apenas 55,5%.

O algoritmo SVM encontrou um valor de classificação ótimo, diferente do que foi encontrado pelo algoritmo kNN. Neste caso, o valor de acurácia e as matrizes de confusão se comportam na maioria das vezes superior ao que foi encontrado no algoritmo anterior chegando à 77,7%. Quanto aos tipos de kernel o que obteve desempenho superior foi o kernel RBF isto ocorreu em todas as etapas de experimentos, tanto na atribuição de parâmetros quanto nos testes. Sendo assim confirmado como a melhor opção no reconhecimento de padrões. Na literatura pesquisada, este kernel já era apontado como uma opção promissora em diversas áreas de pesquisa. Destaca-se a relevante influência da escolha dos atributos (características) usados ao treinar as bases de dados.

## 5.2 Trabalhos Futuros

Apesar de satisfatórios acredita-se que os resultados podem ser melhorados significativamente uma vez que há outros arquivos a serem explorados em nosso con-

junto de dados. Devido ao tempo não pode-se trabalhar com os dados "marcadores" o mesmo pode fornecer informações relevantes e enriquecedoras ao nosso classificador podendo aumentar a taxa de acerto e obter um melhor desempenho.

Além de uma análise mais aprofundada e consistente dos dados é interessante explorar a utilização de novos algoritmos de classificação e até mesmo regressão, já que temos dados fornecidos pode-se procurar padrões ou tendências específicas para então usar os resultados dessa análise para definir os parâmetros ideais e criar um conjunto de heurística e cálculos para um modelo de mineração de dados.

## Referências

- AMORIM, D. G. *Redes art com categorias internas de geometria irregular* [Tese de doutoramento]. [S.l.]: Univ Santiago de Compostela, 2006. Citado na página 31.
- BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. Citado na página 20.
- BRAGA, A. d. P.; CARVALHO, A.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: Livros Técnicos e Científicos, 2000. Citado na página 28.
- BROOMHEAD, D. S.; LOWE, D. *Radial basis functions, multi-variable functional interpolation and adaptive networks*. [S.l.], 1988. Citado na página 27.
- BUHMAN, M. D. *Radial basis functions: theory and implementations*. [S.l.]: Cambridge university press, 2003. v. 12. Citado na página 29.
- CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, IEEE, v. 20, n. 3, p. 542–542, 2009. Citado na página 17.
- COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE transactions on information theory*, IEEE, v. 13, n. 1, p. 21–27, 1967. Citado na página 23.
- DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern classification*. [S.l.]: John Wiley & Sons, 2012. Citado na página 22.
- FUKUCHI, R. K.; FUKUCHI, C. A.; DUARTE, M. A public dataset of running biomechanics and the effects of running speed on lower extremity kinematics and kinetics. *PeerJ*, PeerJ Inc., v. 5, p. e3298, 2017. Citado 3 vezes nas páginas 7, 42 e 43.
- HAYKIN, S. S. *Neural networks: a comprehensive foundation*. [S.l.]: Tsinghua University Press, 2001. Citado 2 vezes nas páginas 30 e 32.
- HAYKIN, S. S. et al. *Neural networks and learning machines*. [S.l.]: Pearson Upper Saddle River, NJ, USA:, 2009. v. 3. Citado na página 31.
- HINO, A. A. F. et al. Prevalência de lesões em corredores de rua e fatores associados. *Rev Bras Med Esporte*, v. 15, n. 1, p. 36–9, 2009. Citado na página 15.
- JAIN, A. K.; MAO, J.; MOHIUDDIN, K. M. Artificial neural networks: A tutorial. *Computer*, IEEE, v. 29, n. 3, p. 31–44, 1996. Citado na página 21.

- JANG, J.-S. Self-learning fuzzy controllers based on temporal backpropagation. *IEEE Transactions on neural networks*, IEEE, v. 3, n. 5, p. 714–723, 1992. Citado na página [28](#).
- KANEVSKI, M.; POZDNUKHOV, A.; TIMONIN, V. Machine learning algorithms for geospatial data. applications and software tools. 2008. Citado na página [32](#).
- KURAMOCHI, M.; KARYPIS, G. Gene classification using expression profiles: a feasibility study. *International Journal on Artificial Intelligence Tools*, World Scientific, v. 14, n. 04, p. 641–660, 2005. Citado na página [24](#).
- LANTZ, B. *Machine learning with R*. [S.l.]: Packt Publishing Ltd, 2013. Citado na página [26](#).
- LINDSAY, T. R.; BOLLT, E. M.; MCGREGOR, S. J. Running gait classification: Support vector machine identification of distinct patterns between trained and untrained participants. *Medicine & Science in Sports & Exercise*, Medicine & Science in Sports & Exercise, v. 46, n. 5S, p. 811, 2014. Citado na página [15](#).
- LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007. Citado na página [32](#).
- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. *Sistemas Inteligentes-Fundamentos e Aplicações*, v. 1, n. 1, 2003. Citado 4 vezes nas páginas [14](#), [17](#), [18](#) e [19](#).
- OPPENHEIM, A. V. *Sinais e sistemas*. [S.l.]: Prentice-Hall, 2010. Citado na página [39](#).
- POWELL, M. Radial basis function methods for interpolation to functions of many variables. In: *HERCMA*. [S.l.: s.n.], 2001. p. 2–24. Citado 2 vezes nas páginas [22](#) e [29](#).
- RANGWALA, H.; KARYPIS, G. Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, Oxford University Press, v. 21, n. 23, p. 4239–4247, 2005. Citado na página [32](#).
- REILLY, J.; GHENT, J.; McDONALD, J. Investigating the dynamics of facial expression. *Advances in Visual Computing*, Springer, p. 334–343, 2006. Citado na página [32](#).
- ROBERTSON, S.; WOODS, C.; GASTIN, P. Predicting higher selection in elite junior australian rules football: The influence of physical performance and

anthropometric attributes. *Journal of Science and Medicine in Sport*, Elsevier, v. 18, n. 5, p. 601–606, 2015. Citado na página [14](#).

SCHÖLKOPF, B.; BURGES, C. J.; SMOLA, A. J. *Advances in kernel methods: support vector learning*. [S.l.]: MIT press, 1999. Citado na página [33](#).

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and computing*, Springer, v. 14, n. 3, p. 199–222, 2004. Citado 3 vezes nas páginas [33](#), [34](#) e [37](#).

TAY, B.; HYUN, J. K.; OH, S. A machine learning approach for specification of spinal cord injuries using fractional anisotropy values obtained from diffusion tensor images. *Computational and mathematical methods in medicine*, Hindawi Publishing Corporation, v. 2014, 2014. Citado na página [25](#).

THEODORIDIS, S.; KOUTROUMBAS, K. et al. *Pattern recognition*. [S.l.]: Academic press London, 1999. Citado 2 vezes nas páginas [20](#) e [25](#).

WETTSCHERECK, D.; AHA, D. W.; MOHRI, T. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. In: *Lazy learning*. [S.l.]: Springer, 1997. p. 273–314. Citado na página [23](#).

# APÊNDICE

A.1 Base de dados brutos

A.2 Código da Transformada de Fourier

A.3 Código do Algoritmo kNN

A.4 Código do Algoritmo SVM

A.5 Código preparacaoDados

A.6 Base de dados dadoAtletas.csv

A.7 Base de dados RBDSinformacoes.csv