NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATHEMATICS

A SPHERICAL SUPPORT VECTOR MACHINE FOR CLASSICAL AND INTERVAL-VALUED DATA

RUI JORGE FERNANDES MALHA

Master in Teaching of Mathematics

DOCTORATE IN MATHEMATICS

NOVA University Lisbon
September, 2024

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATHEMATICS

A SPHERICAL SUPPORT VECTOR MACHINE FOR CLASSICAL AND INTERVAL-VALUED DATA

RUI JORGE FERNANDES MALHA

Master in Teaching of Mathematics

Adviser: Paula Alexandra da Costa Amaral
Associate Professor at NOVA School of Science and Technology of NOVA University Lisbon

Examination Committee

Chair: João Jorge Ribeiro Soares Gonçalves de Araújo
Full Professor at NOVA School of Science and Technology of NOVA University Lisbon

Rapporteurs: Frédéric Messine
Full Professor at University of Toulouse, France

Eligius M. T. Hendrix
Full Professor at Málaga University, Spain

Adviser: Paula Alexandra da Costa Amaral
Associate Professor at NOVA School of Science and Technology of NOVA University Lisbon

Members: João Jorge Ribeiro Soares Gonçalves de Araújo
Full Professor at NOVA School of Science and Technology of NOVA University Lisbon

Maria do Carmo Proença Caseiro Brás
Associate Professor at NOVA School of Science and Technology of NOVA University Lisbon

Miguel dos Santos Fonseca
Assistant Professor at NOVA School of Science and Technology of NOVA University Lisbon

DOCTORATE IN MATHEMATICS

NOVA University Lisbon
September, 2024

A Spherical Support Vector Machine for Classical and Interval-valued Data

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.29) [59].

To my beloved granddaughters, Beatriz and Sofia, and my future grandsons and granddaughters I hope to meet one day.

This work is for you. May it remind you that knowledge is a treasure without bounds and that curiosity can unlock doors to worlds you have yet to imagine.

As you walk through life, I hope you find the courage to ask bold questions, the resilience to seek answers, and the humility to learn from every step of the way.

Know that your potential is limitless, and the pursuit of knowledge will always lead you to greater heights.

You are my legacy and my greatest hope for the future.

May this achievement serve as a beacon for your own dreams, reminding you that dedication and passion can make the impossible possible.

With all my love,

Rui Jorge Fernandes Malha

Acknowledgements

I could not have accomplished this immense task without the unwavering support of two extraordinary women.

To Professor Paula, my deepest gratitude. First, for believing in me and my ability to carry out this project, and then, for guiding me through the paths of research and helping me to build each step of this work with your constant support. I am profoundly thankful for your generosity, your availability, and your well-timed words of encouragement, especially when my motivation began to waver. Your guidance was essential in making this journey possible.

To my wife Clara, my life partner and greatest source of emotional support throughout this journey. Thank you for supporting my decision to embark on this PhD and for always being by my side, through moments of joy as well as discouragement. You were my confidant, sharing in my successes and helping me face the moments when everything seemed insurmountable. Thank you for giving me the strength to continue, for never letting me give up, and for being the foundation upon which I built every step of this achievement.

”””*“”You must do the things you think you cannot do”.”*

— Eleanor Roosevelt

Abstract

The application of automatic classification methods has significantly advanced real-time fraud detection and anomaly monitoring, streamlined and modernized administrative processes, and supported decision-making across various domains. Despite these advantages, it is crucial to acknowledge the inherent risks of using these methods as "black boxes" that operate without sufficient scrutiny or interpretability. Support Vector Machines (SVM) offer a more transparent alternative compared to techniques like deep neural networks and random forests. This study aims to harness the potential of SVMs while avoiding the complexity introduced by kernel functions, thus preserving the classification model within the original feature space and minimizing excessive parameter tuning.

In the classical SVM approach, the separation between classes is determined by a hyperplane. Recently, generalizations of this concept have emerged, incorporating non-linear separations through kernel functions or alternative schemes, such as non-linear functions and polytopes. In this work, we propose a generalization of the SVM method, where the separator is a curve with a spherical shape. However, key SVM principles, such as margin maximization and the use of soft margins, are retained. We model this spherical classification method as a quadratic optimization problem and introduce a linear relaxation. These models were applied to classical data sets available from a benchmark repository. Furthermore, we extend this approach to interval-valued data, within the framework of Symbolic Data Analysis (SDA). In Symbolic Data, features are represented by sets, intervals, or histograms, rather than conventional data arrays, an approach that has gained increasing relevance, particularly in the era of Big Data.

As with the classical case, a relaxation is also presented for interval-value data. The performance of these formulations was tested against traditional classification methods, yielding highly positive results. This demonstrates the potential of the proposed approach for enhancing classification accuracy and interpretability, particularly in complex data scenarios.

Keywords: SVM, Automatic Classification, Non-linear SVM, Histogram-valued Data, Symbolic Data, Spherical separation

Resumo

A aplicação de métodos de classificação automática tem alcançado avanços significativos na deteção de fraudes em tempo real e na monitorização de anomalias, simplificando e modernizando os processos administrativos, e apoiando a tomada de decisões em diversos domínios. Apesar destas vantagens, é crucial reconhecer os riscos inerentes à utilização destes métodos como "caixas negras", que operam sem a devida análise ou interpretabilidade. *Support Vector Machines* (SVM) oferece uma alternativa mais transparente em comparação com técnicas como *Deep Neural Networks* e *Random Forest*. Este estudo tem como objetivo aproveitar o potencial das SVM, evitando, contudo, a complexidade introduzida pelas funções *Kernel*, preservando assim o modelo de classificação no espaço original das características e minimizando o ajuste excessivo de parâmetros.
Na abordagem clássica das SVM, a separação entre classes é determinada por um hiperplano. Recentemente, surgiram generalizações deste conceito, incorporando separações não lineares através de funções de *Kernel* ou esquemas alternativos, como funções não lineares e politopos. Neste trabalho, propomos uma generalização do método SVM, onde o separador é uma curva com uma forma esférica. No entanto, são mantidos os princípios-chave das SVM, como a maximização da margem e a utilização de margens suaves. Modelamos este método de classificação esférica como um problema de otimização quadrática e introduzimos uma relaxação linear. Estes modelos foram aplicados a dados clássicos. Além disso, estendemos esta abordagem a intervalos, no âmbito da Análise de Dados Simbólicos (SDA). Nos Dados Simbólicos, as características são representadas por conjuntos, intervalos ou histogramas, em vez de matrizes de dados convencionais, uma abordagem que tem ganho relevância crescente, particularmente na era de Big Data.
Tal como no caso clássico, uma relaxação também é apresentada para intervalos. O desempenho destas formulações foi testado em comparação com métodos tradicionais de classificação, com resultados altamente positivos. Isto demonstra o potencial da abordagem proposta para melhorar a precisão e a interpretabilidade da classificação, especialmente em cenários de dados complexos.

Palavras-chave: SVM, Classificação automática, SVM não linear, Histogramas, Dados simbólicos, Separação esférica,

Contents

List of Figures

List of Tables

Acronyms

***k*NN**	*K*-Nearest Neighbors *(pp. 5, 11, 12)*
AI	Artificial Intelligence *(pp. 2–4)*
ANN	Artificial Neural Networks *(p. 3)*
CNN	Convolutional Neural Networks *(p. 15)*
DL	Deep Learning *(p. 3)*
DNN	Deep Neural Networks *(pp. 3, 15)*
DT	Decision Trees *(pp. 13, 14)*
FNN	Feedforward Neural Network *(p. 15)*
GPU	Graphics Processing Unit *(p. 16)*
IOT	Internet of Things *(p. 1)*
LS-TSVH	Least Squares Twin Support Vector Hypersphere *(p. 27)*
LSSSVM	Soft Margin Spherical SVM Linear Relaxation *(pp. 36, 38)*
LSSSVMMI	Soft Margin Spherical SVM Linear Relaxation for Multiple Interval *(p. 53)*
ML	Machine Learning *(pp. 2–5)*
NB	Naive Bayes *(p. 12)*
NLP	Natural Language Processing *(p. 16)*
NN	Neural Networks *(pp. 14, 15)*
PCA	Principal Component Analysis *(p. 5)*

RF Random Forest *(p. 13)*
RL Reinforcement Learning *(p. 6)*
RNN Recurrent Neural Networks *(p. 15)*

SDA Symbolic Data Analysis *(pp. v, 20)*
SSSVM Soft Spherical Support Vector Machines *(pp. 36, 38)*
SSSVMMI Soft Margin Spherical SVM for Multiple Interval *(pp. 49, 53)*
SSVM Spherical Support Vector Machines *(pp. 2, 32, 47)*
SSVMI Spherical Support Vector Machines for Interval-valued Data *(p. 2)*
SSVMMI Spherical Support Vector Machine for Multiple Interval *(p. 48)*
SVDD Support Vector Data Description *(pp. 26, 27)*
SVM Support Vector Machine *(pp. 1, 2, 5, 17, 23, 26, 40, 43, 56, 58)*

TSVH Twin Support Vector Hypersphere *(p. 27)*

1

Introduction

"You must do the things you think you cannot do"
Eleanor Roosevelt

In the contemporary era of Big Data, 4th industrial revolution, Internet of Things (IOT), smart cities, smart cars, smart appliances, the sheer volume, velocity, and variety of data being generated are unprecedented. This explosion of data is produced from numerous sources such as social media, sensors, transaction records, and more. Today we have sensors connected to houses, bodies, cars, machines, trees, animals, etc. The act of monitoring reached a hyperbolic state posing significant challenges in terms of storage, processing, and, crucially, interpretation.
Interpreting and classifying these types of data requires good and effective classification methods which are essential to derive meaningful information from this vast sea of data. They enable organizations to transform raw data into valuable insights, driving innovation and efficiency across various domains. As data continues to grow in complexity and scale, the development and application of robust classification techniques will remain a cornerstone of data science and analytics.
Big Data involves enormous data sets that traditional data processing tools cannot manage efficiently. Classification methods enable the organization and simplification of these large data sets into manageable categories, facilitating more effective analysis and decision-making processes. Also identifying relevant patterns and trends is critical. Classification methods help in sorting through heterogeneous data, allowing analysts to pinpoint significant correlations and patterns that might otherwise remain hidden. In business, healthcare, finance, and various other fields, rapid and accurate decision-making is crucial. Classification algorithms can quickly process and analyze large data sets, providing actionable insights that support informed decision-making and strategic planning. In predictive analytics to forecast future trends based on historical data is crucial. Automatic classification enable the construction of predictive models that can analyze past behavior and predict future outcomes with a high degree of accuracy at demand.
Motivated by this framework, and by the lack of research in this area, we investigate in this work, a machine learning classification model, Support Vector Machine (SVM) based,

using hyperspheres, the Spherical Support Vector Machines (SSVM). Our research focused then in modifying and adjusting this classification model to interval-valued data in the scope of symbolic data analysis, the Spherical Support Vector Machines for Interval-valued Data (SSVMI). Since these formulations are nonlinear programming problems (quadratic problems), we also developed a linear relaxation for both formulations.
In this chapter we will introduce the Machine Learning (ML) paradigm and its impact on society, the diferent types of ML, the advantages and disadvantages of blackbox models versus interpretative models and some basic concepts in ML classification and interpretability. We briefly introduce the most common classification models in ML including the SVM which will be described in more detail in chapter 2. Also Symbolic Data is introduced.
To finalize this chapter, we present a short description of the structure of the thesis.

1.1 The Machine Learning Paradigm

1.1.1 What is Machine Learning

Artificial Intelligence (AI), is the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings [73]. AI systems are therefore machine-based systems with varying levels of autonomy that can, for a given set of human-defined objectives, make predictions, recommendations or decisions. AI techniques are increasingly using massive amounts of alternative data sources and data analytics referred to as 'big data' [82]. Such data feed ML models which are computer programs that use such data to learn and improve predictability and performance automatically through experience and data, without being explicitly programmed to do so by humans [70].
ML is therefore an approach to achieve AI [5].

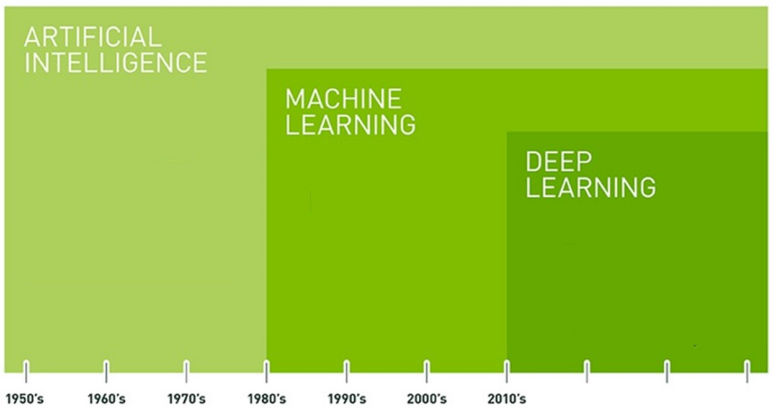

Figure 1.1: Machine Learning

ML teaches computers to do what comes naturally to humans and animals: learn from

experience. ML algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases.
The development of ML spans many decades, influenced by advancements in mathematics, statistics, computer science, and AI. Below is a chronological overview of key milestones in the history of machine learning:

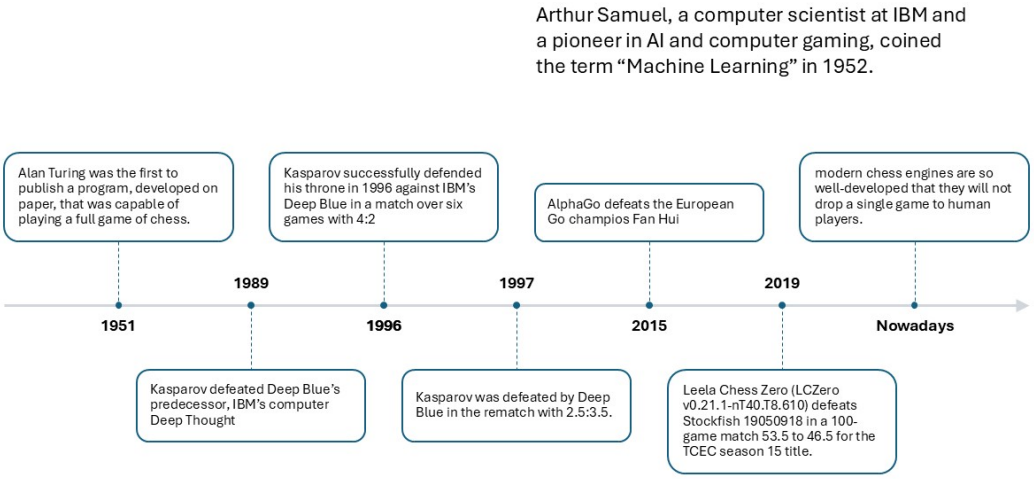

Figure 1.2: Machine Learning milestones

During the last decades, the field of ML has brought forth a variety of remarkable advancements in sophisticated learning algorithms and efficient pre-processing techniques. One of these advancements was the evolution of Artificial Neural Networks (ANN) towards increasingly Deep Neural Networks (DNN) architectures with improved learning capabilities summarized as Deep Learning (DL) [48]. DL is therefore a specialized area within ML that uses neural networks with multiple layers for more complex learning tasks.

1.1.2 Machine Learning Social impact

AI and ML in particular are transforming numerous sectors of society, affecting how industries operate, how services are delivered, and how decisions are made. Their societal impact is growing rapidly in both positive and negative but challeging ways [86].
From healthcare and finance to education, transportation, and beyond, its applications are vast. In healthcare, ML assists in medical diagnosis, enables individualized treatment plans, accelerates the drug discovery process and optimizes the hospital resource management. In finance ML algorithms analyze transaction data for fraud detection, assess

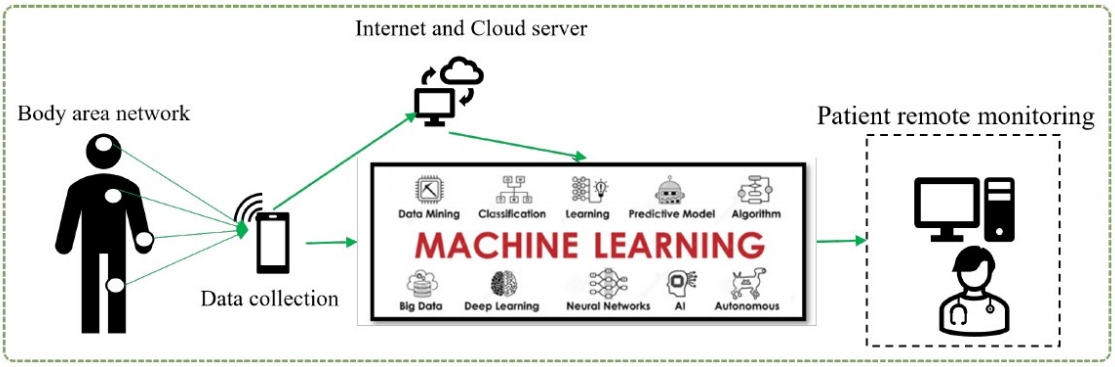

Figure 1.3: Concept of machine learning in healthcare area [6]

credit risk for loans and mortgages, optimize investment strategies by analyzing historical data and market trends, and ML is also used in banking customer service like Chatbots and virtual assistants.
However, if ML has many positive social impacts, it also comes with significant negative aspects and challenges. These drawbacks often arise due to issues related to ethics, privacy, bias, and the societal consequences of automation.
As mentioned above, ML models require large amounts of data to function effectively, which can lead to the collection of sensitive personal information. This raises concerns about privacy and surveillance, particularly when personal data is used without consent or awareness. While ML boosts efficiency in industries like manufacturing, transportation, and customer service, it also leads to job displacement for workers in roles that can be automated. Replacing human workers in certain industries, entire sectors could face job losses, particularly in routine or manual labor jobs. While new jobs may be created in AI and tech, these often require specialized skills that displaced workers may not possess. Also many ML models function as a black box, meaning their decision-making processes are not easily interpretable [57]. This lack of transparency causes decisions made by ML systems difficult to understand and interpret. For example, if a loan application is denied by a machine learning model, the applicant might not know why or how the decision was made, leading to frustration and loss of trust.

1.1.3 Machine Learning Fundamentals

Machine learning starts with data — numbers, photos, or text, like bank transactions, pictures of people or even bakery items, repair records, time series data from sensors, or sales reports. The data is gathered and prepared to be used as training data, or the information the machine learning model will be trained on. The more data we have for training the model, the better will be the model ability to have a good performance.
Some data is held out from the training data to be used as evaluation data, which tests how accurate the machine learning model is when it is shown new data. The result is a

model that can be used in the future with different sets of data.
ML techniques are generally categorized into three main types [98]: supervised learning, unsupervised learning, and reinforcement learning (Figure1.4). Each of these techniques addresses different types of tasks and is used based on the nature of the data and the problem to be solved.

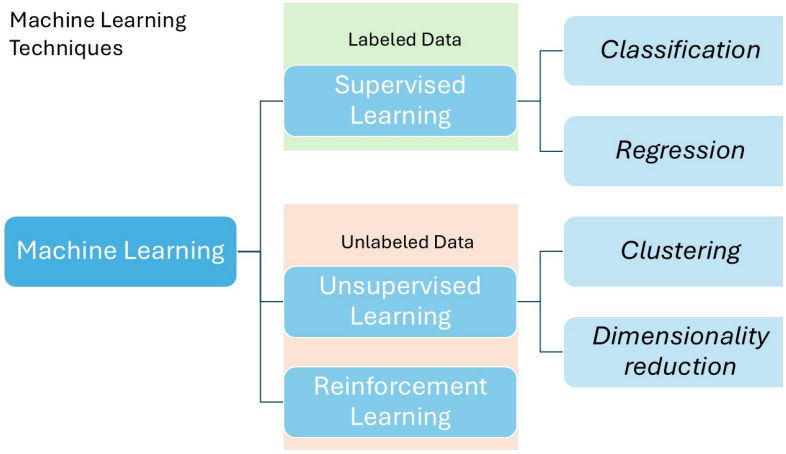

Figure 1.4: Supervised, Unsupervised and Reinforcement Machine Learning

In Supervised Learning [24], the most common type used today, the model is trained on a labeled dataset. Each instance is described by a vector of atributes together with a corresponding known category label, allowing the model to learn a mapping between input features and output labels and then generalize this learning to unseen data.
Classification and regression problems are examples of supervised learning. In Classification, the model predicts a discrete label (e.g., recognizing if an image is of a cat or a dog). In Regression the model predicts a continuous value (e.g., predicting house prices). Some algorithms used in supervised learning are: Linear Regression; Logistic Regression; Support Vector Machines (SVM); Decision Trees; Random Forests; K-Nearest Neighbors (kNN); Neural Networks.
Example: Predicting whether an email is spam or not based on its content. The model is trained on emails that have been labeled as "spam" or "not spam" and then used to predict the label of new emails.
In Unsupervised Learning [39], the model is trained on an unlabeled dataset. The goal is to discover hidden patterns, structures, or relationships in the data, without any explicit output labels. It is intended that the model learns the underlying structure of the data, group similar data points, or reduce the dimensionality of the dataset.
Clustering, dimension reduction and anomaly detection are examples of unsupervised machine learning. Clustering consists in grouping data points into clusters based on their similarities (e.g., customer segmentation). Dimension Reduction consists in reducing the number of features while maintaining important information (e.g., Principal Component Analysis (PCA) [2]). Some common algorithms are: K-Means Clustering [18]; Hierarchical Clustering [68]; DBSCAN (Density-Based Spatial Clustering) [58]; PCA; Autoencoders

[65]; t-SNE (t-distributed Stochastic Neighbor Embedding) [90].
Example: Market basket analysis, where the goal is to group similar customer purchase patterns without predefined categories.
Finally, Reinforcement Learning (RL) [49] involves an agent interacting with an environment and learning to make decisions by taking actions and receiving feedback in the form of rewards or penalties.

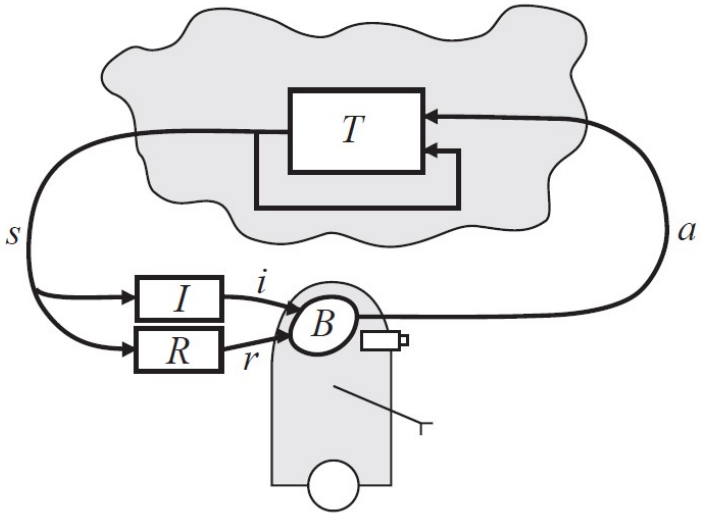

Figure 1.5: The standard reinforcement learning model [49]

The goal is to learn an optimal strategy or policy that maximizes cumulative rewards over time by exploring different actions and learning from the results.
In the standard reinforcement learning model, an agent is connected to its environment via perception and action, as depicted in Figure 1.5. In each step of interaction, the agent receives as input, i, some indication of the current state, s, of the environment; the agent then chooses an action, a, to generate an output. The action changes the state of the environment, and the value of this state transition is communicated to the agent through a scalar reward signal, r. The agent's behavior, B, should choose actions that tend to increase the long-run sum of values of the reward signal. It can learn to do this over time by systematic trial and error, guided by a wide variety of algorithms. The Figure also includes an input function I, which determines how the agent views the environment state. The agent's job is to find a policy π, mapping states to actions, that maximizes some long-run measure of reward.
Reinforcement learning differs from the more widely studied problem of supervised learning in several ways. The most important difference is that there is no presentation of input/ output pairs. Instead, after choosing an action, the agent is told the immediate reward and the subsequent state, but is not told which action would have been in its best long-term interests. It is necessary for the agent to gather useful experience about the possible system states, actions, transitions and rewards actively to act optimally.

Common algorithms are: Q-Learning [94]; Deep Q-Networks (DQN) [44]; Policy Gradient Methods [78]; Proximal Policy Optimization (PPO) [80]; SARSA (State-Action-Reward-State-Action) [66].
An example is training a self-driving car to navigate through a city. The car (agent) takes actions like steering, braking, or accelerating, and the reward might be based on factors like staying on the road or avoiding obstacles.

1.2 Classification

Classification in machine learning is a type of supervised learning where the goal is to allocate some object into one of the given categories called classes. For a specific classification problem, a classifier, which is an algorithm, is developed so that new objects are classified correctly with reasonably good accuracy. The model is trained using labeled data where each instance has a known category.
The input to the classifier is x the vector of attributes that describe the object and the output is y the class label (Figure1.6).

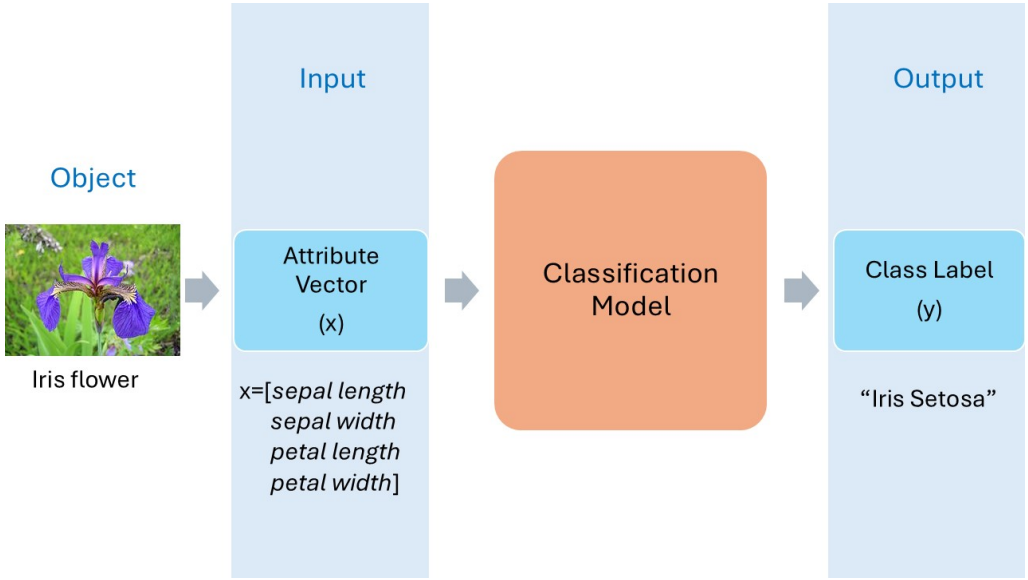

Figure 1.6: The classification problem

For the classification task, data is needed as a collection of observations. Each observation, also known as example or instance, is denoted by (x, y), where x is the vector of attributes and y is the class label or category of the observation (Table1.1). Common examples of classification tasks include email spam detection, image recognition, and medical diagnosis. To find a Classification Model is the process of learning a function f that matches each attribute vector x with one of previously defined class labels y. The function f depends on parameters θ that must be estimated based on a set of data called training data.

Table 1.1: Some examples from Iris Dataset (from UCI Machine Learning Repository [87])

Sepal length	Sepal width	Petal length	Petal width	Label	Designation
(cm)	(cm)	(cm)	(cm)		
x_1	x_2	x_3	x_4	y	
5.1	3.5	1.4	0.2	1	Iris Setosa
4.9	3	1.4	0.2	1	Iris Setosa
4.5	2.3	1.3	0.3	1	Iris Setosa
6.4	3.2	4.5	1.5	2	Iris Versicolor
5.5	2.3	4	1.3	2	Iris Versicolor
6.3	3.3	4.7	1.6	2	Iris Versicolor
7.6	3	6.6	2.1	3	Iris Virginica
4.9	2.5	4.5	1.7	3	Iris Virginica
7.3	2.9	6.3	1.8	3	Iris Virginica

For example, considering binary classification one such function could be:

$$f(x, \theta) \leq 0, \ x \in Class1 \tag{1.1}$$

$$f(x, \theta) > 0, \ x \in Class2. \tag{1.2}$$

1.2.1 Classification performance

In training a classifier, usually we try to maximize classification performance for the training data. However, if the classifier is too fit for the training data, the classification ability for unknown data, i.e., the **generalization** ability is degraded. This phenomenon is called **overfitting**. Namely, there is a trade-off between the generalization ability and fitting to the training data. Various methods have been proposed to prevent overfitting [32]. Figure1.7 illustrates the underfitting and overfitting issue in the context of classification.

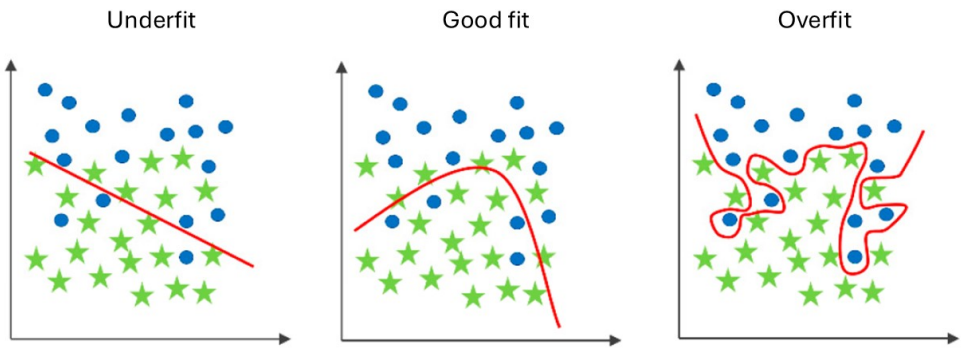

Figure 1.7: Illustration of the underfitting/overfitting issue [54]

Cross-validation is a widely used technique to estimate the generalization error of a classifier and therefore ensure that the model is not underfitting or overfitting. In k-fold cross validation, training data are randomly divided into, approximately, k equally sized subsets, and a classifier is trained using $k-1$ subsets and tested using the remaining subset, the testing set. Figure1.8 shows an example for a 4-fold cross validation [63].
Training is repeated k times, and the total recognition rate for all the k subsets that are

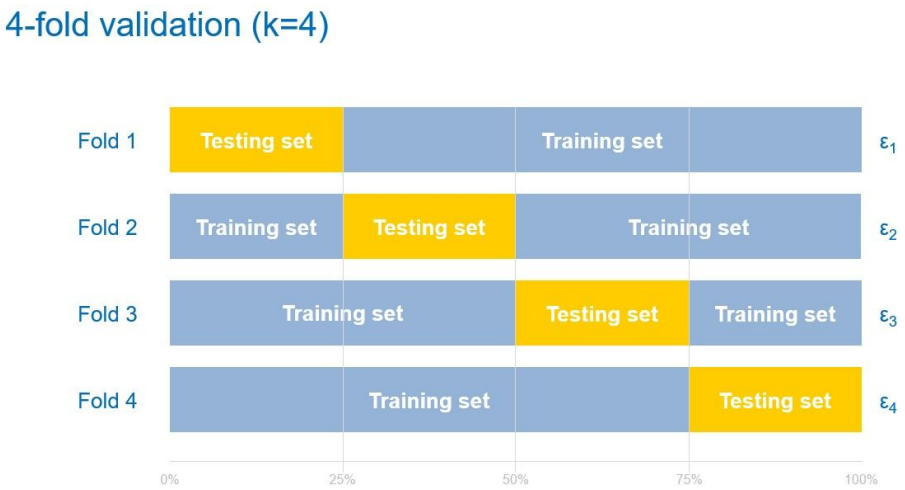

Figure 1.8: 4-fold cross validation

not included in the training data is calculated. A leave-one-out method (LOO) is a special case of cross validation ($l = M - 1$) and k-fold cross validation.
To evaluate the performance of a classification model, a Confusion Matrix is a fundamental tool. It provides a detailed breakdown of how well the model predictions align with the actual outcomes by displaying the counts of correct and incorrect predictions across different classes. This matrix helps in understanding not just the overall accuracy of the model but also the types of errors it is making, which can be crucial for improving model performance. Figure1.9 outlines the structure of a Confusion Matrix.
For a binary classification problem (where there are two classes, such as "Positive" and "Negative"), the confusion matrix is typically a 2x2 table structured as follows:

- True Positive (TP): the model correctly predicts the positive class.
- True Negative (TN): the model correctly predicts the negative class.
- False Positive (FP): the model incorrectly predicts the positive class (Type I error).
- False Negative (FN): the model incorrectly predicts the negative class (Type II error).

For multi-class classification (where there are more than two classes), the confusion matrix expands to an n x n table, where n is the number of classes. Each column represents the actual class, and each row represents the predicted class, allowing for a comprehensive view of how instances of each class are being classified.

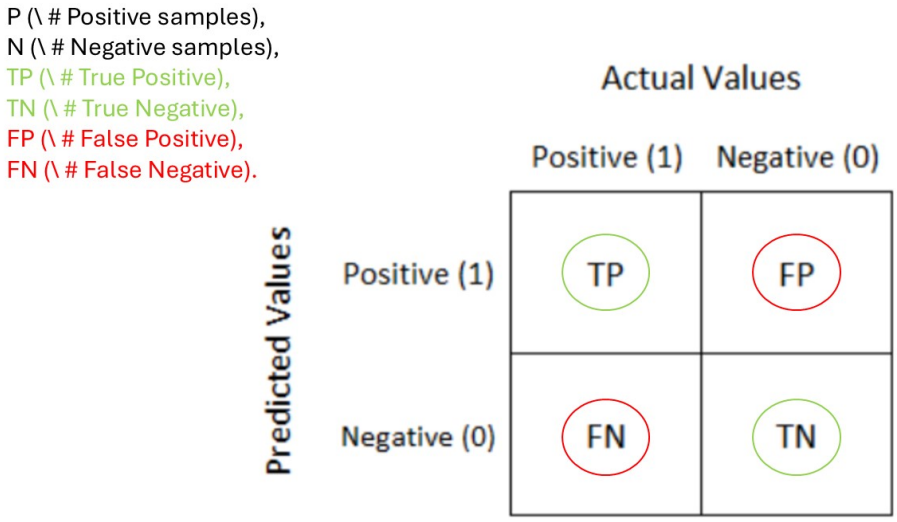

Figure 1.9: The Confusion Matrix

Table1.2 provides the most common metrics for model evaluating and comparison. In this work the accuracy metric is mainly used.

Table 1.2: Metrics

Metric	**Formula**	**Evaluation Focus**
Accuracy	$ACC = \frac{TP+TN}{TP+TN+FP+FN}$	Overall effectiveness of a classifier
Error Rate	$ERR = \frac{FP+FN}{TP+TN+FP+FN}$	Classification error
Precision	$PRC = \frac{TP}{TP+FP}$	Class agreement of the data labels with the positive labels given by the classifier
Sensitivity	$SNS = \frac{TP}{TP+FN}$	Effectiveness of a classifier to identify positive labels
Specificity	$SPC = \frac{TN}{TN+FP}$	How effectively a classifier identifies negative labels
ROC	$ROC = \frac{\sqrt{SNS^2+SPC^2}}{\sqrt{2}}$	Combined metric based on the Receiver Operating Characteristic (ROC) space
F_1score	$F_1 = 2\frac{PRC.SNS}{PRC+SNS}$	Combination of precision (PRC) and sensitivity (SNS) in a single metric
Geometric Mean	$GM = \sqrt{SNS.SPC}$	Combination of sensitivity (SNS) and specificity (SC) in a single metric

1.2.2 Classification models

For a specific classification problem, we have to choose what model to use and as a result of the research in this area, there are several classification methods available and new ones are proposed every day. We can mention among others: Logistic Regression, Decision

trees, *K*-Nearest Neighbors, Naive Bayes, Neural Networks, Support Vector Machines and Ensemble Learning (Random Forest [72], Gradient Boosting [67]) .
In the sequel, a short introduction to several of these models will be provided.

1.2.2.1 *K*-Nearest Neighbors (*k*NN)

The *k*NN algorithm was first introduced by Evelyn Fix and Joseph Hodges in 1951. They presented it in their technical report titled "Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties", which was part of the research at the US Air Force School of Aviation Medicine [38].
*k*NN is a simple yet effective supervised machine learning algorithm used primarily for classification and regression tasks. It is based on the principle of similarity, where it classifies a new data point based on how its neighbors are classified.
Each data point in *k*NN is represented in an n-dimensional space, where n is the number of features. When a new data point (query) is introduced, *k*NN calculates its distance from other points in the dataset. Common distance metrics include the Euclidean Distance, the Manhattan Distance or the Minkowski Distance.
In classification, *k*NN identifies the k nearest data points (neighbors) to the query point, using the chosen distance metric and assigns the query point to the class that is most frequent among its nearest neighbors.
The basic idea is shown in Figure 1.10 [25], which depicts a 3-Nearest Neighbor Classifier, $k = 3$, on a two-class problem in a two-dimensional feature space. In this example, the decision for $q1$ is straightforward - all three of its nearest neighbors are of class O, so it is classified as an O. The situation for $q2$ is a bit more complicated, as it has two neighbors of class X and one of class O. This can be resolved by simple majority voting or by distance weighted voting. So, *k*NN classification has two stages: the first is the determination of the nearest neighbors, and the second is the determination of the class using those neighbours.

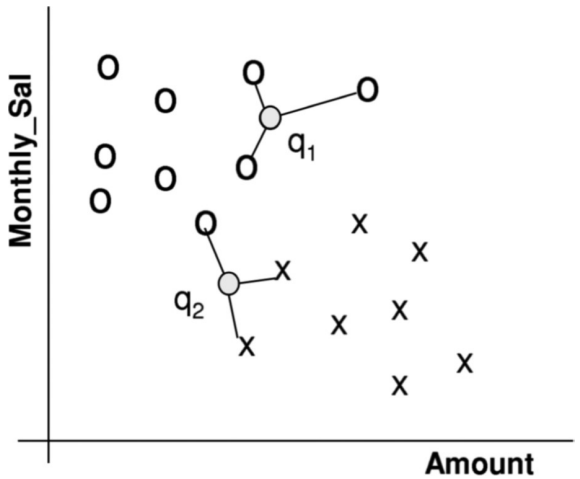

Figure 1.10: 3-Nearest Neighbor Classification in a 2D feature space [25]

The number k determines how many neighbors influence the decision. Choosing an appropriate value for k is crucial. A small value of k makes the model sensitive to noise, while a large value may smooth out predictions too much. Cross-validation is often used to determine the optimal value of k, balancing underfitting and overfitting.
The choice of the distance metric depends on the problem and data. Euclidean distance is commonly used, but may not always be suitable. Optionally, you can weigh neighbors based on their distance to the query point (closer neighbors having more influence).
KNN is easy to understand and implement. It performs well for smaller datasets and when a clear definition of "closeness" (via distance) can be made. No model is built during the training phase, instead, all the computation happens during prediction. It is non-parametric, meaning it makes no assumption about the underlying data distribution.
It has some disadvantages. kNN can be slow and computationally expensive, especially for large datasets, because it requires calculating the distance between the query and every point in the dataset during prediction. Also, if the feature space is large, if data contains irrelevant features or if data is unbalanced where one class significantly outnumbers others, kNN may not perform well.
It may be used for Image Recognition, Recommendation Systems and Medical Diagnosis.

1.2.2.2 Naive Bayes

Naive Bayes (NB) is a probabilistic classification algorithm [97] based on Bayes' Theorem, which describes the probability of an event occurring based on prior knowledge of related conditions.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \tag{1.3}$$

The "naive" part of NB refers to its assumption that each feature contributes independently to the outcome, which is often not the case in real-world data.
The way NB works is simple. It learns the prior probabilities of each class from the training data and estimates the likelihoods for each feature given the class. To classify a new instance, it calculates the posterior probability for each class using Bayes' Theorem and assigns the class with the highest probability.

1.2.2.3 Logistic Regression

Unlike linear regression, which is used to predict continuous quantities, logistic regression [53] is mainly used to predict discrete class labels. A logistic regression algorithm predicts probability with two possible categories for binary classification problems. Logistic regression uses a logistic function to classify the label in a binary outcome between 0 and 1 presented in Figure 1.11. [6]
It is easy to implement and understand (interpretable). Probabilistic regression provides well-calibrated probabilities for the classes, performs well on small and medium-sized

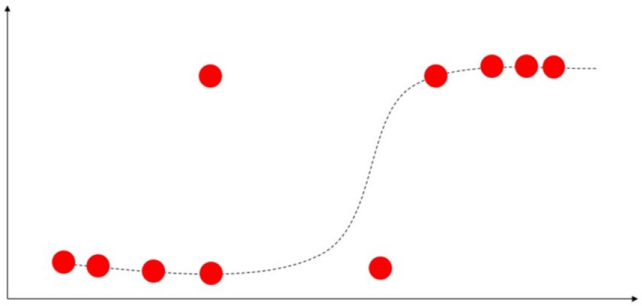

Figure 1.11: Demonstration of logistic regression with s-curve line.

datasets and can be extended with regularization techniques to avoid overfitting. Nevertheless it has some limitations. Logistic Regression assumes a linear relationship between the input variables and the log-odds of the outcome, which may not work well for complex problems, it is best suited for binary classification tasks and can be sensitive to the scale of the input features, so it may require feature scaling. Logistic regression provides a powerful yet simple tool for classification, especially when interpretability and understanding the relationship between input features and the target outcome are important. It can be applied in healthcare predicting disease outcomes based on patient data, credit risk modeling, spam detection and sentiment analysis.

1.2.2.4 Decision Trees

Decision Trees (DT) are a flexible, interpretable, and intuitive tool in machine learning used for classification and regression tasks [26]. It is a flowchart-like structure where the root node is the first node of the tree that represents the entire dataset. It is the feature that best splits the data based on a certain criterion. An internal node represents a feature (attribute), and split the data into subgroups based on a decision rule represented by a branch, and leaf nodes are terminal nodes that provide the predicted class label or value. For classification tasks, each leaf represents a class, while for regression, it represents a predicted value. The data is recursively split into subsets based on the features that provide the most significant difference in the target variable. The goal is to choose features and thresholds that create the purest possible groups.
DT are easy to visualize and interpret, they naturally handle non-linear relationships between variables and can be used for both classification (Classification Trees) and regression (Regression Trees). DT can grow excessively, leading to overfitting resulting in poor performance on new data. Also they may become biased when some classes dominate the dataset. Small changes in the data can result in a completely different tree structure, making the model less stable.
They are often used with advanced variants like Random Forest (RF). This is a collection of decision trees used to overcome overfitting by averaging the predictions of many trees.

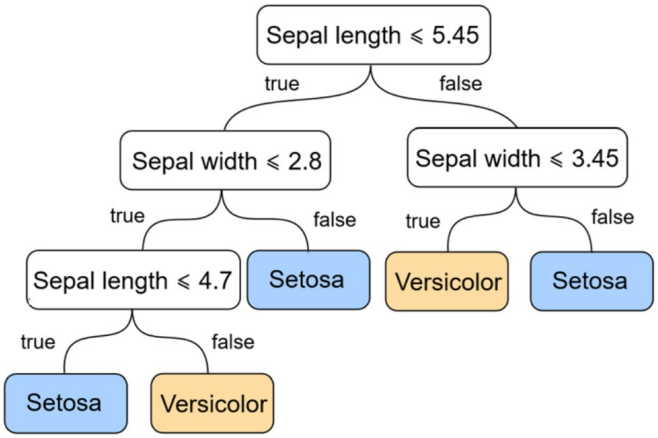

Figure 1.12: Decision Tree flowchart example [23]

An alternative combination is with Gradient Boosting Trees. They are built sequentially, where each new tree corrects errors from the previous one, to enhance their performance and reduce overfitting.
We find applications of DT, for example, in credit scoring used by financial institutions to decide whether to grant loans, in diagnosing diseases based on patient symptoms and test results and for marketing customer segmentation and determining purchase decisions.

1.2.2.5 Neural Networks

Neural Networks (NN) [69] are computational models inspired by the structure and function of biological neurons in the human brain. They are one of the most important and widely used techniques in machine learning and artificial intelligence, and have become the foundation for various deep learning applications such as image recognition, speech processing and natural language understanding.
It consists of interconnected layers of nodes (neurons). Each node or neuron receives one or more inputs, processes them, and outputs a result. Neurons in a network can be thought of as simple functions. Different weights can be applied to each edge connecting the neurons. At each neuron, an activation function is applied to a weighed input signal to generate an output signal. A sigmoidal function is often used, consisting of a first order lowpass filter of a unit step function. In Figure1.13 is the smallest building block in every neural network.

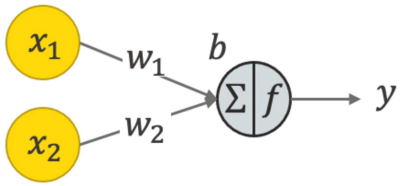

Figure 1.13: Image of an artificial neuron

Neurons are further subdivided into an input layer that receives the raw data or features (it does not process the data but just passes it into the next layer), hidden layers that process inputs by applying weights, biases, and an activation function to introduce non-linearity, and an output layer that produces the final prediction or classification result. In classification tasks, NN usually provide a probability distribution over different classes. There are different types of NN. The Feedforward Neural Network (FNN) (Figure1.14) is the simplest type, where information flows only in one direction, from input to output. There are no cycles or loops.

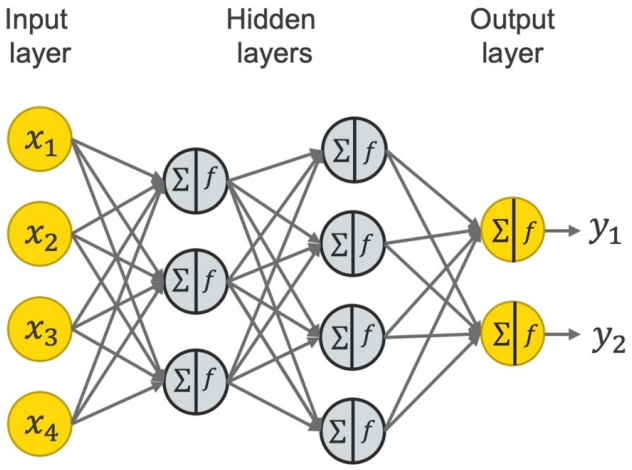

Figure 1.14: Example of a fully connected, feedforward Neural Network

Other types are Convolutional Neural Networks (CNN) [71] using convolutional layers to automatically detect features such as edges, textures, or objects from images, Recurrent Neural Networks (RNN) [64] designed for sequential data (like time series or text) and DNN [52] with many hidden layers leading to the development of Deep Learning.
The training process of Neural networks is done by adjusting weights based on the error between the network's predictions and the actual targets. In the forward pass (forward propagation), input data flows through the network from the input layer to the output layer, producing a prediction. The network's prediction is compared to the true value using a loss function, which measures how far off the prediction is. The error from the loss function is propagated back (back propagation) through the network, and the weights are updated to reduce this error. The gradient descent algorithm is often used to minimize the loss by updating the weights in the direction of the steepest descent of the error function. The training data is passed through the network multiple times, known as epochs, to allow the model to learn effectively.
Neural Networks have the ability to Learn Complex Patterns and can be applied to a wide variety of tasks, including classification, regression, and clustering. With non-linear activation functions, they can model highly complex real-world relationships that simpler models cannot. To counterbalance these advantages, they require large amounts of data for training, which can be a challenge for some tasks, training deep networks

can be computationally expensive and may require specialized hardware (like Graphics Processing Unit (GPU)) and if not properly regularized, neural networks can overfit to the training data, leading to poor generalization to new data.
We find applications of Neural Networks [4] in various fields like Image Recognition, Natural Language Processing (NLP), Speech Recognition and Autonomous Systems.

1.2.2.6 Ensemble Methods

Ensemble methods [31] are not a single machine learning algorithm, rather, they combine multiple individual models to enhance generalization performance of machine learning models, which have shown a remarkable potential (Figure1.15). There are plenty of real-world applications of ensemble learning such as object detection, object recognition, fault detection and diagnosis, semantic segmentation, edge detection, and for black box optimization.

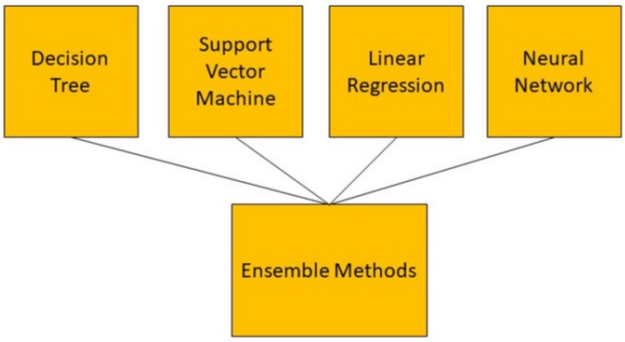

Figure 1.15: Demonstration of ensemble methods which combine different machine learning algorithms.

One advantage of ensemble methods is that they have high predictive accuracy that can be achieved in machine learning. However, the model's training process may be more complicated, where efficiency may not be possible. There are two standard ensemble learning algorithms currently: bagging-based algorithms and boosting-based algorithms. Bagging-based representative algorithms include random forest and boosting-based representative algorithms include Adaboost, GBDT, and XGBOOST [56]. There are a few advantages to using ensemble methods. Firstly, they can avoid the overfitting problem. A single machine learning algorithm can easily find many different hypotheses that can ideally forecast all the training data with less accuracy prediction for unseen examples when using a small data size. Thus, using combined algorithms (the different hypotheses of Averaging) minimizes the risk of selecting unsuitable hypotheses, thus improving overall forecasting performance. Secondly, ensemble methods have the advantage of computation. Ensemble methods can reduce the risk of reaching a local minimum by combining several algorithms as a single algorithm to perform a local search that may fall into the optimal solution. In any single model of an algorithm, the optimal hypothesis may go outside of space. Ensemble methods can extend the search space to fit the data by

integrating different algorithm models. The ensemble method can suit complex problems with large datasets [6].

1.2.2.7 Support Vector Machines (SVM)

Interpreting and classifying classical numerical data requires good classification methods. A SVM is a supervised method proposed by Vapnik [22] which has been used with success in classification [3].

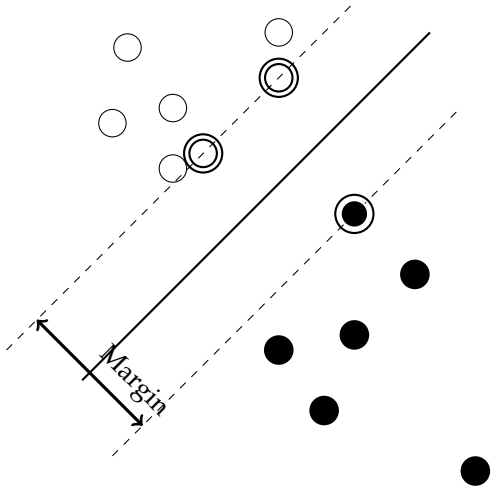

Figure 1.16: SVM with two features ($\mathbb{R}^2$)

SVM uses two sets of labeled data, one to find a frontier that separates two or more classes of data (the training examples) and the other to test the accuracy of the classification (the testing examples). In the classical SVM the frontier is defined by an hyper-plane. The frontier is defined so that the instances of the separate categories are divided by a clear gap, the margin in Figure1.16, that is as wide as possible, which enhances the model's generalization capability on unseen data. Support vectors are the data points that lie closest to the hyperplane and are critical in defining the position and orientation of the hyperplane.
New points are automatically predicted to belong to a category based on the side of the plane on which they fall. In addition to performing linear classification, SVM can efficiently perform a nonlinear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.
Several generalizations of the boundary function have been proposed in the literature. In this work, inspired by SVM, we propose a spherical separation, maintaining the concept of maximal margin. The key feature consists in letting the radius out of the minimization purpose so that, the sphere can approximate a linear separation by enlarging the radius and putting the center away from the centroid of the points. Most methods seek to minimize the ratio and so they have a poor performance for data with linear shape. We

formulate and solve this nonlinear problem for a set of generated data and for set of real data from a repository commonly used for this type of studies. This leads to a quadratic optimization problem. Further we propose a linear relaxation formulation that can give an approximate solution to this quadratic problem.

1.3 Interpretability, Explainability and Counterfactual Analysis

If a machine learning model performs well enough and has an acceptable predictive performance, why do we not just trust the model and disregard why it made a certain decision? Doshi-Velez and Kim answer this question by stating that "the problem is that a single metric, such as classification accuracy, is an incomplete description of most real-world tasks" [33]. Therefore, human-understandable explanations for machine-produced decisions are advantageous in several ways. Taking for example the case of applicants applying for loans, the benefits of an explanation would include:

- An explanation can be beneficial to those whose life is impacted by the decision. For example, it helps an applicant understand which of their attributes were strong drivers in determining a decision.
- Further, it can help an applicant challenge a decision if they feel an unfair treatment has been meted, e.g., if one's race was crucial in determining the outcome. This can also be useful for organizations to check for bias in their algorithms.
- In some instances, an explanation provides the applicant with feedback that they can act upon to receive the desired outcome at a future time.
- Explanations can help the machine learning model developers identify, detect, and fix bugs and other performance issues.
- Explanations help in adhering to laws surrounding machine-produced decisions.

Machine Learning Models whose internal decision-making processes are complex and not easily interpretable are called black box models, while models that allow humans to understand the logic behind decisions are interpretable models.
Interpretability [19] in machine learning refers to the degree to which a human can understand the cause of a decision made by an ML model, while **explainability** refers to the ability of a model to provide clear reasons or explanations for its predictions. It is closely related to interpretability, but is more focused on providing human-friendly explanations that clarify why a model made a particular decision. Examples of inherently interpretable models due to their simple structure, include logistic regression, decision trees, KNN or Naive Bayes. Examples of black box models that need post-hoc explanations include random forest, support vector machines (SVM) using kernel methods and neural networks, see Figure1.17. The lack of interpretability of these models can create issues

in applications such as healthcare, finance, or criminal justice, where understanding the reasoning of the decision is crucial.

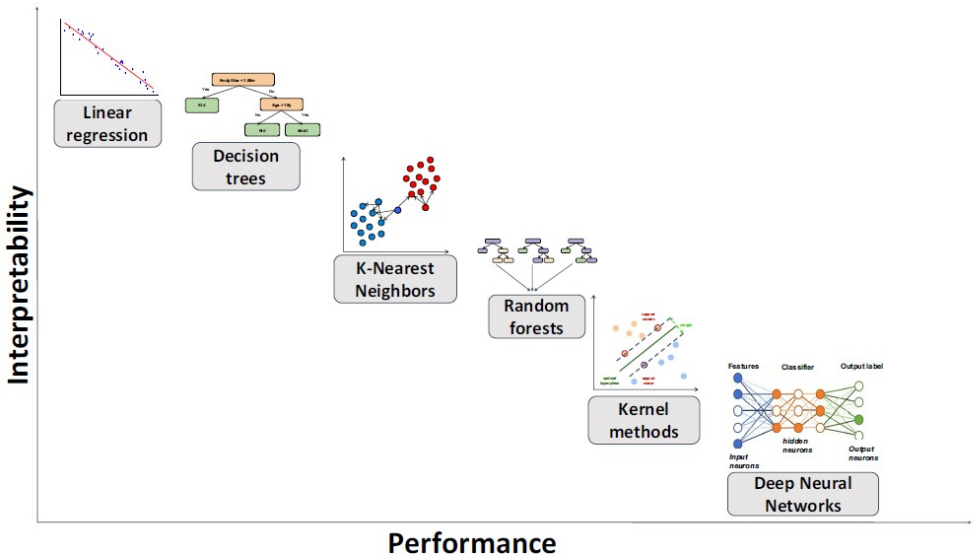

Figure 1.17: Overview of ML methods. The spectrum of available methods ranges from simpler and more interpretable to more advanced algorithms with potentially higher performance at the expense of less interpretability [11].

Nevertheless, it is worth taking into account that not all ML systems require interpretability, as there are situations where being able to provide high predictive performance is enough, with no need for explaining decisions. For that reason, according to Doshi-Velez and Kim [33], there are two kinds of situations where interpretability and, thus, explanations are not necessary: (1) when there is no significant impact or severe consequences for incorrect results and (2) when the problem is well-studied enough and validated in real applications that we trust the system's decisions, even if the system is not perfect. The former situation refers to low-risk systems, such as product recommenders and advertisement systems, where a mistake has no severe or even fatal consequences. The latter refers to well-studied systems that have been used for a while, including postal code sorting and aircraft collision systems, which compute their output without human intervention.
However, in most cases, it is also very important to know why a certain prediction was made, because a correct prediction only partially solves the original problem. **Counterfactual analysis** [92] in ML refers to understanding how changes in the input would alter the model prediction. Refering to our previous example in credit scoring, counterfactual analysis could answer questions like: "If an applicant was denied a loan, what is the smallest change in their financial history that would have led to approval?" So, it helps in understanding the decision boundary of a model showing what could be changed to ensure a positive outcome.

1.4 Introduction to Symbolic Data

Symbolic Data was proposed by Diday [30] to extend classic variables to more complex data, closer to reality, such as "events", "assertions","hordes" and "synthesis" objects. For instance in the study of birds, color is a variable of interest and different birds can be characterized by different conjunctions of colors. The concept of symbolic variables evolved to incorporate data defined by intervals and more recently by distributions, histograms. Symbolic data can be represented in complex tables where in each cell we may find not only values and categories but also sets, intervals or distributions. In these tables the rows refer to entities and columns are the corresponding features. We may have first or high order level units depending if we are considering in the first case objects or individuals or in the former classes. The features may be qualitative or quantitative, which in turn can be represented by single values (single-valued variables), sets of values (multi-valued variables), intervals (interval-valued variables) or probability/frequency/weight distributions (modal-valued variables).
In this work, we are particularly interested in modal-valued variables in the form of empirical distributions: histograms (histogram-values variables). Note that an interval can be considered an histogram with one class and frequency equal to 1 and in that sense histogram-valued variables are a generalization of interval-valued ones.
For instance, the entities may be beaches as in Table 1.3 and the register data can be interval-valued (Temperature of water), qualitative (Life watcher), a set of values (Facilities), or histogram-valued (Visitors).

Table 1.3: Symbolic data example

Beaches	Temperature of water	Life watcher	Number of facilities	Visitors
A	[14,22]	No	3	{[0,50], 0.2; [51,100], 0.4;[101,200], 0.3;[201,500], 0.1;}
B	[13,25]	Yes	2	{[0,50], 0.1; [51,100], 0.35;[101,200], 0.35;[201,500], 0.2;}
C	[16,28]	Yes	4	{[0,100], 0.3; [101,200], 0.5;[201,500], 0.2;}

One obvious importance of histogram data in the era of Big Data, that explains the increasing interest towards this subject is related to the ability to summarize large sets of data from databases, the web or generated in streaming by sensors. Reducing this array of data to just the average, implies losing information. So statistical methods were developed to extend Classic Statistical Analysis to symbolic variables, creating a new area of research defined as Symbolic Data Analysis (SDA) [15]. The focus has been more visible for interval data. Only recently new statistics and techniques for the analysis of histogram-valued data have been proposed. In comparison to interval data, histograms have the additional property of allowing to keep the variability information in the data description.

Different representations of histogram-valued data are illustrated in Figure 1.18, histograms, cumulative distribution functions and quantile functions.

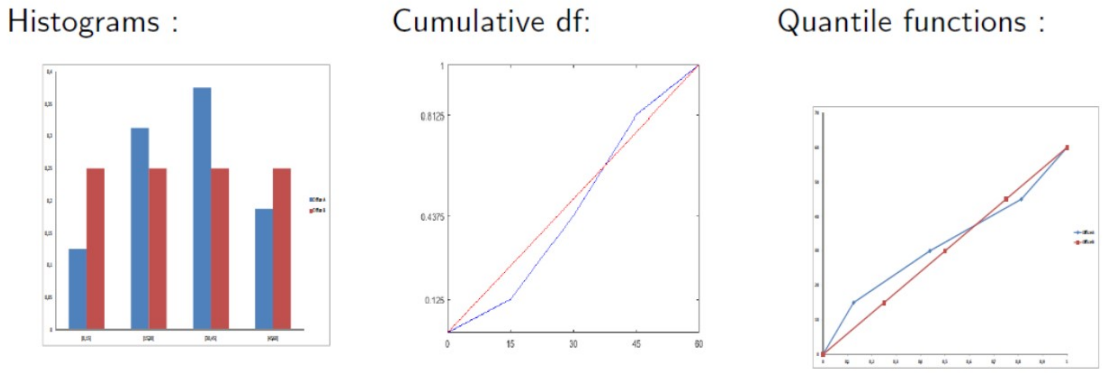

Figure 1.18: Representation of Histogram-valued Data

A classification of statistical methods for Symbolic Data is shown in Figure 1.19.

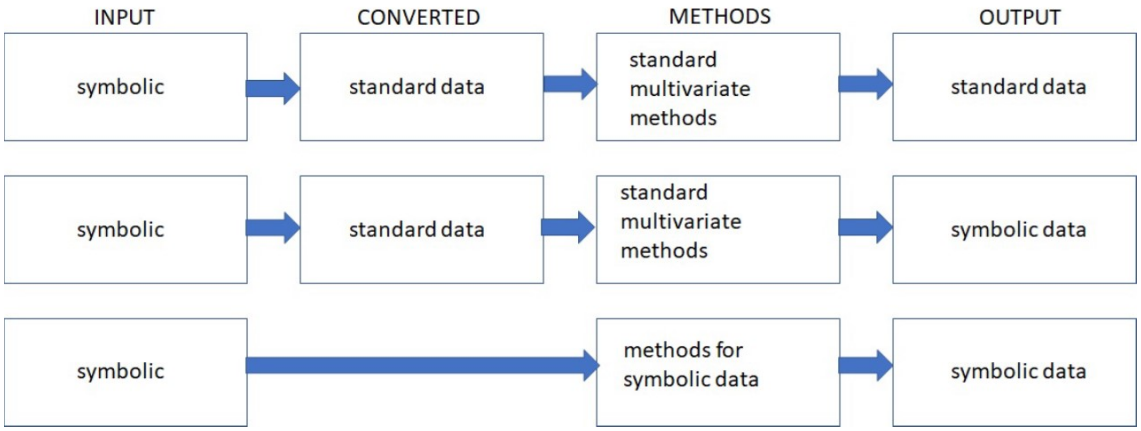

Figure 1.19: Statistical methods for SD

The classification method for inter-valued data proposed in this work and described in Chapter 3, is a symbolic-symbolic-symbolic method.

1.5 Motivation and Organization of the thesis

The challenges posed by machine learning are numerous. Alongside the virtue of possible applications in finance, medicine, engineering, management, and many other fields, comes the responsibility to address the potential dangers of its use. In particular, in sensitive areas, it is necessary to deal with possible bias in the developed model. Black-box models have had enormous success in certain areas, but they make it impossible to unequivocally determine and understand what defines the decision boundary of a classification model. For this reason, in this thesis, an effort was made to study ways to apply separation

models that are explainable and interpretable, and for which it is possible to establish how variations in the data can lead to an opposite outcome.
Deep learning and random forest models do not apply techniques to understand the internal classification mechanisms. The structure created by the model is too complex to allow for a reverse interpretation, from the output back to the input. In addition, other factors that hinder the interpretability and explainability of the model include the existence of many parameters and the use of kernel functions, which shift the data from its original space. This fact is a motivation for exploring new approaches to Support Vector Machine (SVM) models that can be developed in the original data space. The idea of using a curve defined by a hypersphere instead of a hyperplane came from the fact that, to define the hypersphere, only the center (whose structure is exactly the same as the data) and a radius (which is a positive real number) are needed. This allowed the model to be applied in the original data space, which is particularly useful for interval-based applications. This was the main motivation for this work, along with the fact that SVMs allow for the exploration of interesting optimization techniques. Also from a more academic point of view, it was also motivating to explore these models.
In adition to this chapter, the Introduction, this thesis is composed of three more chapters. The remainder of the thesis is organized as follows. In chapter 2 we present the classical SVM method introducing the formulation from a pure optimization perspective without resorting to classical formulas of distance of a point to a line. This helps to understand the formulation of the classical SVM. This is followed by a survey in spherical SVM to highlight the contribution of the methodology proposed in this work. Next we develop our method, the Spherical Support Vector Machine (SSVM). We also present a linear relaxation for the quadratic SSVM formulation and finally we present the results from the numerical computational experience.
In chapter 3 we introduce the basic concepts of Symbolic Data and the algebraic operations with histograms and quantile functions. A review of the different distance measures between functions is presented. Then we travel through the state of the art related to interval-valued data classification. The spherical SVM model for interval-valued data is presented, first considering only one feature (SSVMI), then multiple features (SSVMMI) for cases where the data is separable. Then we discuss the situations where the data is non-separable considering soft margins (SSSVMMI). A linear relaxation for this formulation is also presented. Finally a discussion of computational results is presented.

2

A maximal margin hypersphere SVM

In this chapter, we propose a Support Vector Machine (SVM) with spherical separation, maintaining the concept of maximal margin. The key feature consists of letting the radius out of the minimization purpose, so that the sphere can approximate a linear separation by enlarging the radius and putting the center away from the centroid of the points. Most methods seek to minimize the ratio and so they have a poor performance for data with linear shape. We formulate and solve this nonlinear SVM for a set of generated data and for a set of real data from a benchmark repository commonly used for this type of studies. This leads to a quadratic optimization problem and we also propose a linear formulation that can give an approximate solution to this quadratic problem.

2.1 The standard SVM

In this section, we are going to review the SVM method and describe the fundamentals of the theory behind it, using a pure optimization interpretation.

SVM is a supervised classification method. It uses two sets of labeled data, one to find a frontier that separates two or more classes of data (the training examples) and the other to test the accuracy of the classification (the testing examples). In the classical SVM the frontier is defined by an hyperplane. The frontier is defined so that the instances of the separate categories are divided by a clear gap, the margin in Figure 2.1, that is as wide as possible. New points are automatically predicted to belong to a category based on the side of the hyperplane on which they fall. In addition to performing linear classification, SVM can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

In the next section, we are going to explain the classic SVM using a pure optimization perspective. We will ignore the formula of the euclidean distance of a point to a line

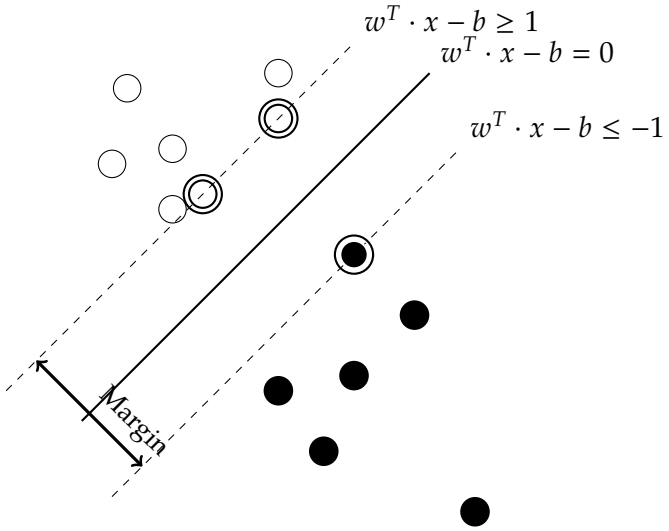

Figure 2.1: SVM with two features ($\mathbb{R}^2$)

in $\mathbb{R}^2$ and deduce it from an optimization point of view.

2.1.1 The distance of a point to a plane as an optimization problem

The distance of a point (x_j^1, x_j^2) to a line $r \equiv w_1 x_1 + w_2 x_2 = b$ is a known algebraic formula that can be obtained by an optimization perspective. We want to find the closest point in r to (x_j^1, x_j^2) according to some measuring distance $D(\cdot)$.

$$\begin{aligned} &\min_{x_1,x_2} D\left((x_j^1, x_j^2), (x_1, x_2)\right) \\ &s.t.\ w_1 x_1 + w_2 x_2 = b \\ &x_1, x_2 \in \mathbb{R} \end{aligned}$$

Using for instance the half of the square of the euclidean distance we obtain:

$$P:\ z = \min \frac{1}{2}\left((x_j^1 - x_1)^2 + (x_j^2 - x_2)^2\right) \tag{2.1}$$

$$s.t.\ w_1 x_1 + w_2 x_2 = b \tag{2.2}$$

$$x_1, x_2 \in \mathbb{R} \tag{2.3}$$

For a general optimization problem

$$z = \min f(x) \tag{2.4}$$

$$s.t.\ Ax = b \tag{2.5}$$

the optimal solution can be found by solving the system

$$\begin{cases} \nabla f &= A^T \lambda \\ Ax &= b \end{cases} \tag{2.6}$$

where λ are the Lagrangian multipliers. For problem P we obtain

$$\begin{cases} \nabla f = \begin{bmatrix} (x_j^1 - x_1) \\ (x_j^2 - x_2) \end{bmatrix} = \lambda \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \\ w_1 x_1 + w_2 x_2 = b \end{cases} \tag{2.7}$$

so using 2.7 in 2.1 we obtain

$$2z^* = \lambda^2 w_1^2 + \lambda^2 w_2^2 = \lambda^2 \|w\|^2 \tag{2.8}$$

solving 2.7 in order of x_1, x_2

$$x_1 = x_j^1 - \lambda w_1 \tag{2.9}$$

$$x_2 = x_j^2 - \lambda w_2 \tag{2.10}$$

$$\tag{2.11}$$

and replacing it in $w_1 x_1 + w_2 x_2 = b$

$$w_1(x_j^1 - \lambda w_1) + w_2(x_j^2 - \lambda w_2) = b \tag{2.12}$$

$$\tag{2.13}$$

we obtain

$$\lambda = \frac{w_1 x_j^1 + w_2 x_j^2 - b}{\|w\|^2} \tag{2.14}$$

Finally, from 2.8 and 2.14 we obtain the distance D_j of a point (x_j^1, x_j^2) to the line $w_1 x_1 + w_2 x_2 = b$

$$D_j = \sqrt{2z^*} = |\lambda| \|w\| = \frac{|w_1 x_j^1 + w_2 x_j^2 - b|}{\|w\|} \tag{2.15}$$

2.1.2 The SVM problem

Given a set of training examples (x_j, y_j) for $j \in N = \{1, \ldots, n\}$, corresponding to two classes, one class with $y_j = 1$ for $j \in N_1$ and the other with $y_j = -1$ for $j \in N_2$, in order to find a hyperplane defined by a vector w that maximizes the margin, and such that the points from distinct classes lie on opposite sides of the hyperplane, we need to solve the following optimization problem:

$$SVM: \; v = \max_{j \in N} \min_{w_1, w_2} D_j \tag{2.16}$$

$$s.t. \; w_1 x_j^1 + w_2 x_j^2 - b \geq 1 \text{ for } j \in N_1 \tag{2.17}$$

$$w_1 x_j^1 + w_2 x_j^2 - b \leq -1 \text{ for } j \in N_2 \tag{2.18}$$

$$D_j = \frac{|w_1 x_j^1 + w_2 x_j^2 - b|}{\|w\|} \text{ for } j \in N \tag{2.19}$$

This maxmin problem can be reformulated as

$$SVM: \ v = \max \delta \tag{2.20}$$
$$s.t. \ w_1 x_j^1 + w_2 x_j^2 - b \geq 1 \text{ if } y_j = 1 \tag{2.21}$$
$$w_1 x_j^1 + w_2 x_j^2 - b \leq -1 \text{ if } y_j = -1 \tag{2.22}$$
$$\frac{|w_1 x_j^1 + w_2 x_j^2 - b|}{\|w\|} \geq \delta \text{ for } j \in N \tag{2.23}$$

which is equivalent to

$$SVM: \ v = \max \delta \tag{2.24}$$
$$s.t. \ y_j(w_1 x_j^1 + w_2 x_j^2 - b) \geq 1 \text{ for } j \in N \tag{2.25}$$
$$|w_1 x_j^1 + w_2 x_j^2 - b| \geq \delta \|w\| \text{ for } j \in N. \tag{2.26}$$

Since together with constraints (2.25) $|w_1 x_j^1 + w_2 x_j^2 - b| = yi(w_1 x_j^1 + w_2 x_j^2 - b)$ we may combine (2.25) and (2.26) and obtain

$$SVM: \ v = \max \delta \tag{2.27}$$
$$s.t. \ y_j(w_1 x_j^1 + w_2 x_j^2 - b) \geq \delta \|w\| \text{ for } j \in N. \tag{2.28}$$

Finally since the problem is invariant to re-scaling of the parameters, we may constrain $\delta\|w\|$ to be equal to any constant in particular 1 and in that case $\delta = \frac{1}{\|w\|}$. Considering that maximizing $\frac{1}{\|w\|}$ is equivalent minimizing $\|w\|$ we obtain the well known SVM problem formulation

$$SVM: \ v = \min \|w\| \tag{2.29}$$
$$s.t. \ y_j(w_1 x_j^1 + w_2 x_j^2 - b) \geq 1 \text{ for } j \in N. \tag{2.30}$$

There will be some points, in each class for which the margin in Figure 2.1 is attained. They are called the support vector and are signaled by a double circle in Figure 2.1.

2.2 SVM with hyperspheres for classification - A small survey

Inspired by SVM [22] proposed by Vapnik, some efforts have been made to improve SVM, in particular, using hyperspheres for classification. Table 2.1 resumes some of the papers that constitute a milestone in the development of generalization of SVM based on spheres.

Spherical classifiers were first introduced into pattern classification by Cooper in 1962 [21] and subsequently developed and generalized by many other researchers. Support Vector Data Description (SVDD), [83–85], is a one-class (the target class) classification method. The goal is to devise a closed decision boundary that may distinguish a member of this

Table 2.1: Cronological list of papers.

Year	Title	reference
1999	Support vector domain description	[85]
2002	Uniform object generation for optimizing one-class classifiers	[83]
2004	Support vector data description	[84]
2005	Pattern classification via single-spheres	[93]
2007	A new maximal margin spherical structured multi-class support vector machine	[43]
2009	A fixed-center spherical separation algorithm with kernel transformations	[10]
2009	A new maximal-margin spherical-structured multi-class support	[43]
2010	Least Squares Twin Support Vector Hypersphere (LS-TSVH) for pattern recognition	[74]
2010	DC models for spherical separation	[8]
2012	Margin maximization in spherical separation	[9]
2013	Sphere Support Vector Machines for large classification tasks	[81]
2013	A twin-hypersphere support vector machine classifier and the fast learning algorithm	[76]
2014	Twin Support Vector Hypersphere (TSVH) classifier for pattern recognition	[77]
2016	A maximum margin and minimum volume hyperspheres machine with pinball loss for imbalanced data classification	[96]
2016	Twin pinball loss support vector hyper-sphere classifier for pattern recognition	[42]
2017	Steel surface defect classification using multiple hyperspheres support vector machine with additional information	[41]
2017	Linear approach for twin-Hypersphere support vector machine	[50]
2018	Multi-class classification method based on support vector machine with hyper-sphere for steel surface defects	[40]
2019	Support vector machine with quantile hyperspheres for pattern classification	[60]
2019	A spheres-based support vector machine for pattern classification	[75]

class from the rest of the feature space, characterized by center and radius $R > 0$ and described by the support vectors. The boundary can also be used to detect novel data or outliers. The model in [83–85] seeks to find a minimum radius hypersphere containing almost all the data from this class. To allow the possibility of outliers in the training set, the distance from data points to the center need not to be strictly smaller than the radius, but these larger distances should be penalized introducing slack variables measuring the distance to the boundary, if an object is outside the description. The sum of the slack variables are included in the objective function parameterized by a constant as a trade-off between the volume of the hypersphere and the errors.

Several versions of classifiers have been extended from SVDD. These classifiers include maximal-margin spherical-structured multi-class SVM (MSM-SVM) [43], Twin support vector hypersphere (TSVH) [77] and Twin-hypersphere support vector machine (THSVM)

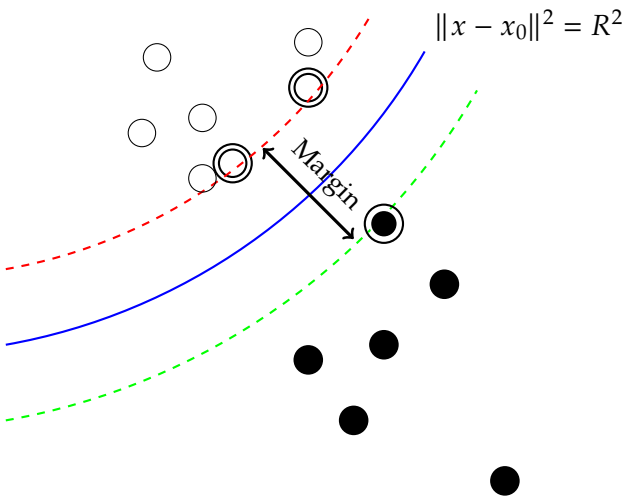

Figure 2.2: Non-Linear SVM with two features ($\mathbb{R}^2$)

[76]. These models find two hyperspheres, one for each class, and classify points according to which hypersphere a given point is relatively closest to. The samples of one class are as much as possible covered by the corresponding hypersphere, whereas, the samples in the other class are as far as possible from this hypersphere.
Other approaches related tho spherical separation [96] use maximum margin and minimum volume hyperspheres machine with pinball loss (Pin-M3HM) and least squares twin support vector hyperspheres (LS-TSVH) [74] .
A support vector machine with quantile hyperspheres (QHSVM) for binary classification [60] was proposed by Maoxiang et al. (2019). Here, the idea is similar to SVDD. However, it needs to build two hypersheres with the same center. Considering two boundary hyperspheres (BHSVM) where target samples are inside a hypershere with radius R+ and negative samples are outside a hypersphere with radius R-. BHSVM maximizes the shortest distance between two classes by minimizing the volume of the first and maximizing the volume of the second, to keep margin as big as possible. QHSVM has been developed by introducing the pinball loss in BHSVM.
In this work, contrary to other approaches we do not minimize the radius. We propose to solve a SVM that is similar to the classic SVM in the sense that we attempt to maximize the margin. The only difference is that the separation boundary is a curve and so it can better adjust to the shape of points as depicted if Figure 2.2. Locally, the larger the radius and the further away the center is from the centroid of the points, the closer the boundary is to a line. In this sense it is a generalization of the classic SVM, but that works well even if the data from one class lies in the interior of the second class.

2.3 A maximal margin hyphersphere SVM method

Consider a set of $N = \{1, 2, \ldots, n\}$ points each one with $M = \{1, 2, \ldots, m\}$ features and distributed in two separable classes C_1 and C_2, such that $N_1 = \{1, 2, \ldots, |C_1|\}$ and $N_2 = \{1, 2, \ldots, |C_2|\}$.
The goal is to find a maximal separation function $f(x, \theta)$, where θ are the parameters of the function, and $\delta \geq 0$ measures the separation, such that:

$$f(x, \theta) \leq b - \delta \text{ if } x \in C_1$$

$$f(x, \theta) \geq b + \delta \text{ if } x \in C_2.$$

In general we may consider $b = 0$, so to find the optimal θ and δ, an optimization problem of the following type must be solved:

$$G: \ v_G = \max h(\delta) \tag{2.31}$$

$$s.t. f(x_i, \theta) \leq -\delta, \ \text{ if } i \in N_1 \tag{2.32}$$

$$f(x_j, \theta) \geq +\delta, \ \text{ if } j \in N_2, \tag{2.33}$$

$$\delta \geq 0 \tag{2.34}$$

where $h(\delta)$ is a strictly increasing function on $\mathbb{R}^+$.

Spherical SVM have been proposed in the literature to define a non-linear separation between two sets of data with different labels $y = -1$ for $x \in C_1$ and $y = 1$ for $x \in C_2$. The common procedure is to find a sphere centered at some point inside the cloud of points defined by one set, such as the centroid. Sometimes two spheres are proposed each one with center in the centroid of each set. However these methods will have a poor performance if the points are distributed along a line. In this work we propose to define a sphere were the center is not necessarily rooted at the center of the points and may even be far outside the centroid if necessary. In this way, depending on the center and the radius, the separation curve may approximate, or not, a linear separator. A margin may be defined as in the linear SVM which we aim to maximize.

2.3.1 SVM with spherical boundary maximizing the separation margin

Our proposal is to find a radius R and a center x_0 of a hypersphere such that the points of one class C_1 fall inside and the points in C_2 fall outside the hypersphere, as in Figure 2.2.

$$\|x - x_0\|^2 \leq R^2 \rightarrow x \in C_1$$

$$\|x - x_0\|^2 > R^2 \rightarrow x \in C_2,$$

and there is a margin δ separating the points in each class

$$\|x - x_0\|^2 \leq (R - \delta)^2 \rightarrow x \in C_1$$

$$\|x - x_0\|^2 > (R + \delta)^2 \rightarrow x \in C_2.$$

So the goal is to find x_0 and positive R that maximizes non-negative δ.

$$S_0 : \ v_0 = \max \ h(\delta) \tag{2.35}$$
$$s.t. \|x_i - x_0\|^2 \leq (R - \delta)^2, \ \text{ if } i \in N_1 \tag{2.36}$$
$$\|x_i - x_0\|^2 \geq (R + \delta)^2, \ \text{ if } i \in N_2 \tag{2.37}$$
$$R \geq \delta$$
$$\delta, R \geq 0.$$

We are going to consider in (2.35) $h(\delta) = \delta$ and we assign a label $y = \pm 1$ to differentiate the classes. The definiton of the label of each class is irrelevant, so we may assume that for $x \in C_1$ we have $y = 1$ and for $x \in C_2$ we have $y = -1$. So from

$$\|x - x_0\|^2 - R^2 - \delta^2 \leq -2R\delta \Leftrightarrow x \in C_1 \Leftrightarrow y\left(\|x - x_0\|^2 - R^2 - \delta^2\right) \leq -2R\delta \text{ for } y = 1$$

$$\|x - x_0\|^2 - R^2 - \delta^2 > 2R\delta \Rightarrow x \in C_2 \Leftrightarrow y\left(\|x - x_0\|^2 - R^2 - \delta^2\right) \leq -2R\delta \text{ for } y = -1$$

we obtain, the following equivalent problem:

$$S_1 : \ v = \max \ \delta \tag{2.38}$$
$$s.t. \ y_i\left(\|x_i - x_0\|^2 - R^2 - \delta^2\right) \leq -2R\delta \text{ for } i \in N. \tag{2.39}$$

Consider $z = \begin{bmatrix} x_0 \\ \delta \\ R \end{bmatrix}$, $Q_i = \begin{bmatrix} y_i I(n) & 0 & 0 \\ 0 & -y_i & 1 \\ 0 & 1 & -y_i \end{bmatrix}$, $v_i = \begin{bmatrix} -y_i x_i \\ 0 \\ 0 \end{bmatrix}$.

Let e be a vector of dimension $n + 2$ of zeros except in the component $n + 1$ which is equal to one. S_1 can be rewritten as:

$$S_1 : \ v = \max \ e^T z \tag{2.40}$$
$$s.t. \ z^T Q_i z + 2v_i^T z \leq -y_i x_i^T x_i \text{ for } i \in N. \tag{2.41}$$

The Lagrangian function corresponding to problem (2.40), (2.41) is given by (with $\lambda_i \geq 0$):

$$L(z, \lambda) = e^T z - \sum_{i \in N} \lambda_i \left(z^T Q_i z + 2v_i^T z + y_i x_i^T x_i\right) \tag{2.42}$$

and

$$\nabla_z L(z, \lambda) = e - \sum_{i \in N} \lambda_i \left(2Q_i z + 2v_i\right) \tag{2.43}$$

and from $\nabla_z L(z, \lambda) = 0$ we obtain considering $X^y = [y_i x_i]$

$$z = \left(\sum_{i \in N} \lambda_i Q_i\right)^{-1} \left(\frac{e}{2} - \sum_{i \in N} \lambda_i v_i\right) = \left(\sum_{i \in N} \lambda_i Q_i\right)^{-1} \left(\frac{e}{2} - V\lambda\right). \tag{2.44}$$

Considering $\lambda^+ = \sum_{i\in N} \lambda_i$ and $\lambda_y^+ = \sum_{i\in N} y_i\lambda_i$ then

$$\left(\sum_{i\in N} \lambda_i Q_i\right)^{-1} = \begin{bmatrix} \frac{1}{\lambda_y^+} I(n) & 0 & 0 \\ 0 & \frac{\lambda_y^+}{(\lambda^+)^2-(\lambda_y^+)^2} & \frac{\lambda^+}{(\lambda^+)^2-(\lambda_y^+)^2} \\ 0 & \frac{\lambda^+}{(\lambda^+)^2-(\lambda_y^+)^2} & \frac{\lambda_y^+}{(\lambda^+)^2-(\lambda_y^+)^2} \end{bmatrix}$$

and so

$$z = \begin{bmatrix} x_0 \\ \delta \\ R \end{bmatrix} = \begin{bmatrix} \frac{X^y\lambda}{\lambda_y^+} \\ \frac{\lambda_y^+}{(\lambda^+)^2-(\lambda_y^+)^2} \\ \frac{\lambda^+}{(\lambda^+)^2-(\lambda_y^+)^2} \end{bmatrix}$$

and

$$R - \delta = \frac{1}{\lambda^+ + \lambda_y^+} = \frac{1}{\lambda_1^+} \tag{2.45}$$

$$R + \delta = \frac{1}{\lambda^+ - \lambda_y^+} = \frac{1}{\lambda_2^+} \tag{2.46}$$

where $\lambda_1^+ = \sum_{i\in N_1} \lambda_i$ and $\lambda_2^+ = \sum_{i\in N_2} \lambda_i$.

Notice that the points x_i for $i \in N$ with $\lambda_i > 0$ are the support vectors. If $i \in N_1$ they fall on the curve $\|x - x_0\|^2 = (R - \delta)^2$ and if $i \in N_2$, they fall on the curve $\|x - x_0\|^2 = (R + \delta)^2$. Since there is at least one support vector for each class then the denominators (2.45) and (2.46) do not vanish. Also, given that the number of support vectors is small, we expect most of the dual variables to be zero. Since the problem is not symmetric, contrary to the linear case, the formulation must be solved twice, considering $y_i = 1$ for $i \in N_1$, $y_i = -1$ for $i \in N_2$ and also for $y_i = -1$ for $i \in N_1$, $y_i = 1$ for $i \in N_2$.

2.3.2 Soft Margins

If data is not separable, we have to allow the existence of soft margins and the introduction of slack variables ϵ_i as follows (with parameter $\mu > 0$):

$$SSVM: \ v_0 = \max \delta - \mu \sum_{i=1}^{n} \epsilon_i \tag{2.47}$$

$$s.t. \ \|x_i - x_0\|^2 \le (R - \delta + \epsilon_i)^2, \ \text{ if } i \in N_1 \tag{2.48}$$

$$\|x_i - x_0\|^2 \ge (R + \delta - \epsilon_i)^2, \ \text{ if } i \in N_2 \tag{2.49}$$

$$R \ge \delta$$

$$\delta \ge 0 \ , x_0 \in \mathbb{R}^m$$

$$\epsilon_i \ge 0, \ i \in (N_1 \cup N_2)$$

The variables ϵ_i and ϵ_j allow for points in Class 1 to be outside the sphere, and points of Class 2 to fall inside the sphere, respectively, counting in both cases with the margin.

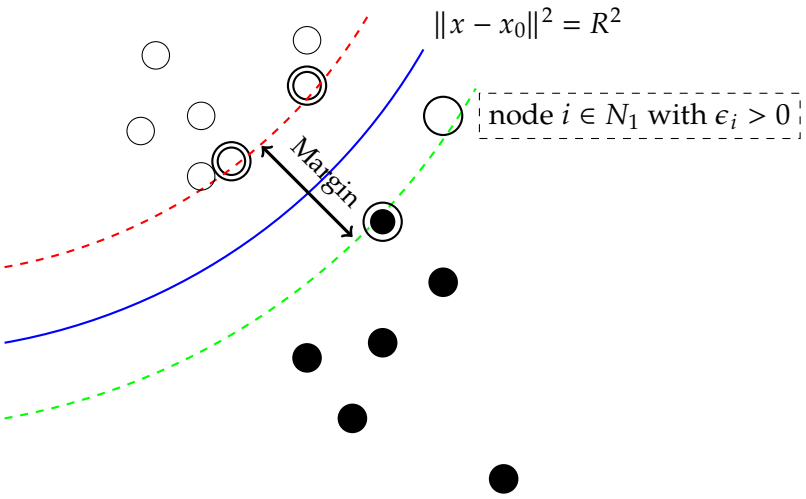

Figure 2.3: Soft margins in $\mathbb{R}^2$ (two features)

Figure 2.3 depicts the case of a point that lies outside the boundary defined by the red dashed line.

2.3.3 A linear relaxation of the quadratic formulation

The quadratic Spherical Support Vector Machines (SSVM) problem is nonconvex, so for large scale problems a global optimal solution may be difficult to achieve computationally even with the best solvers. In that sense we developed a relaxation problem, from which a feasible solution for the original problem can be retrieved. The relaxation obtained is a linear programming problem and so can be efficiently solved even for large scale problems. The relaxation can also be used to find an upper bound for the optimal value. In a heuristic approach this value can be used to define optimality gaps. The upper bound will be a key feature in a branch and bound approach in a future work.
Instead of formulation (2.47),(2.48) and (2.49), to facilitate calculations and without lost of generality we will use the folllowing alternative formulation (with parameter $\mu > 0$):

$$SSSVM: \ v_0 = \max \ \delta - \mu(\sum_{i=1}^{n1} \epsilon_i + \sum_{j=1}^{n2} \epsilon_j), \ n_1 = \left|C_1\right|, \ n_2 = \left|C_2\right| \tag{2.50}$$

$$s.t. \ \|x_i - x_0\|^2 \leq (R - \delta)^2 + \epsilon_i, \ i \in N_1 \tag{2.51}$$

$$\|x_j - x_0\|^2 \geq (R + \delta)^2 - \epsilon_j, \ j \in N_2 \tag{2.52}$$

$$R \geq \delta$$

$$\delta \geq 0$$

$$\epsilon_i, \epsilon_j \geq 0, \ i \in N_1, j \in N_2$$

$$x_0 \in \mathbb{R}^m$$

From the constraints formulation (2.51) and (2.52) we obtain:

$$\|x_i - x_0\|^2 + 2R\delta - \epsilon_i \leq R^2 + \delta^2,\ i \in N_1 \tag{2.53}$$

$$\|x_j - x_0\|^2 - 2R\delta + \epsilon_j \geq R^2 + \delta^2,\ j \in N_2 \tag{2.54}$$

and so from (2.53) and (2.54), for $i \in N_1, j \in N_2$ we have:

$$\|x_i - x_0\|^2 + 2R\delta - \epsilon_i \leq R^2 + \delta^2 \leq \|x_j - x_0\|^2 - 2R\delta + \epsilon_j \Leftrightarrow$$

$$\Leftrightarrow \|x_i\|^2 - 2x_i^T x_0 + \|x_0\|^2 + 2R\delta - \epsilon_i \leq R^2 + \delta^2 \leq \|x_j\|^2 - 2x_j^T x_0 + \|x_0\|^2 - 2R\delta + \epsilon_j \Leftrightarrow$$

$$\Leftrightarrow \|x_i\|^2 - 2x_i^T x_0 + 2R\delta - \epsilon_i \leq R^2 + \delta^2 - \|x_0\|^2 \leq \|x_j\|^2 - 2x_j^T x_0 - 2R\delta + \epsilon_j.$$

Because each feasible solution of $SSSVM$ fulfils condition

$$\|x_i\|^2 - 2x_i^T x_0 + 2R\delta - \epsilon_i \leq \|x_j\|^2 - 2x_j^T x_0 - 2R\delta + \epsilon_j \text{ for } i \in N_1, j \in N_2,$$

we have that

$$4R\delta + 2(x_j - x_i)^T x_0 - \epsilon_i - \epsilon_j \leq \|x_j\|^2 - \|x_i\|^2,\ \text{ for } i \in N_1, j \in N_2.$$

In order to have a linear problem we take $\theta = R\delta$ and obtain the following relaxation problem for $SSSVM$.

$$SSSVML: \max \theta - \mu(\sum_{i=1}^{n1} \epsilon_i + \sum_{j=1}^{n2} \epsilon_j),\ n_1 = \left|C_1\right|,\ n_2 = \left|C_2\right| \tag{2.55}$$

$$\text{s.t. } 4\theta + 2(x_j - x_i)^T x_0 - \epsilon_i - \epsilon_j \leq \|x_j\|^2 - \|x_i\|^2,\ \text{ for } i \in N_1, j \in N_2 \tag{2.56}$$

$$\epsilon_i, \epsilon_j \geq 0,\ i \in N_1, j \in N_2$$

$$\theta \geq 0.$$

However, this problem may be unbounded, because, due to some particular geometry of data points, from

$$4\theta \leq \|x_j\|^2 - \|x_i\|^2 - 2(x_j - x_i)^T x_0 + \epsilon_i + \epsilon_j,$$

if $2(x_j - x_i)^T x_0 < 0$, then x_0 can be chosen such that $-2(x_j - x_i)^T x_0$ is arbitrarily large.

In fact, considering the case where $\epsilon_i = 0$ and $\epsilon_j = 0$ and given a direction d:

$$d = \begin{bmatrix} d_1 \\ \\ d_0 \end{bmatrix} \tag{2.57}$$

with $d_1 \in R^+$, $d_0 \in R^n$.

If d is a direction of unboundedness, for $\gamma \in R_0^+$ we have:

$$4(\theta + \gamma d_1) + 2(x_j - x_i)^T (x_0 + \gamma d_0) \leq \|x_j\|^2 - \|x_i\|^2,\ \text{ for } i \in N_1, j \in N_2 \tag{2.58}$$

and

$$\theta + \gamma d_1 \geq 0. \tag{2.59}$$

So we get

$$4\gamma d_1 + 2(x_j - x_i)^T \gamma d_0 \leq 0, \quad \text{for } i \in N_1, j \in N_2. \tag{2.60}$$

Since $\gamma > 0$, dividing by 4γ and considering $\alpha^{ij} = -1/2(x_j - x_i)^T$ we get:

$$d_1 - \alpha^{ij} d_0 \leq 0, \quad \text{for } i \in N_1, j \in N_2 \Leftrightarrow \tag{2.61}$$

$$\Leftrightarrow \alpha^{ij} d_0 \geq d_1, \quad \text{for } i \in N_1, j \in N_2. \tag{2.62}$$

Without loosing generality we can set $d_1 = 1$ and obtain the condition:

$$\alpha^{ij} d_0 \geq 1. \tag{2.63}$$

Introducing a slack variable $F^{i,j}$ we get

$$\alpha^{ij} d_0 - F^{i,j} = 1, F^{i,j} \geq 0. \tag{2.64}$$

So, if the solution of the following problem

$$\min \sum \beta^{i,j} \tag{2.65}$$

$$\text{s.t. } \alpha^{ij} d_0 - F^{i,j} = 1 - \beta^{i,j}, \quad \text{for } i \in N_1, j \in N_2 \tag{2.66}$$

$$\beta^{i,j} \geq 0 \tag{2.67}$$

$$F^{i,j} \geq 0 \tag{2.68}$$

is $\min \sum \beta^{i,j} = 0$, then d is a direction of unboundedness.
To overcome this issue, we propose introducing an indirect regularization technique. Instead of adding to the objective function $-k\|x_0\|$, where k is a parameter, we introduce a new variable ρ and a new constraint, in the above formulation obtaining (with parameters $k > 0$ and $\mu > 0$):

$$SSSVML(k): \ \max \theta + k\rho - \mu(\sum_{i=1}^{n1} \epsilon_i + \sum_{j=1}^{n2} \epsilon_j), \ n_1 = \left|C_1\right|, \ n_2 = \left|C_2\right| \tag{2.69}$$

$$\text{s.t. } 4\theta + 2(x_j - x_i)^T x_0 - \epsilon_i - \epsilon_j \leq \|x_j\|^2 - \|x_i\|^2 \text{ for } i \in N_1, j \in N_2 \tag{2.70}$$

$$4\theta + 2(x_j - x_i)^T x_0 - \epsilon_i - \epsilon_j \geq \rho \text{ for } i \in N_1, j \in N_2 \tag{2.71}$$

$$\theta \geq 0.$$

$$\epsilon_i, \epsilon_j \geq 0, \ i \in N_1, j \in N_2$$

The solution of this linear problem is $(\theta, \rho, x_0, \epsilon)$. Therefore, we still have to find R and δ and for that we have two possible ways, which we will describe below.

Since we have optimum x_0, to get optimum R and δ, we can solve the following linear problem (with $\mu > 0$):

$$SRL: \ \max \delta - \mu(\sum_{i=1}^{n_1} \varphi_i + \sum_{j=1}^{n_2} \varphi_j), \ n_1 = \left|C_1\right|, \ n_2 = \left|C_2\right| \tag{2.72}$$

$$\text{s.t.} \ \ R - \delta + \varphi_i \geq \|x_i - x_{0_{optim}}\| \text{ for } i \in N_1, j \in N_2 \tag{2.73}$$

$$R + \delta - \varphi_j \leq \|x_j - x_{0_{optim}}\|, \text{ for } i \in N_1, j \in N_2 \tag{2.74}$$

$$R \geq \delta$$

$$\delta \geq 0$$

$$\varphi_i \geq 0, \text{ for } i \in N_1$$

$$\varphi_j \geq 0, \text{ for } j \in N_2$$

In alternative to this procedure, and considering that at the support vectors, we have:

$$\theta_{optim} + 2(x_{j_{sv}} - x_{i_{sv}})^T x_{0_{optim}} - \epsilon_{i_{sv}} - \epsilon_{j_{sv}} = \|x_{j_{sv}}\|^2 - \|x_{i_{sv}}\|^2, \ \text{ for } i \in N_1, j \in N_2$$

After finding the support vectors, we calculate R and δ.

$$R = \frac{\sqrt{\|x_{j_{sv}} - x_{0_{optim}}\|^2 + \epsilon_{j_{sv}}} + \sqrt{\|x_{i_{sv}} - x_{0_{optim}}\|^2 - \epsilon_{i_{sv}}}}{2}$$

$$\delta = \frac{\sqrt{\|x_{j_{sv}} - x_{0_{optim}}\|^2 + \epsilon_{j_{sv}}} - \sqrt{\|x_{i_{sv}} - x_{0_{optim}}\|^2 - \epsilon_{i_{sv}}}}{2}$$

The next example ilustrates the application of the quadratic and linear formulation.

Example 2.3.1 *Consider the set of points in $\mathbb{R}^2$ from two classes, each with 6 elements.*

Class1

x_1	3	5	6	6	7	9
x_2	2	3	2	3	3	2

Class2

x_1	2	3	5	7	9	10
x_2	3	4	5	5	4	3

For the quadratic formulation, we obtained $x_0 = (6, -1)$, $r = 4,9497$ and the margin, $\delta = 0,7071$. The support vectors are $(3, 2)$ and $(9, 2)$ for Class 1 and $(2, 3)$ and $(10, 3)$, for Class 2. This solution is depicted in Figure 2.4.
For the linear formulation we obtained $x_0 = (6, -1.5)$, $\theta = 3,75$, $r = 5,3153$ and the margin, $\delta = 0,7055$. The support vectors are $(3, 2)$ and $(9, 2)$ for Class 1 and $(2, 3)$ and $(10, 3)$, for Class 2. This solution is depicted in Figure 2.5.

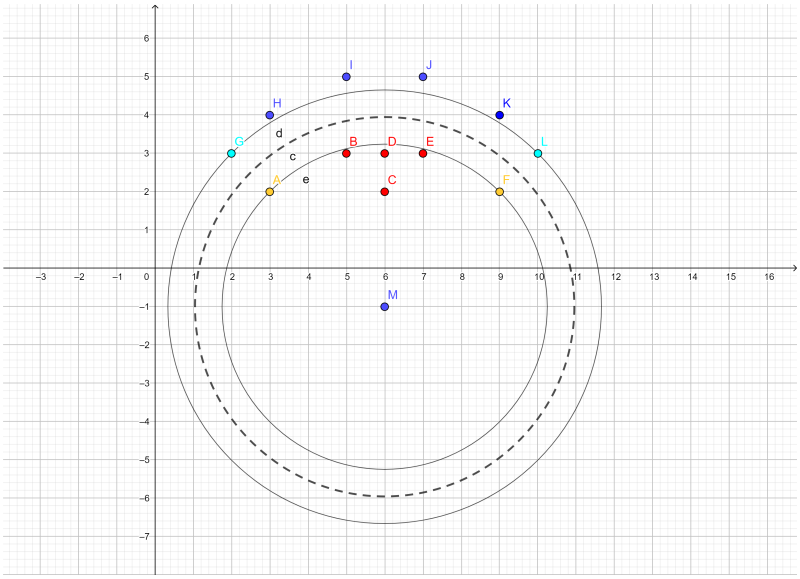

Figure 2.4: Spherical separation using the quadratic formulation

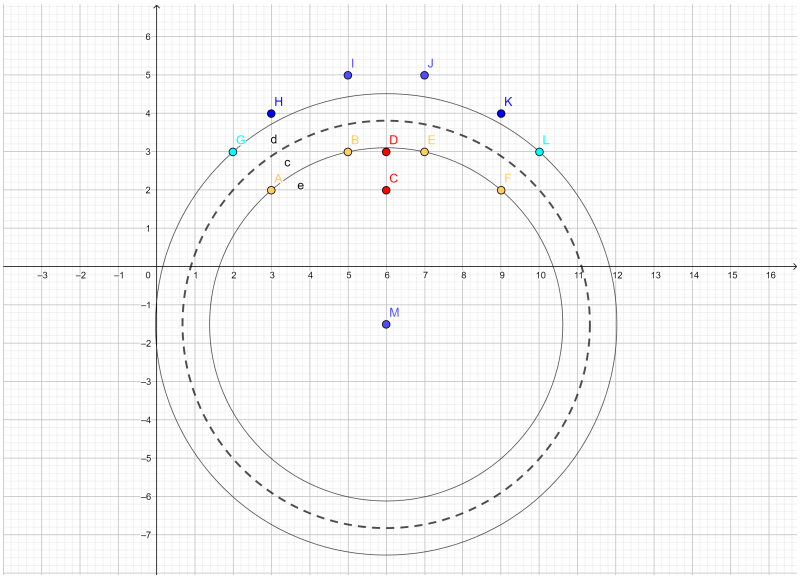

Figure 2.5: Spherical separation using the linear formulation

2.4 Computational Experience

The goal of the computational experience is to test the classification accuracy of the quadratic Soft Spherical Support Vector Machines (SSSVM) and the Soft Margin Spherical SVM Linear Relaxation (LSSSVM) models by comparing them against the accuracy results obtained with all the 34 classifiers available in MATLAB's R2024a Classification Learner App (Table I.1), for real-world and generated datasets.

Experiments were run on a computer with:

- Intel(R) Core(TM) i7-8550U processor, 1.80GHz
- 8.00 GB of RAM
- 64-bit operating system running on Windows 10, version 20H2.

- We code our method on Matlab R2024a, and we used BARON software, version 13.0.1, running on Matlab with MATLAB/BARON interface version v1.89.
- The BARON solver was used to solve the quadratic formulation.
- To run all 34 Matlab classifiers we used the Machine Learning and Deep Learning APP from MATLAB.

We used two types of datasets, which properties are summarized in Table 2.2.

Simul is a simulated data set, built in order to have two classes, linearly not separated and

Table 2.2: Attributes and Number of Data Set Samples

Data Set	Attributes	class1	class 2
Simul	2	100	100
Iris	4	50	50
Thyroid	5	150	35
Seeds	7	70	140
Twonorm	20	200	200

also spherically not separated, but having a cloud of points spherically shaped, each one with two attributes. For class A we consider 100 points $a_i = (a_{i_1}, a_{i_2})$ so that: a_{i_1} satisfies a Gaussian distribution $N(0, 2)$ and $a_{i_2} = \sqrt{100 - a_{i_1}^2} + \gamma_i$, with γ_i satisfying a Gaussian distribution $N(0, 0.1)$. For class B we considered also 100 points $b_j = (b_{j_1}, b_{j_2})$ so that: b_{j_1}satisfies a Gaussian distribution $N(0, 2)$ and $b_{j_2} = \sqrt{105 - b_{j_1}^2} + \gamma_j$, with γ_j satisfying a Gaussian distribution $N(0, 0.1)$. This dataset is represented in Figure 2.6.

Iris, Seeds, Thyroid and Twonorm datasets are real-world numerical datasets available

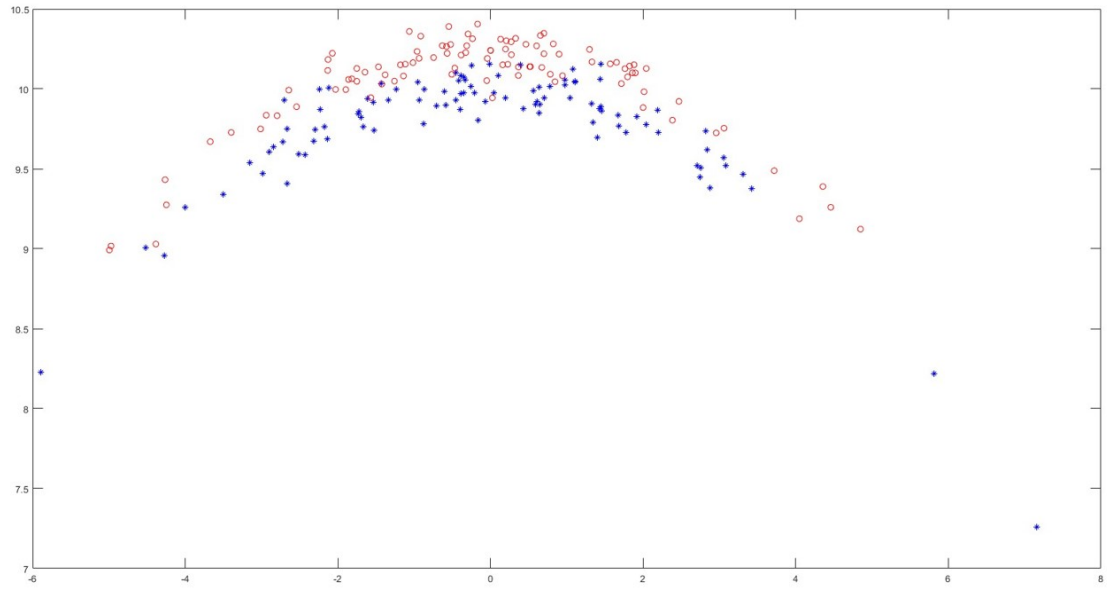

Figure 2.6: Simulated data set

from UCI Machine Learning Repository and commonly used in classification literature. The accuracy results from the 34 Matlab classifiers were condensed in intervals in which the lower limit refers to the minimum accuracy and the upper limit to the maximum accuracy value obtained. Also for SSSVM and LSSSVM we calculate the gap from the best Matlab accuracy value. The accuracy results obtained are summarized in Tables 2.3, 2.4 and 2.5.

Table 2.3: 5-fold cross validation accuracy results

5-fold cross validation					
Data Set	Accuracy%			Gap%	
	MATLAB Classifiers	SSSVM	LSSSVM	SSSVM	LSSSVM
Simul	[54.00 ; 86.50]	88.50	88.00	-	-
Iris	[50,00 ; 97,00]	96.00	95.00	1.03	2.06
Seeds	[66.67 ; 97,14]	95.71	95.24	1.47	1.96
Thyroid	[81.08 ; 99.46]	95.13	93.51	4.35	5.98
Twonorm	[68.00 ; 98.00]	96.25	94.75	1.79	3.32

Table 2.4: Complete set accuracy results

complete set validation					
Data Set	Accuracy%			Gap%	
	MATLAB Classifiers	SSSVM	LSSSVM	SSSVM	LSSSVM
Simul	[60.50 ; 100.00]	89.00	88.00	11.00	12.00
Iris	[50.00 ; 100.00]	98.00	100.00	2.00	0.00
Seeds	[66.67 ; 100.00]	92.38	96.19	7.62	3.81
Thyroid	[81.08 ; 100.00]	100.00	97.30	0.00	2.70
Twonorm	[79.75 ; 100.00]	100.00	98.25	0.00	1.75

Table 2.5: 10-fold cross validation accuracy results

10-fold cross validation accuracy					
Data Set	Accuracy%			Gap%	
	MATLAB Classifiers	SSSVM	LSSSVM	SSSVM	LSSSVM
Simul	[55.00 ; 85.00]	87.50	88.00	-	-
Iris	[50.00 ; 100.00]	90.00	90.00	10.00	10.00
Seeds	[66.67 ; 100.00]	95.24	95.24	4.76	4.76
Thyroid	[78.95 ; 100.00]	100.00	100.00	0.00	0.00
Twonorm	[65.00 ; 100.00]	97.50	95.00	2.50	5.00

Figure 2.7 shows a comparison graph of the SSSVM and LSSSVM 5-fold cross validation accuracy for iris dataset with all the 34 classifiers available in MATLAB.

With complete set validation, we got 100% accuracy with SSSVM, in Thyroid and Twonorm datasets. The results with Simul data set, for SSSVM as well as for LSSSVM, present a

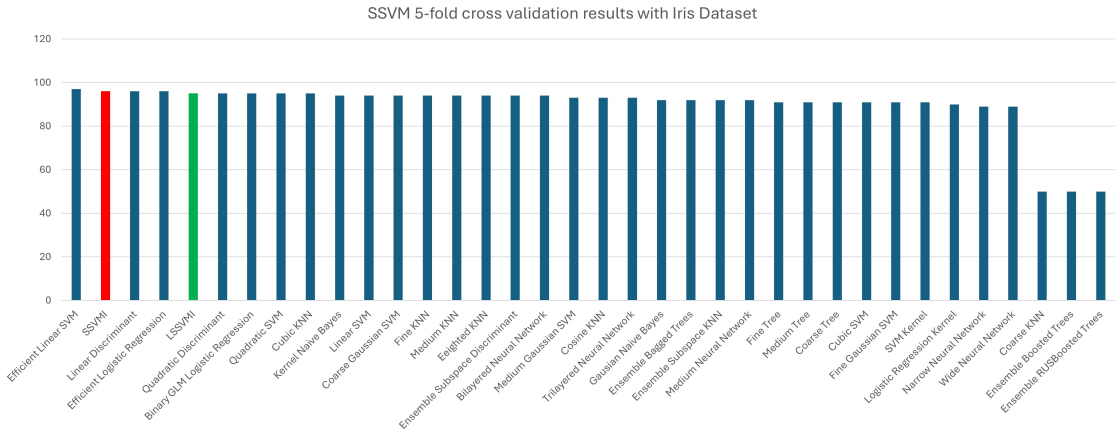

Figure 2.7: SSSVM 5-fold Cross Validation Results with Iris Dataset

bigger gap, but comparing these results with the good performance obtained with 5-fold cross validation, we can conclude that we only do not get overfitting but also have a good generalization ability for both formulations.

3

Interval-valued Data Classification

3.1 Introduction

SVM have demonstrated considerable success in data classification tasks, either employed directly or serving as a foundational approach for the creation of related methodologies [3, 17, 20, 35, 37, 43, 84]. The SVM algorithm is valued for its ability to handle high-dimensional spaces. Researchers have proposed numerous enhancements and generalizations of the SVM boundary function to improve its performance [62] and adapt it to various application domains.

These innovations have expanded the applicability of SVM, making them a crucial tool in the analysis of complex data sets generated by modern monitoring systems, such as sensors, which are widespread in agriculture, economics, security, and industry in general. Interpreting and classifying this type of data requires adaptation of previous methods.

Consider for instance that we have data collected from sensors for several items (machines, people, cities, etc) and we want to create a classification model that will allow to differentiate between the malfunction and operational units. The malfunction state can be interpreted in a broad sense, as anything opposed to normality. It can be an actual malfunction of a device, a disease, a fraud, and so on. We may consider to use the full array of data, for each unit, which is not a good option for several reasons. First we cannot define as a feature the i-th reading of the sensor. A feature is a measurement that has an unique and equal meaning for all instances and that can be compared among them: price, weight, height, pressure, temperature. If our study has a chronological component and each measurement for all units is taken in the same period of time, then it may be meaningful to consider a feature that is the i-th reading of the units regarding some period of time. For instance, it can be the reading in the i-th day, the i-th week, or the i-th year. If not, then it is meaningless and can conduct to wrong interpretations. In addition we may have different numbers of readings for each unit. Their number may be too high as is the case of sensor data. In these situations, it is common to use the average instead of the original vector of data. Replacing the vector of data by a single measure of central tendency

may work well in some cases, for instance, if what differentiates a good from a malfunction behavior in our devices is a shift in the average. However, different distributions of data can be an indication of a malfunction without affecting the mean as is illustrated for the case in Figure 3.1.

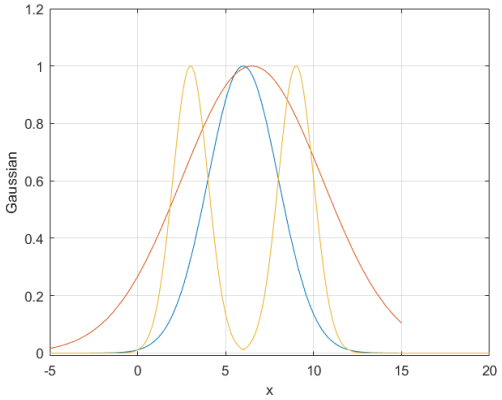

Figure 3.1: Distribution of data.

In some cases, the malfunction may induce a shift of values away from the average in both sides (increase of the variance), or values that are just low or high (bimodal distribution). In the first case, using the mean will not be effective to interpret a malfunction and if using the mean and variance may work for this case, in the bi-modal pattern malfunction case, that will not be enough.

Examples of bimodal distribution as an indicator of a clear differentiation between two distinct classes can be found in many contexts. For instance, a two peak in an age distribution of a cancer is an important distinction, often implying that what was considered one type of cancer can actually be two different cancers, with similar morphology but with distinctive clinical features and different methods of treatment [12]. Another example arises in the context of spam and co-bursting detection [55], where reviewer posting rates (number of reviews written in a period of time) in fake or spam reviews to secretly promote or demote some target products and services, are bimodal. In the scope of fraud detection, bimodal distributions are often an indication of fraud in exams or elections [51]. In [36] detection of climate changes using bimodal distributions is also reported.

Considering that the features of distinction between classes are not known *apriori*, methods that take as much information as possible into account about the distribution of the data are important. Such general method will work well in all situations described and for automatic classification it is important to have methods with a large spectrum of applicability. These methods work with variables more complex than arrays, data with structures known as *Symbolic Data*.

3.2 Interval-Valued Data

Statistical methods for symbolic data can be classified according to the input-data method-output types [46, 91].

- symbolic-numerical-numerical: symbolic data in input are transformed into standard data in order to apply classic multivariate techniques. The result is classic data.
- symbolic-numerical-symbolic: symbolic data in input are analyzed according to a classic multivariate techniques and the result is symbolic data.
- symbolic-symbolic-symbolic: symbolic data in input are transformed using generalization/specialization operators and the result is symbolic data.

A typical symbolic-numerical-numerical method for intervals $[a, b]$ could use $v = (a, b)$ or $v = b - a$ instead of the interval and apply standard statistical methods to v.
If, for instance the interval is transformed into $v = ((a+b)/2, (b-a)/2)$, standard calculations can be performed for v. If a response of the same structure is obtained this allows the reconstruction of an output interval. This is the sort of symbolic-numerical-symbolic method.
For symbolic-symbolic-symbolic the methods and operations must be adapted to deal with symbolic objects. This kind of approach is more complex and recent.
The first proposals of basic univariate and bivariate statistics for histogram-valued data were due to Bertrand and Goupil [13] and later integrated and extended by Billard and Diday [14]. More recently, Irpino and Verde [47] proposed a novel set of univariate and bivariate statistics that better take the sources of variability in data into account and extend some properties of the classic basic statistics to those for multi-valued numeric data. In the context of discriminant analysis, Silva and Brito [34] made a contribution for interval-valued data and Dias and Brito [28] and Dias, Brito and Amaral [29] for histogram-valued variables.

3.3 State of the art

In general, most existing published research articles related to interval-valued data mainly focus on clustering analysis, regression analysis, and feature selection and less on classification tasks.
The first proposals of basic univariate and bivariate statistics for histogram-valued data were due to Bertrand and Goupil [13] and later integrated and extended by Billard and Diday [14].
Irpino and Verde [47] proposed a novel set of univariate and bivariate statistics that better take the sources of variability in data into account and extend some properties of the classic basic statistics to those for multi-valued numeric data. In the context of discriminant

analysis, Silva and Brito [34] made a contribution for interval-valued data and Dias and Brito [28] and Dias, Brito and Amaral [29] for histogram-valued variables.
Utkin and Coolen [88] presented a new approach for constructing regression and classification models for interval-valued data including the extension of the support vector machine method for interval-valued data. Dias [27] proposed new linear regression models for histogram-valued data and interval-valued data represented by their quantile functions.
Utkin and Zhuk [89] proposed a one-class classification support vector machine model by interval-valued training data. Qi [79] proposed an Interval-Valued Data classification method based on the Unified Representation Frame which does not only take the mid point and radius of the interval-valued data into account but also the relationship between them.
Recently, Ma [61] proposed a method for interval-valued data classification based on multiview learning which is a machine learning paradigm that deals with learning from multiple data representations (views) of the same underlying information. The aim of the proposed algorithm is to classify multiview information extracted from the interval-valued observations using SVM, random forests, and neural networks as the basic structures of the multiview classifier.

3.4 Algebraic operations with Histograms and Quantile functions

The development of methods for symbolic data requires the definition of algebraic operations and notions of distance between elements. In this section we will address arithmetic operations between histograms. We will discuss the difficulties in using such arithmetic and introduce quantile functions as a more convenient representation of histograms, simplifying the definition of these operations.
The outcome associated with a histogram-valued variable, may take the form:

$$H_X = \{I_{X_1}, p_{X_1}; ...; I_{X_n}, p_{X_n}\}$$

where I_{X_i}, represents the subinterval i; p_{X_i} is the weight associated with the subinterval I_{X_i} and $\sum_{i=1}^{n} p_{X_i} = 1$ with n the number of subintervals.

Example 3.4.1 *Considering two histograms,*

$$H_1 = \{[1, 3[, 0.1; [3, 5[, 0.6; [5, 8[, 0.3\}$$

$$H_2 = \{[0, 1[, 0.8; [1, 4[, 0.2\}$$

Basic operations become:

$$
\begin{aligned}
H_1 + H_2 =& \{[1,2[, 0.0267; [2,3[, 0.0307; [3,4[, 0.1907; [4,5[, 0.18807; [5,6[, 0.2480; \\
& [6,7[, 0.0980; [7,9[, 0.1880; [9,12[, 0.0300\} \\
H_1 + 2 =& \{[3,5[, 0.1; [5,7[, 0.6; [7,10[, 0.3\} \\
H_1 - 2 =& \{[-1,1[, 0.1; [1,3[, 0.6; [3,6[, 0.3\} \\
2H_1 =& \{[2,6[, 0.1; [6,10[, 0.6; [10,16[, 0.3\} \\
-H_1 =& \{[-8,-5[, 0.3; [-5,-3[, 0.6; [-3,-1[, 0.1\} \\
H_1 - H_1 =& \{[-7,-5[, 0.0120; [-5,-4[, 0.0420; [-4,-3[, 0.0570; [-3,-2[, 0.0720; [-2,0[, 0.3170; \\
& [0,2[, 0.3170; [2,3[, 0.0720; [3,4[, 0.0570; [4,5[, 0.0420; [5,7[, 0.0120\}.
\end{aligned}
$$

These operations are not suited to be used in an algorithm that implies complex operations and steps. Besides they present undesirable properties as for instance the fact that $\frac{H+H}{2} \neq H$. As alternative, quantile functions were proposed in [7], [45], to represent histograms with a much simpler system of basic operations. A quantile function is a piece-wise function given by the inverse of the cumulative functions.

$$
\Psi_X^{-1}(t) = \begin{cases}
\underline{l}_{X_1} + \frac{(t-w_0)}{w_1-w_0}\left(\overline{l}_{X_1} - \underline{l}_{X_1}\right) & \text{if } w_0 \leq t < w_1 \\
\underline{l}_{X_2} + \frac{(t-w_1)}{w_2-w_1}\left(\overline{l}_{X_2} - \underline{l}_{X_2}\right) & \text{if } w_1 \leq t < w_2 \\
\vdots & \\
\underline{l}_{X_m} + \frac{(t-w_{m-1})}{w_m-w_{m-1}}\left(\overline{l}_{X_m} - \underline{l}_{X_m}\right) & \text{if } w_{m-1} \leq t < w_m \\
\vdots &
\end{cases}
$$

where $w_j = \sum_{k=1}^{j} p_j$ so $w_m = 1$ and $w_0 = 0$.
Using the centers and half rays to define the histogram we have:

$$
\Psi_X^{-1}(t) = \begin{cases}
c_{X_1} + \left(\frac{2(t-w_0)}{w_1-w_0} - 1\right) \times r_{X_1} & \text{if } w_0 \leq t < w_1 \\
c_{X_2} + \left(\frac{2(t-w_1)}{w_2-w_1} - 1\right) \times r_{X_2} & \text{if } w_1 \leq t < w_2 \\
\vdots & \\
c_{X_m} + \left(\frac{2(t-w_{m-1})}{w_m-w_{m-1}} - 1\right) \times r_{X_m} & \text{if } w_{m-1} \leq t \leq w_m
\end{cases}
$$

Given the histograms in Example 3.4.1, the cumulative functions are given by:

$$\Psi_X(x) = \begin{cases} 0.1\frac{x-1}{2} & \text{if } 1 \le x < 3 \\ 0.1 + 0.6\dfrac{x-3}{2} & \text{if } 3 \le x < 5 \\ 0.7 + 0.1\dfrac{x-5}{3} & \text{if } 5 \le x \le 8 \end{cases}, \Psi_Y(x) = \begin{cases} 0.8x & \text{if } 0 \le x < 1 \\ 0.8 + 0.2\dfrac{x-1}{3} & \text{if } 1 \le x \le 4 \end{cases}$$

and their inverse the quantile functions are,

$$\Psi_X^{-1}(t) = \begin{cases} 1 + \dfrac{t}{0.1} \times 2 & \text{if } 0 \le t < 0.1 \\ 3 + \dfrac{t-0.1}{0.6} \times 2 & \text{if } 0.1 \le t < 0.7 \\ 5 + \dfrac{t-0.7}{0.3} \times 3 & \text{if } 0.7 \le t \le 1 \end{cases}, \Psi_Y^{-1}(t) = \begin{cases} \dfrac{t}{0.8} & \text{if } 0 \le t < 0.8 \\ 1 + \dfrac{t-0.8}{0.2} \times 3 & \text{if } 0.8 \le t \le 1 \end{cases}$$

These quantile functions are represented in Figure 3.2.

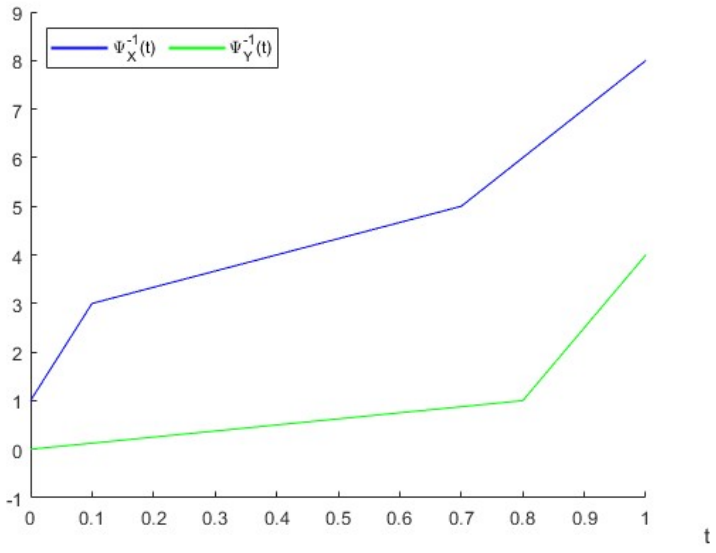

Figure 3.2: Quantile functions Ψ_X^{-1}, Ψ_Y^{-1}

For quantile functions, the basic operations are defined as usual in functions:

- Addition: $\Psi_X^{-1}(t) + \Psi_Y^{-1}(t) = (\Psi_X^{-1} + \Psi_Y^{-1})(t)$
- Product by a real number: $\alpha\Psi_X^{-1}(t) = (\alpha\Psi_X^{-1})(t)$

However, notice that the set of quantile functions with the above operations is not a subspace. In fact if $\alpha < 0$ then $\alpha\Psi_X^{-1}$ is not a quantile function. To allow for general linear combinations of quantile functions, Dias and Brito [28] proposed to use a quantile function given by $-\Psi_X^{-1}(1-t)$ representing $-H_X$ the symmetric of histogram H_X.

3.5 Distance measures

Most of the methods for supervised or unsupervised classification require the definition of a distance metric between objects. In the case of histogram value data, for which intervals are a particular case, several measures were presented in literature to measure the distance between two quantile functions (see e.g. [16]).

Table 3.1: Divergency measures between distributions

Divergency measures	
Kullback-Leibler	$D_{KL}(f,g) = \int_{\mathbb{R}} log\left(\frac{f(x)}{g(x)}\right) f(x)dx$
Jeffreys	$D_J(f,g) = D_{KL}(f,g) + D_{KL}(g,f)$
χ^2	$D_{\chi^2}(f,g) = \int_{\mathbb{R}} \frac{\lvert f(x)-g(x)\rvert^2}{g(x)}dx$
Hellinger	$D_H(f,g) = \left[\int_{\mathbb{R}} \left(\sqrt{f(x)} - \sqrt{g(x)}\right) dx\right]^{\frac{1}{2}}$
Total variation	$D_{var}(f,g) = \int_{\mathbb{R}} \lvert f(x)-g(x)\rvert dx$
Kolmogorov	$D_K(f,g) = \max_{\mathbb{R}} \lvert F(x)-G(x)\rvert$
Wasserstein	$D_W(f,g) = \int_0^1 \lvert F^{-1}(t) - G^{-1}(t)\rvert dt$
Mallows	$D_M(f,g) = \sqrt{\int_0^1 \left(F^{-1}(t) - G^{-1}(t)\right)^2 dt}$

From the measures given in Table 3.5, where $f(x)$ and $g(x)$ are probability density functions defined for $x \in \mathbb{R}$, $F(x)$ and $G(x)$ are probability distribution functions and $F^{-1}(x)$ and $G^{-1}(x)$ are the respective quantile functions, the Mallows distance (3.1) has been considered as an adequate measure to evaluate the similarity between distributions.

$$D_M\left(\Psi^{-1}_{Y_1(i)}, \Psi^{-1}_{Y_2(i)}\right) = \sqrt{\int_0^1 (\Psi^{-1}_{Y_1(i)}(t) - \Psi^{-1}_{Y_2(i)}(t))^2 dt}. \tag{3.1}$$

This distance has been successfully used in cluster analysis for histogram data [45], in forecasting histogram time series [7], and in linear regression with histogram/interval-valued variables [7, 28, 29, 47].

For histogram-valued data, with K pieces, under the uniformity hypothesis, and considering a fixed weight decomposition (same weights, different intervals), (3.1) can be rewritten,

using the centers and half rays to define the classes of the histograms, as (see [45]):

$$D_M^2(\Psi_{Y_1(i)}^{-1}, \Psi_{Y_2(i)}^{-1}) = \sum_{\ell=1}^{K} p_\ell \left[(c_{Y_1(i)} - c_{Y_2(i)})^2 + \frac{1}{3}(r_{Y_1(i)} - r_{Y_2(i)})^2 \right]. \tag{3.2}$$

3.6 A Spherical SVM for Interval-Valued Data Classification

3.6.1 One feature- SSVMI

In this section we describe an extension of SSVM for interval-valued data. We will start with the univariate case $M = 1$ of just one variable X which is a random interval-valued variable. For each observation i, X_i can be represented by an interval with center c_{X_i} and radius r_{X_i}:

$$I_{X_i} = [c_{X_i} - r_{X_i}, c_{X_i} + r_{X_i}] \tag{3.3}$$

and the quantile function $\Psi_{X_i}^{-1}$

$$\Psi_{X_i}^{-1}(t) = c_{X_i} + (2t - 1)r_{X_i}, \quad \text{for } 0 \leq t \leq 1. \tag{3.4}$$

The classification model, analogue to (2.37) is addressed by an optimization problem where the goal is to find $\Psi_{X_o}^{-1}$ and positive R that maximizes non-negative δ. The formulation is as follows:

$$\begin{aligned} SSVMI: \max\ & \delta \\ \text{s.t. } & D^2(\Psi_{X_i}^{-1}, \Psi_{X_o}^{-1}) \leq (R - \delta)^2, \quad \text{for } i \in N_1 \\ & D^2(\Psi_{X_j}^{-1}, \Psi_{X_o}^{-1}) \geq (R + \delta)^2, \quad \text{for } j \in N_2 \\ & R \geq \delta \\ & \delta \geq 0. \end{aligned} \tag{3.5}$$

Using for the distance D the Mallows distance

$$D_M^2(\Psi_{X_i}^{-1}, \Psi_{X_0}^{-1}) = (c_{X_i} - c_{X_0})^2 + \frac{1}{3}(r_{X_i} - r_{X_0})^2 \tag{3.6}$$

where c_{X_0} and r_{X_0} are respectivelly the center and the radius of the quantile function $\Psi_{X_0}^{-1}$ representing the center of the sphere with radius R, we obtain the following problem formulation:

$$
\begin{aligned}
SSVMI : \max\ & \delta \\
\text{s.t. } & (c_{X_i} - c_{X_o})^2 + \frac{1}{3}(r_{X_i} - r_{X_o})^2 \leq (R - \delta)^2, \ \text{ for } i \in N_1 \\
& (c_{X_j} - c_{X_o})^2 + \frac{1}{3}(r_{X_j} - r_{X_o})^2 \geq (R + \delta)^2, \ \text{ for } j \in N_2 \\
& r_{X_o} \geq 0 \\
& R \geq \delta \\
& \delta \geq 0
\end{aligned} \tag{3.7}
$$

using $y_i = 1$ for class 1 and $y_i = -1$ for class 2 we obtain the equivalent formulation

$$
\begin{aligned}
SSVMI : \max\ & \delta \\
\text{s.t. } & y_i(c_{X_i} - c_{X_o})^2 + \frac{y_i}{3}(r_{X_i} - r_{X_o})^2 \leq y_i(R - y_i\delta)^2, \ \text{ for } i \in N \\
& r_{X_o} \geq 0 \\
& R \geq \delta \\
& \delta \geq 0.
\end{aligned} \tag{3.8}
$$

3.6.2 Multiple features - SSVMMI

Now to extend the previous formulation to more than one feature, each observation $i \in N$ incorporates L intervals, $I_{X_i}(1)$ to $I_{X_i}(L)$, each one defined by a center and radius.

$$
I_{X_i}(l) = [c_{X_i(l)} - r_{X_i(l)}, c_{X_i(l)} + r_{X_i(l)}], \ i \in N, \ l = 1...L \tag{3.9}
$$

The separation sphere has a center to be determined, defined by an array of L intervals $I_{X_0}(1), \ldots, I_{X_0}(L)$ and a radius R. Let $c_{X_o(l)}$ and $r_{X_o(l)}$ be the center and radius of interval $I_{X_0}(l)$. Using the Mallows distance for the multivariate case with intervals

$$
D_M^2(\Psi_X^{-1}, \Psi_Y^{-1}) = \sum_{l=1}^{L} \left[(c_{X(l)} - c_{Y(l)})^2 + \frac{1}{3}(r_{X(l)} - r_{Y(l)})^2 \right], \ l = 1...L \tag{3.10}
$$

we have the following formulation for the Spherical Support Vector Machine for Multiple Interval (SSVMMI).

$$
\begin{aligned}
SSVMMI : \max\ & \delta \\
\text{s.t. } & \sum_{l=1}^{L} \left[(c_{X_i(l)} - c_{X_o(l)})^2 + \frac{1}{3}(r_{X_i(l)} - r_{X_o(l)})^2 \right] \leq (R - \delta)^2, \ \text{ for } i \in N_1 \\
& \sum_{l=1}^{L} \left[(c_{X_j(l)} - c_{X_o(l)})^2 + \frac{1}{3}(r_{X_j(l)} - r_{X_o(l)})^2 \right] \geq (R + \delta)^2, \ \text{ for } j \in N_2 \\
& R \geq \delta \\
& \delta \geq 0 \\
& r_{X_o(l)} \geq 0, \ l = 1...L.
\end{aligned} \tag{3.11}
$$

As in 3.7, identifying class membership using y_i labels we obtain:

$$SSVMMI : \max \delta$$

$$\text{s.t. } y_i \sum_{l=1}^{L} \left[(c_{X_i(l)} - c_{X_o(l)})^2 + \frac{1}{3}(r_{X_i(l)} - r_{X_o(l)})^2 \right] \leq y_i(R - y_i\delta)^2, \text{ for } i \in N \tag{3.12}$$

$$R \geq \delta$$

$$\delta \geq 0$$

$$r_{X_o(l)} \geq 0, \ l = 1...L.$$

3.6.3 Soft Margins in the Multiple features - SSSVMMI

If data, distributed in two classes C_1 and C_2, such that $N_1 = \{1, 2, \ldots, |C_1|\}$ and $N_2 = \{1, 2, \ldots, |C_2|\}$, is not separable, we have to consider the concept of soft margins and the introduction of slack variables ϵ_i and ϵ_j as follows (with parameter $\mu > 0$):

$$SSSVMMI : \max \delta - \mu(\sum_{i=1}^{n_1} \epsilon_i + \sum_{j=1}^{n_2} \epsilon_j), \ n_1 = \left|C_1\right|, \ n_2 = \left|C_2\right| \tag{3.13}$$

$$\text{s.t. } \sum_{l=1}^{L} \left[(c_{X_i(l)} - c_{X_o(l)})^2 + \frac{1}{3}(r_{X_i(l)} - r_{X_o(l)})^2 \right] \leq (R - \delta)^2 + \epsilon_i, \text{ for } i \in N_1 \tag{3.14}$$

$$\sum_{l=1}^{L} \left[(c_{X_j(l)} - c_{X_o(l)})^2 + \frac{1}{3}(r_{X_j(l)} - r_{X_o(l)})^2 \right] \geq (R + \delta)^2 - \epsilon_j, \text{ for } j \in N_2 \tag{3.15}$$

$$R \geq \delta$$

$$\delta \geq 0$$

$$r_{X_o(l)} \geq 0, \ l = 1...L$$

$$\epsilon_i \geq 0, \text{ for } i \in N_1$$

$$\epsilon_j \geq 0, \text{ for } j \in N_2.$$

3.6.4 A linear relaxation of the SSSVMMI

The quadratic Soft Margin Spherical SVM for Multiple Interval (SSSVMMI) problem is nonconvex, so for large scale problems a global optimal solution may be difficult to achieve computationally even with the best solvers. In that sense we developed a relaxation problem, from which a feasible solution for the original problem can be retrieved. The relaxation obtained is a linear programming problem and so can be efficiently solved even for large scale problems. The relaxation can also be used to find an upper bound for the optimal value. In a heuristic approach this value can be used to define optimality gaps.

The upper bound will be a key feature in a branch and bound approach in a future work. So, from (3.14) and (3.15) we get:

$$SSSVMMI : \max \delta - \mu(\sum_{i=1}^{n_1} \epsilon_i + \sum_{j=1}^{n_2} \epsilon_j),\ n_1 = \left|C_1\right|,\ n_2 = \left|C_2\right| \tag{3.16}$$

$$\text{s.t.} \quad \sum_{\ell=1}^{L} \left[c^2_{X_i(l)} - 2c_{X_i(l)}c_{X_0(l)} + c^2_{X_0(l)} + \frac{1}{3}(r^2_{X_i(l)} - 2r_{X_i(l)}r_{X_0(l)} + r^2_{X_0(l)}) \right]$$

$$\leq R^2 + \delta^2 - 2R\delta + \epsilon_i, \quad \text{for } i \in N_1 \tag{3.17}$$

$$\sum_{\ell=1}^{L} \left[c^2_{X_j(l)} - 2c_{X_j(l)}c_{X_0(l)} + c^2_{X_0(l)} + \frac{1}{3}(r^2_{X_j(l)} - 2r_{X_j(l)}r_{X_0(l)} + r^2_{X_0(l)}) \right]$$

$$\geq R^2 + \delta^2 + 2R\delta - \epsilon_j, \quad \text{for } j \in N_2 \tag{3.18}$$

$$R \geq \delta$$

$$\delta, \epsilon_i, \epsilon_j \geq 0, \quad \text{for } i \in N_1, j \in N_2.$$

And rearranging the inequations we get

$$SSSVMMI : \max \delta - \mu(\sum_{i=1}^{n_1} \epsilon_i + \sum_{j=1}^{n_2} \epsilon_j),\ n_1 = \left|C_1\right|,\ n_2 = \left|C_2\right| \tag{3.19}$$

$$\text{s.t.} \quad \sum_{\ell=1}^{L}(c^2_{X_i(l)} + \frac{1}{3}r^2_{X_i(l)}) + \sum_{\ell=1}^{n}(c^2_{X_0(l)} + \frac{1}{3}r^2_{X_0(l)}) - 2(\sum_{\ell=1}^{n} c_{X_i(l)}c_{X_0(l)} + \frac{1}{3}r_{X_i(l)}r_{X_0(l)})$$

$$\leq R^2 + \delta^2 - 2R\delta + \epsilon_i, \quad \text{for } i \in N_1 \tag{3.20}$$

$$\sum_{\ell=1}^{L}(c^2_{X_j(l)} + \frac{1}{3}r^2_{X_j(l)}) + \sum_{\ell=1}^{n}(c^2_{X_0(l)} + \frac{1}{3}r^2_{X_0(l)}) - 2(\sum_{\ell=1}^{n} c_{X_j(l)}c_{X_0(l)} + \frac{1}{3}r_{X_j(l)}r_{X_0(l)})$$

$$\geq R^2 + \delta^2 + 2R\delta - \epsilon_j, \quad \text{for } j \in N_2 \tag{3.21}$$

$$R \geq \delta$$

$$\delta, \epsilon_i, \epsilon_j \geq 0, \quad \text{for } i \in N_1, j \in N_2.$$

Let us introduce some notation that will help to simplify the writing of the model. We define:

$$\|x\|^2_M = \sum_{\ell=1}^{L}(c^2_{X(l)} + \frac{1}{3}r^2_{X(l)}) \tag{3.22}$$

$$(x^T y)_M = \sum_{\ell=1}^{L} c_{X(l)}c_{Y(l)} + \frac{1}{3}\sum_{\ell=1}^{L} r_{X(l)}r_{Y(l)} \tag{3.23}$$

and so from (3.20) and (3.21), taking $\theta = R\delta$ we have (with parameter $\mu > 0$):

$$SSSVMMI(\theta): \max \delta - \mu(\sum_{i=1}^{n_1} \epsilon_i + \sum_{j=1}^{n_2} \epsilon_j),\ n_1 = \left|C_1\right|,\ n_2 = \left|C_2\right| \tag{3.24}$$

$$\text{s.t. } \|x_i\|_M^2 - 2(x_i^T x_0)_M + 2\theta - \epsilon_i \leq R^2 + \delta^2 - \|x_0\|_M^2, \text{ for } i \in N_1 \tag{3.25}$$

$$\|x_j\|_M^2 - 2(x_j^T x_0)_M - 2\theta + \epsilon_j \geq R^2 + \delta^2 - \|x_0\|_M^2, \text{ for } j \in N_2 \tag{3.26}$$

$$R \geq \delta \tag{3.27}$$

$$\delta \geq 0 \tag{3.28}$$

$$\epsilon_i \geq 0, \text{ for } i \in N_1 \tag{3.29}$$

$$\epsilon_j \geq 0, \text{ for } j \in N_2. \tag{3.30}$$

Now, since the right sides of both inequations 3.25 and 3.26 are the same, we have

$$\|x_i\|_M^2 - 2(x_i^T x_0)_M + 2\theta - \epsilon_i \leq R^2 + \delta^2 - \|x_0\|_M^2 \leq \|x_j\|_M^2 - 2(x_j^T x_0)_M - 2\theta + \epsilon_j, \text{ for } i \in N_1, j \in N_2 \tag{3.31}$$

Because each feasible solution of $SSSVMMI$ fulfils condition

$$\|x_i\|_M^2 - 2(x_i^T x_0)_M + 2\theta - \epsilon_i \leq \|x_j\|_M^2 - 2(x_j^T x_0)_M - 2\theta + \epsilon_j, \text{ for } i \in N_1, j \in N_2 \tag{3.32}$$

we have that

$$4\theta + 2(x_j^T x_0)_M - 2(x_i^T x_0)_M - \epsilon_j - \epsilon_i \leq \|x_j\|_M^2 - \|x_i\|_M^2, \text{ for } i \in N_1, j \in N_2 \tag{3.33}$$

equivalent to:

$$4\theta + 2[(x_j - x_i)^T x_0]_M - \epsilon_j - \epsilon_i \leq \|x_j\|_M^2 - \|x_i\|_M^2, \text{ for } i \in N_1, j \in N_2 \tag{3.34}$$

So, we come to the linear relaxation problem formulation as follows

$$LSSSVMMI_0: \max \theta - \mu(\sum_{i=1}^{n_1} \epsilon_i + \sum_{j=1}^{n_2} \epsilon_j),\ n_1 = \left|C_1\right|,\ n_2 = \left|C_2\right|$$

$$\text{s.t. } 4\theta + 2[(x_j - x_i)^T x_0]_M - \epsilon_j - \epsilon_i \leq \|x_j\|_M^2 - \|x_i\|_M^2, \text{ for } i \in N_1, j \in N_2$$

$$\theta \geq 0$$

$$\epsilon_i \geq 0, \text{ for } i \in N_1$$

$$\epsilon_j \geq 0, \text{ for } j \in N_2.$$

Because this problem may be unbounded, as shown in 2.3.3, we use an indirect regularization technique introducing a new variable ρ and a new constraint in the above

formulation (with parameters $\mu > 0$ and $k > 0$)

$$
\begin{aligned}
LSSSVMMI : \max\ & \theta + k\rho - \mu(\sum_{i=1}^{n_1} \epsilon_i + \sum_{j=1}^{n_2} \epsilon_j),\ n_1 = \left|C_1\right|,\ n_2 = \left|C_2\right| \\
\text{s.t. } & 4\theta + 2[(x_j - x_i)^T x_0]_M - \epsilon_j - \epsilon_i \leq \|x_j\|_M^2 - \|x_i\|_M^2,\ \text{ for } i \in N_1, j \in N_2 \\
& 4\theta + 2[(x_j - x_i)^T x_0]_M - \epsilon_j - \epsilon_i \geq \rho,\ \text{ for } i \in N_1, j \in N_2 \\
& \theta \geq 0 \\
& \rho \geq 0 \\
& \epsilon_i \geq 0,\ \text{ for } i \in N_1 \\
& \epsilon_j \geq 0,\ \text{ for } j \in N_2.
\end{aligned}
$$

The solution of this linear problem is $(\theta, \rho, x_0, \epsilon)$. Therefore, we still have to find R and δ and for that we have two possible ways, which we will describe below.
Since we have optimum x_0, to get optimum R and δ, we can solve the following linear problem (with $\mu > 0$):

$$
\begin{aligned}
SRLI :\ & \max \delta - \mu(\sum_{i=1}^{n_1} \varphi_i + \sum_{j=1}^{n_2} \varphi_j),\ n_1 = \left|C_1\right|,\ n_2 = \left|C_2\right| && (3.35) \\
\text{s.t. } & R - \delta + \varphi_i \geq \|x_i - x_{0_{optim}}\|_M \text{ for } i \in N_1, j \in N_2 && (3.36) \\
& R + \delta - \varphi_j \leq \|x_j - x_{0_{optim}}\|_M,\ \text{for } i \in N_1, j \in N_2 && (3.37) \\
& R \geq \delta \\
& \delta \geq 0 \\
& \varphi_i \geq 0,\ \text{for } i \in N_1 \\
& \varphi_j \geq 0,\ \text{for } j \in N_2
\end{aligned}
$$

In alternative to this procedure, and considering that at the support vectors, we have:

$$
\theta_{optim} + 2[(x_{j_{sv}} - x_{i_{sv}})^T x_{0_{optim}}]_M - \epsilon_{i_{sv}} - \epsilon_{j_{sv}} = \|x_{j_{sv}}\|_M^2 - \|x_{i_{sv}}\|_M^2,\ \text{ for } i \in N_1, j \in N_2
$$

After finding the support vectors, we calculate R and δ.

$$
\begin{aligned}
R &= \frac{\sqrt{\|x_{j_{sv}} - x_{0_{optim}}\|_M^2 + \epsilon_{j_{sv}}} + \sqrt{\|x_{i_{sv}} - x_{0_{optim}}\|_M^2 - \epsilon_{i_{sv}}}}{2} \\
\delta &= \frac{\sqrt{\|x_{j_{sv}} - x_{0_{optim}}\|_M^2 + \epsilon_{j_{sv}}} - \sqrt{\|x_{i_{sv}} - x_{0_{optim}}\|_M^2 - \epsilon_{i_{sv}}}}{2}
\end{aligned}
$$

3.7 Computational Experience

The goal of the computational experience is to test the classification accuracy of the quadratic SSSVMMI and the Soft Margin Spherical SVM Linear Relaxation for Multiple Interval (LSSSVMMI) models by comparing them against the accuracy results obtained with all the 34 classifiers available in MATLAB's R2024a Classification Learner App (Table I.1), for real-world and generated datasets.
Experiments were run on a computer with:
- Intel(R) Core(TM) i7-8550U processor, 1.80GHz
- 8.00 GB of RAM
- 64-bit operating system running on Windows 10, version 20H2.
- We code our method on Matlab R2024a, and we used BARON software, version 13.0.1, running on Matlab with MATLAB/BARON interface version v1.89.
- The BARON solver was used to solve the quadratic formulation.
- To run all 34 Matlab classifiers we used the Machine Learning and Deep Learning APP from MATLAB. We used three types of interval-valued datasets. The properties of these datasets are summarized in Table 3.2.

Table 3.2: Attributes and Number of Data Set Samples

Data Set	Attributes	class1	class 2
Sim1	1	10	10
Sim2	1	10	10
Iris	4	50	50
Thyroid	5	150	35
Seeds	7	70	140
Twonorm	20	200	200
Mushroom	5	30	20

Sim1 and Sim2 are simulated interval valued datasets.
Sim1 was built as follows: for class 1 we made a 10x10 array of random real numbers taken from a Gaussian distribution N(7,1), then for each row we took the minimum value for the lower interval limit and the maximum value for the upper interval limit. For class 2 the procedure is the same but with real numbers taken from N(13,1). So we have two classes with different average values and similar variance values. In figure 3.3 we can see the quantil functions of this data set.
Sim2 was built the same way as Sim1 but for class 1 we used a 10x10 array of random real numbers taken from N(6,1) and for class 2, real numbers taken from N(6,3). So we have two classes with similar average values and different variance values. In figure 3.4 we can see the quantil functions of this data set.
The Mushroom dataset was extracted from site - The Funji of California [95], and is a real-world interval-valued dataset.

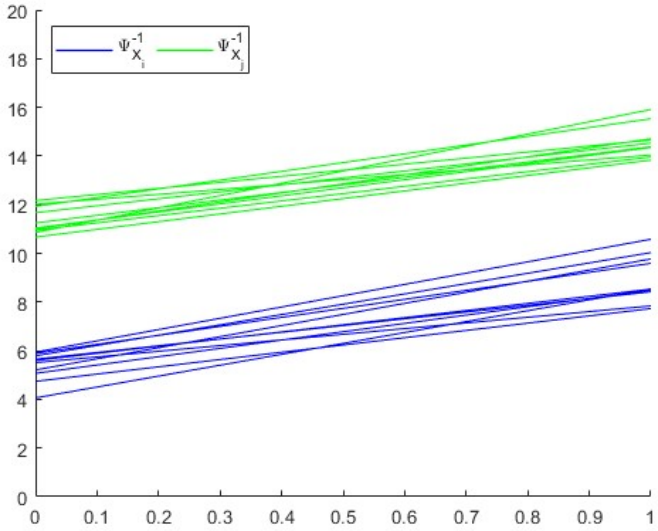

Figure 3.3: Sim1 Quantile functions

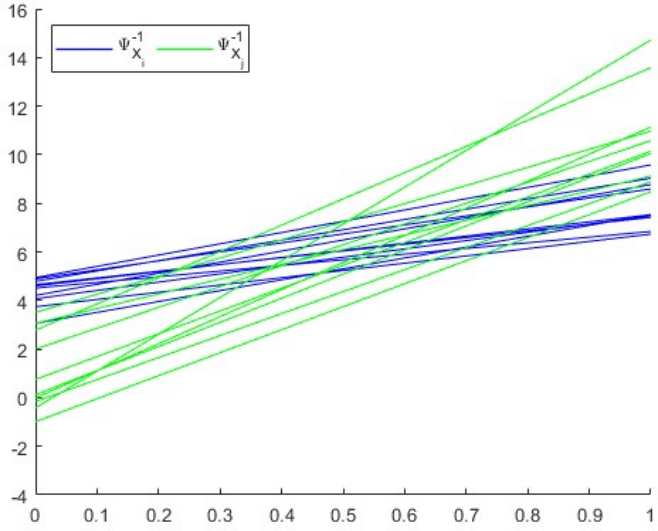

Figure 3.4: Sim2 Quantile functions

It contains 50 species of two fungi genera, 30 of genera Agaricus and 20 of genera Amanita. There are five interval-valued features: the pileus cap width Pw, the stipe length Sl, the stipe thickness St, the spores major axis length Sma, and the spores minor axis length Smi. Some instances of the mushroom dataset are shown in Table 3.3. The goal of our experiment on this dataset is to predict the genera.

Iris, Thyroid, Seeds and Twonorm interval-valued datasets were constructed from real-world numerical datasets available from UCI Machine Learning Repository.

The procedure to get interval-valued data from numerical data is as follows:

The lower and upper limit of intervals, for each feature, were obtained as reducing or

Table 3.3: Mushroom interval-valued data

Genera	Species	Pw (cm)	Sl (cm)	St (cm)	Sma (cm)	Smi (μm)
Agaricus	Moronii	[6; 12]	[2; 7]	[1.5; 3]	[6; 7.5]	[4; 5]
Agaricus	Arorae	[3; 8]	[4; 9]	[0.5; 2.5]	[4.5; 5]	[3; 3.5]
Agaricus	Fissuratus	[6; 21]	[4; 14]	[1; 3.5]	[6.5; 9]	[4.5; 6]
Amanita	Augusta	[4; 12]	[5; 15]	[1; 2]	[8; 12]	[6; 8]
Amanita	Calyptroderma	[8; 25]	[10; 20]	[1.5; 4]	[8; 11]	[5; 6]

increasing the original numerical data by a random value between zero and 10 percent of the average of all values for that feature.
From the three classes of Iris dataset we used only two, versicolor and virginica.
In Table 3.4 we show some instances of the new generated interval-valued Iris dataset
Figure 3.5 provides the intervals for two features (petal lenght and petal width) of classes

Table 3.4: Iris Interval-valued Data

	sepal length	sepal width	petal length	petal width
Iris versicolor 1	[6.49; 7.57]	[3.16; 3.46]	[4.39; 4.75]	[1.35; 1.49]
Iris versicolor ...	...	...	...	...
Iris versicolor 50	[5.19; 5.86]	[2.63; 2.81]	[3.89; 4.25]	[1.27; 1.33]
Iris virginica 1	[6.04; 6.36]	[3.13; 3.44]	[5.66; 6.34]	[2.39; 2.51]
Iris virginica ...	...	...	..	...
Iris virginica 50	[5.34; 6.40]	[2.79; 3.01]	[5.06; 5.14]	[1.67; 1.96]

versicolor and virginica from Iris dataset.
For comparision, three methods of validation were applied to the datasets: 5-fold cross-validation; classification using the complete set; train with 90% test with 10% only one run.
For classification with Matlab we used the mid point of each interval-valued data. The accuracy results from the 34 Matlab classifiers were condensed in intervals in which the lower limit refers to the minimum accuracy and the upper limit to the maximum accuracy value obtained. Also for SSSVMMI and LSSSVMMI we calculate the gap from the best Matlab accuracy value.
The accuracy results obtained are summarized in Tables 3.5, 3.6 and 3.7.

Using 5-fold cross-validation, the gap to the best Matlab classifier accuracy varies from 0.98% to 5.21% with model SSSVMMI and from 1.98% to 10.64 with model LSSSVMMI.

With complete set, we got 100% accuracy in five datasets.

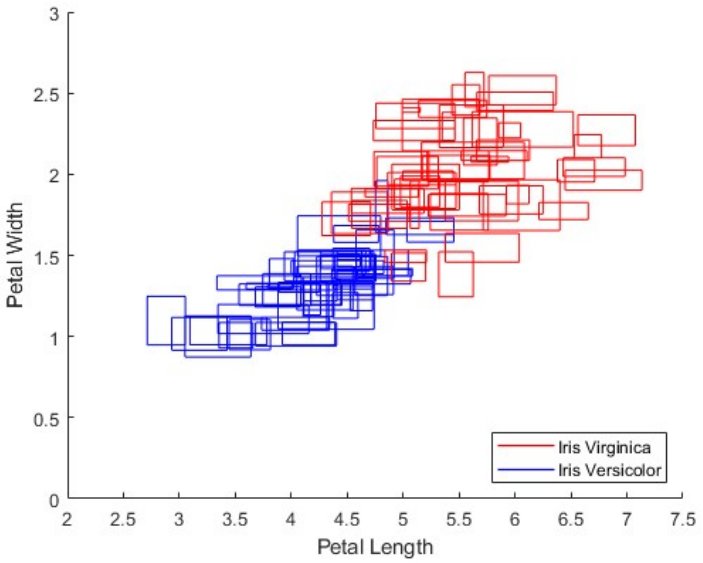

Figure 3.5: Iris intervals (two features)

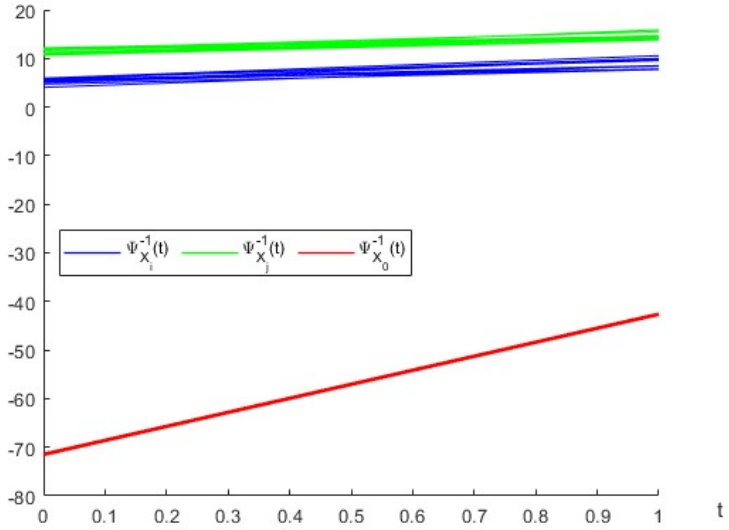

Figure 3.6: Sim1 Quantile functions with classifier

The 20% gap in mushroom dataset is due to the fact that we used a very few number of instances (only 5) for the test set and therefore one misclassified instance represents 20%.

3.8 Results Discussion

In this chapter, we developed a model inspired by SVM, with spherical separation, with the advantage that the separation depends on a center whose structure is an interval, and a radius. Defining a hyperplane presents significant challenges, because the linear

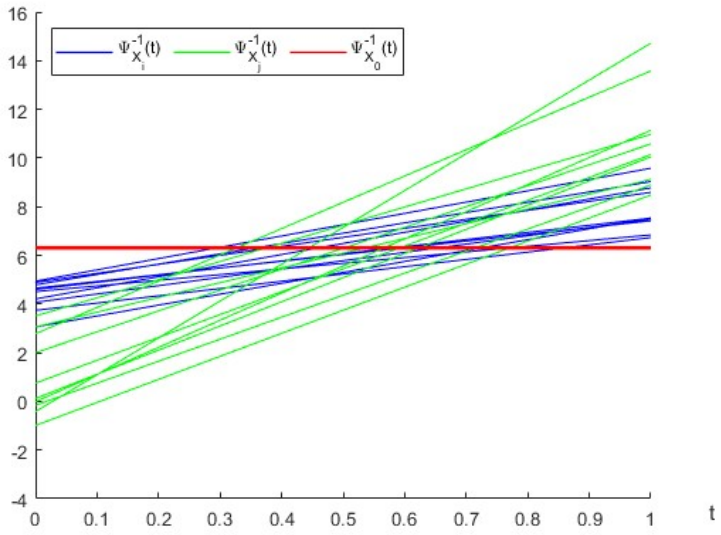

Figure 3.7: Sim2 Quantile functions with classifier

Table 3.5: 5-fold cross validation accuracy results

5-fold cross validation					
Data Set	Accuracy%			Gap%	
	MATLAB Classifiers	SSSVMMI	LSSSVMMI	SSSVMMI	LSSSVMMI
Sim1	[50,00 ; 100,00]	100.00	100.00	0	0
Sim2	[50,00 ; 65,00]	100.00	95.00	-	-
Iris	[50,00 ; 96,00]	91.00	92.00	5.21	4.17
Seeds	[66.67 ; 96,67]	95.72	94.76	0.98	1.98
Thyroid	[81.08 ; 98.92]	98.38	94.05	3.91	3.91
Twonorm	[69.00 ; 98.25]	96.50	94.25	1.78	4.07
Mushrooms	[60.00 ; 94.00]	90.00	84.00	4.26	10.64

Table 3.6: Complete set accuracy results

complete set validation					
Data Set	Accuracy%			Gap%	
	MATLAB Classifiers	SSSVMMI	LSSSVMMI	SSSVMMI	LSSSVMMI
Sim1	[50.00 ; 100.00]	100.00	100.00	0	0
Sim2	[50.00 ; 100.00]	100.00	100.00	0	0
Iris	[50.00 ; 100.00]	99.00	95.00	1.00	5.00
Seeds	[66.67 ; 100.00]	93.81	93.33	6.19	6.67
Thyroid	[81.08 ; 100.00]	100.00	98.92	0.00	1.08
Twonorm	[56.25 ; 100.00]	100.00	97.25	0.00	2.75
Mushrooms	[60.00 ; 100.00]	100.00	94.00	0.00	6.00

combination of quantile functions is not a quantile function. A quantile function (which is non-decreasing) multiplied by a negative coefficient is no longer a quantile function. Defining a hypersphere centered on an interval avoids this problem. The suggested model

Table 3.7: 90% train 10% test accuracy results

90% train 10% test one run					
Data Set	Accuracy%			Gap%	
	MATLAB Classifiers	SSSVMMI	LSSSVMMI	SSSVMMI	LSSSVMMI
Iris	[50.00 ; 90.00]	90.00	90.00	0.00	0.00
Seeds	[66.67 ; 100.00]	95.24	95.24	4.76	4.76
Thyroid	[78.95 ; 100.00]	94.74	100.00	2.26	0.00
Twonorm	[65.00 ; 100.00]	100.00	92.50	0.00	7.50
Mushrooms	[60.00 ; 100.00]	80.00	80.00	20.00	20.00

is inspired by SVM with the definition of a margin (which is maximized) and soft margins. This model is a quadratic optimization model, and we developed a linear relaxation of it. As in the model for classical data, the solution to the linear relaxation allows for defining a solution to the original model.

The results of this model were compared with several classification models in Matlab. Both simulated and real datasets, commonly used in classification literature, were tested. Of the instances used, only one was truly defined by intervals. For the rest, we applied a methodology followed by authors of interval classification models, which consists of transforming classical data into intervals. The results were very satisfactory, especially considering that the model has no more than a single multi-parameter (to account for deviations and margin in the objective function) and allows for working within the original data space.

4

Conclusions and Future Work

The challenges posed by machine learning are numerous. Alongside the virtue of possible applications in finance, medicine, engineering, management, and many other fields, comes the responsibility to address the potential dangers of its use. In particular, in sensitive areas, it is necessary to deal with possible bias in the developed model. Black-box models have had enormous success in certain areas, but they make it impossible to unequivocally determine and understand what defines the decision boundary of a classification model. For this reason, in this thesis, an effort was made to study ways to apply separation models that are explainable and interpretable, and for which it is possible to establish how variations in the data can lead to an opposite outcome. For example, if a loan application is denied based on the response of a classification model, it should be communicated to the involved person what changes in the application would lead to a successful outcome. This property is known as counterfactual analysis.

Deep learning and random forest models do not enable applying techniques to understand the internal classification mechanisms. The structure created by the model is too complex to allow for a reverse interpretation, from the output back to the input. In addition, other factors that hinder the interpretability and explainability of the model include the existence of many parameters and the use of kernel functions, which shift the data from its original space. This fact was one of the motivations for exploring new approaches to Support Vector Machine (SVM) models that can be developed in the original data space. The idea of using a curve defined by a hypersphere instead of a hyperplane came from the fact that, to define the hypersphere, only the center (whose structure is exactly the same as the data) and a radius (which is a positive real number) are needed. This facilitates application of the model in the original data space, which proved particularly useful for interval-based applications. This was the main motivation for this work, along with the fact that SVMs allow for the exploration of interesting optimization techniques. A motivation, from an academic pont of view, is the investigation of the properties of these models.

In the first phase, this method was explored for classical data, defined by tables of values. The mathematical formulation of this model involves defining a non-convex quadratic

problem, and for this reason, an effort was made to define a relaxation of this model, which is easier to solve and whose solution could be transferred to the original model. This model was tested in comparison to classical models defined in Matlab's automatic classification package. Next, the model was applied to symbolic data. Symbolic data are defined by structures such as intervals or histograms. This new paradigm was proposed in the 1980s by Diday and gained much greater prominence from the second half of the 21st century with the advent of Big Data.

Symbolic data, such as histograms and intervals, allow for a more suitable description of the data. For example, assuming that for a sample of various patients, we have a reading of certain biometric data for each individual, such as blood pressure, heart rate, etc., in order to use a classic classification model, this data vector is converted into an average. However, it would be more expressive to consider the range of values or a data histogram instead of just the average. Working with symbolic data presents greater challenges than with classical data, but the scientific community has made several proposals for the development of classic models for data analysis, regression, clustering, discriminant analysis, and others. In general, these methods are applied to quantile functions, which are defined by the inverse of the cumulative distribution function. In case of a histogram, this function is defined over [0,1] and is a piecewise linear function. For intervals, this function is a straight line segment. To define the distance between points, the Mallows distance was used.

In this thesis, we applied a model inspired by SVM, with spherical separation, with the advantage that the separation depends on a center whose structure is an interval and a radius. Defining a hyperplane would present significant challenges, because the linear combination of quantile functions is not a quantile function. A quantile function (which is non-decreasing) multiplied by a negative coefficient is no longer a quantile function. Defining a hypersphere centered on an interval allows us to overcome this problem. The suggested model is inspired by SVM with the definition of a margin (which is maximized) and soft margins. This model is a quadratic optimization model, and we developed a linear relaxation of it. As in the model for classical data, the solution to the linear relaxation allows for defining a solution to the original model.

The results of this model were compared with several classification models in Matlab. Both simulated and real datasets, commonly used in classification literature, were used in testing the developed method. From the instances used, only one was truly defined by intervals. For the rest, we applied a methodology followed by authors of interval classification models, which consists of transforming classical data into intervals. The results were very satisfactory, especially considering that the model has no more than a single multi-parameter (to account for deviations and margin in the objective function) and allows for working within the original data space.

Thus, we can conclude that the objective of this work was successfully achieved, and it constitutes a valid proposal for continuing the development of models we refer to as White Box, in contrast to black-box models.

As future work, we intend to apply this method to histograms. For this structure, some significant modifications will need to be made. We also aim to explore proposals for counterfactual analysis and discuss in-depth issues related to interpretability and explainability. Another proposal for development is the construction of methods for global optimization of the quadratic problem, based on upper bounds derived from copositive reformulations of the problem.

To conclude, I would like to express that this journey of developing a doctoral thesis that started to be a big challenge, has been very rewarding. From the personal point of view, it showed me that we can really achieve our objectives if we don´t give up. As academic, all that involved the research for these new models led me to greatly enlarge the horizons of my knowledge and is an incentive to continue research.

Bibliography

[1] .

[2] H. Abdi and L. J. Williams. "Principal component analysis". In: *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010), pp. 433–459 (cit. on p. 5).

[3] S. Abe. *Support Vector Machines for Pattern Classification*. Springer-Verlag London, 2005-01, pp. 400–473. ISBN: 978-1-85233-929-6. DOI: 10.1007/1-84628-219-5 (cit. on pp. 17, 40).

[4] O. I. Abiodun et al. "State-of-the-art in artificial neural network applications: A survey". In: *Heliyon* 4.11 (2018) (cit. on p. 16).

[5] E. Alpaydin. *Machine learning*. MIT press, 2021 (cit. on p. 2).

[6] Q. An et al. "A Comprehensive Review on Machine Learning in Healthcare Industry: Classification, Restrictions, Opportunities and Challenges". In: *Sensors* 23.9 (2023). ISSN: 1424-8220. DOI: 10.3390/s23094178. URL: https://www.mdpi.com/1424-8220/23/9/4178 (cit. on pp. 4, 12, 17).

[7] J. Arroyo and C. Maté. "Forecasting histogram time series with k-nearest neighbours methods". In: *International Journal of Forecasting* 25.1 (2009), pp. 192–207. URL: https://EconPapers.repec.org/RePEc:eee:intfor:v:25:y:2009:i:1:p:192-207 (cit. on pp. 44, 46).

[8] A. Astorino, A. Fuduli, and M. Gaudioso. "A fixed-center spherical separation algorithm with kernel transformations for classification problems". In: *Journal of Global Optimization* 48 (2010), pp. 657–669. DOI: 10.1007/s10898-010-9558-0 (cit. on p. 27).

[9] A. Astorino, A. Fuduli, and M. Gaudioso. "Margin maximization in spherical separation". In: *Computational Optimization and Applications* 53 (2012), pp. 301–322. DOI: 10.1007/s10589-012-9486-7 (cit. on p. 27).

[10] A. Astorino and M. Gaudioso. "A fixed-center spherical separation algorithm with kernel transformations for classification problems". In: *Computational Management Science* 6 (2009), pp. 357–372. DOI: 10.1007/s10287-007-0051-2 (cit. on p. 27).

[11] S. Badillo et al. "An introduction to machine learning". In: *Clinical pharmacology & therapeutics* 107.4 (2020), pp. 871–885 (cit. on p. 19).

[12] J. j. Berman. *Logic and critical thinking in biomedical sciences*. 2020 (cit. on p. 41).

[13] P. Bertrand and F. Goupil. "Descriptive statistics for symbolic data". In: *Analysis of symbolic data*. Springer, 2000, pp. 106–124 (cit. on p. 42).

[14] L Billard and E Diday. *Symbolic Data Analysis: Conceptual Statistics and Data Mining John Wiley*. 2006 (cit. on p. 42).

[15] L. Billard. "Symbolic data analysis: what is it?" In: *Compstat 2006-Proceedings in Computational Statistics: 17th Symposium Held in Rome, Italy, 2006*. Springer. 2006, pp. 261–269 (cit. on p. 20).

[16] H. Bock and E. Diday. *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. 2000-01. DOI: 10.1007/978-3-642-57155-8 (cit. on p. 46).

[17] I. Bomze et al. "Feature selection in linear SVMs via hard cardinality constraint: a scalable SDP decomposition approach". In: *arXiv preprint arXiv:2404.10099* (2024) (cit. on p. 40).

[18] J. Burkardt. "K-means clustering". In: *Virginia Tech, Advanced Research Computing, Interdisciplinary Center for Applied Mathematics* (2009) (cit. on p. 5).

[19] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso. "Machine learning interpretability: A survey on methods and metrics". In: *Electronics* 8.8 (2019), p. 832 (cit. on p. 18).

[20] M. A. Chandra and S. Bedi. "Survey on SVM and their application in image classification". In: *International Journal of Information Technology* 13.5 (2021), pp. 1–11 (cit. on p. 40).

[21] P. W. Cooper. "The hypersphere in pattern recognition". In: *Information and Control* 5.4 (1962), pp. 324–346. ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(62)90641-1. URL: https://www.sciencedirect.com/science/article/pii/S0019995862906411 (cit. on p. 26).

[22] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine Learning* 20 (1995), pp. 273–297. DOI: 10.1007/BF00994018. URL: https://doi.org/10.1007/BF00994018 (cit. on pp. 17, 26).

[23] V. Costa and C. Pedreira. "Recent advances in decision trees an updated survey". In: *Artificial Intelligence Review* 56.5 (2023), pp. 4765–4800 (cit. on p. 14).

[24] P. Cunningham, M. Cord, and S. J. Delany. "Supervised learning". In: *Machine learning techniques for multimedia: case studies on organization and retrieval*. Springer, 2008, pp. 21–49 (cit. on p. 5).

[25] P. Cunningham and S. J. Delany. "K-nearest neighbour classifiers-a tutorial". In: *ACM computing surveys (CSUR)* 54.6 (2021), pp. 1–25 (cit. on p. 11).

[26] B. De Ville. "Decision trees". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 5.6 (2013), pp. 448–455 (cit. on p. 13).

[27] S. Dias. "Linear Regression With Empirical Distributions". PhD thesis. University of Porto, 2014 (cit. on p. 43).

[28] S. Dias and P. Brito. "Linear Regression Model with Histogram-Valued Variables". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8.2 (2015), pp. 75–113. URL: https://ci.nii.ac.jp/naid/10026575900/en/ (cit. on pp. 42, 43, 45, 46).

[29] S. Dias, P. Brito, and P. Amaral. "Discriminant analysis of distributional data via fractional programming". In: *European Journal of Operational Research* 294.1 (2021), pp. 206–218 (cit. on pp. 42, 43, 46).

[30] E. Diday. "The symbolic approach in clustering and related methods of Data Analysis". In: *Proceedings of IFCS, Classification and Related Methods of Data Analysis, 1988* (1988), pp. 673–384. URL: https://ci.nii.ac.jp/naid/10026575900/en/ (cit. on p. 20).

[31] T. G. Dietterich. "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems*. Springer. 2000, pp. 1–15 (cit. on p. 16).

[32] T. Dietterich. "Overfitting and undercomputing in machine learning". In: *ACM computing surveys (CSUR)* 27.3 (1995), pp. 326–327 (cit. on p. 8).

[33] F. Doshi-Velez and B. Kim. "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608* (2017) (cit. on pp. 18, 19).

[34] A. P. Duarte Silva and P. Brito. "Linear Discriminant Analysis for Interval Data". In: *Computational Statistics* 21.2 (2006), pp. 289–308 (cit. on pp. 42, 43).

[35] F. D'Onofrio et al. "Margin optimal classification trees". In: *Computers & Operations Research* 161 (2024), p. 106441 (cit. on p. 40).

[36] B. Fallah and S. Sodoudi. "Bimodality and regime behavior in atmosphere–ocean interactions during the recent climate change". In: *Dynamics of Atmospheres and Oceans* 70 (2015), pp. 1 –11. ISSN: 0377-0265. DOI: https://doi.org/10.1016/j.dynatmoce.2015.02.002. URL: http://www.sciencedirect.com/science/article/pii/S0377026515000135 (cit. on p. 41).

[37] M. Fathi et al. "A machine learning approach based on SVM for classification of liver diseases". In: *Biomedical Engineering: Applications, Basis and Communications* 32.03 (2020), p. 2050018 (cit. on p. 40).

[38] E. Fix and J. Hodges. *Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties*. USAF School of Aviation Medicine, 1951. URL: https://books.google.pt/books?id=4XwytAEACAAJ (cit. on p. 11).

[39] Z. Ghahramani. "Unsupervised learning". In: *Summer school on machine learning*. Springer, 2003, pp. 72–112 (cit. on p. 5).

[40] R. Gong, C. Wu, and M. Chu. "Multi-class Classification Method Based on Support Vector Machine with Hyper-sphere for Steel surface Defects". In: *2018 37th Chinese Control Conference (CCC)*. 2018, pp. 9197–9202. DOI: 10.23919/ChiCC.2018.8483656 (cit. on p. 27).

[41] R. Gong, C. Wu, and M. Chu. "Steel surface defect classification using multiple hyper-spheres support vector machine with additional information". In: *Chemometrics and Intelligent Laboratory Systems* 172 (2017-12). DOI: 10.1016/j.chemolab.2017.11.018 (cit. on p. 27).

[42] R. Gong et al. "Twin pinball loss support vector hyper-sphere classifier for pattern recognition". In: *2016 Chinese Control and Decision Conference (CCDC)*. 2016, pp. 6551–6556. DOI: 10.1109/CCDC.2016.7532177 (cit. on p. 27).

[43] P.-Y. Hao, J.-H. Chiang, and Y.-H. Lin. "A new maximal-margin spherical-structured multi-class support vector machine". In: *Applied Intelligence* 30.2 (2009), pp. 98–111. DOI: 10.1007/s10489-007-0101-z (cit. on pp. 27, 40).

[44] Y. Huang. "Deep Q-networks". In: *Deep reinforcement learning: fundamentals, research and applications* (2020), pp. 135–160 (cit. on p. 7).

[45] A. Irpino and R. Verde. "A new Wasserstein based distance for the hierarchical clustering of histogram symbolic data". In: 2006, pp. 185–192 (cit. on pp. 44, 46, 47).

[46] A. Irpino and R. Verde. "Dimension reduction techniques for distributional symbolic data". In: *Advances in Latent Variables-Methods, Models and Applications*. 2013 (cit. on p. 42).

[47] A. Irpino and R. Verde. "Basic statistics for distributional symbolic variables: a new metric-based approach". In: *Advances in Data Analysis and Classification* 9.2 (2015-06), pp. 143–175. ISSN: 1862-5355. DOI: 10.1007/s11634-014-0176-4. URL: https://doi.org/10.1007/s11634-014-0176-4 (cit. on pp. 42, 46).

[48] C. Janiesch, P. Zschech, and K. Heinrich. "Machine learning and deep learning". In: *ELECTRONIC MARKETS* 31.3 (2021-09), pp. 685–695. ISSN: 1019-6781. DOI: 10.1007/s12525-021-00475-2 (cit. on p. 3).

[49] L. P. Kaelbling, M. L. Littman, and A. W. Moore. "Reinforcement learning: A survey". In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285 (cit. on p. 6).

[50] S. Ketabchi, H. Moosaei, and M. Razzaghi. "Linear approach for twin-Hypersphere support vector machine". In: *Advanced Modeling and Optimization* 19 (2017-01), pp. 79–85 (cit. on p. 27).

[51] P. Klimek et al. "Statistical detection of systematic election irregularities". In: *Natl Acad Sci U S A*. Vol. 109. 41. 2012, pp. 16469–16473 (cit. on p. 41).

[52] N. Kriegeskorte and T. Golan. "Neural network models and deep learning". In: *Current Biology* 29.7 (2019), R231–R236 (cit. on p. 15).

[53] M. P. LaValley. "Logistic regression". In: *Circulation* 117.18 (2008), pp. 2395–2399 (cit. on p. 12).

[54] T. Ledzinski and G. Grzesk. "Artificial Intelligence Technologies in Cardiology". In: *Journal of Cardiovascular Development and Disease* 10.5 (2023). ISSN: 2308-3425. DOI: 10.3390/jcdd10050202. URL: https://www.mdpi.com/2308-3425/10/5/202 (cit. on p. 8).

[55] H. Li et al. "Bimodal Distribution and Co-Bursting in Review Spam Detection". In: *Proceedings of the 26th International Conference on World Wide Web*. WWW '17. Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, 1063–1072. ISBN: 9781450349130. DOI: 10.1145/3038912.3052582. URL: https://doi.org/10.1145/3038912.3052582 (cit. on p. 41).

[56] T. Li et al. "Prospectivity and Uncertainty Analysis of Tungsten Polymetallogenic Mineral Resources in the Nanling Metallogenic Belt, South China: A Comparative Study of AdaBoost, GBDT, and XgBoost Algorithms". In: *Natural Resources Research* 33.3 (2024), pp. 1049–1071 (cit. on p. 16).

[57] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. "Explainable ai: A review of machine learning interpretability methods". In: *Entropy* 23.1 (2020), p. 18 (cit. on p. 4).

[58] P. Liu, D. Zhou, and N. Wu. "VDBSCAN: varied density based spatial clustering of applications with noise". In: *2007 International conference on service systems and service management*. IEEE. 2007, pp. 1–4 (cit. on p. 5).

[59] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[60] C. M et al. "Support vector machine with quantile hyperspheres for pattern classification". In: *PLoS ONE* 14.2 (2019). DOI: 10.1371 (cit. on pp. 27, 28).

[61] G. Ma et al. "Multiview Classification Through Learning From Interval-Valued Data". In: *IEEE Transactions on Neural Networks and Learning Systems* (2024), pp. 1–15. DOI: 10.1109/TNNLS.2024.3421657 (cit. on p. 43).

[62] R. Malha and P. Amaral. "A Maximal Margin Hypersphere SVM". In: *International Conference on Computational Science and Its Applications*. Springer. 2021, pp. 304–319 (cit. on p. 40).

[63] MathWorks. *Machine Learning Cross Validation*. 2024. URL: https://www.mathworks.com/discovery/cross-validation.html (visited on 2024-09-28) (cit. on p. 9).

[64] L. R. Medsker, L. Jain, et al. "Recurrent neural networks". In: *Design and Applications* 5.64-67 (2001), p. 2 (cit. on p. 15).

[65] U. Michelucci. "An introduction to autoencoders". In: *arXiv preprint arXiv:2201.03898* (2022) (cit. on p. 6).

[66] A. Momenikorbekandi and M. Abbod. "Intelligent Scheduling Based on Reinforcement Learning Approaches: Applying Advanced Q-Learning and State-Action-Reward-State-Action Reinforcement Learning Models for the Optimisation of Job Shop Scheduling Problems". In: *Electronics* 12.23 (2023), p. 4752 (cit. on p. 7).

[67] A. Natekin and A. Knoll. "Gradient boosting machines, a tutorial". In: *Frontiers in neurorobotics* 7 (2013), p. 21 (cit. on p. 11).

[68] F. Nielsen and F. Nielsen. "Hierarchical clustering". In: *Introduction to HPC with MPI for Data Science* (2016), pp. 195–211 (cit. on p. 5).

[69] A. Nigrin. *Neural networks for pattern recognition*. MIT press, 1993 (cit. on p. 14).

[70] OECD. *Artificial Intelligence, Machine Learning and Big Data in Finance*. 2021, p. 72. DOI: https://doi.org/https://doi.org/10.1787/98e761e7-en. URL: https://www.oecd-ilibrary.org/content/publication/98e761e7-en (cit. on p. 2).

[71] K O'Shea. "An introduction to convolutional neural networks". In: *ariv preprint ariv:1511.08458* (2015) (cit. on p. 15).

[72] A. Parmar, R. Katariya, and V. Patel. "A review on random forest: An ensemble classifier". In: *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*. Springer. 2019, pp. 758–763 (cit. on p. 11).

[73] P. Patel. "ARTIFICIAL INTELLIGENCE/ROBOTICS: CHALLENGES AND PROSPECTS". In: *No. 22, Issue 87, APRIL-JUNE 2020* (), p. 202067 (cit. on p. 2).

[74] X. Peng. "Least squares twin support vector hypersphere (LS-TSVH) for pattern recognition". In: *Expert Systems with Applications* 37.12 (2010), pp. 8371–8378. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2010.05.045. URL: https://www.sciencedirect.com/science/article/pii/S0957417410004562 (cit. on pp. 27, 28).

[75] X. Peng. "A spheres-based support vector machine for pattern classification". In: *Neural Computing and Applications* 31 (2019-04), pp. 379–396. DOI: 10.1007/s00521-017-3004-x (cit. on p. 27).

[76] X. Peng and D. Xu. "A twin-hypersphere support vector machine classifier and the fast learning algorithm". In: *Information Sciences* 221 (2013), pp. 12–27. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2012.09.009. URL: https://www.sciencedirect.com/science/article/pii/S0020025512005919 (cit. on pp. 27, 28).

[77] X. Peng and D. Xu. "Twin support vector hypersphere (TSVH) classifier for pattern recognition". In: *Neural Computing and Applications* 24 (2014-04), pp. 1207–1220. DOI: 10.1007/s00521-012-1306-6 (cit. on p. 27).

[78] J. Peters. "Policy gradient methods". In: *Scholarpedia* 5.11 (2010), p. 3698 (cit. on p. 7).

[79] X. Qi et al. "An Interval-Valued Data Classification Method Based on the Unified Representation Frame". In: *IEEE Access* 8 (2020), pp. 17002–17012. DOI: 10.1109/ACCESS.2020.2967780 (cit. on p. 43).

[80] J. Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017) (cit. on p. 7).

[81] R. Strack et al. "Sphere Support Vector Machines for large classification tasks". In: *Neurocomputing* 101 (2013-02), 59–67. DOI: 10.1016/j.neucom.2012.07.025 (cit. on p. 27).

[82] C. Taurion. *Big data*. Brasport, 2013 (cit. on p. 2).

[83] D. M. J. Tax and R. P. W. Duin. "Uniform Object Generation for Optimizing One-Class Classifiers". In: *J. Mach. Learn. Res.* 2 (2002-03), pp. 155–173. ISSN: 1532-4435 (cit. on pp. 26, 27).

[84] D. M. J. Tax and R. P. W. Duin. "Support vector data description". In: *Machine Learning* 54.1 (2004), pp. 45–66 (cit. on pp. 26, 27, 40).

[85] D. M. Tax and R. P. Duin. "Support vector domain description". In: *Pattern Recognition Letters* 20.11 (1999), pp. 1191–1199. ISSN: 0167-8655. DOI: https://doi.org/10.1016/S0167-8655(99)00087-2. URL: https://www.sciencedirect.com/science/article/pii/S0167865599000872 (cit. on pp. 26, 27).

[86] R. Tiwari. "Ethical and societal implications of AI and machine learning". In: *International Journal of Scientific Research in Engineering and Management* 7.01 (2023) (cit. on p. 3).

[87] *UC Irvine Machine Learning Repository*. URL: https://archive-beta.ics.uci.edu/ (cit. on p. 8).

[88] L. Utkin and F. Coolen. "Interval-valued regression and classification models in the framework of machine learning". In: *ISIPTA 2011 - Proceedings of the 7th International Symposium on Imprecise Probability: Theories and Applications* (2011-01) (cit. on p. 43).

[89] L. V. Utkin and Y. A. Zhuk. "An one-class classification support vector machine model by interval-valued training data". In: *Knowledge-Based Systems* 120 (2017), pp. 43–56. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2016.12.022. URL: https://www.sciencedirect.com/science/article/pii/S0950705116305251 (cit. on p. 43).

[90] L. Van Der Maaten. "Accelerating t-SNE using tree-based algorithms". In: *The journal of machine learning research* 15.1 (2014), pp. 3221–3245 (cit. on p. 6).

[91] R. Verde and A. Irpino. "Basic statistics for probabilistic symbolic variables: a novel metric-based approach". In: *arXiv preprint arXiv:1110.2295* (2011) (cit. on p. 42).

[92] S. Verma, J. Dickerson, and K. Hines. "Counterfactual explanations for machine learning: A review". In: *arXiv preprint arXiv:2010.10596* 2 (2020), p. 1 (cit. on p. 19).

[93] J. Wang, P. Neskovic, and L. N. Cooper. "Pattern Classification via Single Spheres". In: *Discovery Science*. Ed. by A. Hoffmann, H. Motoda, and T. Scheffer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 241–252 (cit. on p. 27).

[94] C. J. Watkins and P. Dayan. "Q-learning". In: *Machine learning* 8 (1992), pp. 279–292 (cit. on p. 7).

[95] M. Wood and F. Stevens. *California Mushrooms*. 2024. URL: https://www.mykoweb.com/CAF/ (visited on 2024-05-25) (cit. on p. 53).

[96] Y. Xu et al. "A maximum margin and minimum volume hyper-spheres machine with pinball loss for imbalanced data classification". In: *Knowledge-Based Systems* 95 (2016), pp. 75–85. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2015.12.005. URL: https://www.sciencedirect.com/science/article/pii/S0950705115004773 (cit. on pp. 27, 28).

[97] F.-J. Yang. "An implementation of naive bayes classifier". In: *2018 International conference on computational science and computational intelligence (CSCI)*. IEEE. 2018, pp. 301–306 (cit. on p. 12).

[98] Z.-H. Zhou. *Machine learning*. Springer nature, 2021 (cit. on p. 5).

I

MATLAB Classifiers

List of classifiers in Classification Learner APP - MATLAB R2024a.

MATLAB Classifiers	
Decision Trees	Fine Tree
	Medium Tree
	Coarse Tree
Discriminant Analysis	Linear Discriminant
	Quadratic Discriminant
Logistic Regression	Binary GLM Logistic Regression
	Efficient Logistic Regression
Efficiently Trained Linear Classifier	Efficient Linear SVM
Naive Bayes	Gausian Naive Bayes
	Kernel Naive Bayes
Support Vector Machines	Linear SVM
	Quadratic SVM
	Cubic SVM
	Fine Gaussian SVM
	Medium Gaussian SVM
	Coarse Gaussian SVM
k-Nearest Neighbor	Fine KNN
	Medium KNN
	Coarse KNN
	Cosine KNN
	Cubic KNN
	Weighted KNN
Ensemble Methods	Ensemble Boosted Trees
	Ensemble Bagged Trees
	Ensemble Subspace Discriminant
	Ensemble Subspace KNN
	Ensemble RUSBoosted Trees
Neural Networks	Narrow Neural Network
	Medium Neural Network
	Wide Neural Network
	Bilayered Neural Network
	Trilayered Neural Network
Kernel Approximation Classifiers	SVM Kernel
	Logistic Regression Kernel

Table I.1: MATLAB's R2024a Classification Learner APP Classifiers