A MODIFIED WEIGHTED SUPPORT VECTOR MACHINE (WSVM) TO REDUCE NOISE DATA IN CLASSIFICATION PROBLEM

SYARIZUL AMRI MOHD DZULKIFLI

A thesis submitted in
fulfillment of the requirement for the award of the
Doctor of Philosophy in Information Technology

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

DECEMBER, 2021

ACKNOWLEDGEMENTS

All praise to the Almighty Allah, for the good health and wellbeing that were necessary to complete this study. I would like to express my sincere gratitude to Assoc. Prof. Dr. Mohd Najib bin Mohd Salleh, my supervisor, for the continuous support of my PhD study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I take this opportunity to express gratitude to Faculty of Computer Science and Information Technology (FSKTM) and Centre for Graduate Studies (CGS) of Universiti Tun Hussein Onn Malaysia (UTHM) for providing good facilities and inspiring environment for me to complete this study.

I also place on record, my sense of gratitude to one and all, who directly or indirectly, have lent their helping hands in this study.

Last but not least, deepest appreciation to my parents, Mohd Dzulkifli bin A. Jalil and Shaikhah binti Ahmad, and my siblings, Syahrun Neizam bin Mohd Dzulkifli, Dzunnur Zaily binti Mohd Dzulkifli and Syafiee Syukrie bin Mohd Dzulkifli for their support and give me a lot of encouragement, guidance and motivation.

ABSTRACT

Classification refers to a predictive modeling problem where a class label is predicted for a given example of input data. Data is everywhere and the amount of digital data that exists is growing exponentially. However, data is rarely perfect and there are many inconsistencies that affect data quality such as noise data. Nowadays, the use of SVM is very perspective for the big data classification. SVM provides a global solution for data classification but SVM is highly sensitive to noise data and may not be effective when the level of noise data is high. When noise exists in training data, the decision boundary of SVM would deviate from the optimal hyperplane severely. To overcome SVM drawback for noise data problem, WSVM using KPCM algorithm was used but WSVM using kernel-based learning algorithm such as KPCM algorithm suffer from training complexity, expensive computation time and storage memory when noise data contaminate training data. Thus, through a simple pruning and speed-up method such as clustering method, WKM-SVM has been proposed. However, WKM-SVM has several limitations that are related to *k*-Means Clustering. One of the limitations of WKM-SVM is the clustering centers may not suitably represent original data structures which can potentially cause poor prediction results. Therefore, this research work proposes a modified WSVM utilized with instance selection method and weighted learning to improve WSVM training and classification accuracy. The modification of WSVM will reduce noise data by producing multiple hyperplanes and selecting the optimal hyperplane based on the lowest noise data. The overall result shows that the proposed method outperforms WSVM, OWSVM and WKM-SVM in all datasets in terms of classification accuracy. Specifically, the proposed method produces classification accuracy equal to or higher than 85% for three datasets and lower than 85% for six datasets. However, the performance of the proposed method for test data may not be as good as anticipated since most of the datasets produced classification accuracy lower than 85%.

ABSTRAK

Pengkelasan merujuk kepada masalah pemodelan ramalan yang mana label kelas diramalkan untuk contoh tertentu bagi kemasukan data. Data ada di mana-mana dan jumlah data digital yang wujud telah berkembang dengan pesat. Walau bagaimanapun, data jarang sempurna dan terdapat banyak ketidakseragaman yang mempengaruhi kualiti data seperti hingar data. Masa kini, penggunaan SVM adalah sangat perspektif bagi pengkelasan data besar. SVM menyediakan penyelesaian umum untuk pengkelasan data tetapi SVM amat sensitif terhadap hingar data dan mungkin tidak efektif jika hingar data adalah tinggi. Apabila hingar wujud dalam data latihan, sempadan keputusan SVM akan tersasar jauh dari sempadan optimum. Bagi mengatasi kekurangan SVM dalam masalah hingar data, WSVM yang menggunakan algoritma KPCM telah digunakan tetapi WSVM yang menggunakan algoritma pembelajaran berasaskan *kernel* seperti algoritma KPCM mempunyai masalah dalam latihan, masa pengiraan yang tinggi dan ruang memori apabila hingar data mengubah data latihan. Dengan demikian, melalui kaedah mempercepat dan pengurangan mudah seperti kaedah pengelompokan, WKM-SVM telah dicadangkan. Namun begitu, WKM-SVM mempunyai beberapa kekangan berkaitan dengan *k-Means Clustering*. Salah satu kekangan tersebut adalah pusat pengelompokan tidak sesuai mewakili struktur data asal yang berpotensi menyebabkan keputusan ramalan yang rendah. Lantaran itu, kerja penyelidikan ini mencadangkan agar WSVM diubah suai menggunakan kaedah *instance selection* dan *weighted learning* untuk meningkatkan latihan WSVM dan ketepatan pengkelasan. Pengubahsuaian WSVM akan mengurangkan hingar data melalui penghasilan sempadan keputusan yang pelbagai dan pemilihan sempadan keputusan berdasarkan jumlah hingar data yang rendah. Hasil keseluruhan keputusan menunjukkan bahawa kaedah yang dicadangkan mengatasi WSVM, OWSVM dan WKM-SVM berdasarkan pada ketepatan pengkelasan dalam semua set data. Secara khususnya, kaedah yang dicadangkan memperoleh ketepatan pengkelasan sama atau lebih tinggi dari 85% untuk tiga set data dan lebih rendah dari 85% untuk enam set data. Namun begitu, pencapaian bagi kaedah yang dicadangkan untuk data ujian tidak sebaik yang dijangkakan kerana kebanyakan set data memperoleh ketepatan pengkelasan lebih rendah dari 85%.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ML	-	Machine Learning
AI	-	Artificial Intelligence
w	-	Weight vector
ξ_i	-	Slack variables
b	-	Bias
x	-	Input vector
ρ	-	Margin for a hyperplane
T	-	Transpose
α	-	Lagrange multiplier
S	-	i where $\alpha_i > 0$
C	-	Regularization parameter
h_0/h_1	-	hyperplane
m	-	Distance between hyperplane h_0 and h_1
ω_{ij}	-	Weight function
QP	-	Quadratic Programming
k	-	Number of hyperplanes that will produced depends on the subsets
V	-	Difference slope value between two hyperplanes
R	-	Ranking of the subset
m_i	-	Classification loss
p	-	Orthogonal projection
RBF	-	Radial Basis Function
KKT	-	Karush-Kuhn-Tucker
KPCM	-	Kernel-based Possibilistic C-Means
EP	-	Emerging Patterns
PCA	-	Principal Component Analysis
CSA	-	Cuckoo Search algorithm

BA	-	Bat algorithm
FPA	-	Flower Pollination algorithm
FFA	-	Firefly algorithm
CSISA	-	Cuckoo Search Instance Selection Algorithm
BISA	-	Bat Instance Selection Algorithm
FPISA	-	Flower Pollination Instance Selection Algorithm
FFISA	-	Firefly Instance Selection Algorithm
DT	-	Decision Tree
ANN	-	Artificial Neural Network
KNN	-	*K*-Nearest Neighbors
LUPI	-	Learning Using Privileged Information
NCAR	-	Noise Completely at Random
NAR	-	Noise at Random
NNAR	-	Noise Not at Random
SRM	-	Structural Risk Minimization
SMO	-	Sequential Minimal Optimization
SVM	-	Support Vector Machine
SV	-	Support vector
WSVM	-	Weighted Support Vector Machine
KEEL	-	Knowledge Extraction based on Evolutionary Learning
DASH	-	Difference Average Slope Hyperplane
FPR	-	false positive rate
FNR	-	false negative rate
ROC	-	receiver operating characteristic
AUC	-	area under the ROC curve
TP	-	true positives
TN	-	true negatives
FP	-	false positives
FN	-	false negatives
PPV	-	positive predictive value
TPR	-	true positive rate
PC	-	personal computer
CPU	-	Central Processing Unit

RAM	-	Random Access Memory
OWSVM	-	One-step Weighted Support Vector Machine
IWSVM	-	Iteratively Weighted Support Vector Machine
KM-SVM	-	Support Vector Machine using *k*-Means Clustering
WKM-SVM	-	Weighted Support Vector Machine using *k*-Means Clustering
MHI	-	Multiple Hyperplanes and Instance-weighted
MWSVM-MHI	-	Modified WSVM using Multiple Hyperplanes and Instance-weighted

LIST OF PUBLICATIONS

Journals:

(i) Syarizul Amri Mohd Dzulkifli and Mohd Najib Mohd Salleh (2020). Modified Weighted Support Vector Machine (WSVM) algorithm using Multiple Hyperplanes and Instance-Weighted for class noise. International Conference on Interdisciplinary Computer Science and Engineering 2020 (ICICSE2020).

Proceedings:

(i) Syarizul Amri Mohd Dzulkifli, Mohd Najib Mohd Salleh and Abdul Mutalib Leman (2017). Customer and performance rating in QFD using SVM classification. 3rd Electronic and Green Materials International Conference 2017 (EGM 2017).

(ii) Syarizul Amri Mohd Dzulkifli, Mohd Najib Mohd Salleh and Kashif Hussain Talpur (2019). Improved Weighted Learning Support Vector Machine (SVM) for High Accuracy. 2nd International Conference on Computational Intelligence and Intelligent Systems (CIIS 2019).

(iii) Syarizul Amri Mohd Dzulkifli, Mohd Najib Mohd Salleh and Ida Aryanie Bahrudin (2020). A Comparison of Weighted Support Vector Machine (WSVM), One-Step WSVM (OWSVM) and Iteratively WSVM (IWSVM) for Mislabeled Data. 4th edition of Soft Computing and Data Mining International Conference (SCDM 2020).

CHAPTER 1

INTRODUCTION

1.1 Background of Research

Machine learning (ML) is a continuously developing field, given that some considerations need to be taken into account in working with machine learning methodologies or analysing the impact of machine learning processes (Tagliaferri, 2017). ML applications are highly automated and self-modifying and continue to improve over time with minimal human intervention as they learn with more data. According to a recent study, ML algorithms are expected to replace 25% of global jobs in the next decade (Mathews & Aasim, 2021). The basic objective of ML is to allow computers to automatically learn to recognise complex patterns, make intelligent decisions, and improve performance over time based on the input data. Most often, this involves using a set of historical outcomes to make predictions about future outcomes. ML is also seen as a discipline in artificial intelligence (AI) that consists of designing and developing algorithms. Generally, ML aims to find patterns in data and subsequently use a model that recognizes those patterns in making predictions on new data.

However, ML has its own unique challenges compared to other approaches (Nair, 2017); the first challenge is understanding which processes need automation. Intelligent process automation is about robotic process automation fundamentals combined with ML capabilities to robotize the tasks and learn to perform a job even better (Joshi, 2019). The most straightforward processes to automate are the ones that are performed each day manually with no variable output. The complicated processes require further self-analysis before automation. Though, while ML can help automate some processes, not all automation problems need ML. The second challenge is the lack of good data. Noise in data is a significant concern for many ML techniques used in modeling data (Atla *et al.*, 2011). The solution is to evaluate data through data acquisition, data integration, and data exploration until generating a good dataset. The third challenge is the inadequate infrastructure. For most organisations, managing the various aspects of the infrastructure surrounding ML activities can become a significant challenge (Dean, 2017). The solution to handle ML is to upgrade storage accompanied by hardware acceleration and distributed computing.

The fourth challenge is implementation. Various data driven organizations have spent many years developing successful analytics platforms for implementing ML. Implementing a ML algorithm will provide a deep and practical appreciation for how the algorithm works. This knowledge can also help to internalize the mathematical description of the algorithm (Novikov, 2020). Moreover, integrating newer ML methodologies into existing methodologies is a complicated task. Though maintaining proper interpretation and documentation is an excellent solution to ease the implementation of new methodologies, the last challenge results from the limited skilled resources. With the rapid growth of big data and the availability of programming tools, ML is becoming increasingly popular for data scientists (Mathews & Aasim, 2021). Data scientists often need a combination of domain experience and in-depth knowledge of science, technology, and mathematics. Consequently, the recruitment for data scientists requires companies to pay large salaries since these jobs are often in high demand. This is due to the emergence of big data and how data is being generated and consumed by companies (Das, 2020).

Given these challenges, the second challenge is related to this research as, over the last few decades, noise data has attracted a considerable amount of interest and attention from researchers. The research community has developed several techniques and algorithms to address this issue (Prati *et al.*, 2019). ML can assist people who are frequently susceptible to making mistakes during analyses and trying to establish relationships between multiple features and improve the efficiency of systems and the designs of machines. ML also provides knowledge on making more informed, data driven decisions faster than traditional approaches. Having said that, there are three types of ML: supervised learning, unsupervised learning, and reinforcement learning (Atul, 2019).

In supervised learning, the algorithms are designed to learn by example. When training a supervised learning algorithm, the training data consists of inputs paired with the expected outputs. The training data can accept any type of data as an input, such as values of a database row, the pixels of an image, or an audio frequency histogram. During training, the algorithm searches for patterns in the data that correlate with the expected outputs; then, after training, the algorithm will take in new unseen inputs and determine the label for the new inputs classified based on prior training data. The supervised learning model aims to predict the correct label for newly presented input data (Wilson, 2019a).

Unlike supervised learning, unsupervised learning does not use labeled data but focuses on the data's attributes. Unsupervised learning will frequently find subgroups or detect hidden patterns based on the typical characteristics of the input data within the dataset. In unsupervised learning, the targeted outputs are not subjects of concern as making predictions is not the desired outcome of unsupervised learning algorithms (Wilson, 2019b).

On the other hand, reinforcement learning is considered a hit and trial method of learning. This type of ML is the training of ML models to make a sequence of decisions. To get the machine to do what the programmer wishes, the AI gets either rewards or penalties for the actions. The goal is to maximise the total reward. Moreover, the model has to determine how to perform the task to maximize the reward, beginning from random trials and finishing with advanced tactics (Błażej & Konrad, 2018).

The majority of practical ML uses supervised learning (Brownlee, 2016). There is also no single learning algorithm that works best on all supervised learning problems. A broad range of supervised learning algorithms is available, each with strengths and weaknesses. Supervised learning has been successful in real world applications, divided into two categories: classification and regression (Jaiswal, 2018). Classification predicts discrete values such as true or false and male or female, while regression predicts continuous values such as price, salary, or age. This research focuses on classification because classification is an important technique used in data mining and data analysis applications (Pruengkarn *et al.*, 2015).

In classification, reliability depends on correctly detecting the class label (Sarangam, 2021). Classification refers to a predictive modeling problem where a class label is predicted for a given input data example (Brownlee, 2020). The success of prediction values for the class label aims to measure the overall accuracy, percentage of data for which the class label is correctly predicted. Moreover, classification algorithms have been designed to achieve the maximum possible number of correct class label predictions. In addition, classification seek to predict the target class by analyzing the training data (Priyadarshiny, 2019) and make good predictions on unseen data (Pérez-Ortiz *et al.*, 2016). Attributes and class labels typically characterize the quality of training data, in which the quality of attributes represents how well attributes describe the data for the training purposes.

In contrast, the quality of the class label indicates whether the label of each data is correctly assigned (Nazari *et al.*, 2018). However, having said that, data is rarely perfect, as many inconsistencies affect data quality, such as noise data (Garcia, 2016). Noise data is also considered one of the most challenging classification problems (Farid *et al.*, 2013). Even with extreme efforts to avoid noise, it is challenging to ensure a data acquisition process without errors. Noise data tend to increase the complexity of the classification problem (Napolitano, 2009) within a wide range of research areas. Several studies have concluded that, even in controlled environments, there are at least 5% of noise and errors in a dataset (Maletic & Marcus, 2000). Even though there are various strategies and techniques to manage and deal with noise data, it is often difficult to determine if a given data is indeed noisy or not.

Support Vector Machine (SVM) is a promising and powerful tool for solving practical binary classification problems (Cervantes *et al.*, 2020) and provides a global solution for data classification (Abdiansah & Wardoyo, 2015). One way to learn classification algorithms in the presence of noise data is to correct the labels on the noise data and subsequently to learn the classification algorithm. SVM treats all training data of a given class equally and relies on convex quadratic programming (QP), whose computational complexity is commonly subject to data size. Various studies have indicated that noise data have several consequences, such as significantly reducing the classification accuracy of the classifier (Li *et al.*, 2019), increase in the numbers of necessary training data, increase the classification model building time, alterations in the observed frequencies of the possible classes (Frénay & Kabán, 2014) and increasing the size and interpretability of the classifier (Rani & Rao, 2019). Therefore, this research emphasizes dealing with noise data by using SVM and reducing the high computational complexity of SVM training.

1.2 Problem Statement

Nowadays, the use of SVM is very perspective for the big data classification (Demidova *et al.*, 2016). Training SVM from extremely large and difficult datasets has become an issue given the high training time and memory complexity of SVM training (Nalepa & Kawulok, 2018). SVM requires all training data to be stored in memory during the training when the model's parameters are learned. Once the model parameters are identified, SVM only depends on a subset of training data commonly referred to as support vectors (SV) that lie near the margin. Here, the complexity of the classification task with SVM depends on the number of SV rather than the dimensionality of the input space (Awad & Khanna, 2015). The number of SV retained from the original dataset is data-dependent and varies depending on the complexity of the data, which is captured by the data dimensionality and class. When the data have noise, it is possible that these SV could be construed as noise as well.

Noise data causes decreased performance on SVM given SVM is highly sensitive to noise data (Almasi & Rouhani, 2016) and may not be effective when the level of noise data is high (Li *et al.*, 2013). The performance of SVM can also dramatically decrease with a relatively small number of noise data, which will make the decision boundary deviate from the optimal hyperplane severely (Zhu *et al.*, 2016). Figure 1.1 shows that the noise data influences the decision boundary severely. The thin solid line is the decision plane with no noises, while the bold dotted line is the decision plane with some noises. Circles denote noise data.

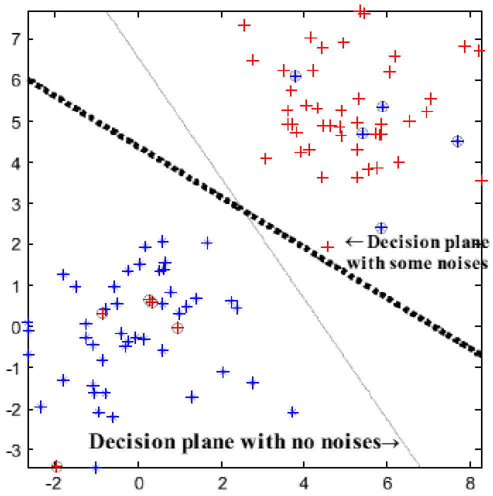


Figure 1.1: Noise data influence the decision boundary severely (Zhu *et al.*, 2016)

In addressing SVM drawback for noise data problem, Yang *et al.* (2007) discovered one of the considered solutions by proposing Weighted SVM (WSVM) using a Kernel-based Possibilistic C-Means (KPCM) algorithm. The KPCM algorithm generates the weights used in WSVM, and these weights will be given to noise data to reduce the effect of noise data as if they do not exist in training data. Indeed, different data have different impacts on the learning of the decision boundary, and the function of weight can make noise data contribute differently. If the data are already associated with the weights, the information can be directly utilized to train the data. As a result, the effect of noise data on the decision boundary is reduced during the training. However, WSVM using kernel-based learning algorithms such as the KPCM algorithm suffer from training complexity, expensive computation time and storage memory when noise data contaminate training data. Nevertheless, it can be reduced through a simple pruning and speed-up method.

Thus, through a simple pruning and speed-up method such as the clustering method, WSVM using *k*-Means Clustering (WKM-SVM) was proposed by Bang & Jhun (2014) and Kim (2016) to reduce noise data. However, WKM-SVM has several limitations related to *k*-Means Clustering. Considering the limitations of WKM-SVM, the intention of scaling down the training data by selecting support vector candidates using a small subset to reduce SVM training time while assigned weight of each noise data for a different penalty of misclassification is considered in this work. The instance selection method is a set of techniques that reduce the quantity of data by selecting a subset of data that resembles the original data. Instance selection method is also intended to reduce the computational complexity by reducing the number of data in training data (Leyva *et al.*, 2015). The reduction of training data reduces both the space requirements of the system and the processing time of learning tasks. In a weighted learning scenario, each training data comes with a positive weight. If classes have different misclassification errors that incur different penalties, prior knowledge can be applied in the form of instance weights. Assigning high weight to data suggests that the learning algorithm should attempt to correctly classify the data (Lapin *et al.*, 2014).

Additionally, the weighted classifier can deliver better classification performance than the unweighted classifier (Wu & Liu, 2013). Thus, the weighted classifier could refine the decision boundary with robustness to noise data. In conjunction with the advantages of both the instance selection method (reduce high computational complexity) and weighted learning (classify the data correctly), this research aims to propose a modified WSVM utilized with an instance selection method and weighted learning.

1.3 Research Objectives

The main objectives of this research are:

(a) To propose a modified WSVM using the instance selection method and weighted learning to improve WSVM training.

(b) To design Multiple Hyperplanes and Instance-weighted (MHI) and incorporate modified WSVM to improve classification accuracy.

(c) To evaluate the performance of the proposed method with other existing methods of WSVM based on classification accuracy and weighted learning performance.

1.4 Research Scope

The study will focus on the implementation of the modified WSVM using MHI for class noise. This research focuses on binary classification and reduction of class noise in training data. The class noise focused on in this research is class noise located on and near the margin of the hyperplane. Four binary classification datasets, four multi-class classification datasets and a real flood dataset with 20% of class noise are used for training data. The results are compared with WSVM, OWSVM and WKM-SVM and evaluated based on their classification accuracy and weighted learning performance.

1.5 Significance of Research

This research provides the following contributions to misclassification in the field of ML, especially WSVM;

(a) The modification of WSVM will reduce class noise by producing multiple hyperplanes, selecting the optimal hyperplane based on the lowest class noise, and achieving high classification accuracy.

(b) The proposed modified WSVM can reduce the computational complexity of SVM training by searching and choosing appropriate subsets of data as if the overall data has been used. It can also handle a large amount of training data which related to big data problems.

(c) The weight for each class noise of the proposed modified WSVM that makes the class noise tend to become error SV produced optimal weight values that can helped obtain better performance on weighted learning.

1.6 Organization of Thesis

This thesis is organized and divided into five chapters. The background of the research, problem statement, research objectives, research scope and significance of the research is highlighted in the first chapter. The second chapter presents the classification task, an overview of class noise, the theoretical concept of SVM that consists of hard margin and soft margin, the theoretical concept of WSVM, determination of SVM training and weighted learning, and the research gap. The third chapter addresses the research framework, introducing the proposed method named modified WSVM using MHI. The fourth chapter presents the experimental results and discussions, followed by the last chapter concluding the research and providing suggestions for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter introduces content related to the research. ML is the process of finding valuable results from real world datasets. Usually, real world datasets contain noise data that can significantly affect various data analysis tasks such as classification. The need to address noise data is evident since it is detrimental to almost any form of data analysis. This chapter describes one of the noise data, which is class noise. The theoretical concept of SVM and WSVM used in this research are also described in detail. SVM training and the learning process of weight are determined by the instance selection method and weighted learning. Also discussed in this chapter are the related works in existing studies of WSVM presenting the research gap(s) in line with the problems revealed from existing studies of WSVM.

2.2 Classification Task

Classification is the task of assigning objects to one of several predefined categories, which requires the use of ML algorithms that learn how to assign a class label to data samples from the problem domain. Classification refers to a predictive modeling problem where a class label is predicted for a given input data sample (Brownlee, 2020). From a modeling perspective, classification requires training data with many samples of inputs and outputs from which to learn.

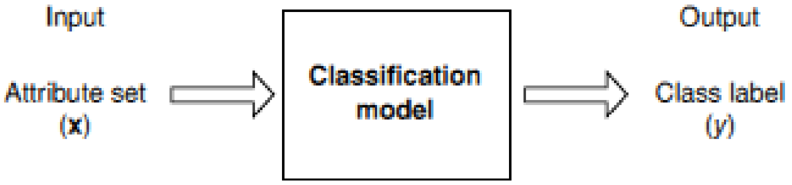


Figure 2.1: A schematic illustration of a classification task (Tan *et al.*, 2019)

The function concerning classification can be explained by mapping each attribute set (x) to one of the predefined class label (y). Figure 2.1 illustrates the general idea surrounding classification. A classification model is an abstract representation of the relationship between the attribute set and the class label, in which the model will classify data correctly if *f*(x) = y. A classification model offers two important roles in data mining, and this model is created using a given set of data known as training data. First, it is used as a predictive model to classify previously unlabeled data. Second, it is used as a descriptive model to identify the characteristics that distinguish data from different classes (Tan *et al.*, 2019). Classification models that implement classification are known as classifiers. A classifier utilizes some training data to understand how given input variables relate to the class. The general intention of a classifier is to separate the classes of the problem using only training data.

Generally, there are two types of classification problems: binary classification problems and multi-class classification problems (Jha *et al.*, 2019). In binary classification, there are only two possible label classes in which an algorithm utilizes some training data to understand how given input variables relate to the class. Multi-class classification refers to cases where there are more than two label classes (Asiri, 2018). However, the difficulty of classification problem can be attributed to three main aspects: uncertainty among the classes, the complexity of the separation between the classes and the data sparsity and dimensionality (Garcia, 2016). Table 2.1 shows examples of attribute sets and class labels for various classification tasks. Spam filtering and tumor identification are examples of binary classification problems, while Galaxy classification is example of multi-class classification problem.

Table 2.1: Examples of classification tasks

Task	Attribute set	Class label
Spam filtering	Features extracted from email message header and content	Spam or non-spam
Tumor identification	Features extracted from magnetic resonance imaging (MRI) scans	Malignant or benign
Galaxy classification	Features extracted from telescope images	Elliptical, spiral or irregular-shaped

This research focuses on binary classification problems, given that it has a discrete value as its output. The problem is standard practice to represent the output of a classification as an integer number such as 0 or 1. Various classifiers are developed for binary classification: Decision Tree, Artificial Neural Network, *K*-Nearest Neighbors and Support Vector Machine (Ortner, 2020).

Decision Tree (DT) is an easily interpretable method with fast prediction, can be adapted to deal with missing data and follows a similar pattern to that of human thinking (Ganegedara, 2018). This method can also be constructed from any size of dataset with many attributes. DT has three main components: nodes, leaves and edges. Moreover, DT could be applied for a random number of decision nodes, and each branch should end with a leaf node. Occasionally this method is a relatively unstable model leading to a complex tree structure (Sen *et al.*, 2020).

Artificial Neural Network (ANN) is an example of a non-linear prediction method that is frequently applied to various fields (Nkoana, 2011). ANN is a mathematical model of human perception which can be trained for performing a particular task based on available empirical data that includes a number of neurons or nodes working in parallel to transform the input data into output categories. However, one of the disadvantages of ANN is that it is challenging for a decision-maker to analyse the structure of the resulting ANN and relate it to the outputs (Solomatine & Dulal, 2003).

K-Nearest Neighbors (KNN) is an example of instance based learning based on the similarity between the new data and available data and places the unique data point into the most similar category to the general categories. This method also assumes that similar things exist in close proximity (Prem, 2021). The classification rule of KNN is simple. For each new data, the class will be assigned according to the majority vote of its KNN in training data, if K = 1, the algorithm only considers the nearest neighbor. Usually, this method leads to consuming high computational time, and the value of K needs to be determined correctly for a lower error rate (Sen *et al.*, 2020).

Another widely used classification model is SVM, which is a representation of the training data since points in space separated into categories by a clear gap that is as wide as possible (Jiang *et al.*, 2012). However, the training time of SVM is relatively high, and if the dataset is very large, then the prediction task is slow (Sen *et al.*, 2020). SVM is ahead of other methods mentioned because SVM is specifically designed for binary classification (Stanevski & Tsvetkov, 2005; Mushtaq & Mellouk, 2017; Brownlee, 2020) and generates the best overall accuracy result from research, as mentioned by Sen *et al.* (2020).

Data is everywhere and the amount of digital data that exists is growing exponentially (Monnappa, 2021). Moreover, there are many indications in which data will play a significant role in the success of companies. Large companies such as Facebook, Amazon and Google use the power of ML models to give customers a better user experience (Jadari, 2019). The development of high-end technologies has resulted in higher rate in proportion of data that data volume, variety, velocity, and veracity refers as big data.

Big data is a datasets that are so large and complex where traditional data processing technologies are inadequate (Demidova *et al.*, 2016). Traditional ML techniques does not give accurate results for massive datasets, thus many techniques were proposed to detect and eliminate noise so that the efficiency of the algorithm increases (Rani & Rao, 2019). For example, stock market data are constantly generating a large quantity of information in every single seconds. This information impact on different factors such as domestic and international news, government reports and natural disasters, hence it is crucial that the stock market data should be classified appropriately. Big data is also special application of data science.

Data science refers to the extraction of knowledge from data involving a wide range of techniques and theories drawn from many research fields within mathematics, statistics and information technology (Pérez-Ortiz *et al.*, 2016). Data science algorithms also are of great value to improve the performance of different applications, particularly the areas where data are collected daily and extracted to improve current systems. For the task of ML, data scientists study the data structure first and approach the given problem in the best way possible. Usually, the steps involved in data mining are data acquisition, pre-processing, selection and application of ML tools, evaluation, interpretation and presentation of the results obtained, and finally, dissemination and use of new knowledge.

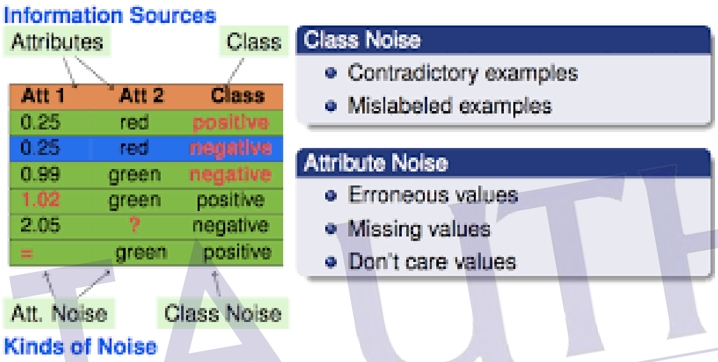
The generation of noise data can be characterized differently (Nettleton *et al.*, 2010). First, it can be characterized by its distribution, such as Gaussian noise. Secondly, it can be characterized by where it is introduced, as input attributes or output class. Finally, it can be characterized by distinguishing whether the magnitude of the generated noise values is relative to each data value of each variable. Thus, noise data make it more difficult for ML algorithms in forming accurate models from the data. However, producing good training data or high classification accuracy often leads to high computational complexity (Blachnik, 2015).

There are several definitions of what noise is in the context of data. One definition draws a distinct line between two main categories: attribute noise and class noise (Abdel Maksoud *et al.*, 2019). Attribute noise is defined as errors that affect the observed values of the attribute, whereas, in contrast, class noise alters the observed labels assigned to instances by incorrectly setting a negative label on a positive instance in binary classification. Frénay & Verleysen (2014) revealed that class noise is potentially more harmful than attribute noise.

2.3 Class Noise

Class noise is known as labelling error when the incorrect class label is assigned to data (Nazari *et al.*, 2018) and may significantly impact the learning process. Class noise usually occurs on the boundaries of the classes where the samples may have similar characteristics. Most of the research on class noise tends to focus on the influence of the classification performances (Pelletier *et al.*, 2016).

Class noise exists due to various reasons such as errors or the subjective nature of the data labeling process, inadequate information to determine the true label of a given example and mistakes made during data entry (Prati *et al.*, 2019). The subjective nature of data labeling process may arise when observations need to be ranked, such as when the information used to label an object is distinctly different from the information to which the learning algorithm will have access. The problem of information adequacy in determining the true label of a given example arises when the information used to label each observation is not sufficient, and there is an inadequate amount of information to determine the true class label of a given example. The most frequent errors are mistakes made during data entry that eventuate when transforming information on paper to computerized forms due to illegible or unclear handwriting. Occasionally experts often make mistakes during labeling though nowadays, since automated classification devices are increasingly used, classification errors are not always due to human experts. Figure 2.2 shows the class noise.



Att 1	Att 2	Class
0.25	red	positive
0.25	red	negative
0.99	green	negative
1.02	green	positive
2.05	?	negative
=	green	positive

Figure 2.2: Class noise (Burgos & Lorite, 2001)

There are two possible sources for class noise: contradictory samples and misclassifications (Morales *et al.*, 2017). Contradictory samples signify data that appear more than once in the dataset but with different class labels, while misclassifications are data are labeled with the incorrect classes. This research focuses on misclassification since this type of noise is more common and disruptive than contradictory samples (Nazari *et al.*, 2018).

A taxonomy of class noise mechanisms was offered by Frénay & Verleysen (2014), based on four random variables: X is the feature vector, Y is the true class, $\acute{Y}$ is the observed class, and E is a binary variable that indicates if the noise is present or not. This taxonomy is only applicable to binary and multi-class problems. According to the statistical dependencies among these four variables, the class noise occurrence is believed to be a stochastic process, and the probability of a data mislabeled is categorized into three groups (Prati *et al.*, 2019) as follows:

(a) Noise Completely at Random (NCAR)

This type of noise occurs in a completely stochastic way, and the probability of data mislabeled does not depend on the class nor the other predictive attributes.

(b) Noise at Random (NAR)

The probability of a data mislabeled is dependent on the value of the actual class (it can assume different values for different classes). NCAR class noise is a particular case of NAR class noise where the probabilities are the same for all classes.

(c) Noise Not at Random (NNAR)

Both NCAR and NAR models assumed that class noise applies to all data, but this is not always the case. The probability of data mislabeled depends on the feature space. For example, the data near class boundaries are likely to be noisier.

For a real world dataset, endeavouring to cleanse the data in some way or form is entirely out of the question, given the amount of person hours involved. A manual process of data cleansing is also time consuming, requires hard work and is prone to errors (Zhu & Wu, 2004). Powerful tools that can manage and assist in the data cleansing process are necessary and may be the only practical and cost effective means in achieving a reasonable quality level in an existing dataset.

Having said that, the problem of learning in noisy environments has been the main focus of many research studies in ML. Most learning algorithms have a mechanism to enhance their learning abilities in a noisy environment (Nazari *et al.*, 2018). However, despite the strategies and techniques in dealing with noisy data, some research studies reveal that the presence of class noise can still can have a negative impact on the performance of ML algorithms concerning classification accuracy (Saseendran *et al.*, 2019; Gupta & Gupta, 2019; Nazari *et al.*, 2018).

The consequences of class noise on the behavior of a classifier can be relatively severe such as the performance of the classifier may be significantly deteriorated, the learning algorithm can be easily affected given cardinality of the training data may increase to compensate for class noise, and the final model of the algorithm can be more complex than it should be (Nalepa & Kawulok, 2018). Class noise can also lead to severe overfitting and dramatically reduce accuracy (Yi & Wu, 2019).

There are three main approaches to address the class noise (Frénay & Verleysen, 2014): class noise-robust, class noise cleansing, and class noise-tolerant methods. The first approach relies on an algorithm that is naturally robust to class noise. The learning of the classifier is presumed to be not too sensitive to the presence of class noise. Several studies have indicated that some algorithms are less influenced than others by class noise, and the performances of classifiers inferred by class noise-robust algorithms continue to be affected by class noise.

The second approach is to improve the quality of training data using filter approaches. Noisy data can either be relabeled or simply removed. Filter approaches are relatively cheap and easy to implement, but some of them are likely to remove a substantial amount of data. Several studies have observed that by simply removing noisy data is more efficient than relabeling the data. However, research from Matić *et al.* (1992) revealed that over cleansing might reduce the performances of classifiers.

The third approach is that there exist algorithms that directly model class noise during learning or the model, modified to consider class noise in making existing methods less sensitive to the influence of class noise. The advantage of this approach is in separating the classification model and the class noise model. However, the main problem of this approach is that the complexity of learning algorithms will be increased given the additional parameters of the training data model.

Accordingly, different models should be used for training and testing in the presence of class noise. A complete model of the training data will consist of a class noise model and classification model, both used during training, but only the classification model is helpful for prediction. The learning process of the classification model is intended to be robust or tolerant to class noise in producing a good classification model.

One of the important issues for some of the approaches mentioned above is to prove their efficiency. Most experiments assess the efficiency of the approaches in dealing with class noise regarding accuracy (Brodley & Friedl, 1996; Brodley & Friedl, 1996) since a decrease in accuracy is one of the main outcomes of class noise. Another issue is the model complexity. Less complex models are considered better since they are less susceptible to overfitting.

Overfitting occurs when the ML model aims to achieve zero error on training data. SVM uses Structural Risk Minimization (SRM), an inductive principle that selects a model for learning from a finite training data. When SVM is being trained, SV will also be optimized; minimizing the SRM, SVM avoids overfitting.

2.4 Theoretical Concept of Support Vector Machine (SVM)

SVM became a popular and effective algorithm in ML given its high ability in generalization and good performance (Nazari & Kang, 2015) in many real-life applications such as bioinformatics, electrical load forecasting, pattern recognition, image processing and field of hydrology (Parikh & Shah, 2016). SVM is also one of the best-known margin-based learner models (Sabzevari, 2015) based on statistical learning theory (Vapnik, 1995).

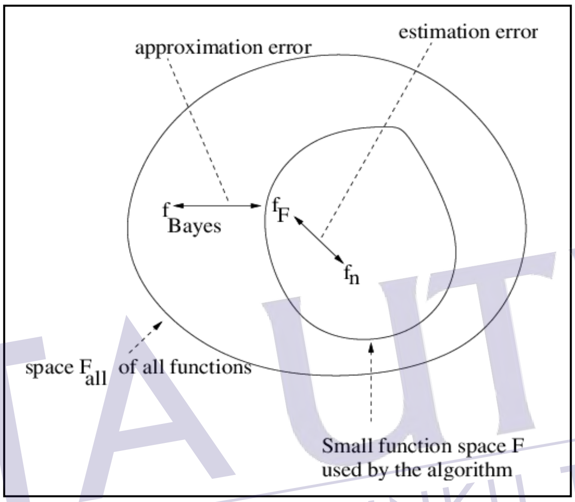

approximation error
estimation error
f_{Bayes}
f_F
f_n
space F_{all} of all functions
Small function space F used by the algorithm

Figure 2.3: Approximation and estimation error (Luxburg & Schölkopf, 2011)

The goal of modeling the statistical learning theory refers to two error terms, as shown in Figure 2.3. The two error terms include:

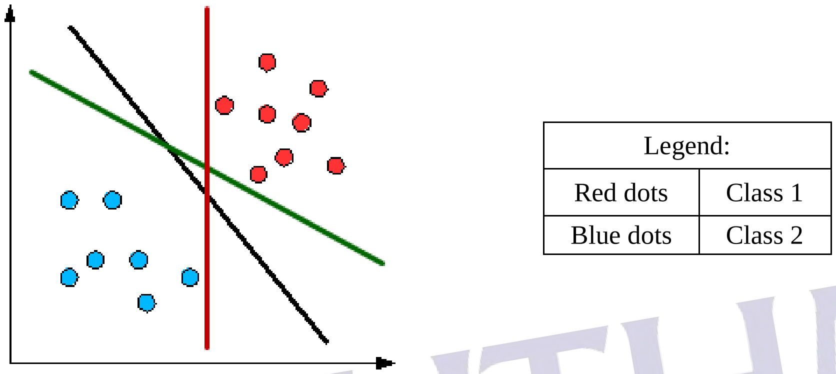
(a) Approximation Error

This error term is not influenced by any random quantities but deals with the error made by looking for the best function in a small function space rather than looking for the best function in the entire space of all functions.

(b) Estimation Error

This error term deals with the uncertainty introduced by the random sampling process. This error measures the variation of the risk of the function f_n estimated on the sample.

The goal of SVM is to find an optimal separating hyperplane so that data with different labels are located on different sides of the hyperplane, and the margin of separation is maximized (Ding & Xu, 2015). Consider the example in Figure 2.4. Many possible linear classifiers can separate the data, but there is only one that maximizes the margin. One particular linear classifier, the linear SVM, turns out to be particularly well suited with high dimensionality (Bersimis & Varlamis, 2019).



Legend:	
Red dots	Class 1
Blue dots	Class 2

Figure 2.4: Possible linear classifiers (Welch, 2017)

For SVM, *w*, *b* and *x* are the weight vector, bias and input vector of the optimal hyperplane, respectively. The separating function can be written as follows:

$$w^T x + b = 0 \tag{2.1}$$

where *T* denotes transpose, and according to function (2.1) there can be infinite number of solutions using various scaling factors. The boundary function of the separating margin from function (2.1) can be defined with (2.2):

$$\begin{gathered} w^T x + b = 1, \\ w^T x + b = -1 \end{gathered} \tag{2.2}$$

SVM was introduced by Vapnik (1995) for the initial idea of the separable case (hard margin SVM), where SVM constructs a hyperplane with the maximum margin that correctly classifies all the input (Figure 2.5). The maximum margin hyperplane is determined by the parameters *w* and *b* through solving the convex optimization problem as follows:

$$\min_{w,b} \frac{1}{2}||w||^2 \tag{2.3}$$

For all the data in training data, the following constraints must be satisfied:

$$\text{Subject to: } y_i(wx_i + b) \geq 1, for\ i = 1, \dots, n \tag{2.4}$$

where the constraints ensure that each example is correctly classified and minimizing $||w||^2$ is equivalent to maximizing the margin. The formula above describes a quadratic optimization problem to efficiently solve such optimization problems for millions of examples or dimensions.

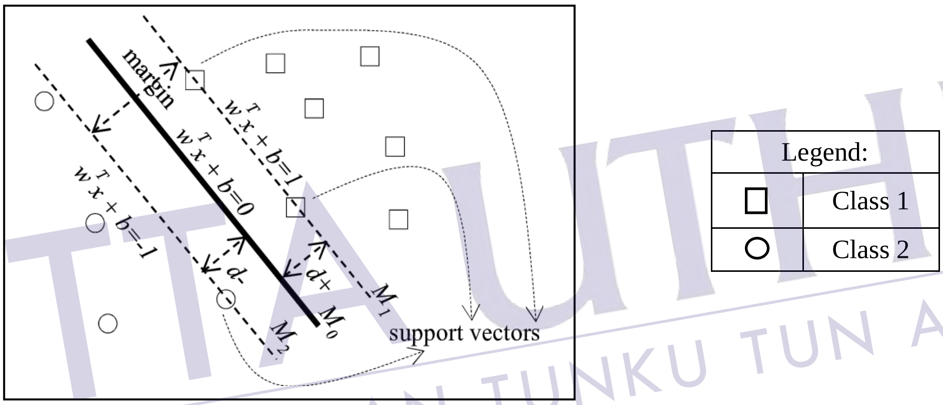


Figure 2.5: Hard margin SVM (Duong & Truong Hoang, 2019)

Maximizing the value of the separating margin is equal to minimizing the value of $||w||^2$. Generally, to solve the constrained optimization problem is carried out by using Lagrange multiplier. The following Lagrange function has been constructed as follows:

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(wx_i + b) - 1] \tag{2.5}$$

The following function can be obtained:

$$w = \sum_{i=1}^{n} \alpha_i\, y_i x_i$$
$$b = \frac{1}{S}\sum_{i=1}^{S} (y_i - w^T x) \tag{2.6}$$

where α denotes the Lagrange multiplier. S is determined by finding the indices i where $\alpha_i > 0$. The data point with $\alpha_i > 0$ is called SV. Function (2.6) will be substituted into the Lagrange function (2.5) to obtain the corresponding dual problem as follows:

$$\mathrm{W}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{2.7}$$

$$Subject\ to: \sum_{i=1}^{n} \alpha_i y_i = 0\ \ and\ 0 \leq \alpha_i \leq C, for\ i = 1, \dots, n \tag{2.8}$$

The dual problem is a typical convex QP optimization problem. Although, in many real world problems, a clear separating hyperplane could not be found to differentiate the data given the complexity of dataset (noise data).

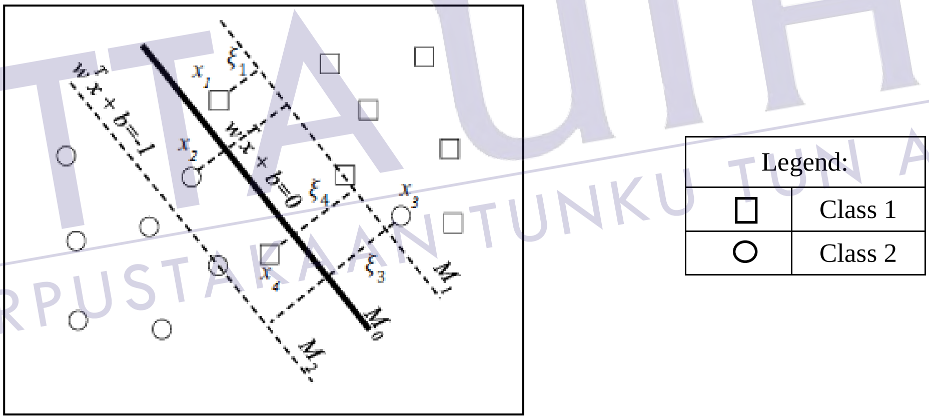


Figure 2.6: Soft margin SVM (Duong & Truong Hoang, 2019)

Thus, Cortes & Vapnik (1995) expand the idea of the separable case to the non-separable case by introducing positive slack variables referred to as soft margin SVM or standard SVM (Figure 2.6). For non-separable case, SVM maps the data to a higher dimensional feature space using an appropriate kernel function. Soft margin SVM makes it extremely effective in many applications given its high capability in generalization. Generalization refers to the ability to correctly classify unseen data (Bagchi, 2014). In such circumstances, a few data that exist on the wrong side of the separating hyperplane is allowed. The goal of soft margin SVM is to improve the generalization ability of SVM. The soft margin SVM optimization problem with slack variables can be formulated as follows:

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \tag{2.9}$$

Subject to: $y_i(wx_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \; for \; i = 1, \ldots, n$ (2.10)

where slack variables (ξ_i) is used to control the penalty associated with class noise, and parameter C is the regularization parameter that controls the tradeoff between the complexity and the number of allowed class noise. The penalty term $\sum_{i=1}^{n}\xi_i$ can be considered as a measure of the number of total misclassifications of the model. The objective function of (2.9) has two goals: (1) to maximize the margin and (2) to minimize the number of misclassifications. When trained with noise data, the decision hyperplane might deviate from optimal hyperplane due to the slack variables. The dual problem of soft margin SVM is equivalent to the dual problem of (2.7).

The constraint $y_i(wx_i + b) \geq 1 - \xi_i$ can be written more concisely as $y_i f(x_i) \geq 1 - \xi_i$ where together with $\xi_i \geq 0$ is equivalent to:

$$\xi_i = [1 - y_i f(x_i)] \tag{2.11}$$

SVM can also be fit in the general regularization framework using loss function as follows:

$$\min_{\mathrm{w,b}} \frac{1}{2}||w||^2 + \mathrm{C}\sum_{i=1}^{n}[1 - y_i f(x_i)] \tag{2.12}$$

where $[1 - y_i f(x_i)]$ is known as the hinge loss function. Function (2.12) is equivalent to function (2.9). Finally, the decision function, decided by the Lagrange multiplier and SVs, is represented as follows:

$$f(x) = sign\left(\sum_{i=1}^{n}\alpha_i\, y_i(x_i, x) + b\right) \tag{2.13}$$

where α_i is the Lagrange multiplier and the data points with $\alpha_i > 0$ is called SV. SV identification is referred to in Section 2.4.2.

```
For a given training data,
Input: training data $x_i$, labels $y_i$,
Output: sum of weight vector, α array, $b$ and SV
    Initialize: $\alpha_i = 0, f_i = -y_i$
    Compute: $b_{high}, I_{high}, b_{low}, I_{low}$ in (2.14 and 2.15)
    Update $\alpha_{I_{high}}$ and $\alpha_{I_{low}}$
    Repeat
    Update $f_i$ in 2.13
    Compute: $b_{high}, I_{high}, b_{low}, I_{low}$
    Update $\alpha_{I_{high}}$ and $\alpha_{I_{low}}$
    Until $b_{low} \leq b_{high} + 2\tau$
    Update the bias $b$
    Store the new α1 and α2 values
    Update the weight vector $w$
Return: SVM model
```

Figure 2.7: The pseudo-code of SVM

Figure 2.7 shows the pseudo-code of SVM. The data representation is in dot products, and a kernel is a function that calculates the dot product of two training vectors. The kernel function enables operations to be performed in the input space rather than the potentially high dimensional feature space. Different kernel functions are listed below:

(a) Linear kernel

The linear kernel is the simplest kernel function. It is given by the inner product <x,y> plus an optional parameter *C*.

(b) Polynomial kernel

The polynomial kernel is a non-stationary kernel. Polynomial kernels are well suited for problems where all the training data is normalized.

(c) Exponential Radial Basis Function kernel

The exponential kernel is closely related to the Gaussian kernel. It is also known as radial basis function (RBF) kernel.

(d) Gaussian Radial Basis Function kernel

The Gaussian kernel is an example of a RBF kernel. The adjustable parameter sigma plays a significant role in the performance of the kernel and should be carefully tuned. The exponential will behave almost linearly if overestimated, and the higher-dimensional projection will begin to lose its non-linear power. If underestimated, the function will lack regularization, and the decision boundary will become highly sensitive to noise in training data.

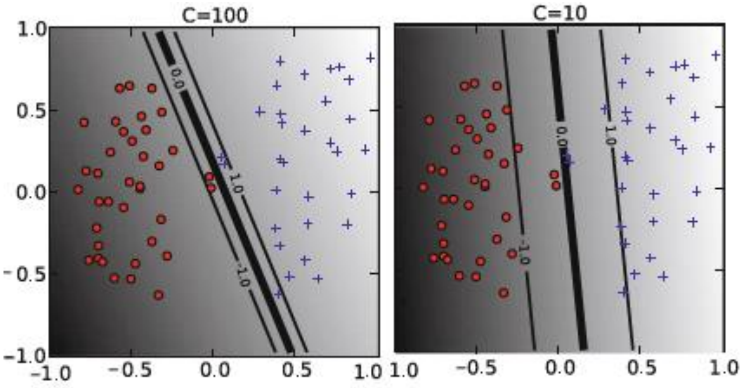
The kernel function plays a critical role in SVM and its performance using Kernel Hilbert Spaces (Paulsen & Raghupathi, 2016). The kernel symbolises a legitimate inner product in the feature space. The training data is not linearly separable in the input space, but are linearly separable in the feature space, referred to as "kernel trick". Using the kernel function, the inner product in the mapped feature space can be replaced with the kernel. The principle is to substitute the inner products in the feature space with inner products in the original data space.

2.4.1 SVM Parameters

In SVM, the parameter *w*, *b* and *C* are very important as these parameters will be computed to produce a hyperplane. Parameter *w* is the weight vector that can be explicitly retrieved and signifies the separating hyperplane between the two classes. Parameter *b* is a special parameter in SVM, called the bias value. Parameter *b* is the intercept of the hyperplane from the origin. SVM does not give the optimal separating hyperplane if it does not happen to pass through the origin unless it has the bias term. Bias b_{high} and b_{low} can be defined with their associated indices:

$$\begin{gathered} b_{high} = \min\{f_i : i \in I_0 \cup I_1 \cup I_2\} \\ I_{high} = \arg\min_i f_i \\ b_{low} = \max\{f_i : i \in I_0 \cup I_3 \cup I_4\} \\ I_{low} = \arg\, max_i\, f_i \end{gathered} \tag{2.14}$$

Parameter *C* adds a penalty for each class noise. The value of parameter *C* is fixed, and all training data are treated equally throughout the training. There is no rule of thumb in choosing the value of parameter *C*. If the value of parameter *C* is large, a large penalty is assigned to misclassified data, the margin decreases, and the classifier may overfit the data resulting in low generalization ability. In contrast, if the value of parameter C is small, the penalty for misclassified data is low, thus the margin increases and many errors occur. The effect of parameter C on the decision boundary is shown in Figure 2.8.



Legend:	
Red circle	Class 1
Blue cross	Class 2

Figure 2.8: The effect of the parameter C on the decision boundary (Ben-Hur *et al.*, 2008)

2.4.2 SV Identification

The Lagrange multipliers in the context of SVM, usually denoted as α, is a vector of the weights of all the training data referred to SV. SV are the training data that obtain a non-zero coefficient. These data are the most difficult data to classify and provide the most information regarding classification (Samanta, 2018). Assumed that there are n training examples. Then α is a vector of size n, and any ith element of α is α_i. A higher value of α_i means that ith training example has a higher contribution to the weight vector. Most $\alpha_i = 0$ is a direct consequence of the Karush-Kuhn-Tucker (KKT) dual complementarity conditions. The following index set I is defined denoting the training data pattern as follows:

REFERENCES

Abbasi, Z., & Rahmani, M. (2019). An Instance Selection Algorithm Based on ReliefF. *International Journal on Artificial Intelligence Tools*, *28*(1), 14.

Abdel Maksoud, E. A., Barakat, S., & Elmogy, M. (2019). Medical Images Analysis Based on Multilabel Classification. In *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging* (pp. 209–245). Elsevier Inc.

Abdiansah, A., & Wardoyo, R. (2015). Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM. *International Journal of Computer Applications*, *128*(3), 0975–8887.

Aich, A. (2019). Overfitting and Underfitting With Algorithms in Machine Learning. Retrieved March 20, 2021, from https://www.knowledgehut.com/blog/data-science/overfitting-and-underfitting-in-machine-learning

Akinyelu, A. A., & Adewumi, A. O. (2017). Improved instance selection methods for support vector machine speed optimization. *Security and Communication Networks*, *1*(1), 11.

Alcalá-Fdez, J., Sánchez, L., García, S., Jesus, M. J. del, Ventura, S., Garrell, J. M., … Herrera, F. (2009). KEEL: A software tool to assess evolutionary algorithms for data mining problems. In *Soft Computing* (pp. 307–318).

Almasi, O. N., & Rouhani, M. (2016). Fast and de-noise support vector machine training method based on fuzzy clustering method for large real world datasets. *Turkish Journal of Electrical Engineering and Computer Sciences*, *24*(1), 219–233.

Altidor, W., Khoshgoftaar, T. M., & Van Hulse, J. (2011). Robustness of filter-based feature ranking: A case study. In *Proceedings of the 24th International Florida Artificial Intelligence Research Society, FLAIRS - 24* (pp. 453–458).

Andronicus, A. A. (2017). *Intelligent Instance Selection Techniques for Support Vector Machine Speed Optimization with Application to e-Fraud Detection*. University of KwaZulu-Natal, Durban, South Africa.

Arefi, M. (2018). *Data-Driven Diagnostics of Issues Related to Power System Dynamics Using PMU Measurement*. University of North Carolina.

Arnaiz González, Á. (2018). *Estudio de métodos de selección de instancias*. Universidad De Burgos.

Asiri, S. (2018). Machine Learning Classifiers. Retrieved September 9, 2020, from https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

Atla, A., Tada, R., Sheng, V., & Singireddy, N. (2011). Sensitivity of different machine learning algorithms to noise. *Journal of Computing Sciences in Colleges*, *26*(5), 96–103.

Atul, H. (2019). Machine Learning Tutorial for Beginners. Retrieved July 22, 2020, from https://www.edureka.co/blog/machine-learning-tutorial/

Awad, M., & Khanna, R. (2015). Support Vector Machines for Classification. In *Efficient Learning Machines* (pp. 39–66). Apress, Berkeley, CA.

Bagchi, T. P. (2014). SVM Classifiers Based On Imperfect Training Data. In *POMS Conference* (pp. 1–7).

Bang, S., & Jhun, M. (2014). Weighted Support Vector Machine Using k-Means Clustering. *Communications in Statistics - Simulation and Computation*, *43*(10), 2307–2324.

Barros De Almeida, M., De Pádua Braga, A., & Braga, J. P. (2000). SVM-KM: Speeding SVMs learning with a priori cluster selection and k-means. In *Brazilian Symposium on Neural Networks, SBRN* (pp. 162–167).

Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B., & Rätsch, G. (2008). Support vector machines and kernels for computational biology. *PLoS Computational Biology*, *4*(10), 1–8.

Bersimis, F. G., & Varlamis, I. (2019). Use of health-related indices and classification methods in medical data. In *Classification Techniques for Medical Image Analysis and Computer Aided Diagnosis* (pp. 31–66). Academic Press.

Bhandari, A. (2020). AUC-ROC Curve in Machine Learning Clearly Explained. Retrieved August 4, 2020, from https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Blachnik, M. (2015). Reducing time complexity of SVM model by LVQ data compression. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)* (pp. 1–9).

Błażej, O., & Konrad, B. (2018). What is reinforcement learning? The complete guide. Retrieved July 25, 2020, from https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/#:~:text=Reinforcement learning is the training,faces a game-like situation.

Brodley, C. E., & Friedl, M. A. (1996). Improving automated land cover mapping by identifying and eliminating mislabeled observations from training data. In *International Geoscience and Remote Sensing Symposium (IGARSS)* (Vol. 2, pp. 1379–1381).

Brodley, Carla E., & Friedl, M. A. (1996). Identifying and eliminating mislabeled training instances. *Proceedings of the National Conference on Artificial Intelligence*, *1*(1), 799–805.

Brownlee, J. (2014). Feature Selection to Improve Accuracy and Decrease Training Time. Retrieved May 20, 2018, from https://machinelearningmastery.com/feature-selection-to-improve-accuracy-and-decrease-training-time/

Brownlee, J. (2016). Supervised and Unsupervised Machine Learning Algorithms. Retrieved January 21, 2019, from https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

Brownlee, J. (2019). Information Gain and Mutual Information for Machine Learning. Retrieved December 20, 2019, from https://machinelearningmastery.com/information-gain-and-mutual-information/

Brownlee, J. (2020). 4 Types of Classification Tasks in Machine Learning. Retrieved June 7, 2020, from https://machinelearningmastery.com/types-of-classification-in-machine-learning/

Burgos, S. A., & Lorite, F. J. C. (2001). Noisy Data in Data Mining. Retrieved February 10, 2020, from https://sci2s.ugr.es/noisydata

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*(10), 189–215.

Chaudhary, M. (2020). Silhouette Analysis in K-means Clustering. Retrieved September 19, 2021, from https://medium.com/@cmukesh8688/silhouette-analysis-in-k-means-clustering-cefa9a7ad111

Chelliah, I. (2020). An Introduction to Support Vector Machine. Retrieved December 24, 2020, from https://towardsdatascience.com/an-introduction-to-support-vector-machine-3f353241303b

Chen, J., Zhang, C., Xue, X., & Liu, C. L. (2013). Fast instance selection for speeding up support vector machines. *Knowledge-Based Systems*, *45*(8), 1–7.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297.

Das, S. (2020). Reasons, Why There Is A Shortage Of Data Scientists In The Industry. Retrieved August 18, 2020, from https://analyticsindiamag.com/reasons-why-there-is-a-shortage-of-data-scientists-in-the-industry/

De Haro-García, A., & García-Pedrajas, N. (2009). A divide-and-conquer recursive approach for scaling up instance selection algorithms. *Data Mining and Knowledge Discovery*, *18*(3), 392–418.

Dean, J. (2017). 5 Machine Learning Mistakes – and How To Avoid Them. Retrieved November 3, 2020, from https://www.sas.com/en_us/insights/articles/big-data/5-machine-learning-mistakes.html

Delavar, A. G. N., & Jafari, Z. (2016). One Method to Reduce Data Classification Using Weighting Technique in SVM+. *Modern Applied Science*, *10*(9), 1913–1844.

Demidova, L., Nikulchev, E., & Sokolova, Y. (2016). Big Data Classification Using the SVM Classifiers with the Modified Particle Swarm Optimization and the

SVM Ensembles. *International Journal of Advanced Computer Science and Applications*, *7*(5), 294–312.

Ding, H., & Xu, J. (2015). Random Gradient Descent Tree: A Combinatorial Approach for SVM with Outliers. In *Twenty-Ninth AAAI Conference on Artificial Intelligence* (pp. 2561–2567).

Duong, H. T., & Truong Hoang, V. (2019). A Survey on the Multiple Classifier for New Benchmark Dataset of Vietnamese News Classification. In *2019 11th International Conference on Knowledge and Smart Technology, KST 2019* (pp. 23–28).

Eichelberger, R. K., & Sheng, V. S. (2013). An empirical study of reducing multi-class classification methodologies. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 505–519).

Fan, H., & Ramamohanarao, K. (2005). A weighting scheme based on emerging patterns for weighted support vector machines. *IEEE International Conference on Granular Computing*, *2*(2), 435–440.

Fan, R. E., Chen, P. H., & Lin, C. J. (2005). Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, *6*(12), 1889–1918.

Farid, D. M., Maruf, G. M., & Rahman, C. M. (2013). A new approach of Boosting using decision tree classifier for classifying noisy data. In *International Conference on Informatics, Electronics and Vision, ICIEV* (pp. 1–4).

Frénay, B, & Kabán, A. (2014). A Comprehensive Introduction to Label Noise. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 23–25.

Frénay, Benoît, & Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(5), 845–869.

Ganegedara, T. (2018). Intuitive Guide to Understanding Decision Trees. Retrieved May 5, 2020, from https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-understanding-decision-trees-adb2165ccab7

Garcia, L. P. F. (2016). *Noise detection in classification problems*. University of Sao Paulo.

Glasmachers, T., & Igel, C. (2006). Maximum-Gain Working Set Selection for SVMs. *Journal of Machine Learning Research*, *7*, 1437–1466.

Glen, S. (2019). Weighting Factor, Statistical Weight: Definition, Uses. Retrieved April 4, 2020, from https://www.statisticshowto.com/weighting-factor/#:~:text=1.,Weight and the Weighting Factor.&text=It is usually used for,medicine for calculating effective doses.

Gonzalez, W. (2020). Today's AI Solutions Require Quality, Unbiased Training Data. Retrieved October 20, 2020, from https://www.forbes.com/sites/forbesbusinesscouncil/2020/08/26/todays-ai-solutions-require-quality-unbiased-training-data/

Gupta, S., & Gupta, A. (2019). Dealing with noise problem in machine learning data-sets: A systematic review. In *Procedia Computer Science* (pp. 466–474).

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The Elements of Statistical Learning Data Mining, Inference, and Prediction - Second Edition*. Springer, New York.

Helmenstine, A. M. (2020). What Is the Difference Between Accuracy and Precision? Retrieved December 22, 2020, from https://www.thoughtco.com/difference-between-accuracy-and-precision-609328

Huang, X., Shi, L., & Suykens, J. A. K. (2015). Sequential minimal optimization for SVM with pinball loss. *Neurocomputing*, *149*(3), 1596–1603.

Jadari, S. (2019). *Finding mislabeled data in datasets: A study on finding mislabeled data in datasetsby studying loss function*. Uppsala Universitet.

Jaiswal, S. (2018). Regression vs. Classification in Machine Learning. Retrieved November 13, 2020, from https://www.javatpoint.com/regression-vs-classification-in-machine-learning

Jha, A., Dave, M., & Madan, S. (2019). Comparison of Binary Class and Multi-Class Classifier Using Different Data Mining Classification Techniques. In *ICACM* (pp. 1–10).

Jiang, L., Luo, S., & Li, J. (2012). An approach of household power appliance monitoring based on machine learning. In *Proceedings - 2012 5th International Conference on Intelligent Computation Technology and Automation, ICICTA 2012* (pp. 577–580).

Joshi, N. (2019). Robotic Process Automation Just Got “Intelligent” Thanks to Machine Learning. Retrieved February 2, 2019, from https://www.forbes.com/sites/cognitiveworld/2019/01/29/robotic-process-automation-just-got-intelligent-thanks-to-machine-learning/?sh=1ee7156b52d8

Kao, W., Chung, K., Sun, C., & Lin, C. (2004). Decomposition methods for linear support vector machines. *Neural Computation*, *16*, 1689–1704.

Karasuyama, M., Harada, N., Sugiyama, M., & Takeuchi, I. (2012). Multi-parametric solution-path algorithm for instance-weighted support vector machines. *Machine Learning*, *88*(3), 297–330.

Karthikeyan, T., & Revathy, N. P. (2014). Improved Edge Detection Method by using Weighted Support Vector Machines. *International Journal of Advanced Research in Computer Science*, *5*(6), 255–260.

Kim, S. (2016). Weighted K-means support vector machine for cancer prediction. *SpringerPlus*, *5*(1), 1162.

Lapin, M., Hein, M., & Schiele, B. (2014). Learning using privileged information: SVM+ and weighted SVM. *Neural Networks*, *53*, 95–108.

Lee, H. G., Kim, Y. S., Jeong, C. Y., Han, S. W., & Nam, T. Y. (2005). Multi-Level Objectionable Text Classification Using SVM and Non-Harmful Document Screen. In *The 4th International Conference on Asian Language Processing and Information Technology* (pp. 1–8).

Leyva, E., González, A., & Pérez, R. (2015). Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective. *Pattern Recognition*, *48*(4), 1523–1537.

Li, H. X., Yang, J. L., Zhang, G., & Fan, B. (2013). Probabilistic support vector machines for classification of noise affected data. *Information Sciences*, *221*, 60–71.

Li, J., Wong, Y., Zhao, Q., & Kankanhalli, M. S. (2019). Learning to learn from noisy labeled data. In *Proceedings of the IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition (pp. 1–9).

Li, Y., Hu, Z., Cai, Y., & Zhang, W. (2005). Support vector based prototype selection method for nearest neighbor rules. In *Lecture Notes in Computer Science* (pp. 528–535).

Liu, C., Wang, W., Wang, M., Lv, F., & Konan, M. (2017). An efficient instance selection algorithm to reconstruct training set for support vector machine. *Knowledge-Based Systems*, *116*, 58–73.

Liu, H., & Motoda, H. (2001). Data Reduction via Instance Selection. In *Instance Selection and Construction for Data Mining* (pp. 3–20). Springer US.

Loukas, S. (2020). Is your model overfitting? Or maybe underfitting? An example using a neural network. Retrieved September 2, 2020, from https://towardsdatascience.com/is-your-model-overfitting-or-maybe-underfitting-an-example-using-a-neural-network-in-python-4faf155398d2

Low, J. Y. (2020). The Importance of Good Quality Training Data. Retrieved March 10, 2020, from https://blog.supahands.com/2020/01/31/the-importance-of-good-quality-training-data/

Luxburg, U. von, & Schölkopf, B. (2011). *Statistical Learning Theory: Models, Concepts, and Results*. *Handbook of the History of Logic*. Elsevier North Holland.

Lyhyaoui, A., Martínez, M., Mora, I., Vázquez, M., Sancho, J. L., & Figueiras-Vidal, A. R. (1999). Sample selection via clustering to construct support vector-like classifiers. *IEEE Transactions on Neural Networks*, *10*(6), 1474–1481.

Maletic, J., & Marcus, A. (2000). Data Cleansing: Beyond Integrity Analysis. (pp. 1–10).

Martinez, T. R., & Zeng, X. (2014). *US 8, 788, 439 B2*.

Mathews, B., & Aasim, O. (2021). Common Machine Learning Algorithms for Beginners. Retrieved August 16, 2021, from https://www.dezyre.com/article/common-machine-learning-algorithms-for-beginners/202

Matić, N., Guyon, I., Bottou, L., Denker, J., & Vapnik, V. (1992). Computer aided cleaning of large databases for character recognition. In *Proceedings* -

International Conference on Pattern Recognition (Vol. 2, pp. 330–333).

Maxwell, A. E., Warner, T. A., & Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. *International Journal of Remote Sensing*, *39*(9), 2784–2817.

Mohd Dzulkifli, S. A., Mohd Salleh, M. N., & Leman, A. M. (2017). Customer and performance rating in QFD using SVM classification. In *AIP Conference Proceedings* (pp. 1–8).

Monnappa, A. (2021). Data Science vs. Big Data vs. Data Analytics. Retrieved July 24, 2021, from https://www.simplilearn.com/data-science-vs-big-data-vs-data-analytics-article

Morales, P., Luengo, J., Garcia, L. P. F., Lorena, A. C., de Carvalho, A. C. P. L. F., & Herrera, F. (2017). The Noise Filters R package: Label noise preprocessing in R. *The R Journal*, *9*(1), 219–228.

Mourad, S., Tewfik, A., & Vikalo, H. (2017). Data subset selection for efficient SVM training. In *25th European Signal Processing Conference, EUSIPCO 2017* (pp. 833–837).

Mushtaq, M.-S., & Mellouk, A. (2017). Methodologies for Subjective Video Streaming QoE Assessment. *Quality of Experience Paradigm in Multimedia Services*, 27–57.

Nair, A. (2017). 5 Common Machine Learning Problem and How to Solve Them. Retrieved December 12, 2017, from https://www.provintl.com/blog/5-common-machine-learning-problems-how-to-beat-them

Nalepa, J., & Kawulok, M. (2018). Selecting training sets for support vector machines: a review. In *Artificial Intelligence Review* (pp. 857–900).

Napolitano, A. (2009). *Classification Techniques for Noisy and Imbalanced Data*. Florida Atlantic University.

Nazari, Z., & Kang, D. (2015). Density Based Support Vector Machines for Classification, *4*(4), 69–76.

Nazari, Z., Nazari, M., Danish, M. S. S., & Kang, D. (2018). Evaluation of Class Noise Impact on Performance of Machine Learning Algorithms. *International Journal* *of Computer Science and Network Security*, *18*(8), 148–153.

Nettleton, D. F., Orriols-Puig, A., & Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, *33*(4), 275–306.

Neville, J. (2000). *Iterative Classification*.

Nguyen, M. H., & de la Torre, F. (2010). Optimal feature selection for support vector machines. In *Pattern Recognition* (Vol. 43, pp. 584–591).

Nkoana, R. (2011). *Artificial Neural Network Modelling of Flood Prediction and Early Warning*. University of The Free State Bloemfontein.

Novikov, D. (2020). How to Implement a Machine Learning Algorithm in Code. Retrieved November 2, 2020, from https://resources.experfy.com/ai-ml/how-to-implement-a-machine-learning-algorithm-in-code/

Olson, D. L., & Delen, D. (2008). *Advanced Data Mining Techniques*. Springer Berlin.

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., & Kittler, J. (2010). A review of instance selection methods. In *Artificial Intelligence Review* (pp. 133–143).

Ortner, A. (2020). Top 10 Binary Classification Algorithms. Retrieved July 23, 2020, from https://medium.com/@alex.ortner.1982/top-10-binary-classification-algorithms-a-beginners-guide-feeacbd7a3e2

Panda, N., Chang, E. Y., & Wu, G. (2006). Concept boundary detection for speeding up SVMs. In *ACM International Conference Proceeding Series* (pp. 681–688).

Parikh, K. S., & Shah, T. P. (2016). Support Vector Machine – A Large Margin Classifier to Diagnose Skin Illnesses. In *Procedia Technology* (Vol. 23, pp. 369–375).

Paulsen, V. I., & Raghupathi, M. (2016). *An introduction to the theory of reproducing kernel Hilbert spaces*. *Cambridge Studies in Advanced Mathematics*. Cambridge University Press.

Pearlman, S. (2019). What is Data Preparation? Retrieved April 4, 2021, from https://www.talend.com/resources/what-is-data-preparation/

Pelletier, C., Valero, S., Inglada, J., Champion, N., Sicre, C. M., & Dedieu, G. (2016).

Effect of Training Class Label Noise on Classification Performances for Land Cover Mapping with Satellite Image Time Series. *Remote Sensing*, 1–23.

Pérez-Ortiz, M., Jiménez-Fernández, S., Gutiérrez, P. A., Alexandre, E., Hervás-Martínez, C., & Salcedo-Sanz, S. (2016). A review of classification problems and algorithms in renewable energy applications. *Energies*, *9*(8), 1–27.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods* (pp. 185–208). MIT Press.

Prati, R. C., Luengo, J., & Herrera, F. (2019). Emerging topics and challenges of learning from noisy data in nonstandard classification: a survey beyond binary class noise. In *Knowledge and Information Systems* (Vol. 60, pp. 63–97).

Prem. (2021). A Simple Introduction to the k-Nearest Neighbour (kNN) Algorithm. Retrieved January 12, 2021, from https://www.iunera.com/kraken/fabric/k-nearest-neighbour-knn-algorithm/

Priyadarshiny, U. (2019). Introduction to Classification Algorithms. Retrieved November 26, 2019, from https://dzone.com/articles/introduction-to-classification-algorithms

Pruengkarn, R., Fung, C. C., & Wong, K. W. (2015). Using misclassification data to improve classification performance. In *12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology* (pp. 1–5).

Qi, B., Zhao, C., Youn, E., & Nansen, C. (2011). Use of weighting algorithms to improve traditional support vector machine based classifications of reflectance data. *Optics Express*, *19*(27), 26816.

Qin, T., Zhang, X. D., Wang, D. S., Liu, T. Y., Lai, W., & Li, H. (2007). Ranking with multiple hyperplanes. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'07* (pp. 279–286).

Rani, S. N., & Rao, S. (2019). Study and Analysis of Noise Effect on Big Data Analytics. *International Journal of Management, Technology and Engineering*, *8*(12), 5841–5850.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In *Encyclopedia of*

Database Systems (pp. 532–538).

Riquelme, J. C., Aguilar-Ruiz, J. S., & Toro, M. (2003). Finding representative patterns with ordered projections. *Pattern Recognition*, *36*(4), 1009–1018.

Sabzevari, M. (2015). *Ensemble Learning in the Presence of Noise*. Universidad Autonoma de Madrid.

Sáez, J. A., Galar, M., Luengo, J., & Herrera, F. (2014). Analyzing the presence of noise in multi-class problems: Alleviating its influence with the One-vs-One decomposition. *Knowledge and Information Systems*, *38*(1), 179–206.

Sáez, J. A., Luengo, J., & Herrera, F. (2016). Evaluating the classifier behavior with noisy data considering performance and robustness: The Equalized Loss of Accuracy measure. *Neurocomputing*, *176*, 26–35.

Samanta, D. (2018). Support Vector Machine. *IIT Kharagpur*.

Sanz, H., Reverter, F., & Valim, C. (2020). Enhancing SVM for survival data using local invariances and weighting. *BMC Bioinformatics*, *21*(193), 1–20.

Sarangam, A. (2021). Difference between Classification and Prediction in Data Mining - An Easy Guide in Just 3 Points. Retrieved April 11, 2021, from https://www.jigsawacademy.com/blogs/data-science/classification-and-prediction-in-data-mining/

Sarkar, T. (2019). Clustering metrics better than the elbow-method. Retrieved November 7, 2019, from https://towardsdatascience.com/clustering-metrics-better-than-the-elbow-method-6926e1f723a6

Saseendran, A. T., Setia, L., Chhabria, V., Chakraborty, D., & Roy, A. B. (2019). Impact of Noise in Dataset on Machine Learning Algorithms. In *Machine Learning Research* (pp. 0–8).

Segata, N., Blanzieri, E., Delany, S. J., & Cunningham, P. (2010). *Noise reduction for instance-based learning with a local maximal margin approach*. *Journal of Intelligent Information Systems* (Vol. 35).

Sen, P. C., Hajra, M., & Ghosh, M. (2018). Supervised Classification Algorithms in Machine Learning: A Survey and Review. In *Advances in Intelligent Systems and Computing* (pp. 99–111).

Shanab, A. A., Khoshgoftaar, T. M., & Wald, R. (2011). Impact of noise and data sampling on stability of feature selection. In *Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011* (Vol. 1, pp. 172–177).

Sharma, M. (2019). Generalization in Machine Learning for better performance. Retrieved June 27, 2019, from https://mathanrajsharma.medium.com/generalization-in-machine-learning-for-better-performance-51bed74a3820

Sirohi, K. (2019). Support Vector Machine (Detailed Explanation). Retrieved September 11, 2019, from https://towardsdatascience.com/support-vector-machine-support-vector-classifier-maximal-margin-classifier-22648a38ad9c

Solomatine, D. P., & Dulal, K. N. (2003). Model trees as an alternative to neural networks in rainfall—runoff modelling. *Hydrological Sciences Journal*, *48*(3), 399–411.

Stanevski, N., & Tsvetkov, D. (2005). Using Support Vector Machine as a Binary Classifier. In *International Conference on Computer Systems and Technologies* (pp. 1–5).

Sun, R., Luo, Z.-Q., & Ye, Y. (2020). On the Efficiency of Random Permutation for ADMM and Coordinate Descent. *Mathematics of Operations Research*, *45*(1), 233–271.

Tagliaferri, L. (2017). An Introduction to Machine Learning. Retrieved January 10, 2018, from https://www.digitalocean.com/community/tutorials/an-introduction-to-machine-learning

Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2019). Classification: Basic Concepts, and Techniques. In *Introduction to Data Mining* (p. 839).

Tavara, S. (2018). *High-performance computing for support vector machines*. University of Skovde.

Tian, J., Gu, H., Liu, W., & Gao, C. (2011). Robust prediction of protein subcellular localization combining PCA and WSVMs. *Computers in Biology and Medicine*, *41*(8), 648–652.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. *Statistics for*

Engineering and Information Science. Springer New York.

Vapnik, V., & Vashist, A. (2009). A new learning paradigm: Learning using privileged information. *Neural Networks*, *22*(6), 544–557.

Wang, Q., Li, B., & Hu, J. (2009). Human resource selection based on performance classification using weighted support vector machine. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *13*(4), 407–415.

Welch, D. (2017). Applied Data Mining and Statistical Learning - When Data is Linearly Separable. Retrieved August 22, 2019, from https://online.stat.psu.edu/stat508/lesson/10/10.1

Wen, C., Guyer, D. E., & Li, W. (2009). Local feature-based identification and classification for orchard insects. *Biosystems Engineering*, *104*(3), 299–307.

Wilson, A. (2019a). A Brief Introduction to Supervised Learning. Retrieved December 18, 2020, from https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590

Wilson, A. (2019b). A Brief Introduction to Unsupervised Learning. Retrieved December 18, 2020, from https://towardsdatascience.com/a-brief-introduction-to-unsupervised-learning-20db46445283

Wu, Y., & Liu, Y. (2013). Adaptively Weighted Large Margin Classifiers. *Journal of Computational and Graphical Statistics : A Joint Publication of American Statistical Association, Institute of Mathematical Statistics, Interface Foundation of North America*, *22*(2), 37–41.

Yang, X. S. (2010). A new metaheuristic Bat-inspired Algorithm. In *Studies in Computational Intelligence* (pp. 65–74).

Yang, X. S. (2012). Flower pollination algorithm for global optimization. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 240–249).

Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *World Congress on Nature and Biologically Inspired Computing, Proceedings* (pp. 210–214).

Yang, Xin-she. (2010). *Nature-Inspired Metaheuristic Algorithms*. *University of Cambridge, United Kingdom*. Luniver Press.

Yang, Xulei, Song, Q., & Wang, Y. (2007). A Weighted Support Vector Machine for Data Classification. *International Journal of Pattern Recognition and Artificial Intelligence*, *21*(5), 961–976.

Yi, K., & Wu, J. (2019). Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 7017–7025).

Zhang, H., & Sun, G. (2002). Optimal reference subset selection for nearest neighbor classification by tabu search. *Pattern Recognition*, *35*(7), 1481–1490.

Zhu, F., Yang, J., Gao, J., & Xu, C. (2016). Extended nearest neighbor chain induced instance-weights for SVMs. *Pattern Recognition*, *60*, 863–874.

Zhu, X., & Wu, X. (2004). Class Noise vs. Attribute Noise: A Quantitative Study. *Artificial Intelligence Review*, *22*(3), 177–210.