# A MATHEMATICAL CONTRIBUTION OF STATISTICAL LEARNING AND CONTINUOUS OPTIMIZATION USING INFINITE AND SEMI-INFINITE PROGRAMMING TO COMPUTATIONAL STATISTICS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SÜREYYA ÖZÖĞÜR-AKYÜZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
DEPARTMENT OF SCIENTIFIC COMPUTING

FEBRUARY 2009

Approval of the thesis:

**A MATHEMATICAL CONTRIBUTION OF STATISTICAL LEARNING AND CONTINUOUS OPTIMIZATION USING INFINITE AND SEMI-INFINITE PROGRAMMING TO COMPUTATIONAL STATISTICS**

submitted by **SÜREYYA ÖZÖĞÜR-AKYÜZ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Scientific Computing, Middle East Technical University** by,

Prof. Dr. Ersan Akyıldız _____
Dean, Graduate School of **Applied Mathematics**

Prof. Dr. Bülent Karasözen _____
Head of Department, **Department of Scientific Computing**

Prof. Dr. Gerhard Wilhelm Weber _____
Supervisor, **Department of Scientific Computing, IAM, METU**

Prof. Dr. John Shawe-Taylor _____
Co-supervisor, **Department of Computer Science, UCL**

**Examining Committee Members:**

Prof. Dr. Volkan Atalay _____
Department of Computer Engineering, METU

Prof. Dr. Gerhard Wilhelm Weber, Germany _____
Department of Scientific Computing, IAM, METU

Prof. Dr. John Shawe-Taylor, UK _____
Department of Computer Science, UCL

Assoc. Prof. İnci Batmaz _____
Department of Statistics, METU

Prof. Dr. Aytül Erçil _____
Department of Electronic Engineering, Sabancı University

**Date :** _____

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: SÜREYYA ÖZÖĞÜR-AKYÜZ

Signature :

# ABSTRACT

A MATHEMATICAL CONTRIBUTION OF STATISTICAL LEARNING AND CONTINUOUS OPTIMIZATION USING INFINITE AND SEMI-INFINITE PROGRAMMING TO COMPUTATIONAL STATISTICS

Özöğür-Akyüz, Süreyya
Ph.D., Department of Scientific Computing
Supervisor : Prof. Dr. Gerhard Wilhelm Weber
Co-Supervisor : Prof. Dr. John Shawe-Taylor

February 2009, 135 pages

A subfield of artificial intelligence, machine learning (ML), is concerned with the development of algorithms that allow computers to "learn". ML is the process of training a system with large number of examples, extracting rules and finding patterns in order to make predictions on new data points (examples). The most common machine learning schemes are supervised, semi-supervised, unsupervised and reinforcement learning. These schemes apply to natural language processing, search engines, medical diagnosis, bioinformatics, detecting credit fraud, stock market analysis, classification of DNA sequences, speech and hand writing recognition in computer vision, to encounter just a few. In this thesis, we focus on Support Vector Machines (SVMs) which is one of the most powerful methods currently in machine learning.

As a first motivation, we develop a model selection tool induced into SVM in order to solve a particular problem of computational biology which is prediction of eukaryotic pro-peptide cleavage site applied on the real data collected from NCBI data bank. Based on our biological example, a generalized model selection method is employed as a generalization for all

kinds of learning problems. In ML algorithms, one of the crucial issues is the representation of the data. Discrete geometric structures and, especially, linear separability of the data play an important role in ML. If the data is not linearly separable, a *kernel function* transforms the nonlinear data into a higher-dimensional space in which the nonlinear data are linearly separable. As the data become heterogeneous and large-scale, single kernel methods become insufficient to classify nonlinear data. Convex combinations of kernels were developed to classify this kind of data [8]. Nevertheless, selection of the finite combinations of kernels are limited up to a finite choice. In order to overcome this discrepancy, we propose a novel method of "infinite" kernel combinations for learning problems with the help of *infinite* and *semi-infinite programming* regarding all elements in kernel space. This will provide to study variations of combinations of kernels when considering heterogeneous data in real-world applications. Combination of kernels can be done, e.g., along a homotopy parameter or a more specific parameter. Looking at all infinitesimally fine convex combinations of the kernels from the infinite kernel set, the margin is maximized subject to an infinite number of constraints with a compact index set and an additional (Riemann-Stieltjes) integral constraint due to the combinations. After a parametrization in the space of probability measures, it becomes semi-infinite. We analyze the regularity conditions which satisfy the Reduction Ansatz and discuss the type of distribution functions within the structure of the constraints and our bilevel optimization problem. Finally, we adapted well known numerical methods of semi-infinite programming to our new kernel machine. We improved the discretization method for our specific model and proposed two new algorithms. We proved the convergence of the numerical methods and we analyzed the conditions and assumptions of these convergence theorems such as optimality and convergence.

Keywords: Statistical Learning Theory, Support Vector Machines, Continuous Optimization, Computational Biology, Infinite Programming, Semi-Infinite Programming, Reduction Ansatz, Discretization, Exchange Methods, Numerical Optimization, Regularization, Data Mining, Inverse Problems

# ÖZ

İSTATİSTİKSEL ÖĞRENME VE SÜREKLİ OPTİMİZASYON YÖNTEMLERİNİN SONSUZ VE YARI SONSUZ PROGRAMLAMA KULLANILARAK HESAPLAMALI İSTATİSTİĞE UYGULANMASI

Özöğür-Akyüz, Süreyya
Doktora, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Prof. Dr. Gerhard Wilhelm Weber
Ortak Tez Yöneticisi : Prof. Dr. John Shawe-Taylor

Şubat 2009, 135 sayfa

Makina öğrenimi, yapay zekanın bilgisayarların öğrenimini sağlayan algoritmaların geliştirilmesi ile ilgilenen bir alt alanıdır. Bu yöntem, sisteme ait kuralları ve sablonları çok fazla sayıda örnek ile eğiterek çıktısı bilinmeyen yeni veri noktalarını tahmin etme sürecidir. Yaygın makine öğrenimi problemleri denetlenmiş öğrenim, denetlenmemiş öğrenim, yarı denetlenmiş öğrenim ve desteklenmiş öğrenim vb. alt başlıklardan oluşur. Bu alan doğal dil işleme, arama motorları, medikal diagnoz, bioinformatik, kredi kartı sahtekarlığı tespiti, borsa analizi, DNA dizilerinin sınıflandırılması, konuşma ve el yazısı tanıma ve obje tanıma gibi pek çok uygulamayı içermektedir. Bu tezde, makina ögrenimi alanları içerisinde en güçlü metodlardan biri olan Destekçi Vektor Makinaları (DVM) üzerine yoğunlaşılacaktır.

İlk motivasyon olarak, NCBI veri bankasından derlenmiş gerçel veri üzerinde ökaryotik propeptid kesim yerlerini sorgulayan biyoloji problemini çözmek için SVM metodunun iç erisinde model seçimi yapan bir araç geliştirilmiştir. Biyolojik problem esas alınarak bulunan bir önceki model seçimi yöntemi, çeşitli veri kümelerine de uygulanabilir halde genelleştirilmiştir. Makine ögrenimi algoritmalarında önemli bir unsur da verinin ifade ya da gösterim biçimidir.

Ayrık geometrik şekiller, özellikle verinin doğrusal olarak ayrilabilirliği makine ögrenimi yöntemlerinde önemli rol oynamaktadır. Doğrusal olarak ayrılamayan veri kümelerinde, çekirdek (kernel) fonksiyonu ile doğrusal olmayan veriler yüksek boyutlu uzaya taşınarak lineer ayrılabilir hale getirilmektedir. Çok boyutlu ve heterojen kaynaklı veri kümelerinde tek çekirdekli sınıflandırma algoritmaları doğrusal olmayan veriyi sınıflandırmakta yetersiz kalmaktadır. Bu tür veriyi sınıflandırmak için çekirdeklerin (kernellerin) dısbükey kombinasyonlarından oluşan çoklu çekirdek öğrenim yöntemi geliştirilmiştir [8]. Buna rağmen çoklu çekirdek ögrenimindeki çekirdeklerin seçimi sınırlı sayı ile kısıtlıdır. Bu eksikliğin giderilmesi için bu tezde çekirdek uzayının tüm elemanlarını kapsayan sonsuz ve yarısonsuz programlama ile modellenen sonsuz çekirdek öğrenimi yöntemi önerilmiştir. Sonsuz çekirdek ögrenimi sayesinde gerçel hayat problemlerinde karşımıza çıkan heterojen ve çok boyuttaki veri kümelerinin sınıflandırıldığı durumlarda, olası bütün çeşitleri kapsayan çekirdeklerin (kernel) kombinasyonları incelenmiş olacaktır. Çekirdeklerin kombinasyonları homotopi parametreleri sayesinde ifade edilmiştir. Sonsuz çekirdek uzayında Riemann-Stieltjes integrali ile sonsuz sayıdakı çekirdeğin kombinasyonuna bakılarak, tıkız sonsuz indeks seti altında iki sınıf arasındaki uzaklık maksimize edilmiştir. Sonsuz programlama olarak modellenen sınıflandırma problemi, paramterizasyon ile yarı sonsuz programlamaya indirgenmiştir. İndirgeme ansatz gerekliliklerini sağlayan, düzenlilik koşulları incelenerek, kısıt yapılarıve iki seviyeli optimizasyon problemi içerisinde çeşitli dağılım fonksiyonları analiz edilmiştir. Son olarak yarısonsuz programlamaya uygulanan bilinen nümerik yöntemler önerdiğimiz çekirdek makinasına uyarlanmıştır. Önerilen model için uyarlanan ayrıştırma yöntemini geliştirilip iki ayrı algoritma geliştirilmiştir. Bu problemin nümerik yöntemler ile teorik bazlı analizi yapılmış ve optimal sonucun varlığı ve yakınsaması için gerekli koşullar araştırılmıştır.

Anahtar Kelimeler: İstatistiksel Öğrenme, Destekçi Vektör Makinaları, Sürekli Optimizasyon, Sonsuz Programlama, Yarısonsuz Programlama,, Indirgeme Koşulları, Ayrıştırma, Değişim Metodu, Numerik Optimizasyon, Düzenleştirme, Veri Madenciliği, Ters Problemler

*To my family*

# ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Prof. G.-Wilhelm Weber and my co-supervisor Prof. John Shawe-Taylor for their precious support and guideness through the thesis. This thesis wouldn't have been possible without the inspiration of Prof. G.-Wilhelm Weber and his encouragement. I would like to special thanks to Prof. John Shawe-Taylor for his host and support financially and scientifically through my visit at University College London(UCL). I am grateful for his patience and continuation of his guidance after my visit.

I am grateful to Prof. John Shawe-Taylor's research group for their unlimited help, precious discussion and collaboration. I especially deeply thank to Dr. Zakria Hussain and Dr. David Hardoon for their endless support and help.

I thank to supportive staff at UCL, especially to Wendy Hall and Dawn Bailey for their friendship and kind understanding.

I am grateful to Prof. Z.B. Ögel and Prof. V. Atalay for their valuable discussions, and their precious time and support. I would also like to take this opportunity to thank Prof. Aytül Erçil for her recommendations, motivation and support.

I am also thankful to Prof. Edward Anderson and Prof. Miguel Goberna, Prof. Yıldıray Ozan and Prof. Uluğ Çapar for their valuable discussion and their time.

I am grateful to the head of IAM institute, Prof. Ersan Akyıldız and the head of Scientific Computing Department, Prof. Bülent Karasözen, for introducing me the institute and for their supports during my research.

I thank the supportive IAM staff especially, Nejla Erdoğdu and Rukiye Ekinci who addressed all my queries.

I deeply thank the members of the Institute of Applied Mathematics and my friends for their suggestions during the period of writing the thesis.

Many thanks go to the VPA Lab students and staffs at Sabancı University for their friendship

and patience. I would like to special thank to Ayül Erçil for her support and advices during my stay at Sabancı University.

Finally, I would like to special thank to my husband Akın Akyüz and many thanks to our families for their endless supports. I am grateful that I have such great parents in my life and I am thankful for their guidance and their life experience. I would like to special thanks to my dad and mum for all kinds of supports and for the effort in providing the best conditions anytime.

# TABLE OF CONTENTS

ABSTRACT . . . . . iv

ÖZ . . . . . vi

DEDICATION . . . . . viii

ACKNOWLEDGMENTS . . . . . ix

TABLE OF CONTENTS . . . . . xi

LIST OF TABLES . . . . . xiv

LIST OF FIGURES . . . . . xv

CHAPTERS

1 INTRODUCTION . . . . . 1

1.1 CLASSIFICATION PROBLEM IN MACHINE LEARNING . . . . 1

2 BACKGROUND . . . . . 6

2.1 INTRODUCTION . . . . . 6

2.2 SUPPORT VECTOR MACHINES . . . . . 6

2.3 RIEMANN-STIELTJES INTEGRALS . . . . . 11

2.3.1 INTRODUCTION . . . . . 11

2.3.2 RIEMANN SUMS AND INTEGRABILITY . . . . . 12

2.3.3 PROPERTIES OF RIEMANN-STIELTJES INTEGRALS 13

2.4 SEMI-INFINITE PROGRAMMING . . . . . 16

2.4.1 INTRODUCTION . . . . . 16

2.4.2 PROBLEM DEFINITION . . . . . 17

2.4.3 CONSTRAINT QUALIFICATIONS . . . . . 18

2.4.4 FIRST-ORDER OPTIMALITY CONDITIONS . . . . . 20

2.4.5 CONVEX AND LINEAR SEMI-INFINITE PROGRAMMING . . . . . 21

2.4.6 SECOND-ORDER OPTIMALITY CONDITIONS . . . . 24

3 PATTERN ANALYSIS FOR THE PREDICTION OF FUNGAL PRO-PEPTIDE CLEAVAGE SITES . . . . . 28

3.1 INTRODUCTION . . . . . 28

3.2 BIOLOGICAL BACKGROUND . . . . . 29

3.2.1 PROTEIN SYNTHESIS . . . . . 29

3.2.2 N-TERMINAL PRO-PEPTIDES OF FUNGAL SECRETED PROTEINS . . . . . 31

3.2.3 FUNCTIONS OF PRO-PEPTIDES . . . . . 32

3.2.4 DIBASIC PROCESSING . . . . . 32

3.2.5 MONOBASIC PROCESSING . . . . . 33

3.3 MATERIALS AND METHODS . . . . . 33

3.3.1 INPUT AND OUTPUT FOR THE SVM . . . . . 34

3.3.2 SLIDING WINDOW APPROACH FOR CONSTRUCTING A TEST SET . . . . . 35

3.3.3 KERNEL DEFINITIONS . . . . . 36

3.4 MODEL SELECTION PROCEDURE . . . . . 39

3.5 RESULTS AND DISCUSSION . . . . . 40

3.6 CONCLUSION AND PERSPECTIVES . . . . . 42

4 MODEL SELECTION ALGORITHMS . . . . . 43

4.1 INTRODUCTION . . . . . 43

4.2 CROSS VALIDATION . . . . . 43

4.3 SVM MODEL SELECTION BASED ON OBSERVED MARGIN . 45

4.3.1 INTRODUCTION . . . . . 45

4.4 METHODS . . . . . 45

4.5 DATA SET DESCRIPTION . . . . . 49

4.6 RESULTS . . . . . 50

4.7 CONCLUSION . . . . . 51

5 INFINITE KERNEL LEARNING . . . . . 55

5.1 INTRODUCTION . . . . . 55

5.2 MULTIPLE KERNEL LEARNING . . . . . 57

5.3 LEARNING WITH INFINITE KERNELS . . . . . 59
5.3.1 COMBINATION OF INFINITELY MANY KERNELS . . 59
5.3.2 DUAL PROBLEM . . . . . 63
5.3.3 PRIMAL PROBLEM . . . . . 67
5.4 REGULARIZATION OF INFINITE PROGRAMMING MODEL WITH RESPECT TO KERNEL COEFFICIENTS . . . . . 81
5.5 DIFFERENT PARAMETRIZATION FUNCTIONS FOR INFINITE PROBLEM . . . . . 85
5.6 SUMMARY OF INFINITE KERNEL LEARNING AND ITS ADVANTAGES . . . . . 87
5.7 CONCLUSION . . . . . 88
6 NUMERICAL TREATMENT OF INFINITE KERNEL LEARNING PROBLEM . . . . . 90
6.1 INTRODUCTION . . . . . 90
6.2 DISCRETIZATION . . . . . 91
6.2.1 CONCEPTUAL DISCRETIZATION METHOD . . . . . 95
6.3 EXCHANGE METHOD . . . . . 112
6.3.1 CONCEPTUAL REDUCTION METHOD . . . . . 116
7 CONCLUSION . . . . . 120
REFERENCES . . . . . 124
VITA . . . . . 130

# LIST OF TABLES

TABLES

Table 3.1 Accuracy results of SVM [53]. . . . . 41

Table 3.2 Average training time for the SVM for one of the 72 test sequence [53]. . . 41

Table 4.1 Description of the data sets chosen from UCI machine learning repository. First column represents the name of the data set, second column represents the number of data, third column represents the number of features, the fourth and fifth columns represent the numbers of the positive and the negative examples, respectively. . . . . 50

Table 4.2 $L_\infty$-norm results against SVM with Cross-Validation [52]. . . . . 52

Table 4.3 $L_1$- and $L_2$-norm results [52]. . . . . 53

# LIST OF FIGURES

FIGURES

Figure 2.1 Nonlinear mapping from input space to feature space [19]. . . . . 9

Figure 2.2 Geometric margin [19]. . . . . 10

Figure 2.3 An example illustrating the minimizing constraint function, $g(\bar{\mathbf{x}}, \cdot)$ and the Reduction Ansatz, an example. . . . . 27

Figure 3.1 RNA Translation process [16]. . . . . 30

Figure 3.2 Illustration of pro-peptide cleavage sites [17]. . . . . 31

Figure 3.3 Black parts denote pro-peptide region and white parts stand for the mature part of the protein [17]: ($a$) The window does not containing the cleavage site. ($b$) The window contains the cleavage site at its center. ($c$) The window does not contain the cleavage site. . . . . 36

Figure 3.4 Different real-valued outputs from the SVM. The confidence level is the highest difference between two maximum positive outputs. . . . . 40

Figure 4.1 The graph of $L_\infty$-norm which predicts the example as +1 where actual class is +1. Each bar corresponds to the functional margin value given for that particular SVM model $f$. Bold bars corresponds to $\max\{f_i(x)\}_{i=1}^{\ell}$ and $\min\{f_i(x)\}_{i=1}^{\ell}$ [52]. . . 47

Figure 4.2 Illustration of $L_1-$norm approach. Each bar corresponds to the functional margin value given for that particular SVM model $f$. The $L_1$-norm predicts +1, and the actual class of the example is +1 [52]. . . . . 48

Figure 4.3 Illustration of $L_2$-norm approach. Each bar corresponds to the functional margin value given for that particular SVM model $f$. The $L_2$-norm predicts +1, where the actual class of the example is +1 [52]. . . . . 49

Figure 5.1 On the homotopy between two kernels, example. . . . . 60

Figure 5.2 Homotopy function which starts at $H(0)$ and combines kernels until $H(1)$ is reached (a symbolic illustration). . . . . 60

Figure 5.3 Active constraints, the red dots are degenerate points; two examples. . . . . 70

Figure 5.4 Active constraints with regular points in the perturbed problem; two examples. . . . . 71

Figure 6.1 Illustration of the minimum values with respect to different feasible sets corresponding different discretizations; an example. . . . . 94

Figure 6.2 Transversal intersection (excision) of the feasible set with box; an example taken from [64, 88]. (The surface may come from an equality constraint; the figure implies perturbational arguments of [88].) . . . . . 99

Figure 6.3 Illustration of the transversal cutting around the height function with a box; an example. . . . . 100

Figure 6.4 Illustration of the 2-simplex in $\mathbb{R}^3$. . . . . 105

Figure 6.5 Illustration of the algorithmic way of finding corner points of $A$; an easy example for $l = 3$ (in $\mathbb{R}^3$). . . . . 107

Figure 6.6 Nonuniform sampling of a standard simplex $\Delta^N$, an example in $\mathbb{R}^3$, $\Delta_1 \neq \Delta_2 \neq \Delta_3 \neq \Delta_4$. . . . . 108

Figure 6.7 Transformation of the barycentric coordinates of a polytope to a sphere [91]. 108

Figure 6.8 Discretization of the sphere; an example [91]. . . . . 109

Figure 6.9 Illustration of the local discretization in $\mathbb{H}^k$, $P(f, g^{0,\nu}, u^{\mathcal{P}}, v^{\mathcal{P}})$ is the discretized problem and $P(f, g^{\mathcal{P}}, u^{\mathcal{P}}, v^{\mathcal{P}}, u, v)$ is the primal SIP problem, where $\nu$ is the number of grid points, $f$ is the objective function and $g^{\mathcal{P}}$ is the inequality constraint of the SIP problem; an example [88]. . . . . 111

Figure 6.10 Illustration of the (local) linearization of $A$, with linear $u$ and $v$. . . . . 112

# CHAPTER 1

# INTRODUCTION

This dissertation presents a new approach to model selection and kernel methods in machine learning (ML), specifically for support vector machines (SVMs), by using infinite programming (IP) and semi-infinite programming (SIP). Machine learning is a subfield of artificial intelligence (AI) which deals with extracting rules and patterns from large data sets when training points are provided at input, aiming at generating a prediction on new data sets.

The purpose of this chapter is: 1) to explain classification problems in literature; 2) to introduce the problem addressed in this dissertation; 3) to summarize the current state of SVMs and machine learning domains to discuss the needs for new kernel methods; as well as to provide a concise description of the approach introduced in this work.

In the following section, we point out the main contributions and define the outline of the dissertation.

## 1.1 CLASSIFICATION PROBLEM IN MACHINE LEARNING

The availability in recent years of large databases in biology, chemistry,engineering sciences has posed new problems and challenges to the scientific community. In this context, classification is still a major conundrum and central research topic [19, 52, 53, 54, 71]. Therefore, it is very difficult to analyze and understand the behaviour or structure of the data by human capability. Computers take place the human work load by artificial intelligence techniques. Machine learning is a subfield of artificial intelligence which facilitates (or promotes) learning rules that characterize input data, and provide a framework that allows for predictions when new samples are tested. Classification problems belong to the major tasks in ML which has

many application areas in real-world problems such as bioinformatics, biomedicine, cancer research, image processing, computer vision, finance, marketing and business.

In this thesis, one of the most powerful methods, SVMs, is discussed for the classification task. The target of an SVM is to classify the data by maximizing the margin (distance) between classes by a hyperplane. Mathematically, it corresponds to solve an (optimization) problem which provides to find the best classifier on a given set of examples. The prediction step is done by testing new samples on a best classifier function which has already been derived from optimization problem before. The choose of the best classifier depends on model parameters; this implies the model selection part of the classification problem. If the data are not linearly separable, such a hyperplane cannot be found by the SVM problem. In such cases, a transformation technique of the data points to the higher-dimensional space is given by the so-called *kernel methods*, aiming classification of data linearly by using a nonlinear mapping with a kernel function. As a result, there are two major problems when solving classification problems: 1) *model selection* and 2) *kernel learning*. In real-world applications of heterogeneous source of data and large scale data, multiple kernel learning (MKL) has been developed [8, 71]. MKL allows to enlarge the selection and use the facility of a combination of different kernels for heterogeneous kinds of data. With the help of multiple kernels, the similarity measurement becomes more effective if the data are generated from a heterogeneous source. Detailed information can be found in [8] with real-world examples are also provided. We give a brief introduction on MKL in Chapter 5.

One of the issues in the design of the SVMs application is the model selection phase, i.e., the parameter selection for the best classifier. In statistical learning methods [30], cross validation (CV) (see Chapter 2) or heuristic searches are used to find the optimum parameters. These methods are computationally expensive when the search space is large and the dimension of the data is high. The first contribution of this thesis is on the model selection phase of the SVM which is specifically developed for a biological problem. We also generalize the model selection phase to any kind of data. Our biological problem is to find pro-peptide cleavage sites of fungi proteins for given amino acid sequences. Finding critical positions in amino acid sequences has been studied for many years [7, 9, 12, 22, 23, 31, 48, 50] and still preserves its importance in bioinformatics studies. We collected the data from NCBI[1] protein data bank specifically for fungi proteins (see Chapter 3) and developed a model selection

[1] http://www.ncbi.nlm.nih.gov/

for the prediction of pro-peptide cleavage sites. Our model selection algorithm is based on the predefined *confidence level* for the selection of the best classifier on the *test phase*. Let us note that the classifiers which are provided from the training set are used to decide the class of the new point by a predefined confidence level. Furthermore, the confidence level is measured by using the value of the classifiers on the test points which is resembled by the principle "*The bigger the confidence level is, the better the classifier is*". The confidence level enables us to choose classifiers which are specific to their own protein sequence. In other words, each protein sequence has its own classifier for the prediction of pro-peptide region and each protein sequence is window based analyzed. Training and testing sequences are chosen by a fixed pre-chosen window of amino acids. In fact, the classifier is chosen on the test phase with the confidence interval notion. This methodology saves a lot of training time, and the results show a comparable accuracy when compared with cross validation and with other classification methods (e.g., neural networks [23]).

The second contribution of this thesis is the generalization of the model selection to different kinds of data sets. We generalized the model selection scheme as "classification on observed margin". Our method benefits from all the classifiers of the training process and selects the best according to different norms defined on functional margin. We defined three kinds of $L_p$-norms ($p = 1, 2, \infty$) and compared the results with cross validation with respect to time and also error percentages. With this new generalized model selection method, we saved a lot of time in training when our method is compared with cross validation. Among these norms, $L_\infty$ presented the best error percentage. Furthermore, our new method gives good performance (error rate) for the unbalanced data sets. We also applied our methodology to unbalanced data sets and got meaningful results (see Chapter 4).

The third contribution of the thesis is related to the kernel selection for the SVM. In classical kernel learning methods, a single kernel is used to map the input space to a higher dimensional feature space. But for large scale and heterogeneous data in real-world applications, multiple kernel learning is developed [8, 71]. The main intuition behind multiple kernel learning is to combine finitely many pre-chosen kernels in a convex combination. However, this approach has some limitations on the kernel search space since the combination depends on the problem and the selection of the user from a discrete set of kernels. In [8], a multiple kernel reformulation is modeled by semi-definite programming for selecting the optimum weights of corresponding kernels. This reformulation has some drawbacks in computation time because

of semi-definite programming and this reformulation is developed in [71] by semi-infinite linear programming. We improved the multiple kernel learning and semi-infinite reformulation by enlarging the kernel search space by constituting a continuous domain of kernels in which infinitely many kernels play in a Riemann-Stieltjes integral form. By this new formulation, we have the opportunity of recording ("scanning") all possible choices of kernels from the kernel space and, hence, the uniformity is also preserved. We proposed *homotopy function* to give an idea of infinite combination of kernels. We model this idea by *infinite programming* which has infinitely many constraints, and the problem variables are from an infinite dimensional space which correspond to the kernel coefficients. Let us note that infinitely many kernels correspond with infinitely many coefficients. We defined these kernel coefficient function as an *increasing monotonic function* by means of *positive measures* and, we established Radon measures and Prokhorov distance on these measures.

The regularity conditions are analyzed on the lower level problem of infinite programming. Based on these conditions and some assumptions, we assured the point masses of these infinitely many kernel coefficients. In other words, there exist finitely many active points (finitely many kernels coefficients) because of the assumptions and the regularity conditions (nondegeneracy) given in [88]. Hence, we "scan" all infinite possible kernels from the infinite dimensional search space, and we assure to find the point masses (discrete active constraints) which define the kernel combination under the above assumptions and conditions.

As an alternative way of solving an infinite programming problem, we propose a parametrization of positive measures by *probability density functions*. By this parametrization, our infinite problem turns into a semi-infinite programming problem with infinitely many constraints and having variables from a finite dimensional space. As our last contribution, we propose some numerical methods to solve our parametrized problem. We proposed two new methodologies for the discretization of the infinite index set to be used in these numerical methods. These two strategies are explained with examples and remarks in the last chapter of the thesis. Furthermore, we give a proof of our concept, i.e, convergence of the numerical methods is demonstrated based on some assumptions. These assumptions are analyzed for our problem and the convergence is proved for each numerical method without giving numerical illustration.

We want to remind the scope the thesis: developing new model selection and kernel learning

methods which are established on theoretical foundations, on a mathematical basis and on continuous optimization. In this thesis, we demonstrate and illustrate the idea of our new mathematical model (model selection method) which is applied on a bioinformatics problem and we assure the convergence of the proposed numerical methods for the new infinite kernel learning formulation (by using infinite programming). By this thesis, we introduce a new scientific approach and methodology in statistical learning, and we propose and initialize future research. We want to note that whenever we want to give detailed information we refer "Closer Explanation" throughout the thesis.

# CHAPTER 2

# BACKGROUND

## 2.1 INTRODUCTION

In this chapter, we will give brief and comprehensive background explanations on the mathematical methods which we use to model our classification problem. The fundamental definitions and theorems of three main mathematical and statistical approaches will be introduced which are used in this thesis. We will start with section on Support Vector Machines (SVM) and continue with Riemann-Stieltjes integrals. In the later sections of this chapter, we give the principal theorems of Semi-Infinite Programming. Throughout this chapter, except Section 2, we denote the vectors in bold, i.e., $\mathbf{x}, \mathbf{w}$ and we denote the components by sub indices, i.e., $x_i, w_i$. In Section 2, $x_i$ will denote points on the real line.

## 2.2 SUPPORT VECTOR MACHINES

Data mining is the process of analyzing a massive amount of data and gathering useful information or structure through the analysis. It is a highly demanding area because of the large amount of experimental data in data bases. Various applications of data mining can be found in medicine, finance, business and so forth. There are different types of data mining tools such as statistical analysis, probabilistic methods and machine learning tools.

In recent years, learning methods are desirable because of their reliability and efficiency in real-world problems. In such situations, for instance, in engineering or biological problems, experiments can be very costly and time consuming. In situations like these, accurate and predictive methods are in demand to overcome these difficulties. Furthermore, lots of data

bases are freely accessible at the internet which contain huge data sets. It is important to understand and analyze these data sets to make them beneficial. Different methodologies have been developed to learn the system behaviour in a supervised or in an unsupervised way.

*Supervised learning* is a learning methodology where unseen data can be predicted with the help of observations. Mathematically, for a given set of observations $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$, the classification on new unseen data is performed by a defined function from these observations, where $\mathbf{x}_i \in \mathbb{R}^n \quad (i = 1, 2, \ldots, l)$ are data points (inputs) and $y_i \in \{\pm 1\}$ are corresponding labels (outputs). Here, input vectors, $\mathbf{x}_i$'s are labeled by $y_i = +1$ if $\mathbf{x}_i$'s belong to the positive class, and input vectors, $\mathbf{x}_i$'s are labeled by $y_i = -1$ if $\mathbf{x}_i$'s belong to the negative class. Algorithms are developed based on training examples which consist of input vectors, $\mathbf{x}_i$, and outputs, $y_i$, given to the a learning system that subsequently predicts the outputs for test examples. Training sets are the main resource of supervised learning. Learning is enabled with a function defining a relation between input and output with a functional mapping which is called a *target function*. Estimation of the target function will give a solution of the learning problem which is also called a *decision function*. The solution is selected from a set of *candidate functions* $f \in \mathcal{F}$, where $\mathcal{F}$ is a set of functions [19]:

$$f : \text{input space} \longrightarrow \text{output domain},$$

and these candidate functions are referred to as *hypotheses* [19].

*Binary classification* is frequently performed using linear classification methods.

**Definition 2.2.1** *Let $f$ be a real-valued function defined on a subset $X \subseteq \mathbb{R}^n$,*

$$f : X \longrightarrow \mathbb{R}.$$

*Then, $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is assigned to the* positive class *if $f(\mathbf{x}) \geq 0$, otherwise it is assigned to the* negative class*, i.e., if $f(\mathbf{x}) < 0$. Here, $n$ is the* dimension *of input space.*

In our study, such a function $f(\mathbf{x})$ is requested to be affinely linear, i.e., it can be expressed as

$$\begin{aligned} f(\mathbf{x}) &= \langle \mathbf{w}, \mathbf{x} \rangle + b \\ &= \sum_{i=1}^{n} w_i x_i + b, \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product and $(\mathbf{w}, b)$ consists of the parameters that control the function and decision rule given by $\text{sgn}(f(\mathbf{x}))$,

where

$$\text{sgn}(f(\mathbf{x})) = \begin{cases} 1, & \text{if } f(\mathbf{x}) \geq 0 \\ -1, & \text{if } f(\mathbf{x}) < 0. \end{cases} \tag{2.1}$$

Here, the decision rule refers to the positive class or negative class of the points defined by Definition 2.2.1, and $\mathbf{w}$ is referred to as the *weight vector* and $b$ as the *bias*.

In linear *binary classification*, the two classes are discriminated by a hyperplane defined by $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$.

**Definition 2.2.2** *We define the* (functional) margin *of the examples* $(\mathbf{x}_i, y_i)$ $(i = 1, 2, \ldots, l)$ *with respect to a hyperplane* $(\mathbf{w}, b)$ *to be the quantity*

$$\gamma_i := y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \ \ (i = 1, 2, \ldots, l).$$

Let us note that if $\gamma_i > 0$, then the correct classification is achieved. We know that if $\mathbf{x}_i$ is in the positive class, then $\langle w, \mathbf{x}_i \rangle + b \geq 0$ and if $\mathbf{x}_i$ is in the negative class, then $\langle w, \mathbf{x}_i \rangle + b < 0$ by Definition 2.2.1. The product of $\langle w, \mathbf{x}_i \rangle + b$ with $y_i$ is positive if $\mathbf{x}_i$ is in the positive class since $\langle w, \mathbf{x}_i \rangle + b \geq 0$, and $y_i = 1$. Similarly, the product of $\langle w, \mathbf{x}_i \rangle + b$ with $y_i$ is positive if $\mathbf{x}_i$ is in the negative class since $\langle w, \mathbf{x}_i \rangle + b < 0$ and $y_i = -1$. Hence, $\gamma_i > 0$ for both cases if the correct classification is achieved.

**Definition 2.2.3** *The* geometric margin *is defined by* $\frac{\gamma}{\|w\|}$ *as the distance between the nearest points to the hyperplane (see Figure 2.2).*

Linear separability of the classes of data is one of the essential issues in SVM theory since a hyperplane is a tool to discriminate the classes. The pattern of the data can be discretely nonconvex or some part of the data can belong to one class or group of data, surrounded by the data of the other class. In most of the real-world problems [53], data are not linearly

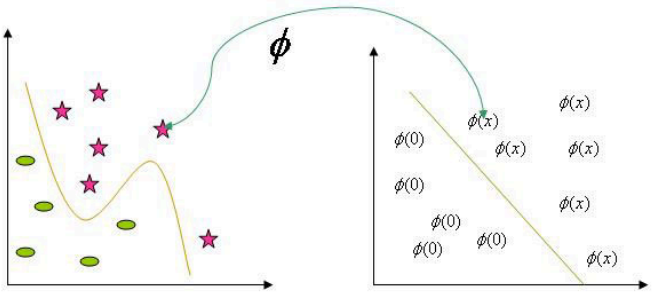


Figure 2.1: Nonlinear mapping from input space to feature space [19].

separable. Thus, the data need to be mapped into another space in which they become linearly separable, see Figure 2.1. The representation of nonlinear data is changed with a nonlinear mapping $\phi$ which transforms the input space into a higher dimensional feature space such that the data points are linearly separable. Then, $f$ and $w$ can be written as:

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b. \tag{2.2}$$

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \phi(\mathbf{x}_i). \tag{2.3}$$

In fact, we can compute $f$ without the explicit feature vectors $\phi(\mathbf{x})$ if we have a direct method for computing $\kappa(\mathbf{x}, \mathbf{z}) := \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$, also referred to as the *kernel* function [19].

SVMs choose the linear classifier that maximizes the geometrical margin on the training data. Since rescaling of $(\mathbf{w}, b)$ does not change classification, we can enforce the margin

$$y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1. \tag{2.4}$$

Then, the functional margin is 1 by normalizing the distance between hyperplane and the points with Hessa normal form . The geometric margin is the distance between the two supportive lines as in Figure 2.2 and it is calculated as $\gamma = \frac{1}{\|\mathbf{w}\|_2}$. Hence, to maximize the margin, it is necessary to minimize $\|\mathbf{w}\|_2^2$ (where $\|\cdot\|_2$ is the Euclidean norm).

Now, we have a convex optimization problem to find the optimum classifier in the following form:

$$\min_{\mathbf{w},b} \langle \mathbf{w}, \mathbf{w} \rangle$$

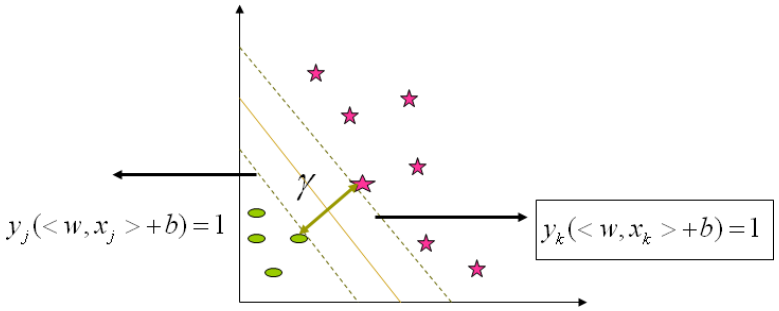


Figure 2.2: Geometric margin [19].

$$\text{such that } \; y_i \cdot (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \geq 1 \; (i = 1, 2, \ldots, l).$$

Usually, the *dual form* of the problem is preferred because of the sparse structure of the dual variable. The dual representation of the classification function can be derived by the help of optimization theory. Using *Lagrangian* and its partial derivatives, with the *Karush-Kuhn-Tucker* (*KKT*) *conditions*, the dual problem is given in the following form [19]:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} y_i y_j \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \\ & \text{subject to} \quad \sum_{i=1}^{l} y_i \alpha_i = 0, \\ & \alpha_i \geq 0 \quad (i = 1, 2, \ldots, l), \end{aligned}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_l)^T$. To solve complex classification problems, it is not enough to apply strictly perfect maximal margin classifiers without any error tolerance. Therefore, new variables are introduced which violates the maximal margin criterion. Then, this classifier is called *a soft margin classifier*.

**Soft Margin Classifier Problem**

Here, a vector of some kind of slack variables $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_l)^T$ is added to the objective function with regularization constant $C$ [19]:

$$\begin{aligned} & \min_{\xi, \mathbf{w}, b} \quad \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{l} \xi_i \\ & \text{subject to} \quad \boldsymbol{\xi} \geq 0 \;\text{ and }\; y_i \cdot (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i \quad (i = 1, 2, \ldots, l). \end{aligned} \tag{2.5}$$

The corresponding dual form can be easily constructed in the same way setting the gradient of the Lagrange function to be equal to zero and writing the KKT conditions [19]. Hence, one

is faced with the following dual optimization problem:

$$\begin{aligned} &\max_{\alpha} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j), \\ &\text{subject to} \quad \sum_{i=1}^{l} y_i \alpha_i = 0, \\ &0 \leq \alpha_i \leq C, \quad (i = 1, 2, \ldots, l), \end{aligned} \tag{2.6}$$

where $\alpha_i$'s are called **support vector coefficients** which are the coefficients of vectors lying on supporting hyperplane. We shall be referring *support vector machine (SVM)* to problem (2.6).

In the following section, we will introduce Riemann-Stieltjes integrals which are used for infinite combination of kernels. We will give the main theorems and definitions without further proofs and details, and refer the reader to [70].

## 2.3 RIEMANN-STIELTJES INTEGRALS

### 2.3.1 INTRODUCTION

*Calculus* studies limits, integrals, derivatives and infinite series, it is historically referred to as *infinitesimals* or *infinitesimal calculus* [39, 79]. It provides us to understand the collection of data using small discrete increments. Calculus has wide applications in science and engineering to solve problems where *algebra* is insufficient. Calculus is developed by manipulating small quantities. On a number line, these infinitesimal quantities correspond to locations which are *not zero but of a zero distance* from zero. No nonzero number is infinitesimal since its distance from zero is positive, and any multiple of an infinitesimal is still infinitely small. From this point of view, calculus is a collection of techniques for manipulating infinitesimals [39, 79]. From this aspect, the infinitesimal changes and infinite combinations of kernel functions in classification problems are one of the main challenges in this thesis.

The idea of infinitesimal is changed by *limits* later in the 19th century. Limits describe the value of a function at a certain input in terms of its values at a nearby input. As in infinitesimals, limits capture the small changes but with ordinary numbers. Based on discrete points and their limits, *integral calculus* is developed. It is established on *Riemann sums* which represents a summation over the infinitesimal collection of intervals. Furthermore, limits provide to get a result of this infinitesimal collection by a number with the help of convergence

analysis. Finally, we use the notion of the *Riemann integral* [4, 39] for the the limit of all corresponding Riemann sums.

### 2.3.2 RIEMANN SUMS AND INTEGRABILITY

In this section, we will provide the key definitions which are necessary to understand the notion of integration and, especially, Riemann-Stieltjes integration. Since the main approach and a method of this thesis is to use Riemann-Stieltjes integral as a tool for our classification model, we will not go into the details of the calculus of integration theory and Riemann-Stieltjes integration. For further information on this subject, please read [4, 70].

We note that we build our model in Chapter 5 with the help of Riemann-Stieltjes integrals and later on we further parametrized the model by probability density functions.

**Definition 2.3.1** *[70]. Given a closed interval $I = [a, b]$, a **partition** of $I$ is any finite strictly increasing sequence of points $\mathcal{P} = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$, $x_0 < x_1 < \ldots < x_{m-1} < x_m$, such that $a = x_0$ and $b = x_n$. The **mesh of the partition** $\{x_0, x_1, \ldots, x_{n-1}, x_n\}$ is defined by*

$$\operatorname{mesh} \mathcal{P} = \max_{1 \le j \le n} (x_j - x_{j-1}).$$

*Each partition of $I$, $\mathcal{P} := \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ decomposes $I$ into $n$ subintervals $I_j = [x_{j-1}, x_j]$ $(j = 1, 2, \ldots, n)$ such that*

$$I_j \cap I_k = \begin{cases} x_j, & \text{if } k = j + 1 \\ \emptyset, & \text{if } k \neq j \text{ or k} \neq \text{(j + 1)}. \end{cases} \tag{2.7}$$

*Each such decomposition of $I$ into subintervals is called a **subdivision of** $I$.*

If $f$ is a function defined on the closed interval $I$ and bounded on the interval $I$, then $f$ has both a least upper bound and a greatest lower bound on $I$ which implies also the same result for each interval of any subdivision of $I$.

**Definition 2.3.2** *[70]. Given a bounded function $f$ defined on the interval $I$, and a partition $\mathcal{P} = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ of $I$, let $I_j = [x_{j-1}, x_j]$. $M_j := \sup_{x \in I_j} f(x)$ and $m_j := \inf_{x \in I_j} f(x)$ $(j = 1, 2, \ldots, n)$. Then, the **upper Riemann sum of $f$ with respect to the partition** $\mathcal{P}$, denoted by*

*$U(\mathcal{P}, f)$, is defined by*

$$U(\mathcal{P}, f) := \sum_{j=1}^{n} M_j \Delta x_j,$$

*and the* ***lower Riemann sum of*** $f$ ***with respect to the partition*** $\mathcal{P}$*, denoted by $L(\mathcal{P}, f)$, is defined by*

$$L(\mathcal{P}, f) := \sum_{j=1}^{n} m_j \Delta x_j,$$

*where $\Delta x_j := x_j - x_{j-1}$ $(j = 1, 2, \ldots, n)$.*

**Definition 2.3.3** *[70]. Suppose that $f$ is a function on $\mathbb{R}$ that is defined and bounded on the interval $I = [a, b]$ and $\tilde{\mathcal{P}} = \tilde{\mathcal{P}}[a, b]$ is the set of all partitions of $[a, b]$. Then the* ***upper Riemann integral*** *and the* ***lower Riemann integral*** *are defined by*

$$\overline{\int_a^b} f(x)dx := \inf_{\mathcal{P} \in \tilde{\mathcal{P}}} U(\mathcal{P}, f),$$

*and*

$$\underline{\int_a^b} f(x)dx := \sup_{\mathcal{P} \in \tilde{\mathcal{P}}} U(\mathcal{P}, f),$$

*respectively. If $\overline{\int_a^b} f(x)dx = \underline{\int_a^b} f(x)dx$, then $f$ is* ***Riemann integrable****, or just* ***integrable****, on $I$, and the common value of the integral is denoted by $\int_a^b f(x)dx$ or $\int_{[a,b]} f(x)dx$.*

### 2.3.3 PROPERTIES OF RIEMANN-STIELTJES INTEGRALS

In the previous subsection, the key definitions were given to establish the Riemann-Stieltjes integration based on similar definitions. Intuitively, the main difference between Riemann integration and Riemann-Stieltjes integration is the difference between the lengths of the partitions. In Riemann-Stieltjes integration, the length of the intervals $\Delta x_j$ depends on some monotonically increasing function, whereas in Riemann integration, the length of the interval for a given partition $\mathcal{P}$ does not depend on anything, i.e., intervals are solely the subtraction of the numbers on a real line. Let us generalize this idea by the following definitions.

**Definition 2.3.4** *[70]. Given a bounded function $f$ on a closed interval $I = [a, b]$, a monotonically increasing function $\beta$ on $I$, and a partition $\mathcal{P} = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ of $I$ with corresponding subdivision $\Delta$, let $M_j := \sup_{x \in I_j} f(x)$ and $m_j := \inf_{x \in I_j} f(x)$, for $I_j := [x_{j-1}, x_j]$ $(j =$*

$1, 2, \ldots, n$). *Then, the* ***upper Riemann-Stieltjes*** *sum of $f$ over $\beta$ with respect to the partition $\mathcal{P}$, denoted by $U(\mathcal{P}, f, \beta)$ or $U(\Delta, f, \beta)$, is defined by*

$$U(\mathcal{P}, f, \beta) := \sum_{j=1}^{n} M_j \Delta\beta_j,$$

*and the* ***lower Riemann-Stieltjes sum*** *of $f$ over $\beta$ with respect to the partition $\mathcal{P}$, denoted by $L(\mathcal{P}, f, \beta)$ or $L(\Delta, f, \beta)$, is defined by*

$$L(\mathcal{P}, f, \beta) := \sum_{j=1}^{n} m_j \Delta\beta_j,$$

*where $\Delta\beta_j := \beta(x_j) - \beta(x_{j-1}) \quad (j = 1, 2, \ldots, n)$.*

After giving definitions of upper and lower Riemann-Stieltjes sums, we will introduce the following upper and lower Riemann-Stieltjes integrals in a similar way as for Riemann integrals.

**Definition 2.3.5** *[70]. Suppose that $f$ is a function on $\mathbb{R}$ that is defined and bounded on the interval $I = [a, b]$, $\hat{\mathcal{P}} = \hat{\mathcal{P}}[a, b]$ is the set of all partitions of $[a, b]$, and $\beta$ is a monotonically increasing function on $I$. Then, the* ***upper Riemann-Stieltjes integral*** *and the* ***lower Riemann-Stieltjes integral*** *are defined by*

$$\overline{\int_a^b} f(x)d\beta(x) := \inf_{\mathcal{P}\in\hat{\mathcal{P}}} U(\mathcal{P}, f, \beta),$$

*and*

$$\underline{\int_a^b} f(x)d\beta(x) := \sup_{\mathcal{P}\in\hat{\mathcal{P}}} L(\mathcal{P}, f, \beta),$$

*respectively. If $\overline{\int_a^b} f(x)d\beta(x) = \underline{\int_a^b} f(x)d\beta(x)$, then $f$ is* ***Riemann-Stieltjes integrable*** *or* ***integrable*** *with respect to $\beta$* ***in the Riemann sense****, on $I$, and the common value of the integral is denoted by*

$$\int_a^b f(x)d\beta(x) \ (or \int_{[a,b]} f(x)d\beta(x)) \ \ or \ \ \int_a^b f d\beta \left(or \int_{[a,b]} f d\beta\right).$$

Throughout the thesis, we will denote the set of all functions that are Riemann-Stieltjes integrable with respect to $\beta$ by $\mathfrak{R}(\beta)$. As Riemann-Stieltjes integrals are not the main concern of the thesis, but a tool for the construction of our model, we will give some major theorems for the Riemann-Stieltjes integrability; for the proofs and detailed explanation, we refer [4, 70].

**Theorem 2.3.6** *[70]. Suppose that $f$ is a bounded function on $I = [a, b]$, $\beta$ is a monotonically increasing function on $I$, and $m \leq f(x) \leq M$ for all $x \in I$. Then,*

$$m \cdot (\beta(b) - \beta(a)) \leq \underline{\int_a^b} f d\beta \leq \overline{\int_a^b} f d\beta \leq M \cdot (\beta(b) - \beta(a)). \tag{2.8}$$

*Furthermore, if $f$ is Riemann-Stieltjes integrable on $I$, then*

$$m \cdot (\beta(b) - \beta(a)) \leq \int_a^b f(x) d\beta(x) \leq M \cdot (\beta(b) - \beta(a)). \tag{2.9}$$

**Theorem 2.3.7** (Integrability Criterion) *[70]. Suppose that $f$ is a bounded function on $I = [a, b]$ and $\beta$ is a monotonically increasing function on $I$. Then, $f \in \mathfrak{R}(\beta)$ on $I$ if and only if for every $\epsilon > 0$ there exists a partition $\mathcal{P}$ of $I$ such that*

$$U(\mathcal{P}, f, \beta) - L(\mathcal{P}, f, \beta) < \epsilon. \tag{2.10}$$

The proof is based on Theorem 2.3.6 [70].

**Corollary 2.3.8** *[70]. If $f$ is a continuous function on the interval $I = [a, b]$, then $f$ is Riemann-Stieltjes integrable on $[a, b]$.*

**Corollary 2.3.9** *[70]. If $f$ is a monotonic function on the interval $I = [a, b]$ and $\beta$ is a continuous and monotonically increasing function on $I$, then $f \in \mathfrak{R}(\beta)$.*

**Theorem 2.3.10** (Algebraic Properties of Riemann-Stieltjes Integrals) *[70].*
*Suppose that functions $f, f_1, f_2 \in \mathfrak{R}(\beta)$ are given on the interval $I = [a, b]$, and let $k \in \mathbb{R}$ be given too.*

1. *If $g(x) = kf(x)$ for all $x \in I$, then $g \in \mathfrak{R}(\beta)$ and*
$$\int_a^b g(x) d\beta(x) = k \int_a^b f(x) d\beta(x).$$
2. *If $h = f_1 + f_2$, then $f_1 + f_2 \in \mathfrak{R}(\beta)$ and*
$$\int_a^b h(x) d\beta(x) = \int_a^b f_1(x) d\beta(x) + \int_a^b f_2(x) d\beta(x).$$
3. *If $f_1(x) \leq f_2(x)$ for all $x \in I$, then*
$$\int_a^b f_1(x) d\beta(x) \leq \int_a^b f_2(x) d\beta(x).$$

4. *If the function $f \in \mathfrak{R}(\beta)$ is also given on $I^* = [b, c]$, then $f$ is Riemann-Stieltjes integrable on $I \cup I^*$ and*

$$\int_a^c f(x)d\beta(x) = \int_a^b f(x)d\beta(x) + \int_b^c f(x)d\beta(x).$$

5. *If $|f(x)| \leq M$ for $x \in I$, then*

$$\left| \int_a^b f(x)d\beta(x) \right| \leq M \cdot [\beta(b) - \beta(a)].$$

6. *If $f \in \mathfrak{R}(\beta^*)$ on $I$, then $f \in \mathfrak{R}(\beta + \beta^*)$ and*

$$\int_a^b f d(\beta + \beta^*) = \int_a^b f(x)d\beta(x) + \int_a^b f(x)d\beta^*(x).$$

7. *If $c$ is any positive real constant, i.e., $c \in \mathbb{R}$ and $c > 0$, then $f \in \mathfrak{R}(c\beta)$ and*

$$\int_a^b f d(c\beta) = c \int_a^b f(x)d\beta(x).$$

**Theorem 2.3.11** *[70]. If $f \in \mathfrak{R}(\beta)$ and $g \in \mathfrak{R}(\beta)$ on $[a, b]$, then $fg \in \mathfrak{R}(\beta)$.*

**Theorem 2.3.12** *[70]. If $f \in \mathfrak{R}(\beta)$ on $[a, b]$, then $|f| \in \mathfrak{R}(\beta)$ and*

$$\left| \int_a^b f(x)d\beta(x) \right| \leq \int_a^b |f(x)|\, d\beta(x).$$

The above theorems and definitions are helpful to understand the model structure in this thesis. For detailed explanation and proofs, we refer the readers to [70]. In the following, we will continue with one of the main optimization methods, semi-infinite programming, for our infinitesimally defined model.

## 2.4 SEMI-INFINITE PROGRAMMING

### 2.4.1 INTRODUCTION

In this section, we introduce *semi-infinite programming (SIP)* and discuss the necessary theorems and definitions. SIP is a class of optimization problems which have infinitely many constraints and finitely many variables, as the name "semi-infinite" actually says. SIP has been studied and developed by researchers over the last 30 years [33, 34, 32, 77, 78, 84, 85,

87, 88, 90]. More than 1000 papers have been published on the theory and the numerical methods of SIP. SIP was originally related with Chebyshev approximation, see [34]. For an excellent review, we refer to [33] and [57], for linear semi-infinite programming, we refer to [27].

### 2.4.2 PROBLEM DEFINITION

**Definition 2.4.1** *A semi-infinite programming (SIP) is an optimization problem which has finite dimensional variable* $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ *on a feasible set described by finitely many equality constraints and infinitely many inequality constraints:*

$$\begin{aligned} &\min_{\mathbf{x}} \ f(\mathbf{x}) \\ &h_i(\mathbf{x}) = 0 \ \ (i \in I), \\ &g(\mathbf{x}, \mathbf{y}) \geq 0 \ \ (\mathbf{y} \in Y). \end{aligned} \tag{2.11}$$

*Here,* $I := \{1, 2, \ldots, s\}$ *is a finite index set,* $Y \subseteq \mathbb{R}^m$ *is an infinite index set and* $\mathbf{y} := (y_1, y_2, \ldots, y_m)^T$.

Throughout this section, $\mathcal{M}$ will denote the feasible set, where

$$\mathcal{M} := \{\mathbf{x} \in \mathbb{R}^n \mid h_i(\mathbf{x}) = 0 \ \ (i \in I) \text{ and } \ g(\mathbf{x}, y) \geq 0 \ \ (y \in Y)\}. \tag{2.12}$$

Let $v := \inf\{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{M}\}$ be the optimal value and $\mathcal{S} := \{\bar{\mathbf{x}} \in \mathcal{M} \mid f(\bar{\mathbf{x}}) = v\}$ be the set of minimizers of the problem (2.11).

**Assumptions**: We assume that $Y \subset \mathbb{R}^m$ is compact, $g(\mathbf{x}, y)$ is $C^2$. Now, we will consider the Chebyshev approximation as an example for SIP where SIP is originally related with [78].

**Example 2.4.2** ***(Chebyshev Approximation)****: Let a function* $f \in C(\mathbb{R}^m, \mathbb{R})$ *in the variable* $\mathbf{y}$ *and a family of approximating functions* $\tilde{f}(\mathbf{x}, \mathbf{y})$, $\tilde{f} \in C(\mathbb{R}^n \times \mathbb{R}^m, \mathbb{R})$ *parametrized by* $\mathbf{x} \in \mathbb{R}^n$, *be given. The problem is to find a best approximation to* $f$ *by our functions* $\tilde{f}(\mathbf{x}, \cdot)$ *in the max-norm (Chebyshev norm)*

$$\|\mathcal{F}\|_\infty := \max_{\mathbf{y} \in Y} |\mathcal{F}(\mathbf{y})|,$$

*on a compact set* $Y \subset \mathbb{R}^m$. *Minimizing the approximation error* $\epsilon$ *motivated by* $\epsilon = \left\|f - \tilde{f}\right\|_\infty$ *is equivalent to the following optimization problem:*

$$\begin{aligned} &\min_{\mathbf{x}, \epsilon} \ \epsilon \\ &\text{such that } \ g^{\pm}(\mathbf{x}, \mathbf{y}) = \pm(f(\mathbf{y}) - \tilde{f}(\mathbf{x}, \mathbf{y})) \leq \epsilon \ \ (\mathbf{y} \in Y). \end{aligned} \tag{2.13}$$

*The problem (2.13) is a semi-infinite problem since the constraint should be satisfied for all* $\mathbf{y} \in Y$ *which makes the number of constraints infinite.*

There are other real-world applications of SIP such as shape optimization problem [20], robotics [28, 33], robust optimization models in economics [2, 10, 63, 72, 42], design centering, gene-environment networks, optimal control [86].

### 2.4.3 CONSTRAINT QUALIFICATIONS

In this subsection, we will discuss the structure of the feasible sets $\mathcal{M}$ of finite, semi-infinite programming and linear semi-infinite programming (LSIP). Furthermore, we will introduce the constraint qualifications both for finitely constrained programming and semi-infinite programming to emphasize the difference between them.

In ***semi-infinite programming (SIP)***, $\mathcal{M}$ is defined by

$$\mathcal{M} = \{\mathbf{x} \in \mathbb{R}^n \mid \ h_i(\mathbf{x}) = 0 \ (i \in I), \ \ g(\mathbf{x}, \mathbf{y}) \geq 0 \ (\mathbf{y} \in Y)\}, \tag{2.14}$$

where $I$ is a finite index set, $Y \subset \mathbb{R}^m$ is an infinite index set, $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}$ are $C^2$ functions.

In ***linear semi-infinite programming (LSIP)*** [27], $\mathcal{M}$ is defined by

$$\mathcal{M} = \{\mathbf{x} \in \mathbb{R}^m \left| a_{\mathbf{y}}^T \mathbf{x} \geq b_{\mathbf{y}} \ (\mathbf{y} \in Y)\right.\}, \tag{2.15}$$

with functions $\mathbf{y} \mapsto a_{\mathbf{y}} \in \mathbb{R}^n$ and $\mathbf{y} \mapsto b_{\mathbf{y}} \in \mathbb{R}$, where $I := \{1, 2, \ldots, s\}$ is a finite index set, $Y \subset \mathbb{R}^m$ is an infinite index set. Here, we exclude the equality constraints in our problem definition since we will not need the equality constraint after the parametrization by probability density functions in Chapter 6.

In ***finite programming (FP)***, $\mathcal{M}$ is defined by

$$\mathcal{M} = \{\mathbf{x} \in \mathbb{R}^n \left| \ h_i(\mathbf{x}) = 0 \ (i \in I), \ g_j(\mathbf{x}) \geq 0 \ (j \in J)\right.\}, \tag{2.16}$$

where $J = \{1, 2, \ldots, m\}$ and $I = \{1, 2, \ldots, s\}$ are finite index sets and $h_i$ $(i \in I)$ and $g_j : \mathbb{R}^n \to \mathbb{R}$ $(j \in J)$ are $C^2$ functions.

Throughout this subsection, we assume that $h_i$, $g_j$ and $g$ are $C^2$-functions. Let us define constraint qualifications for finite and semi-infinite programming.

**Constraint Qualifications in FP:**

The ***Linear Independence Constraint Qualification (LICQ)*** is said to hold at a feasible point $\bar{\mathbf{x}} \in \mathcal{M}$ if the vectors

$$\nabla h_i(\bar{\mathbf{x}}) \ \ (i \in I),$$

$$\nabla g_j(\bar{\mathbf{x}}) \ (j \in J_0(\bar{\mathbf{x}})),$$

form a linearly independent family. Here,

$$J_0(\bar{\mathbf{x}}) := \{j \in J \,\big|\, g_j(\bar{\mathbf{x}}) = 0\} \tag{2.17}$$

is the *active index set*.

The following constraint qualification is weaker than LICQ. In fact, ***Mangasarian Fromovitz Constraint Qualification (MFCQ)*** is said to hold at $\bar{\mathbf{x}} \in \mathcal{M}$ if there exists a vector $\boldsymbol{d} \in \mathbb{R}^n$ such that

$$\nabla^T h_i(\bar{\mathbf{x}})\boldsymbol{d} = 0 \ (i \in I) \ \text{ and } \ g_j(\bar{\mathbf{x}})\boldsymbol{d} > 0 \ (j \in J_0(\bar{\mathbf{x}})).$$

In this thesis, MFCQ will not be the criterion for our analysis but we will use it in a theorem in the next chapters.

**Constraint Qualifications in SIP:** The LICQ is said to hold at $\bar{\mathbf{x}} \in \mathcal{M}$ if the vectors

$$\nabla h_i(\bar{\mathbf{x}}) \ (i \in I),$$

$$\nabla_{\mathbf{x}} g(\bar{\mathbf{x}}, \mathbf{y}) \ (\mathbf{y} \in Y_0(\bar{\mathbf{x}})),$$

form a linearly independent family, where

$$Y_0(\bar{\mathbf{x}}) := \{\mathbf{y} \in Y \mid g(\bar{\mathbf{x}}, \mathbf{y}) = 0\}. \tag{2.18}$$

We note that if LICQ holds at $\bar{\mathbf{x}}$, the active set $Y_0(\bar{\mathbf{x}})$ cannot contain more than $n$ elements, where $n$ is the dimension of the vector space $\mathbb{R}^n$.

In the following, we continue with the definitions of local, global minimizer and theorem for first-order optimality conditions for SIP without giving a proof. For further explanation, we refer to [32, 33, 34, 78, 84].

### 2.4.4 FIRST-ORDER OPTIMALITY CONDITIONS

**Definition 2.4.3** *[78]. A feasible point $\bar{\mathbf{x}} \in \mathcal{M}$ is called a **local minimizer** of SIP if there is some $\epsilon > 0$ such that*

$$f(\mathbf{x}) - f(\bar{\mathbf{x}}) \geq 0 \text{ for all } \mathbf{x} \in \mathcal{M} \text{ with } \|\mathbf{x} - \bar{\mathbf{x}}\|_2 < \epsilon, \tag{2.19}$$

*where $\|\cdot\|_2$ is the Euclidean norm.*

**Definition 2.4.4** *[78]. The minimizer $\bar{\mathbf{x}}$ is **global** if (2.19) holds for any $\epsilon > 0$.*

**Definition 2.4.5** *[78]. We call $\bar{\mathbf{x}} \in \mathcal{M}$ a **strict local minimizer** of order $p > 0$ if there exist some $q > 0$ and $\epsilon > 0$ such that*

$$f(\mathbf{x}) - f(\bar{\mathbf{x}}) \geq q \|\mathbf{x} - \bar{\mathbf{x}}\|_2^p \text{ for all } \mathbf{x} \in \mathcal{M} \text{ with } \|\mathbf{x} - \bar{\mathbf{x}}\|_2 \leq \epsilon.$$

**Remark 1** *It is obvious that around a point $\bar{\mathbf{x}} \in \mathcal{M}$, where the active index set $Y_0(\bar{\mathbf{x}})$ is empty, i.e., $g(\bar{\mathbf{x}}, \mathbf{y}) > 0$ ($\mathbf{y} \in Y$), the SIP problem locally refers to a common unconstrained problem.*

**Theorem 2.4.6** (First-Order Sufficient Condition) *[33, 78, 88].*
*Let $\bar{\mathbf{x}}$ be feasible for (2.11). Suppose that there does not exist a vector $0 \neq \mathbf{d} \in \mathbb{R}^n$ satisfying*

$$\nabla f(\bar{\mathbf{x}})\mathbf{d} \leq 0,\ \nabla_{\mathbf{x}}^T h_i(\bar{\mathbf{x}})\mathbf{d} = 0\ (i \in I),\ \nabla_{\mathbf{x}}^T g(\bar{\mathbf{x}}, \mathbf{y})\mathbf{d} \geq 0\ (\mathbf{y} \in Y_0(\bar{\mathbf{x}})).$$

*Then, $\bar{\mathbf{x}}$ is a strict local minimizer of SIP of order $p = 1$.*

We refer [78] for a proof of Theorem 2.4.6.

**Theorem 2.4.7** *(First-Order Necessary Condition) [78]. Let $\bar{\mathbf{x}}$ be a local minimizer of (2.11). Then the following holds:*

***KKT Condition:** If MFCQ holds at $\bar{\mathbf{x}}$, then there exist multipliers $\lambda_1, \lambda_2, \ldots, \lambda_s$ $\in \mathbb{R}$ and $\mu_1, \mu_2, \ldots, \mu_k \geq 0$ and indices $y_1, y_2, \ldots, y_k \in Y_0(\bar{\mathbf{x}})$, $k \leq n$, such that*

$$\nabla f(\bar{\mathbf{x}}) - \sum_{i=1}^{s} \lambda_i \nabla_{\mathbf{x}}^T h_i(\bar{\mathbf{x}}) - \sum_{j=1}^{k} \mu_j \nabla_{\mathbf{x}}^T g(\bar{\mathbf{x}}, y_j) = 0. \tag{2.20}$$

The proof is omitted here; see [33, 78, 88].

We will represent the equation (2.20) as *Lagrange* functions later on in our problems. Since the assumptions of Theorem 2.4.6 are rather strict, we need second-order information. Before starting to introduce this information, let us briefly discuss a special cases of SIP where the functions are convex and linear. For the detailed explanations, we refer to [10, 11, 27]. In the following, we give definitions on convexity, convex programming, duality and feasibility conditions for SILP cases.

### 2.4.5 CONVEX AND LINEAR SEMI-INFINITE PROGRAMMING

We start with definitions of a convex set and a convex function and we continue with theorems used in this thesis.

**Definition 2.4.8** *[11]. A set $C \subseteq \mathbb{R}^m$ is convex if the line segment between any two points in $C$ also lies in $C$. Mathematically, for any $\mathbf{x}_1, \mathbf{x}_2 \in C$ and any $\delta \in [0, 1]$, we have*

$$\delta \mathbf{x}_1 + (1 - \delta)\mathbf{x}_2 \in C.$$

**Definition 2.4.9** *[11]. A set $C$ is called a cone, or nonnegative homogeneous, if for every $\mathbf{x} \in C$ and $\delta \geq 0$ we have $\delta \mathbf{x} \in C$.*
*A set $C$ is a convex cone if it is convex and a cone, which means that for any $\mathbf{x}_1, \mathbf{x}_2 \in C$ and $\delta_1, \delta_2 \geq 0$, we have*

$$\delta_1 \mathbf{x}_1 + \delta_2 \mathbf{x}_2 \in C.$$

**Definition 2.4.10** *[11]. Suppose $(l + 1)$ points $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_l \in \mathbb{R}^n$ are affinely independent, which means $\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0, \ldots, \mathbf{x}_l - \mathbf{x}_0$ are linearly independent. The simplex determined by them is given by*

$$C = \mathbf{conv}\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_l\} = \{\delta_0 \mathbf{x}_0 + \delta_1 \mathbf{x}_1 + \ldots + \delta_l \mathbf{x}_l \,\big|\, \delta \geq 0,\ \mathbf{1}^T \delta = 1\},$$

*where $\mathbf{1}$ denotes the vector with all entries one*

We will use the simplex definition later in Section 6.2.

**Definition 2.4.11** *[11]. A function $f : \mathbb{R}^n \to \mathbb{R}$ is **convex** if the domain of $f$, $domf$, is a convex set and if for all $x, y \in domf$, and $\delta$ with $0 \leq \delta \leq 1$, we have*

$$f(\delta\mathbf{x} + (1-\delta)\mathbf{y}) \leq \delta f(\mathbf{x}) + (1-\delta)f(\mathbf{y}).$$

**Definition 2.4.12** *[11]. A function $f$ is concave if $(-f)$ is convex.*
*The semi-infinite program is convex if both the objective function $f(\mathbf{x})$ and for the constraint functions $-g(\cdot, y)$ $(\mathbf{y} \in Y)$ are convex.*

**Theorem 2.4.13** *[78]. If $\bar{\mathbf{x}}$ is a feasible point of a convex SIP problem such that KKT conditions given by (2.20) and nonnegativity of the Lagrange multipliers are satisfied, then $\bar{\mathbf{x}}$ is a global minimizer of the convex SIP.*

An important special case of convex SIP is given by the *linear semi-infinite problem (LSIP)* [27], where the objective function $f$ and the function $g$ are linear in $\mathbf{x}$:

$$\begin{aligned} (LSIP): \qquad & \min_{\mathbf{x}} \ \mathbf{c}^T\mathbf{x} \\ & \text{such that} \quad a_{\mathbf{y}}^T\mathbf{x} \geq b_{\mathbf{y}} \ \ (\mathbf{y} \in Y). \end{aligned} \tag{2.21}$$

For the intensive analysis we refer to [27]. We don't need to write equality constraints here, i.e., $I = \emptyset$, since we will not need them in our infinite problem in further chapters. In this section, we consider the strong duality results.

It is well known [11, 27] that any convex optimization problem

$$\begin{aligned} & \min_{\mathbf{x}} \ \mathbf{c}^T\mathbf{x} \\ & \text{such that} \ \ \mathbf{x} \in \mathcal{M} \ \ (\mathcal{M} \subset \mathbb{R}^n \text{ being a closed convex set}), \end{aligned}$$

can be written as a LSIP problem. If $Y$ is compact and the functions $\mathbf{y} \mapsto a_{\mathbf{y}} = a(\mathbf{y})$, $\mathbf{y} \mapsto b_{\mathbf{y}} = b(\mathbf{y})$ are continuous on $Y$, then (2.21) is continuous problem. Throughout the section, we assume that (2.21) is continuous. In a more comprised way, (2.21) can be written as follows:

$$\begin{aligned} (LSIP_{primal}) \qquad & \min_{\mathbf{x}} \ \mathbf{c}^T\mathbf{x} \\ & \text{such that} \ \ A\mathbf{x} \geq \mathbf{b}, \end{aligned} \tag{2.22}$$

where $A$ is a generalized matrix with infinitely many rows $a_{\mathbf{y}}^T$ $(\mathbf{y} \in Y)$ and $\mathbf{b}$ is a vector with infinitely many components $b_{\mathbf{y}}$ $(\mathbf{y} \in Y)$. Let us define the ***dual*** of (2.22) as

$$\begin{aligned} (LSIP_{dual}) \qquad & \max_{u} \ \mathbf{b}^T\mathbf{u} \\ & \text{such that} \ \ A^T\mathbf{u} = \mathbf{c}, \ \mathbf{u} \geq 0. \end{aligned} \tag{2.23}$$

Here, $\mathbf{u} = (u_{\mathbf{y}})_{\mathbf{y}\in Y}$ is the dual variable and it is ***dual feasible*** if $u_{\mathbf{y}} \geq 0$ ($\mathbf{y} \in Y$) and $u_{\mathbf{y}} > 0$ for only ***finitely many*** $\mathbf{y} \in Y$. Hence, $\mathbf{b}^T\mathbf{u} = \sum_{\mathbf{y}\in Y} b_{\mathbf{y}}u_{\mathbf{y}}$ and $A^T\mathbf{u} = \sum_{\mathbf{y}\in Y} a_{\mathbf{y}}u_{\mathbf{y}}$ are finite sums.

Throughout the section, we denote the optimal function values of (2.22) and (2.23) by $v_{PLSIP}$ and $v_{DLSIP}$.

**Definition 2.4.14** *A set $C$ is called a cone, or nonnegative homogeneous, if for every $\mathbf{x} \in C$ and $\theta \geq 0$ we have $\theta\mathbf{x} \in C$.*

**Remark 2** *[78]. Recall from the definition of a cone that $c$ is in the cone of $a_{\mathbf{y}}$, i.e., $\boldsymbol{c} \in \text{cone}\{a_{\mathbf{y}} \mid \mathbf{y} \in Y\}$ if and only if (2.23) is feasible. Furthermore, from Caratheodory's Lemma [62], $c = \sum_{\mathbf{y}\in Y} a_{\mathbf{y}}u_{\mathbf{y}}$ can be expressed as sums with at most n nonzero coefficients $u_{\mathbf{y}}$.*

*As in Linear Programming (LP)* [45, 49], if $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} = (u_{\mathbf{y}})_{\mathbf{y}\in Y}$ are primal and dual feasible, respectively, then

$$\boldsymbol{c}^T\mathbf{x} - u^T\mathbf{b} = u^T(A\mathbf{x} - \mathbf{b}) = \sum_{\mathbf{y}\in Y} u_{\mathbf{y}}(a_{\mathbf{y}}^T\mathbf{x} - b_{\mathbf{y}}) \geq 0.$$

**Theorem 2.4.15** *(**Weak Duality, Complementary Slackness**) [33, 78]. If $\mathbf{x}$ and $\mathbf{u}$ are primal and dual feasible, respectively, then $\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{u}$. If $\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{u}$, then $\mathbf{x}$ and $\mathbf{u}$ are optimal with $v_{PLSIP} = v_{DLSIP}$.*

Note that LSIP does not necessarily have the strong duality property (cf. [78]) unless additional constraint qualifications hold (*Slater constraint qualifications*). Another important property of LSIP is that the existence of the primal and dual feasible solutions need not imply the optimality of these solutions. To guarantee the existence of an optimal solution of (2.22), we may assume strict feasibility of the dual problem. Next, we give the definitions for Slater constraint qualification for the primal and the dual LSIP and the strong duality result.

**Definition 2.4.16** *[27, 78]. The **Slater constraint qualification** for the primal LSIP, (2.22), is said to hold if there exist some strictly feasible primal solution $\hat{\mathbf{x}}$ (Slater point), i.e., there exist a certain $\hat{\mathbf{x}} \in \mathbb{R}^n$ such that $a_{\mathbf{y}}^T\hat{\mathbf{x}} > b_{\mathbf{y}}$ for all $\mathbf{y} \in Y$, and the following relation holds*

$$(SC_{PLSIP}) \qquad \text{There exists some } \hat{\mathbf{x}} \in \mathbb{R}^n \text{ with } A\hat{\mathbf{x}} > b. \tag{2.24}$$

Here, $A$ is a generalized matrix having infinitely many rows which constitute a family of infinitely many functions, and each member of this family, i.e., $A\hat{\mathbf{x}}$, is strictly positive.

**Definition 2.4.17** *[27, 78]. Let the same assumptions hold in Definition 2.24 for the dual LSIP problem (2.23). Then, the dual Slater constraint qualification holds if*

$$(SC_{DLSIP}) \qquad \mathbf{c} \in \text{int cone}\{a_{\mathbf{y}} \mid \mathbf{y} \in Y\}. \tag{2.25}$$

**Theorem 2.4.18** ***(Strong Duality)*** *[27, 78]. Assume that the Slater conditions (2.24) and (2.25) are satisfied, i.e., (2.22) and (2.23) are strictly feasible. Then optimal primal and the dual solutions $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ exists with $v_{PLSIP} = v_{DLSIP}$.*

*Moreover, $\bar{\mathbf{x}}$ is an optimal solution if and only if $\bar{\mathbf{x}}$ is a KKT-point, i.e., there exists a dually feasible $\bar{\boldsymbol{u}}$ satisfying $\bar{\boldsymbol{u}}_{y_j} \geq 0$ for $y_j \in Y_0(\bar{\mathbf{x}})$ $(j = 1, 2, \ldots, k)$, $k \leq n$, and $\bar{\boldsymbol{u}}_{\mathbf{y}} = 0$ otherwise, such that*

$$c = \sum_{j=1}^{k} \bar{\boldsymbol{u}}_{y_j} a_{y_j}.$$

*In particular, $\bar{\boldsymbol{u}}^T(A\bar{\mathbf{x}} - \boldsymbol{b}) = 0$.*

As we stated in Section 2.4.4 that the assumptions of Theorem 2.4.6 are rather strict, we need second-order information for further ways of indicating a local minimizer. In the next section, we will introduce those second-order optimality conditions.

### 2.4.6 SECOND-ORDER OPTIMALITY CONDITIONS

In this subsection, we introduce the *second-order optimality conditions (SOC)* for semi-infinite problem given in (2.11) by applying the so-called ***reduction approach*** (see [90]). *(SOC)* plays a big role in our solution scheme of Chapter 5 and Chapter 6. We will use the same theorems which are adapted to our model later.

Let $\bar{\mathbf{x}} \in \mathcal{M}$ be a feasible point of (2.11) and assume that $u, v, f, g, h \in C^2$ and infinite index set $Y$ is defined as the solution set of equalities and inequalities with functions $u_k$ and $v_\ell$:

$$Y = \{\mathbf{y} \in \mathbb{R}^m \mid u_k(\mathbf{y}) = 0 \ (k \in K),\ v_\ell(\mathbf{y}) \geq 0 \ \ (\ell \in L)\}, \tag{2.26}$$

where $K := \{1, 2, \ldots, r\}$ and $L := \{1, 2, \ldots, q\}$ By definition, any active point $\bar{\mathbf{y}}$ from $Y_0(\bar{\mathbf{x}})$ is a (global) minimizer of the following parametric optimization problem:

$$\begin{aligned} Q(\bar{\mathbf{x}}) \qquad & \min_{\mathbf{y}} \; g(\bar{\mathbf{x}}, \mathbf{y}) \\ & \text{such that } \; u_k(\mathbf{y}) = 0 \; (k \in K) \text{ and } \; v_\ell(\mathbf{y}) \geq 0 \;\; (\ell \in L). \end{aligned} \tag{2.27}$$

The problem (2.27) is called the *lower level problem*; it depends on $\bar{\mathbf{x}}$ as a parameter. Active index set is denoted by $L_0(\bar{\mathbf{y}})$ and defined as the set of indices of active constraints of (2.27), i.e., $L_0(\bar{\mathbf{y}}) = \{\ell \in L \,|\, v_\ell(\bar{\mathbf{y}}) = 0\}$. Let us write the Lagrange function of $Q(\bar{\mathbf{x}})$ at any $\bar{\mathbf{x}} \in \mathcal{M}$ and $\bar{\mathbf{y}} \in Y_0(\bar{\mathbf{x}})$ or $\bar{\mathbf{y}}$ being a local minimizer of our lower level problem:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\zeta}, \boldsymbol{\gamma}) := g(\mathbf{x}, \mathbf{y}) - \sum_{k \in K} \zeta_k u_k(\mathbf{y}) - \sum_{\ell \in L_0(\bar{y})} \gamma_\ell v_\ell(\mathbf{y}), \tag{2.28}$$

where $\zeta_k$ $(k \in K)$ and $\gamma_l$ $(l \in L_0(\bar{\mathbf{y}}))$ are the Lagrange multiplier vectors.

Now, we define *Reduction Ansatz* which has crucial assumptions to have optimal and unique solution.

**Definition 2.4.19** *(**Reduction Ansatz**) [32, 33, 34, 78, 88]. We say Reduction Ansatz holds, if for any $\bar{\mathbf{y}} \in Y_0(\bar{\mathbf{x}})$ or $\bar{\mathbf{y}}$ being a local minimizer of $Q(\bar{\mathbf{x}})$, the following properties holds:*

1. *LICQ: $\nabla_{\mathbf{y}} u_k(\bar{\mathbf{y}})$, $\nabla_{\mathbf{y}} v_\ell(\bar{\mathbf{y}})$ $(k \in K)$, $\ell \in L_0(\bar{\mathbf{y}})$ are linearly independent family.*

2. *Under LICQ, there are unique multipliers $\bar{\boldsymbol{\zeta}}$ and $0 \leq \bar{\boldsymbol{\gamma}} \in \mathbb{R}^{|L_0(\bar{\mathbf{y}})|}$ satisfying*

$$\nabla_y \mathcal{L}(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\boldsymbol{\zeta}}, \bar{\boldsymbol{\gamma}}) = 0.$$

   *We assume $\bar{\gamma}_\ell > 0$ $(\ell \in L_0(\bar{\mathbf{y}}))$ (strict complementary slackness).*

3. *The second order condition (SOC): With $\bar{\gamma}$ in 2.,*

$$\boldsymbol{\eta}^T \nabla^2_{\mathbf{y}} \mathcal{L}(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\boldsymbol{\zeta}}, \bar{\boldsymbol{\gamma}}) \boldsymbol{\eta} > 0, \;\; \textit{for all } \boldsymbol{\eta} \in \mathcal{T}(\bar{\mathbf{y}}) \setminus \{0\},$$

   *where $\mathcal{T}(\bar{\mathbf{y}})$ is the tangent space at $\bar{y}$ defined by*

$$\mathcal{T}(\bar{\mathbf{y}}) = \{\boldsymbol{\gamma} \in \mathbb{R}^m \,\big|\, \nabla^T_{\mathbf{y}} u_k(\bar{\mathbf{y}}) \boldsymbol{\eta} = 0 \; (k \in K), \; \nabla^T_{\mathbf{y}} v_\ell(\bar{\mathbf{y}}) \boldsymbol{\eta} = 0 \; (\ell \in L_0(\bar{\mathbf{y}}))\}.$$

Under the assumptions in Reduction Ansatz, the following theorem is proven in [32, 33, 34, 78, 88, 90].

**Theorem 2.4.20** *[32, 33, 34, 78, 88, 90]. Let the Reduction ansatz at the active index set $Y_0(\bar{\mathbf{x}})$ or the set of local minimizers of $Q(\bar{\mathbf{x}})$ be satisfied at the feasible point $\bar{\mathbf{x}}$ for (2.11) and infinite index set Y given by (2.26) be compact. Then the following holds:*

1. *The active index set $Y_0(\bar{\mathbf{x}}) := \{\bar{\mathbf{y}}_1, \bar{\mathbf{y}}_2, \ldots, \bar{\mathbf{y}}_r\}$ is finite and there exist neighborhoods $U_{\bar{\mathbf{x}}}$ of $\bar{\mathbf{x}}$ and $V_{\bar{\mathbf{y}}_j}$ of $\bar{\mathbf{y}}_j$, corresponding Lagrange multipliers $\bar{\zeta}_j$ and $\bar{\gamma}_k$ and continuous mappings*

$$y_j : U_{\bar{\mathbf{x}}} \to V_{\bar{\mathbf{y}}_j}, \text{ with } y_j(\bar{\mathbf{x}}) = \bar{\mathbf{y}}_j,$$

$$\zeta_k : U_{\bar{\mathbf{x}}} \to \mathbb{R}, \ \zeta_k^j(\bar{\mathbf{x}}) = \bar{\zeta}_k \ (k \in K) \text{ and}$$

$$\gamma_j : U_{\bar{\mathbf{x}}} \to \mathbb{R} \text{ with } \gamma_l^j(\bar{\mathbf{x}}) = \bar{\gamma}_j, \ (l \in L_0(\bar{\mathbf{x}}))$$

   *$(j = 1, 2, \ldots, r)$ such that for every $\mathbf{x} \in U_{\bar{\mathbf{x}}}$ the value $y_j(\mathbf{x})$ is the unique local minimizer of $Q(\mathbf{x})$ in $V_{\bar{\mathbf{y}}_j}$ with corresponding Lagrange multiplier vectors $\gamma_l^j(\mathbf{x})$ and $\zeta_k^j(\mathbf{x})$.*

2. *With the functions in 1. the following finite reduction holds:*

   *$\mathbf{x} \in U_{\bar{\mathbf{x}}} \cap \mathcal{M}$ is a local solution of (2.11) if and only if $\bar{\mathbf{x}}$ is a local solution of the so-called* ***reduced problem***

$$\begin{aligned} P_{red}(\bar{\mathbf{x}}): \quad & \min_{\mathbf{x} \in U_{\bar{\mathbf{x}}}} f(\mathbf{x}) \\ & \text{such that } h_i(\mathbf{x}) = 0 \ (i \in I), \\ & \qquad \tilde{g}_j(\mathbf{x}) := g(\mathbf{x}, y_j(\mathbf{x})) \geq 0 \quad (j = 1, 2, \ldots, r). \end{aligned} \tag{2.29}$$

3. *The functions $h_i(\mathbf{x})$ $(i \in I)$ and $\tilde{g}_j(\mathbf{x}) = g(\mathbf{x}, y_j(\mathbf{x}))$ $(j = 1, 2, \ldots, r)$ are $C^2$-functions in $U_{\bar{\mathbf{x}}}$.*

We refer to [78] for the proof of the theorem.

**Geometric Interpretation of Theorem 2.4.20**

Theorem 2.4.20 obviously presents conditions of a *reduction* property. For a given feasible point $\bar{\mathbf{x}}$, infinitely many constraints are reduced to finitely many constraints by solving the lower level problem $Q(\bar{\mathbf{x}})$; see Figure 2.3. In Figure 2.3, $\tilde{\mathbf{x}}$ represents the small perturbation of $\bar{\mathbf{x}}$, i.e., $\bar{\mathbf{x}} \to \tilde{\mathbf{x}}$. If the infinite index set is compact and the nondegeneracy and continuity assumptions of our model defining functions hold, then by Theorem of Heine-Borel there are finitely many local minima of the lower level problem $Q(\bar{\mathbf{x}})$ (for a careful argumentation see

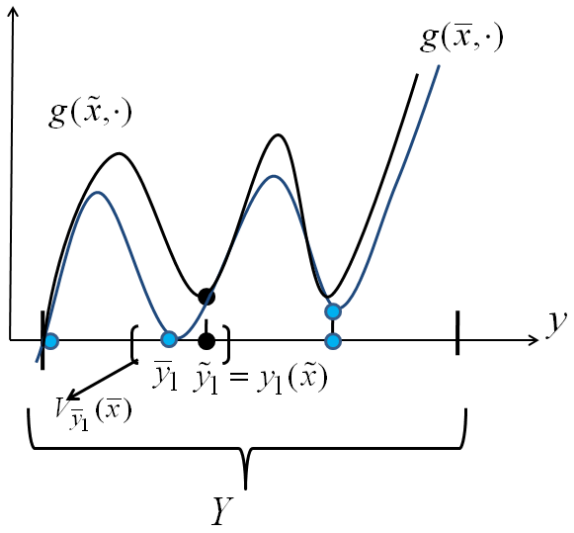


Figure 2.3: An example illustrating the minimizing constraint function, $g(\bar{\mathbf{x}}, \cdot)$ and the Reduction Ansatz, an example.

[88]). Furthermore, finitely many local minima of $Q(\bar{\mathbf{x}})$ lead to finitely many active inequality constraints for the upper level problem. Solving the reduced problem is equivalent to solve *SIP locally*. The drawback of Theorem 2.4.20 is that the indices *implicitly* depend on the variable $\mathbf{x}$. There are some numerical methods which combine the basic idea of the reduction property and a discretization of the infinite index set. This will be explained in Chapter 6.

# CHAPTER 3

# PATTERN ANALYSIS FOR THE PREDICTION OF FUNGAL PRO-PEPTIDE CLEAVAGE SITES

## 3.1 INTRODUCTION

There is a growing interest in the application of machine learning techniques together with optimization to real-world applications such as biological problems [53], engineering problems etc.. This chapter is devoted to solving one of the important problems in peptide biology, namely predicting pro-peptide cleavage site for a given amino acid sequence of a protein by using an SVM which is introduced by a novel *confidence level* model selection algorithm. There have been many studies on predicting peptide regions such as signal peptide [12, 13, 18, 22, 31, 48], pro-peptide [23] solved using neural networks with classical model selection methods such as cross validation (CV) [30].

In this thesis, we have developed an efficient and novel model selection algorithm embedded in a classical SVM to predict pro-peptide cleavage sites in *filamentus fungi* [53]. Prediction results of the confidence level by an SVM are compared with the results achieved by the pro-peptide prediction tool ProP1.0 [23]. **ProP1.0** is a bioinformatics and computational biology tool which predicts pro-peptide cleavage sites on a furin specific based network and a general PC network separately by using a **neural network**. ProP1.0 consists of 227 proteins of all eukaryotes including those of humans and animals. The data set is presented to the neural network by sparsely encoded moving windows. The output of a neural network is assessed by a threshold of 0.5 to determine the potential pro-peptide cleavage site.

This study concentrates more on fungal proteins due to the industrial importance of these organisms in heterologous protein production, including those of humans. The data set is

collected from largely non-homologous fungal proteins consisting of 72 sequences. Our prediction tool, *confidence level* SVM is fed with both binary input vectors and the substitution matrix PAM250 separately and results are reported for both. The sequences are given to the learning machine by encoded sliding windows through each sequence. Each protein is tested with different training sets. Rather than splitting the data set into groups, we have used a different strategy that enables us to use the whole data for both training and testing. This is explained in detail in the next section. The construction of the data set from non-homologous sequences is justified by using ClustalW to construct a phylogenetic tree which is based on multiple sequence alignment.

## 3.2 BIOLOGICAL BACKGROUND

Proteins are very important molecules for a cell because of their role in building cell structures. Moreover, they include enzymes as special proteins that catalyze many metabolic reactions in a cell. Proteins are composed of small units called *amino acids*. They are specified in a code of 20 different letters. The primal structure of proteins is specified by its amino acid sequence that determines the structure of the protein and this structure determines its function. Each amino acid is bind together with a peptide bond to form the amino acid sequence that folds to a different specific structure.

### 3.2.1 PROTEIN SYNTHESIS

Cells produce new proteins either for reproduction or to replace a degraded one. Protein synthesis is a process that occurs in a cell's nucleus. It consists of two main stages, transcription and translation. Transferring of genes from DNA into RNA is called *transcription*.

**Transcription**:

In each cell's nucleus, the DNA strand carries information that controls protein synthesis. In DNA, genes are embedded in chromosomes. During the transcription RNA synthesizes mRNA from DNA. In eukaryotes, after mRNA is synthesized, it moves out of the cell's nucleus through the nuclear pores to the translation [16].

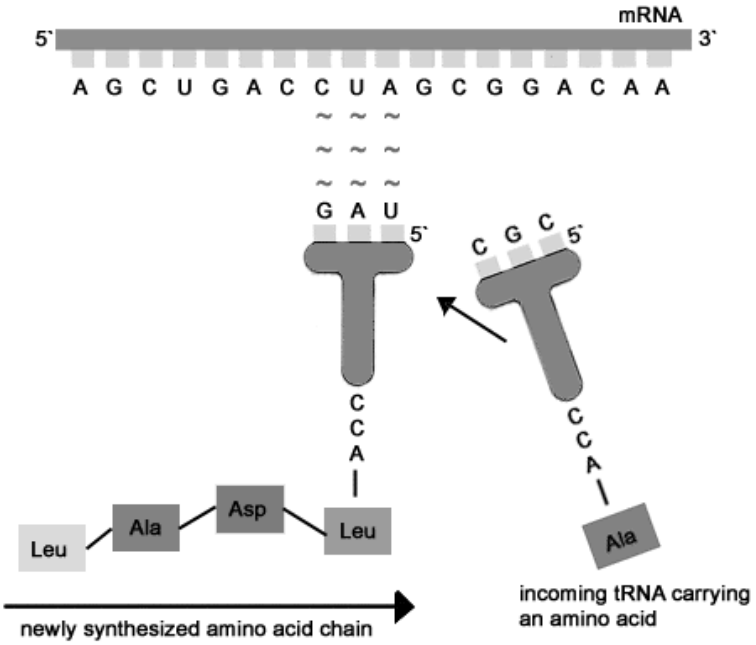


Figure 3.1: RNA Translation process [16].

**Translation**:

After mRNA has been transported through the rough endoplasmic reticulum, it is fed into the ribosomal translation machineries. Ribosomes start to read the mRNA sequence and to convert mRNA into protein; tRNA is the responsible molecule to read the mRNA sequence, 3 nucleotides at a time.

Amino acids are represented by codons, which are 3-nucleotide RNA sequences. The mRNA sequence matches three nucleotides at a time to a complementary set of three nucleotides in the anticodon region of the corresponding **tRNA** molecule. An amino acid is attached to a site opposite to the anticodon region of each tRNA, and as the mRNA is read off, the amino acids on each tRNA are joined together through *peptide bonds*. Figure 3.1 illustrates how tRNA molecules bind to mRNA [16].

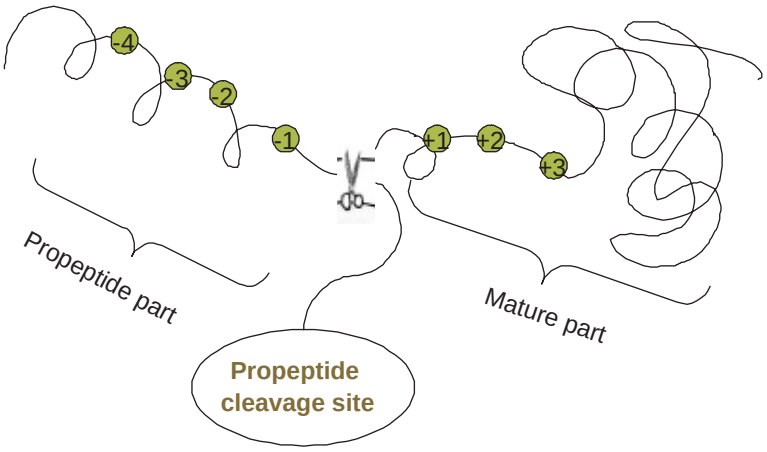


Figure 3.2: Illustration of pro-peptide cleavage sites [17].

**Post Translational Processing**:

Post translational processing is the stage of modification of inactive precursors into their mature and active forms. One such process involves the removal of signal and pro-peptide regions. While signal peptides are required for the secretion of proteins, pro-peptides may have various functions, such as acting as a chaperon and keeping the enzyme in an inactive state. The proteolytic cleavage site is also one of the critical factors in Alzheimer's's disease [25]. The pro-peptide cleavage process is illustrated in Figure 3.2.

### 3.2.2 N-TERMINAL PRO-PEPTIDES OF FUNGAL SECRETED PROTEINS

There is a growing interest to the proteolytic processing events in cell biology due to recent findings related to their critical functions in apoptosis [89], in triggering human diseases such as Alzheimer's [25] as well as their well-known role in cell trafficking [74]. Most of the proteolytic processing events take place at the N-terminus of proteins. Among these processes, signal peptide cleavage is perhaps the most well-known, and software programmes are established by which eukaryotic and prokaryotic signal peptides and cleavage sites can be predicted with high precision such as the SignalP Server [48]. Relatively less is known about the function and mechanism of the post-translational removal of pro-peptides which exist not only in secreted proteins but also in some of the proteins that do not pass through the endoplasmic reticulum.

Fungi are ideal model systems for the study of eukaryotic molecular mechanisms due to their relative simplicity. In general, filamentous fungi are more closely related to higher eukaryotes than the yeasts. Filamentous fungi have also attracted much attention due to their importance as heterologous expression systems [46] although some yeasts such as *Pichia pastoris* are also promising as heterologous hosts. Particularly, Aspergillus and Trichoderma species find wide use as industrial protein factories. While large amounts of fungal proteins are heterologously produced in efficient and safe hosts such as *Aspergillus sojae* [59], still more progress is necessary to enhance the heterologous production of mammalian proteins [36]. This requires more in-depth understanding of the events taking place during the secretion process.

At the N-terminus, transient peptides may consist of only a signal peptide or may also contain one or more additional peptides. Here, signal peptide sequences were not considered, however, only proteins with a predicted signal peptide were selected from the NCBI Genebank database. In general, a single additional peptide is called a pro-peptide. If there are two additional peptides, they are called *pre-peptide* and ***pro-peptide***, respectively.

### 3.2.3 FUNCTIONS OF PRO-PEPTIDES

Pro-peptides have been implicated in a number of cellular processes including their role as an intracellular chaperon [68], in proper folding, in subcellular sorting and in keeping proteins in an inactive configuration. The pro-peptide is removed upon or before departure from the secretory pathway by maturases [9] that reside either in the late stage of the Golgi, the secretory vesicles or are extracellularly anchored to the cytoplasmic membrane with a GPI-anchor [73]. The processing of most of the pro-peptides occurs at either a monobasic or a dibasic cleavage site [9]. Dibasic cleavage is directed by the kexin family of endoproteases whereas monobasic cleavage is conducted in yeast by the yapsin family of endoproteases [35] and by the furin-type of proteases in Trichoderma [58]. A significant group of proteins, including mainly the proteases, is processed by autocatalytic cleavage [50].

### 3.2.4 DIBASIC PROCESSING

Multiple Sequence Alignment results show that fungal pro-peptides of secreted proteins are cleaved following a dibasic site. In the majority of dibasic processing sites, cleavage takes

place following a “Lys-Arg” pair, whereas “Lys-Lys” and “Arg-Arg” pairs are less frequently encountered [50].

### 3.2.5 MONOBASIC PROCESSING

A remarkable number of filamentous fungal extracellular proteins possess a monobasic cleavage site at their leader-mature protein junction. Considering the putative pro-peptides of proteins that are subject to monobasic processing, a common sequence motif does not exist, with the exception of a proline that is consistently present and frequently adjacent to a Leu or Ile. The fact that the pro-peptides contain both hydrophilic and hydrophobic residues and the absence of sequence homology could either indicate processing by different proteases or the importance of conformational determinants for cleavage; in the latter case, the presence of a proline may be highly significant. In filamentous fungi, there are no examples where proline is present immediately before or after the basic residue at the cleavage site. Nevertheless, since the role of proline is suggested to be at the level of a three-dimensional structure, rather than the primary sequence [50], a similar function can still be attributed to the proline residues within the structure of pro-peptides of filamentous fungi where monobasic processing takes place.

## 3.3 MATERIALS AND METHODS

The data set is collected from the NCBI databank based on fungal proteins which are publicly available[1]. 72 fungal sequences are selected among non-homologous protein families. This is one of the reasons for the small number of sequences contained. To reduce further redundancy in the data set and prevent the training and testing from being homologous, we made a phylogenetic tree analysis based on multiple sequence alignment by ClustalW. There, in a phylogenetic tree many individual main branches are resulted (data not shown) indicating that the selected proteins are not homologous. In our learning process by SVM, we chose symmetric windows around possible cleavage sites, where the window length varies between 11 to 21 and the results indicates that the optimum window length lies between 13 and 19. The best accuracy results are found with window length chosen as 15. These parameters can

[1] http://www3.iam.metu.edu.tr/iam/images/1/1a/Datasetsureyya.pdf

vary according to the type of the data set and the kind of problem.

To see the discriminative motifs existing in the sequences, we used *MEME* software[2]. This yielded the motif **KR**. To check this result, *Multiple Sequence Alignment (MLA)*, with the package *ClustalW* is applied to the data set which confirms this observation. The motif **KR** gives us a clue for the preparation of the input sequences for the SVM. With MLA, most of the cleavage site patterns are in the form of either **K**, **R** or **KR**. Therefore, it is sensible to train the SVM restricted to inputs with **K** or **R** residues.

### 3.3.1 INPUT AND OUTPUT FOR THE SVM

There are different ways to represent *text based* data when introducing the data to a learning algorithm. In bioinformatics, these data can be amino acid (a.a.) sequences, DNA sequences, etc.. The most popular method of encoding amino acid sequences into numerical values is given by binary vectors [7]. However, this ignores the *context* information. There has been a lot of research on encoding amino acids to give each individual amino acid a numerical value regarding the biochemical and physiochemical properties [38]. One of the most powerful substitution matrices is PAM250 matrix due to its property of preserving mutations of the sequences. In this study, two types of encoding are considered, namely, a binary encoding matrix and the *PAM250* substitution matrix. Please note that, encoding a.a. by substitution matrices is needed for the input vectors for the SVM. Thus, the windows of a.a. sequences are presented to the SVM with the numerical values corresponding to the input vectors.

There are many similarity matrices developed according to different similarity approaches and gap penalties given between two amino acids. Dayhoff et al. [21] created a table where they aligned the proteins in several families of proteins and constructed phylogenetic trees for each family [21]. The resulting similarity table presents relative frequencies with which amino acids replace each other in a short evolutionary period since each phylogenetic tree is checked for the substitutions found on each branch. The traditional Dayhoff PAM250 matrix assumes the occurrence of 250 point mutations per 100 amino acids or 300 nucleotides in the gene [47].

PAM matrices are theoretically more advantageous than the others. They arise from Day-

[2] http://meme.sdsc.edu/meme/meme-output-example.html

hoff's method [21] which is based on observed evolutionary mutations. Hence, they preserve information given by the processes that generate the mutations. Statistically, PAM matrices and other log-odds matrices are the most accurate description of the changes in the amino acid composition after a given number of mutations. Details about the formulation of log odds matrices and PAM matrices can be found in [1, 21].

Since we have 20 amino acids, we have entries in a 20×20 PAM250 matrix. Each amino acid is represented by a 20 dimensional vector corresponding to the entries in a column of the PAM250 matrix. If there is a sequence of $n$ amino acids, then we will have an $n \times 20$ dimensional real-valued vector as input.

### 3.3.2 SLIDING WINDOW APPROACH FOR CONSTRUCTING A TEST SET

The *sliding window approach* is a method to construct the training and test set with a previously chosen window size. Training windows are chosen from the neighbourhood of the potential cleavage sites in such a way that the cleavage sites are at the center of the window. For example, if we have a window size of 11, then the considered cleavage site is between the 5th and the 6th position of the window. In this way, each sequence contributes one positive window. For the negative class, three windows are chosen from each sequence by selecting positions which have residues **K** or **R** at their center. Here, windows are chosen as symmetric in all cases. A test sequence is constructed by sliding the window through the whole sequence as illustrated in Figure 3.3. In our case, all the sequences have at least one **K** or **R** which are the motifs that we learned from ClustalW through multiple sequence alignment. Sliding windows through the whole sequence generate many test windows, i.e., test inputs. Furthermore, the cleavage window(s) in the test sequence are going to be labeled as a positive class from the output of SVM and the others as a negative class. It is clear that restricting the windows by including to those windows that have **K** or **R** at their center will decrease the number of test examples and, hence, makes it easier to select the positive one(s) (cleavage window(s)) when compared to the high number of windows for a particular test sequence. In other words, if we call the set of all sliding windows $S$ and choose a special subset $A \subseteq S$ which depends on motifs known in advance from a bioinformatics tool, then searching a cleavage window(s) among $A$ will be easier than searching from the bigger set $S$ for a particular test sequence. If the set $A$ is empty, i.e., $A = \emptyset$, the set $S$ which contains all possible windows of the particular

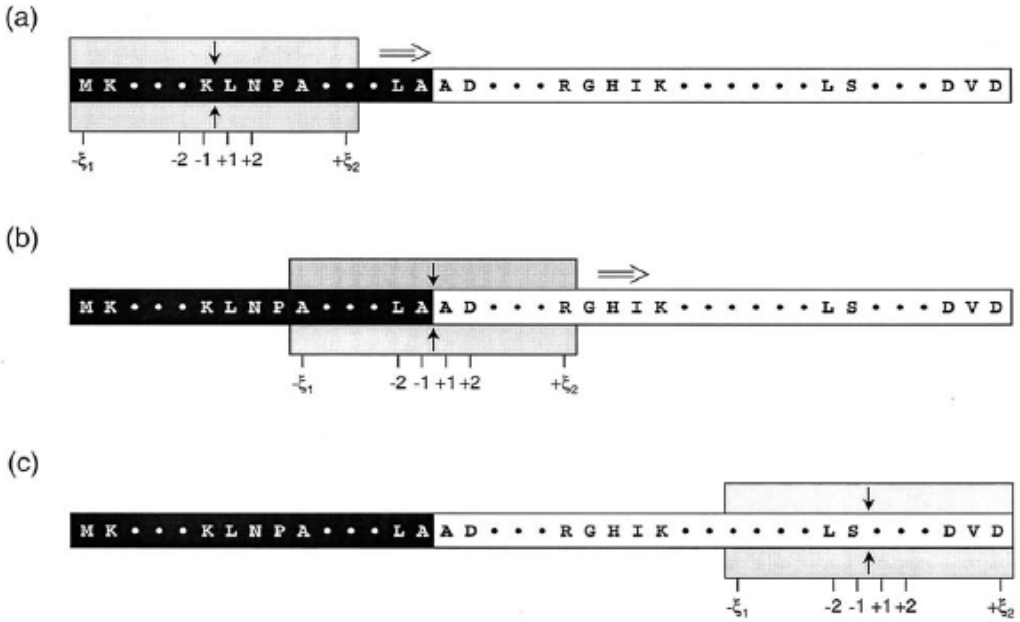


Figure 3.3: Black parts denote pro-peptide region and white parts stand for the mature part of the protein [17]: (*a*) The window does not containing the cleavage site. (*b*) The window contains the cleavage site at its center. (*c*) The window does not contain the cleavage site.

test sequence can be used as test examples. In our special data set on fungal proteins, the subset $A$ of $S$ is nonempty, i.e., $A \neq \emptyset$. Moreover, the cardinality of $A$ is always greater than 3, i.e., $|A| \geq 4$.

Our data set comprises 72 proteins and, hence, 72 amino acid sequences, each giving rise to be one positive window and three negative windows. So, 72 sequences are used for both training and testing using the *leave one out* principle that leaves each sequence in turn as testing while using the remaining 71 for training. In this way, we have trained using 71 sequences and have tested 1 sequence 72 times. The accuracy is calculated as the percentage of the total number of correct predictions over the 72 sequences.

### 3.3.3 KERNEL DEFINITIONS

For string data, an SVM can make use of string kernels which are described in [67]. These types of kernels can be used in text mining, DNA sequences and protein sequences. Since

measuring the similarity between the windows of sequences is one of the most crucial items, a novel kernel function is defined with an explicit mapping $\Phi$ which measures the similarity of windows by counting the number of matching sequences in a neighbourhood of the potential cleavage site and it is shown to be a *Gaussian kernel* in Proposition 3.3.1. Thus, the calculation of very high-dimensional vectors $\Phi$, is avoided by the use of a Gaussian kernel.

In our first method, which is explained in Subsection 3.3.1 with PAM250 matrices, we choose the Gaussian kernel while using the LibSVM package [14]. In order to motivate our second choice of the kernel, we consider counting the number of matching sequences between two windows.

Let us regard windows to be sequences of amino acids indexed by $\{1, 2, \ldots, n\}$. Moreover, let the feature space $F_r$ be indexed by pairs $(s, \mathbf{i})$, where $s$ is a sequence of $r$ amino acids and $\mathbf{i}$=$(i_1, i_2, \ldots, i_r)$ a tuple of $r$ distinct indices, $i_j \in \{1, 2, \ldots, n\}$. We define the mapping $\Phi : w \mapsto (\phi_1(w), \phi_2(w), \ldots, \phi_r(w), \ldots) \in \prod_{r\geq 1} F_r$ by

$$\phi_r(w)_{(s,\mathbf{i})} = \begin{cases} \alpha^{r/2}, & \text{if } w_{\mathbf{i}} = s \\ 0, & \text{otherwise,} \end{cases} \tag{3.1}$$

where $w_{\mathbf{i}} = s$ means $w_{i_j} = s_j$ $(j = 1, 2, \ldots, r)$ and $\alpha \in \mathbb{R}, \alpha > 0$.

The feature space in which the learning will be conducted is $\prod_{r\geq 1} F_r$. It is worth noting that this is a very high dimensional space. For example, $F_5$ has dimension

$$20^5 \binom{n}{5},$$

though, clearly, for $r > n$ all $\binom{n}{r}$ become 0. So, for any fixed $n$, the effective dimension is finite. The feature space makes it possible for the learning to assign weights for each pattern of positions and corresponding choice of amino acids at those positions.

This choice of feature space ensures that the learning can readily identify the salient patterns which indicate the presence of a cleavage site. Naturally, it will not be practical to compute this feature vector explicitly. Now, we show in the next proposition that there is an efficient method of computing the kernel corresponding to the feature map $\Phi$. This opens the way for us to learn in this feature space by using the kernel methods approach introduced above.

Let us consider (with a slight abuse of notation) the feature vector

$$\Psi(w) = [u_{w_1}, u_{w_2}, \ldots, u_{w_n}]^T,$$

where $u_a$ is defined as $u_a = [0 \ldots 010 \ldots 0]_{1\times 20}$ with the value 1 in the position corresponding to the amino acids ($a \in \{w_1, w_2, \ldots, w_n\}$).

**Proposition 3.3.1** *[53] . Using the defined notation above, we have for all windows v, w of size n:*

$$\begin{aligned} k(v,w) &= \langle \Phi(v), \Phi(w) \rangle \\ &= (1+\alpha)^n \exp\left( -\frac{||\Psi(v) - \Psi(w)||_2^2 \ln(1+\alpha)}{2} \right), \end{aligned} \tag{3.2}$$

where $\langle \cdot, \cdot \rangle$ is an inner product.

**Proof 1** *If we fix a number r of matches, we compare two windows by counting the number of r tuples of position that contain the identical set of amino acids. If the number of positions where the sequences of two windows agree is m, then the number of r tuples is given by the binomial coefficient $\binom{m}{r}$. This is the inner product associated with the high-dimensional representation $\phi_r$. Let us denote this kernel by $k_r(v,w) := \alpha^r \langle \phi_r(v), \phi_r(w) \rangle$. Observe that by using a combination of these kernels, we can create our measure of similarity:*

$$k(v,w) = \sum_{r=0}^{\infty} \kappa_r(v,w) = \sum_{r=0}^{\infty} \alpha^r \binom{m}{r} = \langle \Phi(v), \Phi(w) \rangle .$$

*Here, $m := \#\{i : v_i = w_i, i = 1, 2, \ldots, l\}$, which we will denote by $\#[v == w]$, gives the number of positions in which the two sequences agree.*
*Therefore, from the Binomial Theorem we learn*

$$k(v,w) = (1+\alpha)^{\#[v==w]}. \tag{3.3}$$

*We note that $\langle \Psi(v), \Psi(w) \rangle = \#[v == w]$, while $||\Psi(v)||_2^2 = n$.*

*Letting $m = \#[v == w]$, we have*

$$\begin{aligned} k(v,w) &= (1+\alpha)^m \\ &= \exp\left[ m\ln(1+\alpha) - ||\Psi(v)||_2^2 \ln(1+\alpha)/2 - ||\Psi(w)||_2^2 \ln(1+\alpha)/2 \right] (1+\alpha)^n \\ &= (1+\alpha)^n \exp\left( -\frac{||\Psi(v) - \Psi(w)||_2^2 \ln(1+\alpha)}{2} \right). \end{aligned}$$

*Hence, the kernel turns out to be*

$$k(v, w) = (1 + \alpha)^n \exp\left(-\frac{\|\Psi(v) - \Psi(w)\|_2^2 \ln(1 + \alpha)}{2}\right),$$

*as required.*

**Remark 3** *We note that equation (3.2) is a scaled Gaussian kernel with kernel width* $\sigma = \sqrt{\frac{1}{\ln(1+\alpha)}}$ *by the definition of the Gaussian kernel over the features* $\Psi(\cdot)$*. We again consider both normalized and unnormalized versions of the features* $\Psi(\cdot)$*, though this only affects the scaling of the kernel width since* $\|\Psi(v)\|_2^2 =$ *constant.*

## 3.4 MODEL SELECTION PROCEDURE

The definition of the kernel and the SVM algorithm both involve an additional parameter vector $(C_+, C_-, \sigma)$, the parameters $C_+$ and $C_-$ for the SVM and the kernel width $\sigma$ for the Gaussian kernels. The usual way to set these parameters is using cross-validation [30]. This assesses the quality of different parameter settings by dividing the training data into $m$ groups. It then leaves out one group in turn to train the classifier with a range of possible values for the parameters and uses the group left out as a test set. The average accuracy for each parameter setting over all $m$ test groups is then used to select the parameter settings. We employed this approach where we took $m = 71$, i.e., we performed a subround of "leave one out" error estimation on each training set in order to select the parameters to use training for the set of 71 sequences before testing on 72nd left out sequence. Note that this is the only *leave one out* at the level of sequences, since each sequence corresponds to 4 windows, one of which is positive.

Our second method of model selection is a novel approach for problems in which each test involves multiple inputs, but with the additional information that only one is positive: in our case, there are many windows, but only one is a cleavage site. Rather than to pre-select the parameters, we train the SVM on all the training data (other than the single test sequence) with all the parameter settings. For each SVM we compute the real-valued outputs, for all the windows arising from the sequence. We define the confidence of the classifier as the difference between the maximal output and the second largest; see Figure 3.4. Now, we select the parameter settings for which the confidence is largest and identify the window with

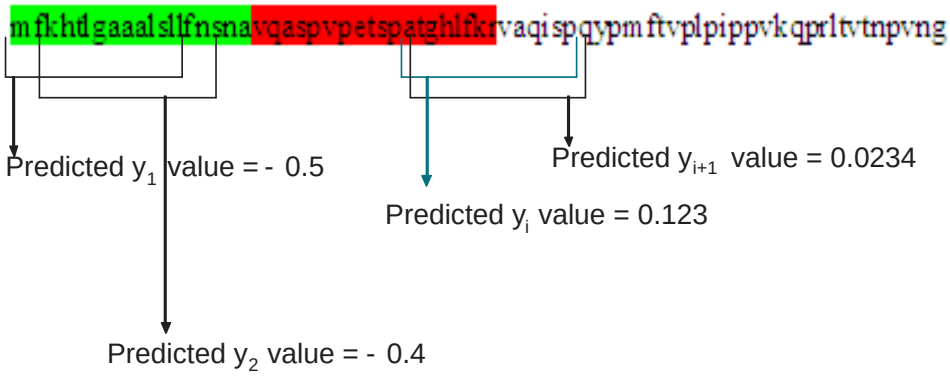


Figure 3.4: Different real-valued outputs from the SVM. The confidence level is the highest difference between two maximum positive outputs.

maximal output as the cleavage site. It should be stated that not every test sequence has to have a cleavage site. It corresponds to having all test window outputs being negative. In such cases, our algorithm outputs that these sequences do not have a cleavage site.

The model selection method involves choosing among a number of support vector machines with different parameter settings for the kernel and regularization. The question of consistency of support vector machines has been studied by a number of authors. For example, Steinwart [76] shows a dependence on the choice of kernel and regularization parameters, so that a priori consistency is not guaranteed for a fixed value of the regularization parameter. It is an interesting question whether our method can choose from an appropriate sequence of regularization parameters to ensure consistency without the need for handcrafted choices. This question is, however, beyond the scope of the thesis.

## 3.5 RESULTS AND DISCUSSION

Our data set consists of 72 sequences from fungal proteins selected among non-homologous proteins with known pro-peptide regions, determined by N-terminal amino acid sequencing. This has limited the number of available protein sequences but, is expected to have enhanced accuracy. We initialize the parameter $C_+$ from 0.5 and increased it by the factor of 2 for 6 iterations for both the confidence level method and cross validation. For each value of $C_+$, $C_-$

was initialized to $C_+/4$ and increased by a factor of 2 for 4 iterations. Likewise, we initialize $\sigma$ to $2^{-8}$ and multiply by a factor of 2 for 6 iterations. Accuracy results are given in Table 3.1. We compare our results with the ProP1.0 server [23] and the full 71 cross validation. As it can be seen in Table 3.1, the best accuracy is achieved with the model selection method proposed in this study by confidence level with SVM and our second approach with normalized binary inputs. We test our data set on **ProP1.0** server and it gave **61%** accuracy on the 72 test sequences. Our novel approach improved on the accuracy of ProP1.0 server [23] by **15%**, although our training data set is 3 times smaller than that used in the neural network approach described in [23] which used 227 proteins. Furthermore, parameter selection with the *confidence level* gives higher accuracy than cross validation.

When we compare our confidence level based approach with cross validation, we see from Table 3.2 that the computational complexity of training times of the confidence level method is significantly shorter than cross validation. Here, we show the average of the elapsed time in training each leave one out phase, i.e., the results in Table 3.2 give the approximate time in seconds per test sequence in training. As it can be seen from Table 3.1 and Table 3.2, the best method both in training and accuracy is the *confidence level* with binary inputs.

Table 3.1: Accuracy results of SVM [53].

| Input type | Cross validation | Confidence level |
|---|---|---|
| Normalized data encoded by binary vec. | 44% | **76%** |
| Not normalized data encoded by binary vec. | 47% | 75% |
| Normalized data encoded by PAM250 | 37% | 58% |
| Not normalized data encoded by PAM250 | 33% | 56% |

Table 3.2: Average training time for the SVM for one of the 72 test sequence [53].

| Input type | Cross Validation | Confidence Level |
|---|---|---|
| Normalized data encoded by binary vec. | 151 sec. | **4** sec. |
| Not normalized data encoded by binary vec. | 163 sec. | 5 sec. |
| Normalized data encoded by PAM250 | 1312 sec. | 23 sec. |
| Not normalized data encoded by PAM250 | 1924 sec. | 34 sec. |

## 3.6 CONCLUSION AND PERSPECTIVES

Our paper has considered the problem of identifying the cleavage site for fungal pro-peptides which can be extended in general to eukaryotic proteins. This task has previously been tackled by a neural network [23]. We presented a kernel-based solution with two novel features: A *kernel* specifically defined for the task enabling the learning to take place using linear functions in a very high-dimensional feature space; and the implementation of *model selection* at the test point evaluation phase, rather than by cross validation. Both of these innovations lead to a significant improvement in classification accuracy on a real-world data set as well as giving results that are an improvement over the earlier approaches. It would be interesting to apply the kernel introduced here to other sequence analysis tasks. The approach to model selection is interesting in that it gives an improved performance with very significantly reduced training times. This approach should be evaluated more widely on standard evaluation tasks and also using results similar to those of [65], the approach can be placed on a sound theoretical footing.

# CHAPTER 4

# MODEL SELECTION ALGORITHMS

## 4.1 INTRODUCTION

This chapter is devoted to the our model selection technique which is based on *maximum margin principle* of SVM [19, 30]. We generalize our *confidence interval* approach (see Chapter 3) with different $L_p$-norms ($p = 1, 2, \infty$) in this chapter and we apply this generalized methodology to various data gathered from *UCI machine learning repository*[1]. This chapter explains the second contribution of the thesis which is compared with well-known model selection algorithm called *cross validation* [30]. We will start with explaining the cross validation method in the first section and continue with our model selection method in the following sections.

Throughout this chapter, we will use $k$ as the number of folds in cross validation method. There should not be any confusion with the kernel functions as $k$ is understood by in other chapters.

## 4.2 CROSS VALIDATION

Cross validation is a method for choosing the best fitted model or function for a given data. The main idea behind is partitioning of the data into subsets such that the analysis is initially performed on a single subset, while the other subset(s) are saved for subsequent use in confirming and validating the initial analysis. The selection of the best model (or hypothesis) depends on the error rates of the validation sets for each subset of the partitioned data. There are three kinds of cross validation methods:

[1] http://archive.ics.uci.edu/ml/

1. *hold-out cross validation*,

2. *k-fold cross validation*, and

3. *leave one-out cross validation*.

**Hold-out cross validation:** Hold-out cross validation is not cross validation in a common sense since the data is not crossed over: The data is splitted into two as training and testing and the function approximator (hypothesis) fits a function using the training set only. Although it does not spend too much time in training, evaluation can have a high variance since it depends on the random selection of training and testing.

***k*-fold cross validation:** $k$-fold cross validation is an improved way of hold out cross validation. It is based on splitting the data set into $k$-folds to minimize the high variance problem in hold out cross validation. It repeatedly chooses one fold as validation and the remaining $(k-1)$-fold for training $k$-times. The disadvantage of this method is that the training algorithm has to be rerun $k$ times, which means that it takes $k$ times as much computation to make an evaluation. The advantage of doing this is that we can independently choose how large each validation set is and how many trials we average over.

**Leave one-out cross validation:** As from its name, leave one-out cross validation chooses one single observation for validation and the remaining portion of the data for the training process. Thus, it is an $l$-fold cross validation where $l$ is the number of the data points (number of observations) in the data set. The evaluation given by leave-one-out cross validation error is good, but it is very expensive to compute.

As it is explained above, the main disadvantage of all the cross validation type is that the computation is expensive as the number of folds increase. Note that there is a trade-off between computation cost and the cross validation accuracy. As the cross validation accuracy increases (number of folds increase), the computation costs increase. We develop a new model selection method based on the maximum margin principle but on the test points. In other words, the selection depends on the maximum distance of the test points to the classifiers on the training points. By this way, we save a lot of training time when it is compared with cross validation. In the following section, we explain our new model selection method with examples on real data.

## 4.3 SVM MODEL SELECTION BASED ON OBSERVED MARGIN

### 4.3.1 INTRODUCTION

Support vector machines (SVMs) carry out binary classification by maximizing the margin of a hyperplane between the two classes of examples and then classifying test points according to the half-spaces in which they reside (irrespective of the distances that may exist between the test examples and the hyperplanes). In cross validation, the principle idea is to find the *one* SVM model and its optimal parameters that help to achieve the smallest training error amongst all of the models that can be constructed. In contrast, we collect *all* of the models found in the model selection phase and produce predictions for test points by finding the SVM models whose hyperplanes achieve the maximum distance from the test points. In this setting, we avoid the complex and time consuming paradigm of model selection via cross validation. Experimental results demonstrate the plausibility of the method proposed and show a significant decrease in computational time as well as a competitive generalization error.

For all kinds of data mining tools, parameter selection is one of the critical questions; it determines the right model for data analysis and prediction. In this chapter, we mainly develop a fast algorithm for model selection which uses the benefit of all hypothesis space by means of functions or models [52]. We apply our model selection approach called *maximum margin* to the binary classification problems by using support vector machines (SVMs) which, as we recall, is one of the most efficient methods in machine learning.

## 4.4 METHODS

In this section, three different norms will be discussed for model selection at the testing phase. Given a set of functions in the variable $\mathbf{x}$, $f_1, f_2, \ldots f_\ell$ being the outputs by the SVM, with $\ell = |C| \cdot |\sigma| = \ell_1 \cdot \ell_2$, then being the number of models that can be constructed from the set of parameter values $C \in \{C_1, C_2, \ldots, C_{\ell_1}\}$ and $\sigma \in \{\sigma_1, \sigma_2, \ldots, \sigma_{\ell_2}\}$, where $C$ is the error constant and $\sigma$ is the Gaussian kernel width. We can use some or a combination of models derived by these parameters in order to make predictions. The first approach which we propose uses the $L_\infty$-norm for choosing which function to use. This is equivalent to evaluating

the distance of a test point according to the function that achieves the largest (functional) margin. For example, let us assume we have three values for $C \in \{C_1, C_2, C_3\}$ and two values for $\sigma \in \{\sigma_1, \sigma_2\}$, respectively. Therefore, we have the following $\ell = 6$ SVM models together with their set of parameter values $\{C, \sigma\}$:

- $f_1 = \text{SVM}_1$: $\{C_1, \sigma_1\}$,
- $f_2 = \text{SVM}_2$: $\{C_1, \sigma_2\}$,
- $f_3 = \text{SVM}_3$: $\{C_2, \sigma_1\}$,
- $f_4 = \text{SVM}_4$: $\{C_2, \sigma_2\}$,
- $f_5 = \text{SVM}_5$: $\{C_3, \sigma_1\}$,
- $f_6 = \text{SVM}_6$: $\{C_4, \sigma_4\}$.

Now, for the functional values, we would compute the functional margins at each test points. For instance, given a test example $\mathbf{x} = \mathbf{x}^0 \in X_{test}$, where $X_{test}$ is a collection of data points to be tested, let us assume the following six functional values:

- $f_1(\mathbf{x}^0) = 1.67$,
- $f_2(\mathbf{x}^0) = 0.89$,
- $f_3(\mathbf{x}^0) = -0.32$,
- $f_4(\mathbf{x}^0) = -0.05$,
- $f_5(\mathbf{x}^0) = 1.1$,
- $f_6(\mathbf{x}^0) = 1.8$.

We assume here, without loss of generality, that the functions $f$ compute the functional margins and not the geometrical margins (hence, the reason that the example values we have presented are not bounded by 1 and -1). Finally, we would predict the class of $\mathbf{x}^0$ by looking for the maximal (positive) and the minimal (negative) value of all functions. This corresponds to $f_6$ and $f_3$, respectively. However, the distance of the test example $\mathbf{x}^0$ from the hyperplane is greater for the $f_6$ =$\text{SVM}_6$ function (model) and, therefore, this example can be predicted as

positive. For this reason, the $L_\infty$ prediction function $F_\infty(x)$ evaluated at our given example $\mathbf{x}^0$ can be defined in the following way:

$$F_\infty(x) \quad := \quad \operatorname{sgn}\left(\max\{f_i(x)\}_{i=1}^{\ell} + \min\{f_i(x)\}_{i=1}^{\ell}\right), \tag{4.1}$$

where sgn denotes the sign function, i.e., positivity or negativity of a function, as defined in Section 2.2 in equation (2.1)

The $L_\infty$-norm approach is also illustrated in Figure 4.1 on a real-world data set which gives the evaluations of 110 SVM models[2] (i.e., functional margin values), sorted in ascending order, for a particular test point. From Figure 4.1 the maximum positive margin (the far right most bar) and minimum negative margin (the far left most bar) are shown in black. Hence, the sign of the sum of these two function margin values will give us the prediction of the test point. This test point is classified as in the positive class.

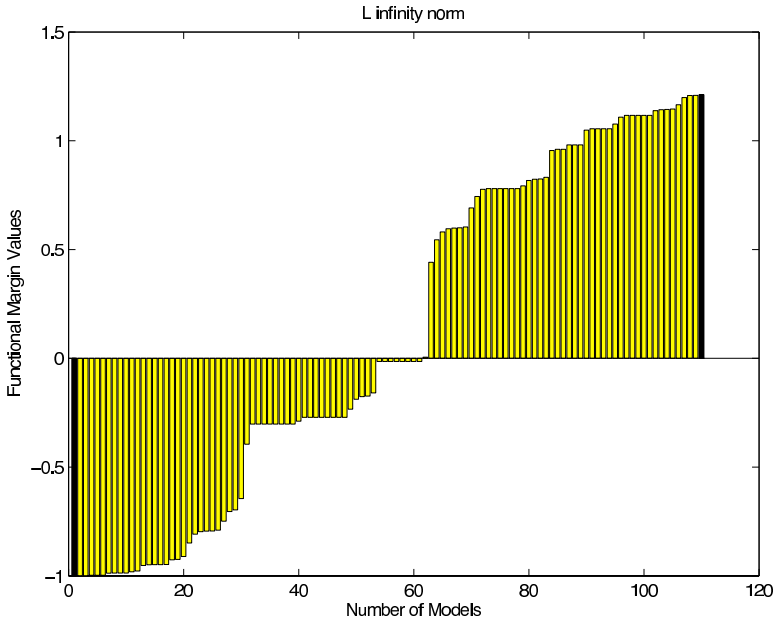


Figure 4.1: The graph of $L_\infty$-norm which predicts the example as +1 where actual class is +1. Each bar corresponds to the functional margin value given for that particular SVM model $f$. Bold bars corresponds to $\max\{f_i(x)\}_{i=1}^{\ell}$ and $\min\{f_i(x)\}_{i=1}^{\ell}$ [52].

The second approach which we introduce is for the $L_1$-norm where the decision depends on the sign of the Riemann sum of all outputs evaluated for a test point [52]. This results in the

[2] There are 11 $C$ values = $\{2^{-5}, 2^{-3}, 2^{-1}, 2, 2^3, 2^5, 2^7, 2^9, 2^{11}, 2^{13}, 2^{15}\}$, and 10 $\sigma$ values: $\{2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2, 2^3\}$

following $L_1$-norm prediction function $F_1$ given a test example $\mathbf{x}$, e.g., $\mathbf{x}^0$:

$$F_1(x) \quad = \quad \text{sgn}\left(\sum_{i=1}^{\ell} f_i(x)\right). \tag{4.2}$$

This is illustrated in Figure 4.2. In the sense of Riemann sums it is obvious that the prediction function looks at the integrals of the two areas (indicated in black) above and below the threshold of 0. Essentially, this equates to summing the above and below bars. In Figure 4.2, it is clear that the summation will be positive since the area of the positive values (above 0) is bigger than the area of the negative values (below 0). This methodology corresponds to summing the weighted average of all the prediction functions with a uniform weighting of 1. Indeed, this is closely related to taking a weighted majority vote [60, 75].

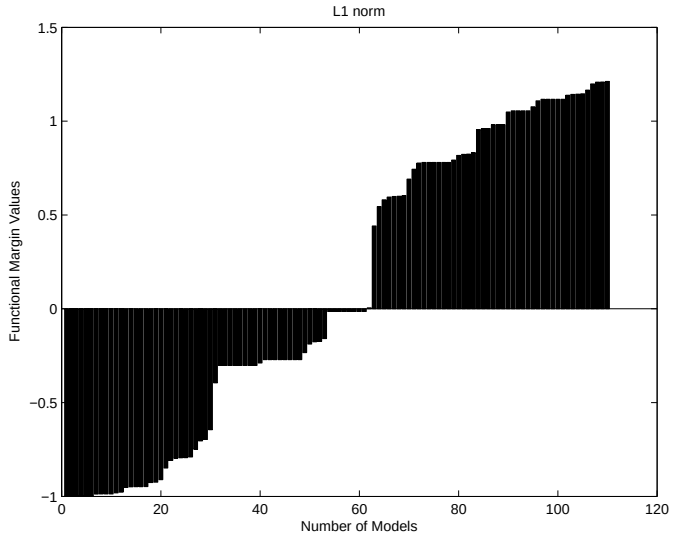


Figure 4.2: Illustration of $L_1-$norm approach. Each bar corresponds to the functional margin value given for that particular SVM model $f$. The $L_1$-norm predicts +1, and the actual class of the example is +1 [52].

Our final approach [52] corresponds to the $L_2$-norm and is similar to the one with the $L_1$-norm discussed above, but with a down-weighting if the absolute values are less than 1 and an up-weighting if they are above 1. This means that we are giving a greater confidence to functions that predict functional values greater than 1 or less than -1 but less confidence to those that are closer to the threshold of 0. Another way of thinking about this approach is that it is equivalent to a weighted combination of functional margins with the absolute values of themselves. Therefore, given a test example $\mathbf{x}$, e.g., $\mathbf{x}^0$, we have the following $L_2$-norm

prediction function $F_2(\mathbf{x})$ defined by

$$F_2(\mathbf{x}) \quad := \quad \text{sgn}\left(\sum_{i=1}^{\ell} f_i(\mathbf{x})|f_i(\mathbf{x})|\right). \tag{4.3}$$

The plot of Figure 4.3 represents the $L_2$-norm solution for the same test point predicted by the $L_2$-norm. As we can see, the yellow region corresponds to the original values of the functions and the black bars are the down-weighted or up-weighted values of the 110 prediction functions. The $L_2$-norm corresponds to summing the weights of the black bars only. It can be seen that the absolute values that are smaller than 1 are down-weighted (decreased) and those greater than 1 are up-weighted (increased). Obviously, values that are close to 1 do not change significantly.

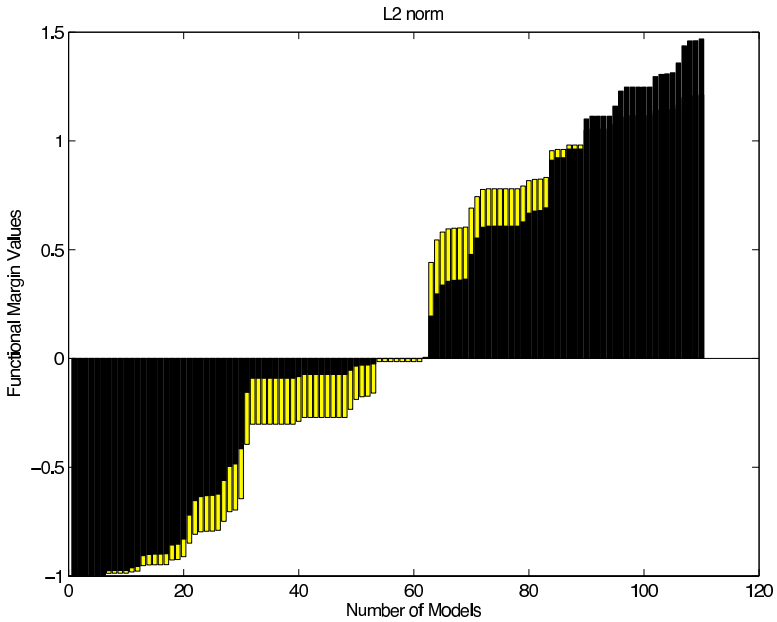


Figure 4.3: Illustration of $L_2$-norm approach. Each bar corresponds to the functional margin value given for that particular SVM model $f$. The $L_2$-norm predicts +1, where the actual class of the example is +1 [52].

## 4.5 DATA SET DESCRIPTION

In this study, we used the well-known *standard UCI machine learning repository*[3]. From the repository, we used the Votes, Glass, Haberman, Bupa, Credit, Pima, BreastW, Ionosphere,

[3] http://archive.ics.uci.edu/ml/

Australian Credit and the German Credit data sets. For the first seven data sets, we removed examples containing unknown values and contradictory labels (this is why the votes data set is considerably smaller than the one found at the UCI website). The number of examples, attributes and class distributions of all the data sets are given in Table 4.1.

Table 4.1: Description of the data sets chosen from UCI machine learning repository. First column represents the name of the data set, second column represents the number of data, third column represents the number of features, the fourth and fifth columns represent the numbers of the positive and the negative examples, respectively.

| Data set | # instances | # attributes | # pos | # neg |
|---|---|---|---|---|
| Votes | 52 | 16 | 18 | 34 |
| Glass | 163 | 9 | 87 | 76 |
| Haberman | 294 | 3 | 219 | 75 |
| Bupa | 345 | 6 | 145 | 200 |
| Credit | 653 | 15 | 296 | 357 |
| Pima | 768 | 8 | 269 | 499 |
| BreastW | 683 | 9 | 239 | 444 |
| Ionosphere | 351 | 34 | 225 | 126 |
| Australian | 690 | 14 | 307 | 383 |
| German | 1000 | 20 | 300 | 700 |

## 4.6 RESULTS

We call our methods the SVM-$L_\infty$, SVM-$L_1$ and SVM-$L_2$ [52] which correspond to using the $L_\infty$-, $L_1$- and $L_2$-methods that we proposed in Section 4.4. We also test our methods against the SVM with cross validation (CV), where we carry out 10-fold cross-validation to estimate the optimal $C$ and $\sigma$ values. We note that in the methods which we propose, we do not need to carry out this parameter tuning phase and, hence, achieve a 10 fold speed-up against the SVM with CV.

Table 4.2 and 4.3 present the results, including the standard deviation (STD) of the error over the 10-folds of cross-validation, the cumulative training and testing time (column "time"') in seconds for all folds of CV, the error as percentages (column "error %"), as numbers (column "error # "), the Area Under the Curve (column "AUC") and the average over all data sets for the entire 10-fold cross-validation process.

The results of the SVM-$L_p$ where $p = \infty, 2, 1$, show a significant decrease in computational

time when compared to the SVM with CV. For example, we can see that the German data set takes approximately 4368 seconds to train and test and that our methods take between 544 and 597 seconds for training and testing purposes. This is approximately 8 times faster than using cross-validation. We can also see from Table 4.2 and that the $L_\infty$-method seems to capture better prediction models compared to the other two $L_p$-norm methods ($p = 1, 2$) given by Table 4.3, but all three methods compare favorably with respect to test error against the SVM with CV. Since several data sets are imbalanced (see Table 4.1), we also report AUC results in Table 4.2 and Table 4.3. It is well known that as the AUC tends to 1, the better the prediction accuracy becomes [26]. In Table 4.2, we can see that the $L_\infty$-norm has greater AUC values than the other $L_p$- norm methods ($p = 1, 2$).

Finally, when comparing the three methods proposed it is clear that the most successful in terms of speed and accuracy is the $L_\infty$-norm. This is perhaps less surprising when viewed from the theoretical motivation of this work, as [65] has proposed a bound that gives higher confidence of correct classification if the test point achieves a large separation from the hyperplane. This is exactly what the $L_\infty$-norm method does. The other $L_p$-norm methods ($p = 1, 2$) do not have such a theoretical justifications.

## 4.7 CONCLUSION

We proposed a novel method for carrying out predictions with the SVM classifiers once they had been constructed using the entire list of regularization parameters (chosen by the user). We showed that we could apply the $L_p$-norms ($p = 1, 2, \infty$) to help picking these classifier(s). Moreover, we introduced the SVM-$L_\infty$, SVM-$L_1$ and SVM-$L_2$ strategies and discussed their attributes with real-world examples [52]. We showed that the $L_\infty$-method would choose a single classifier for prediction, the one that maximizes the distance of a test point from its hyperplane. The $L_1$- and $L_2$-norms were similar to each other and gave predictions using a (weighted) sum of the prediction functions constructed by each SVM function. Finally, in Section 4.6 we gave experimental results that elucidated the methods described in this chapter.

The main benefit of the work proposed [52] can be for imbalanced data sets in which there are great differences between the size of the classes. In such situations, e.g., fraud detection, we may have a very large number of examples but only a small number of examples are fraud

Table 4.2: $L_\infty$-norm results against SVM with Cross-Validation [52].

| Data Set | SVM with CV | | | | | SVM-$L_\infty$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | STD | time | err % | err # | AUC | STD | time | err % | err # | AUC |
| Votes | 0.2115 | 11.54 | **8.33** | 5 | 0.9417 | 0.0991 | **0.94** | **8.33** | **5** | 0.8333 |
| Glass | 0.1222 | 89.93 | 34.85 | 57 | 0.7556 | 0.135 | **16.29** | **31.99** | **52** | 0.7095 |
| Haberman | 0.0437 | 169.51 | **24.81** | **73** | 0.7253 | 0.0363 | **23.18** | 25.17 | 74 | 0.7163 |
| Bupa | 0.0648 | 354.46 | **28.95** | **100** | 0.5437 | 0.0611 | **82.87** | 31.55 | 109 | 0.4535 |
| Credit | 0.1916 | 1812.86 | **13.49** | **88** | 0.9391 | 0.1439 | **370.21** | 18.09 | 118 | 0.9144 |
| Pima | 0.038 | 1300.82 | **23.81** | **183** | 0.7227 | 0.03 | **265.9** | 26.17 | 201 | 0.6649 |
| BreastW | 0.0192 | 414.18 | **3.35** | **23** | 0.9881 | 0.0361 | **111.26** | 4.81 | 33 | 0.9771 |
| Ionosphere | 0.0568 | 172.21 | **6.21** | **22** | 0.9507 | 0.0512 | **76.61** | 8.19 | 29 | 0.9211 |
| Australian | 0.0416 | 1868.16 | **14.80** | **102** | 0.7111 | 0.0333 | **223.33** | **14.97** | 103 | 0.6803 |
| German | 0.0454 | 4378.01 | **23.80** | **238** | 0.8678 | 0.0378 | **544.93** | 26.60 | 266 | 0.8488 |
| Average | 0.0834 | 1057.168 | **18.12** | **89.1** | 0.81 | 0.066 | **171.552** | 19.91 | 99 | 0.77 |

Table 4.3: $L_1$- and $L_2$-norm results [52].

| Data Set | SVM-$L_1$ | | | | | SVM-$L_2$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | STD | time | err % | err # | AUC | STD | time | err % | err # | AUC |
| Votes | 0.095 | 1.84 | 27 | 14 | 0.4875 | 0.0979 | 0.9 | 19.17 | 10 | 0.6875 |
| Glass | 0.1125 | 13.28 | 39.27 | 64 | 0.6401 | 0.1282 | 9.59 | 36.92 | 60 | 0.6363 |
| Haberman | 0.0127 | 23.44 | 25.49 | 75 | 0.7183 | 0.0133 | 13.62 | 25.83 | 76 | 0.7183 |
| Bupa | 0.0089 | 80.59 | 42.02 | 145 | 0.1632 | 0.0089 | 80.59 | 42.02 | 145 | 0.2117 |
| Credit | 0.0984 | 245.43 | 18.52 | 121 | 0.9165 | 0.1036 | 199.31 | 18.22 | 119 | 0.9228 |
| Pima | 0.0101 | 183.89 | 34.64 | 266 | 0.5336 | 0.0101 | 183.89 | 34.64 | 266 | 0.5791 |
| BreastW | 0.0421 | 72.35 | 4.8 | 33 | 0.9721 | 0.0363 | 105.88 | 3.78 | 26 | 0.9796 |
| Ionosphere | 0.0694 | 54.91 | 14.16 | 50 | 0.8673 | 0.0652 | 96.96 | 11.62 | 41 | 0.8980 |
| Australian | 0.0195 | 225.98 | 17.29 | 119 | 0.5928 | 0.0186 | 236.45 | 16.71 | 114 | 0.6067 |
| German | 0.0084 | 569.33 | 29.60 | 296 | 0.8187 | 0.0097 | 596.79 | 28.60 | 286 | 0.8245 |
| Average | 0.0477 | 147.104 | 25.19 | 118.3 | 0.67101 | 0.049 | 152.398 | 23.62 | 114.3 | 0.70645 |

cases (positive examples, i.e., $f(x) = +1$). In this case, using CV can be costly as we will tend to use up these little amount of the fraud cases within a large proportion of non-fraud cases and, hence, have a massive imbalance during training. However, in the models which we have proposed, we can use all of the fraud cases and, hence, a larger proportion during training. We have the opinion that this is an area which could greatly benefit from the work proposed in this thesis. Also, removing the CV dependency for finding parameters greatly improves training and testing times for the SVM algorithm.

A future research direction would be to use other methods for choosing the classifiers for testing. Perhaps, a convex combination of the functions would yield better generalization capabilities. Such a combination of functions could be weighted by a factor in the following way:

$$F_\beta(x) \quad := \quad \operatorname{sgn}\left(\sum_{i=1}^{\ell} \beta_i f_i(x)\right) \tag{4.4}$$

where $\sum_{i=1}^{\ell} \beta_i = 1, \;\; \beta_j \geq 0 \;\; (j = 1, 2, \ldots, \ell)$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_\ell)^T$.

Finally, we believe that tighter margin-based bounds could help to improve the selection of the SVM functions at testing. The bound proposed by [65] suggests the $L_\infty$-method which we proposed in this work [52]. However, from Section 4.6, it is clear that this does not always create a smaller generalization error than the SVM with CV. Therefore, a future research direction is to use a tighter bounding principle for the margin-based bound of [65], such as a PAC-Bayes analysis (due to [44], and extended to margins by [41]). Therefore, we could use the bounds to indicate which classifiers to use at testing. We think that a tighter estimate of the bounds would yield an improved generalization.

# CHAPTER 5

# INFINITE KERNEL LEARNING

## 5.1 INTRODUCTION

Real-world data can be supplied from heterogeneous kinds of sources. In such cases, multiple kernels are more convenient to use for a good accuracy. Recent applications [40] showed the need for a *multiple kernel learning (MKL)* because of its interpretability and efficiency. The common approach to MKL is a convex combination of several kernels. Those kernels are selected before and combined to serve well for the embedding into the feature space to do linear separation there. In [8], the kernel-based SVM is formulated by a combination of multiple kernels and solved by *quadratically-constrained quadratic programming (QCQP)* which is applied to solve a dual conic optimization problem. Likewise, [71] uses adapted multiple kernel learning to large-scale problems which applies the method to biological sequence analysis. Since the biological sequences have different motifs inside and for each subsequence, different types of kernels are used, and the combination is taken over the whole sequence. In [71], kernel coefficients are maximized beyond a minimization with respect to the dual variables, which is a max-min type of a problem. It can become canonically represented as a semi-infinite problem [84, 85]. The classical SVM is solved iteratively with linear programming and increasing the number of constraints iteratively in [71]. A different form of an objective function is proposed in [61] for MKL by adapted weighted $L_2$-norm regularization for each function $f$ induced by kernels $k_\kappa$ ($\kappa = 1, 2, \ldots, K$) instead of using the 1-norm block regularization [71] ($K$ denoting some finite number of kernels). Sparsity of linear combinations of kernels is controlled by adding a 1-norm regularization term on these kernel weights.

**Note** (on *Numerical Aspects*:) In our previous studies [52, 53], data are classified regarding the

margin of the test data points and using all classifiers in the hypothesis set. Thus, this benefits from the information of all classifiers and also from the various kernels by different kernel parameters, e.g., in case of Gaussian kernel, the kernel parameter is a Gaussian width. Hence, using different classifiers in different ways, for example, by voting, by ensemble classifiers, gives comparable accuracy results for each test data point and it also improves the speed [52, 53]. In [52], the classification functions depend on one kernel only, but the classification of the new data depends on the results of different combinations of these classifiers on the test points. This improves the accuracy and the speed of the algorithm in the numerical results.

The finite combinations of kernels are limited up to a *finite* choice. This limitation does not always allow to represent the similarity or dissimilarity of data points, specifically highly nonlinearly distributed and large-scaled ones. A finite combination may fail, here. In order to overcome this, with the motivation of previous studies [52, 53], we propose a combination of *infinitely* many kernels in Riemann-Stieltjes integral form (on that form please cf. Section 2.3), for binary classification to allow an infinite wealth of possible choices of kernels in the kernel space. This makes the problem infinite in both its dimension and its number of constraints; which is so-called *infinite programming (IP)*. Our IP problem formulation consists, in the limiting case of infinitely many kernel coefficients $\beta_\kappa$ where $\kappa \to \infty$; this will become represented by a monotonically increasing function (or a probability measure) $\beta$, and an infinite number of constraints coming from the maximal margin principle of SVM. Here, $\beta$ is indeed a monotonically increasing function, as in Section 2.3. Allowing infinitely many kernels might make our problem ill-posed for real-world problems, because of the enormous complexity of the model resulting which is also called *overfitting*. To penalize this curse of dimensionality, we introduce *regularization* terms and approximate "differentiability" in the penalizing term by first- and second-order difference quotients. On the other hand, to solve IP more tractably, we reduced the IP to a semi-infinite problem, by *parametrizing* infinite variables (measures) by parametric *probability density functions (pdfs)*. We will illustrate this parametrization with examples for pdfs.

The organization of this chapter is as follows: In Section 5.2, we will motivate our approach by giving a brief introduction to MKL. In Section 5.3, we will introduce our approach of so-called *infinite kernel learning (IKL)* written by IP problems, we present optimality conditions and find the regularity conditions of the *reduction ansatz* for the *lower level problems* of both the primal and the dual problem. By the reduction ansatz, we get locally finitely con-

strained problems at the place of IP. Regularity conditions of the reduction ansatz, but also a *neighbourhood* notion of locally optimal solutions on the lower level need to be defined since optimal points are implicitly depending on measures locally. Thus, we will discuss the topology on parameters of the lower level problems, which are defined by measures in IP. In Section 5.4, regularization of "infiniteness" will be discussed by means of adding a term which penalizes complexity caused by infiniteness in the model. In Section 5.5, examples of different parametrizations will be given to reduce the problems of IP. Finally, in Section 5.6, summary of Infinite Kernel Learning and its advantages will be explained an we will give conclusions of this chapter in Section 5.7.

## 5.2 MULTIPLE KERNEL LEARNING

In this section, we will provide an intuition of MKL and problem formulations. Heterogeneous kinds of data in real-world examples have let kernel learning algorithms become generalized by the combination of kernels in a compact form [71]. A weighted combination of kernels allows to define similarity measurement of heterogeneous data. Firstly, we regard a convex combination of kernels $k_\kappa$ $(\kappa = 1, 2, \ldots, K)$:

$$k_\beta(\mathbf{x}_i, \mathbf{x}_j) := \sum_{\kappa=1}^{K} \beta_\kappa k_\kappa(\mathbf{x}_i, \mathbf{x}_j), \tag{5.1}$$

where $\beta_\kappa \geqslant 0$ $(\kappa = 1, 2, \ldots, K)$, $\sum_{\kappa=1}^{K} \beta_k = 1$. Here, the input vectors $\mathbf{x}_i$ $(i = 1, 2, \ldots, l)$ are translated via $K$ mappings $\phi_\kappa : \mathbf{x} \mapsto \phi_\kappa(\mathbf{x}) \in \mathbb{R}^{D_\kappa} (\kappa = 1, 2, \ldots, K)$, from the input space $\mathbb{R}^n$ into $K$ feature spaces $\mathbb{R}^{D_\kappa}$, $D_\kappa$ being the dimension of the $k$-th feature space [71], and $k_\kappa(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \phi_\kappa(\mathbf{x}_i), \phi_\kappa(\mathbf{x}_j) \right\rangle$ $(\kappa = 1, 2, \ldots, K)$.

In [71], the following MKL problem is derived by using the convex combination of kernels (5.1):

$$\begin{aligned} &\textit{Primal Multiple} \quad \min_{\mathbf{w}_\kappa, \xi, b} \ \frac{1}{2}\left(\sum_{\kappa=1}^{K} \|\mathbf{w}_\kappa\|_2\right)^2 + C\sum_{i=1}^{l}\xi_i \\ &\textit{Kernel Problem} \quad \text{subject to } (\mathbf{w}_\kappa \in \mathbb{R}^{D_\kappa} \ (\kappa = (1, 2, \ldots, K)), \ \boldsymbol{\xi} \in \mathbb{R}^l, \ b \in \mathbb{R}) \\ &\qquad\qquad y_i \cdot \left(\sum_{\kappa=1}^{K} \langle \mathbf{w}_\kappa, \phi_\kappa(\mathbf{x}_i)\rangle + b\right) \geqslant 1 - \xi_i \ \ (i = 1, 2, \ldots, l) \ \text{ and } \ \xi_i \geqslant 0, \end{aligned} \tag{5.2}$$

In [8], the dual of the problem (5.2) is expressed with second-order cones as follows:

$$
\begin{aligned}
&\textit{Dual Multiple} \qquad \min_{\gamma,\boldsymbol{\alpha}} \ \frac{1}{2}\gamma^2 - \sum_{i=1}^{l} \alpha_i \quad (\gamma \in \mathbb{R}, \boldsymbol{\alpha} \in \mathbb{R}^l) \\
&\textit{Kernel Problem} \qquad \text{subject to} \quad 0 \leqslant \alpha_i \leqslant C,\ \textstyle\sum_{i=1}^{l} \alpha_i y_i = 0, \\
&\qquad\qquad\qquad\qquad \textstyle\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k_\kappa(\mathbf{x}_i, \mathbf{x}_j) \leqslant \gamma \ \ (\kappa = 1, 2, \ldots, K).
\end{aligned}
\tag{5.3}
$$

A numerical solution for large-scale problems is introduced in [71] by using a *semi-infinite linear programming (LSIP)* [27] (cf. Subsection 2.4) at the place of (5.3) rather than solving an SDP (semidefinitec programming) problem as done in [8]. Indeed, it can be written as maxmin type of problem with respect to kernel coefficients $\beta$ and dual variable $\alpha$ as follows [71]:

$$
\begin{aligned}
&\max_{\boldsymbol{\beta}} \ \min_{\boldsymbol{\alpha}} \ \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\boldsymbol{\alpha}) \quad (\boldsymbol{\alpha} \in \mathbb{R}^l,\ \boldsymbol{\beta} \in \mathbb{R}^K) \\
&\text{subject to} \quad 0 \leqslant \alpha_i \leqslant C,\ \ \boldsymbol{\beta} \geqslant 0 \ \ \text{(componentwise)}, \\
&\qquad\qquad \textstyle\sum_{i=1}^{l} \alpha_i y_i = 0, \text{ and } \sum_{k=1}^{K} \beta_k = 1,
\end{aligned}
\tag{5.4}
$$

where $S_\kappa(\boldsymbol{\alpha}) := \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k_\kappa(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l} \alpha_i$. Let us denote $S(\boldsymbol{\alpha}, \boldsymbol{\beta}) := \sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\boldsymbol{\alpha})$. Problem (5.4) can be represented as an *SIP (semi-infinite programming)* problem by a standard "*epigraph*" argument [71]. Indeed, by maximizing the height variable $\theta$ under the graph of min term, problem (5.4) reduces to the following smooth maximization problem of SILP kind. Indeed, if we had turned from our max to a min term, then we would minimize our height variable over the epigraph which is the area beyond the graph. Now, our *SIP* problem looks as follows:

$$
\begin{aligned}
&\max_{\theta,\boldsymbol{\beta}} \ \ \theta \qquad (\theta \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}^K) \\
&\text{subject to} \ \ \boldsymbol{\beta} \geqslant 0,\ \textstyle\sum_{\kappa=1}^{K} \beta_\kappa = 1 \\
&\qquad\qquad \textstyle\sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\boldsymbol{\alpha}) \geqslant \theta \ \ \forall \boldsymbol{\alpha} \in \mathbb{R}^l \text{ with } 0 \leqslant \boldsymbol{\alpha} \leqslant C\mathbf{1} \text{ and } \ \textstyle\sum_{i=1}^{l} y_i \alpha_i = 0,
\end{aligned}
\tag{5.5}
$$

where, $\mathbf{1} = (1, 1, 1, \ldots, 1)^T \in \mathbb{R}^l$. Let us emphasize that the representation of our maximization problem in this way turned the *feasible set* of (5.4) to the *index set* of inequality constraints of (5.5), constituting the SIP character of the new model representation. By a mere epigraph argument (as for minimization problems), the FP character would have been preserved, i.e., we would have remained in the case of finitely many constraints.

## 5.3 LEARNING WITH INFINITE KERNELS

### 5.3.1 COMBINATION OF INFINITELY MANY KERNELS

Due to the limitation of the selection of multiple kernels from a discrete set of kernels as it is discussed in Section 5.1, we propose a different formulation with the motivation of multiple kernel learning. We introduce infinitely many kernels in the Riemann-Stieltjes integral form [4, 70] which introduces us into an infinite dimensional kernel space. Mathematically, an infinite combination will be represented by the following formula:

$$k_\beta(\mathbf{x}_i, \mathbf{x}_j) := \int_\Omega k(\mathbf{x}_i, \mathbf{x}_j, \omega) d\beta(\omega), \tag{5.6}$$

where $\omega \in \Omega$ is a kernel parameter and $\beta$ is a monotonically increasing function of integral 1, or just a probability measure on $\Omega$. Furthermore, we assume that the kernel function $k(\mathbf{x}_i, \mathbf{x}_j, \omega)$ is a twice continuously differentiable function with respect to $\omega$, i.e., $k(\mathbf{x}_i, \mathbf{x}_j, \cdot) \in C^2$. The infinite combination can be, e.g., a combination of Gaussian kernels with different widths from a set $\Omega$, i.e., $\kappa_\beta(\mathbf{x}_i, \mathbf{x}_j) = \int_\Omega \exp(-\omega \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2) d\beta(\omega)$. It is obvious that the Gaussian kernel is from a family of twice continuously differentiable functions of the variable $\omega$. Hereby, we use the wealth of infinitely many kernels to overcome the limitation of the kernel combination given by finitely pre-chosen kernels. The questions on *which* combination of kernels and on the *structure* of the mixture of kernels could be considered and optimized, and it may, e.g., be answered by *homotopies*. More formally, let us define a function which provides the combination of kernels as follows:

$$H_{\mathbf{x}_i, \mathbf{x}_j}(\omega) := k(\mathbf{x}_i, \mathbf{x}_j, \omega) \ \ (\omega \in [0, 1]). \tag{5.7}$$

In short, we write $H(\omega) := H_{\mathbf{x}_i, \mathbf{x}_j}(\omega)$, and we illustrate such a homotopy by an example.

**Example 5.3.1** *Given* $k(\mathbf{x}_i, \mathbf{x}_j, \omega) = \omega \exp(-w^* \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2) + (1 - \omega)(1 + \mathbf{x}_i^T \mathbf{x}_j)^d$
*with some Gaussian width* $w^*$*, then,*
$H(0) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^d = k^1(\mathbf{x}_i, \mathbf{x}_j)$ *(polynomial kernel),*
$H(1) = \exp(-w^* \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2) = k^2(\mathbf{x}_i, \mathbf{x}_j)$ *(Gaussian kernel).*

Herewith, $\int_\Omega k(\mathbf{x}_i, \mathbf{x}_j, \omega) d\beta(\omega) = k_\beta(\mathbf{x}_i, \mathbf{x}_j)$, where $\Omega = [0, 1]$.

The intuition behind the above Example 5.3.1 is illustrated in Figure 5.1 and Figure 5.2. We can go from "polynomial" to "Gaussian" via a defined homotopy while weighting with infinitesimal coefficients $d\beta(\omega)$.

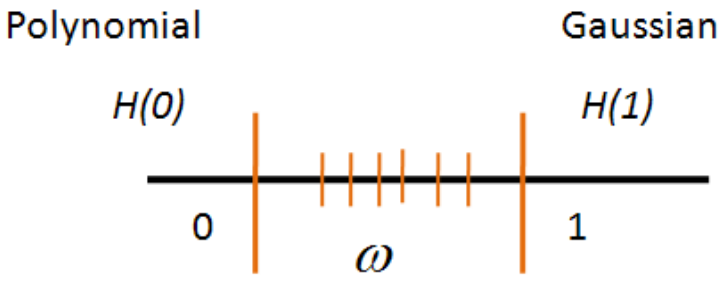


Figure 5.1: On the homotopy between two kernels, example.

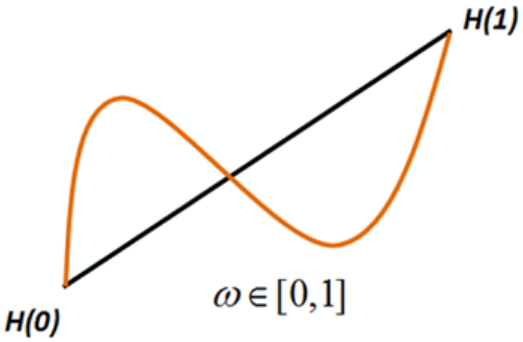


Figure 5.2: Homotopy function which starts at $H(0)$ and combines kernels until $H(1)$ is reached (a symbolic illustration).

To come to infinitely many and infinitesimal coefficients, let us assume that $(\eta_\kappa)_{\kappa\in\mathbb{N}_0}$ is a monotonically increasing sequence in the bounded interval $\Omega := [0, 1]$ tending to 1 as $\kappa \to \infty$ and, say, $\eta_0 = 0$. Then, $\sum_{\kappa=1}^{\infty}(\eta_\kappa - \eta_{\kappa-1}) = 1$. We can refine the summation by a Riemann-Stieltjes integral with any monotonically increasing function $\beta : [0, 1] \longrightarrow \mathbb{R}$, as introduced in Section 2.3, such that $\int_0^1 d\beta(\omega) = 1$. Here, we employ a relation of the kind $eta = \beta(\omega)$. Indeed, we obtain an infinitesimal increment $d\beta(\omega)$ after limit calculus with weights $\beta_\kappa = \beta(\omega_\kappa) - \beta(\omega_{\kappa-1})$, i.e., the incremental weights related to a convex combination $\beta$ of kernels as in the definition (5.1).

Another form of a combination is having just one kernel with its specific (e.g., probabilistic) parameter(s), and to regard it or them as a degree of freedom. More formally, this can be written in the following way:

**Example 5.3.2** *Given a kernel* $k(\mathbf{x}_i, \mathbf{x}_j, \omega) = \exp\left(-\omega \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2\right)$*, the infinite combination of kernels in a Riemann-Stieltjes integral form is*

$$\begin{aligned} k_\beta(\mathbf{x}_i, \mathbf{x}_j) &= \int_\Omega k(\mathbf{x}_i, \mathbf{x}_j, \omega) d\beta(\omega) \\ &= \int_\Omega \exp\left(-\omega \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2\right) d\beta(\omega), \end{aligned}$$

*where* $\Omega = [a, b]$ $(0 \leq a < b)$ *is the set in which* $\omega$ *lies. Here, we allow a different combination of Gaussian widths. We mostly prefer* $\Omega = [0, 1]$ *in later sections and chapters.*

The difference between the Example 5.3.1 and Example 5.3.2 is that, in Example 5.3.1, the Gaussian width is fixed and different types of kernels are combined by a homotopy. But, in Example 5.3.2, the kernel parameter is allowed to be a specific nonlinearly implied variable.

After giving an information about the structure of the combination of infinitely many kernels, we introduce these combinations in the form of Riemann-Stieltjes integrals into the problem (5.5) as follows:

$$\begin{array}{ll} \max\limits_{\theta,\beta} & \theta \quad (\theta \in \mathbb{R},\ \beta : [a, b] \to \mathbb{R},\ \text{monotonically increasing function}) \\ \text{subject to} & \int_\Omega \left(\frac{1}{2} S(\omega, \boldsymbol{\alpha}) - \sum_{i=1}^l \alpha_i\right) d\beta(\omega) \geqslant \theta \ \ \forall \boldsymbol{\alpha} \in \mathbb{R}^l \text{ with } 0 \leqslant \boldsymbol{\alpha} \leqslant C\mathbf{1}, \\ & \sum_{i=1}^l \alpha_i y_i = 0, \ \ \int_\Omega d\beta(\omega) = 1. \end{array} \tag{5.8}$$

Here, $S(\omega, \boldsymbol{\alpha})$ is defined by

$$S(\omega, \boldsymbol{\alpha}) := \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j, \omega). \tag{5.9}$$

Let us introduce $T(\omega, \boldsymbol{\alpha}) := S(\omega, \boldsymbol{\alpha}) - \sum_{i=1}^l \alpha_i$, recall $\Omega = [0, 1]$ and for the index set of inequality constraints we write

$$A := \left\{ \boldsymbol{\alpha} \in \mathbb{R}^l \mid 0 \leqslant \boldsymbol{\alpha} \leqslant C\mathbf{1} \text{ and } \sum_{i=1}^{l} \alpha_i y_i = 0 \right\},$$

where $\mathbf{1} = (1, 1, \ldots, 1)^T$. Herewith, (5.8) turns into the following form with the above abbreviations:

$$\begin{aligned} &\max_{\theta,\beta} \ \theta \quad (\theta \in \mathbb{R},\ \beta: \ \text{a positive measure on } \Omega) \\ &\text{subject to} \quad \theta - \textstyle\int_\Omega T(\omega, \boldsymbol{\alpha}) d\beta(\omega) \leqslant 0 \ \ (\boldsymbol{\alpha} \in A),\ \int_\Omega d\beta(\omega) = 1. \end{aligned} \tag{5.10}$$

Since there are infinitely many inequality constraints and the state variable $\beta$ is from an infinite dimensional space, our problem is one of *infinite programming (IP)* [3]. Now, we get a dual of (5.10) as

$$\begin{aligned} &\min_{\sigma,\rho} \ \sigma \quad (\sigma \in \mathbb{R},\ \rho: \ \text{a positive measure on } A) \\ &\text{subject to} \quad \sigma - \textstyle\int_A T(\omega, \boldsymbol{\alpha}) d\rho(\boldsymbol{\alpha}) \geqslant 0,\ (\omega \in \Omega),\ \int_A d\rho(\boldsymbol{\alpha}) = 1. \end{aligned} \tag{5.11}$$

Because of the conditions, $\int_\Omega d\beta(\omega) = 1$ and $\int_A d\rho(\boldsymbol{\alpha}) = 1$, in latter sections (cf. Section 5.5), we define our positive measures $\beta$(or $\rho$) as probability measures and parametrize these measures with probability density functions. Hence, there is indeed *no* need to write the last equality constraints, $\int_\Omega d\beta(\omega) = 1$ and $\int_A d\rho(\boldsymbol{\alpha}) = 1$, in primal problem, (5.10), and the dual problem, (5.11), respectively, within further definitions or formulations.

**Corollary 5.3.3** *Let us assume that there exist $(\beta, \theta)$ and $(\rho, \sigma)$ which are feasible for their respective problems, and are complementary slack, i.e.,*

$$\sigma = \int_A T(\omega, \boldsymbol{\alpha}) d\rho(\boldsymbol{\alpha}), \quad \theta = \int_A T(\omega, \boldsymbol{\alpha}) d\beta(\omega) \ \text{ and } \ \sigma = \theta.$$

*Then, $\beta$ has measure only where $\sigma = \int_A T(\omega, \boldsymbol{\alpha}) d\rho(\boldsymbol{\alpha})$ and $\rho$ has measure only where $\theta = \int_\Omega T(\omega, \boldsymbol{\alpha}) d\beta(\omega)$ which implies that both solutions are optimal for their respective problems.*[1]

*The interesting theoretical problem with this is to find conditions which ensure that solutions are point masses (i.e., originally, the monotonic $\beta$ is a step function). Under the nondegeneracy assumptions of the Reduction Ansatz since $\Omega$ and $A$ are compact, and with the Heine-Borel theorem, we can assure that there are finitely many local minimizers at the lower levels and, hence, finitely many active points (point masses) [88].*

*From now on, we will look at these conditions for the finitely many local minima. The first conditions is the compactness of the infinite index set. Since the infinite index sets both for the primal problem and the dual problem are compact, this condition is already satisfied. We*

[1] Communication with E.J. Anderson

*need to check the conditions for the nondegeneracy of the critical points and smoothness of the inequality constraint function. Our constraint function g needs to be an element of the $C^2$-family of functions for a given continuous kernel function and the parametrization of the probability measure by a probability density function needs to be done by an element of $C^2$. With these conditions, we can say that there are finitely many local minima of the problems on the lower level sets and among these finite local minima, there are finitely many active ones [88].*

Problem (5.11) is a linear infinite one, i.e., from *ILP (infinite linear programming)*, an SILP (semi-infinite linear programming) one up to the infinite dimensions of $\rho$ space. Because of this insight and problem, and in view of the compactness of the feasible (index) sets at the lower levels, $A$ and $\Omega$, we are interested in the nondegeneracy of the local minima of the lower level problem to get finitely many local minimizers [84]. We note that on the lower levels, $\theta$ and $\sigma$ are just shift terms which do not affect the local solutions there.

For the sake of simplicity and to do real evaluations, from now on, a Gaussian kernel combination will be used in the form given in Example 2. We emphasize that any other kinds of kernels and their combinations could be regarded, too.

### 5.3.2 DUAL PROBLEM

In this section, regularity conditions which are introduced as *reduction ansatz* in Section 2.4.6 will be analyzed for the dual problem on its *lower level*. Let us focus on problem (5.11), employ the language of bilevel programming known from *SIP (semi-infinite programming)*, and introduce the function

$$g((\sigma,\rho),\omega) := \sigma - \int_{\Omega} T(\omega,\boldsymbol{\alpha}) d\rho(\boldsymbol{\alpha}),$$

which is parametric in $(\sigma,\rho)$.

**Lower Level Problem (Dual)**: For a given parameter $(\sigma,\rho)$ we consider

$$\begin{aligned} &\min_{\omega} \quad g((\sigma,\rho),\omega) \\ &\text{subject to} \quad \omega \in \Omega. \end{aligned} \tag{5.12}$$

Indeed, we denote the defining inequality constraint functions of $\Omega$ by $v_1((\sigma,\rho),\omega) := \omega$, $v_2((\sigma,\rho),\omega) := -\omega + 1$. We write $L := \{1,2\}$, $L_0(\omega) := \{\ell \in L \,|\, v_\ell(\omega) = 0\}$ and briefly denote $v_\ell(\omega) := v_\ell((\sigma,\rho),\omega)$ $(\ell = 1,2)$. Consequently, for any critical (and feasible) point $\bar{\omega}$, the Lagrange function reads

$$\mathcal{L}^{\mathcal{D}}(\sigma,\rho;\omega,\gamma) := g((\sigma,\rho),\omega) - \sum_{\ell \in L_0(\bar{\omega})} \gamma_\ell v_\ell(\omega).$$

Here, $\gamma := (\gamma_\ell)_{\ell \in L_0(\bar{\omega})}$. We briefly write $\mathcal{L}^{\mathcal{D}}(\omega,\gamma) := \mathcal{L}^{\mathcal{D}}(\sigma,\rho;\omega,\gamma)$. Since $\Omega$ is compact and $g$ is continuous, for any $(\sigma,\rho)$, local (global) minimizer(s) of (5.12) exists. We analyze the three conditions, of the *nondegeneracy* of a critical point $\bar{\omega}$ of the lower level problem (see [32, 33, 34, 90]) which establish the *reduction ansatz* [33]. For any given $(\sigma,\rho)$ and $\bar{\omega} \in \Omega$ we note:

1. *LICQ*: $\nabla v_\ell(\bar{\omega})$ $(\ell \in L_0(\bar{\omega}))$ is a family with not more than one element since an active $v_\ell$ can either be $\omega$ or $-\omega + 1$ in the interval $[0,1]$ and $\nabla v_1(\omega) = 1$ and $\nabla v_2(\omega) = -1$ do not vanish, respectively. Since LICQ is satisfied, the Lagrange multipliers, referred to in the subsequent points 2. and 3., are *uniquely* determined.

2. *Karush Kuhn-Tucker (KKT) condition with strictly positive Lagrange multipliers*: There exists a multiplier $\bar{\gamma} \in \mathbb{R}^{|L_0(\bar{\omega})|}$ such that $\nabla_\omega \mathcal{L}^{\mathcal{D}}(\bar{\omega},\bar{\gamma}) = 0$ and $\bar{\gamma}_\ell > 0$ $(\ell \in L_0(\bar{\omega}))$. We evaluate this subsequently. If we rewrite $g((\sigma,\rho),\omega)$, it will have the following form:

$$\begin{aligned} g((\sigma,\rho),\omega) &= \sigma - \int_A T(\omega,\boldsymbol{\alpha}) d\rho(\boldsymbol{\alpha}) \\ &= \sigma - \sum_{i,j=1}^{l} k(\mathbf{x}_i,\mathbf{x}_j,\omega) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}) + \int_A \sum_{i=1}^{l} \alpha_i d\rho(\boldsymbol{\alpha}). \end{aligned}$$

Our Lagrange function is parametric in $(\sigma,\rho)$ and, fully, it looks as follows:

$$\mathcal{L}^{\mathcal{D}}(\omega,\gamma) = \sigma - \frac{1}{2} \sum_{i,j=1}^{l} k(\mathbf{x}_i,\mathbf{x}_j,\omega) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}) + \int_A \sum_{i=1}^{l} \alpha_i d\rho(\boldsymbol{\alpha}) - \sum_{\ell \in L_0(\bar{\omega})} \gamma_\ell v_\ell(\omega).$$

Let us find the conditions which satisfy the KKT condition with strictly positive Lagrange multipliers, to ensure the nondegeneracy:

$$\nabla_\omega \mathcal{L}^{\mathcal{D}}(\omega,\gamma) = \nabla Z - \nabla_\omega \Big( \sum_{\ell \in L_0(\bar{\omega})} \gamma_\ell v_\ell(\omega) \Big),$$

where, in this case, gradients are reals, and

$$Z := -\frac{1}{2}\sum_{i,j=1}^{l} k(\mathbf{x}_i, \mathbf{x}_j, \omega) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}),$$

$$\Rightarrow \nabla Z = -\frac{1}{2}\sum_{i,j=1}^{l} \nabla_\omega k(\mathbf{x}_i, \mathbf{x}_j, \omega) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}).$$

To closer illustrate this, as announced above, let us take a Gaussian kernel, i.e., $k(\mathbf{x}_i, \mathbf{x}_j, \omega) = \exp(-\omega \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2)$, and denote

$$\mathcal{I}(\ell \in L_0(\bar{\omega})) := \begin{cases} 1, & \text{if } \ell \in L_0(\bar{\omega}), \\ 0, & \text{if } \ell \notin L_0(\bar{\omega}), \end{cases}$$

at some critical point $\bar{\omega}$. Then, we get

$$\nabla Z = \frac{1}{2}\sum_{i,j=1}^{l} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 \exp(-\omega \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha})$$

and

$$\nabla_\omega \left( \sum_{\ell \in L_0(\bar{\omega})} \gamma_\ell v_\ell(\omega) \right) = \mathcal{I}(1 \in L_0(\bar{\omega})) \cdot \gamma_1 - \mathcal{I}(2 \in L_0(\bar{\omega})) \cdot \gamma_2.$$

Now, we come back to our KKT conditions and evaluate

$$\nabla Z = -\mathcal{I}(1 \in L_0(\bar{\omega})) \cdot \gamma_1 + \mathcal{I}(2 \in L_0(\bar{\omega})) \cdot \gamma_2. \tag{5.13}$$

There are three cases to be discussed to find strictly positive Lagrange multipliers as given below:

**Case 1:** If $v_1(\bar{\omega}) = 0$, i.e., $1 \in L_0(\bar{\omega})$, equation (5.13) will be

$$\frac{1}{2}\sum_{i,j=1}^{l} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 \exp(-\omega \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}) = \gamma_1,$$

$$\gamma_1 > 0 \Leftrightarrow \sum_{i,j=1}^{l} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 \exp(-\omega \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}) > 0. \tag{5.14}$$

**Case 2:** If $v_2(\bar{\omega}) = 0$, i.e., $2 \in L_0(\bar{\omega})$, equation (5.13) will be

$$\frac{1}{2}\sum_{i,j=1}^{l} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 \exp(-\omega \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}) = -\gamma_2,$$

$$\gamma_2 > 0 \Leftrightarrow \sum_{i,j=1}^{l} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 \exp(-\omega \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}) < 0. \tag{5.15}$$

**Case 3**: If $L_0(\bar{\omega}) = \emptyset$, the solution lies in the interior of the feasible region and then the necessary condition for optimality is: $\nabla_\omega g((\sigma, \rho), \bar{\omega}) = 0$. This leads to solve $\bar{\omega}$ from the following:

$$\nabla_\omega \left( \sigma - \sum_{i,j=1}^{l} k(\mathbf{x}_i, \mathbf{x}_j, \bar{\omega}) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}) + \int_A \sum_{i=1}^{l} \alpha_i d\rho(\boldsymbol{\alpha}) \right) = 0.$$

For the following, we introduce

$$\bar{\gamma} := \begin{cases} \gamma_1, & \text{if case 1 holds,} \\ \gamma_2, & \text{if case 2 holds,} \\ 0, & \text{if case 3 holds.} \end{cases} \tag{5.16}$$

3. *Second Order Condition (SOC)*: With our value $\bar{\gamma}$ introduced it has to be fulfilled

$$\eta^{\mathcal{T}} \nabla_\omega^2 \mathcal{L}^D(\sigma, \rho; \bar{\omega}, \bar{\gamma}) \eta > 0, \quad \text{for all } \eta \in \mathcal{T}^{\mathcal{D}}(\bar{\omega}) \setminus \{0\},$$

where $\mathcal{T}^{\mathcal{D}}(\bar{\omega}) := \{\eta \in \mathbb{R} \mid \nabla^T v_\ell(\bar{\omega}) \eta = 0 \ (\ell \in L_0(\bar{\omega}))$.

Let us find the tangent space $\mathcal{T}^{\mathcal{D}}(\bar{\omega})$ for all cases, and evaluate (SOC) with respect to them by the following cases. Here, we write $\mathcal{L}_j^{\mathcal{D}}(\bar{\omega}, \bar{\gamma}), \mathcal{T}_j^{\mathcal{D}}(\bar{\omega})$ $(j = 1, 2, 3)$ according to those cases. (The same later on for the dual case.)

**Case 1**: If $v_1(\bar{\omega}) = 0$, then $\mathcal{T}_1^{\mathcal{D}}(\bar{\omega}) = \{0\}$.

(SOC) $\eta^T \nabla_\omega^2 \mathcal{L}_1^D(\bar{\omega}, \bar{\gamma}) \eta > 0 \ \ \forall \eta \in \mathcal{T}_1^{\mathcal{D}}(\bar{\omega}) \setminus \{0\}$

is fulfilled, since $\forall \eta \in \emptyset$.

**Case 2**: If $v_2(\bar{\omega}) = 0$, then $\mathcal{T}_2^{\mathcal{D}}(\bar{\omega}) = \{0\}$

and, hence,

(SOC) $\eta^{\mathcal{T}} \nabla_\omega^2 \mathcal{L}_2^{\mathcal{D}}(\bar{\omega}, \bar{\gamma}) \eta > 0 \ \forall \eta \in \mathcal{T}_2^{\mathcal{D}}(\bar{\omega}) \setminus \{0\}$

is fulfilled since $\mathcal{T}_2^{\mathcal{D}}(\bar{\omega}) \setminus \{0\} = \emptyset$.

**Case 3**: $L_0(\bar{\omega}) = \emptyset \Rightarrow \mathcal{T}_3^{\mathcal{D}}(\bar{\omega}) = \mathbb{R}$.

Then, the Lagrange function consists only of the objective function $g((\sigma, \rho), \omega)$ which gives

$$\mathcal{L}_3^{\mathcal{D}}(\omega) = \sigma - \sum_{i,j=1}^{l} k(\mathbf{x}_i, \mathbf{x}_j, \omega) y_i y_j \int_A \alpha_i \alpha_j d\rho(\boldsymbol{\alpha}) + \int_A \sum_{i=1}^{l} \alpha_i d\rho(\boldsymbol{\alpha}).$$

$$(SOC) \quad -\tfrac{1}{2}\sum_{i,j=1}^{l}\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_2^4\exp\left(-\bar{\omega}\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_2^2\right)y_iy_j\int_A\alpha_i\alpha_j d\rho(\boldsymbol{\alpha})>0. \tag{5.17}$$

Thus $\bar{\omega}$ is nondegenerate if and only if the sign conditions (on the multipliers) and, in case 3,

$$\sum_{i,j=1}^{l}\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_2^4\exp\left(-\bar{\omega}\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_2^2\right)y_iy_j\int_A\alpha_i\alpha_j d\rho(\boldsymbol{\alpha})<0$$

are fulfilled.

We observe and underline that this essentially depends on the data given. One of the important differences between the dual and primal problem is that the dual problem (5.11) reduces the dimension in the lower level from $l$ to 1. Let us observe that the infinitely many inequality constraints of the dual problem depend on the *one-dimensional* variable $\omega$, whereas in the primal problem they depend on the $l$ dimensional variable $\boldsymbol{\alpha}$. Hence, working with dual problem is analytically more easy and computationally more tractable. However, the interpretation of the classification function for SVM is difficult if we solve (5.11) because of the infinite dimension of the nonlinear mapping $\phi(x)$. This infinity is implicit and parametric in the sense of definition (5.6). For example, even when we have one kernel, in particular, a Gaussian kernel,

$$k(\mathbf{x}_i,\mathbf{x}_j)=\left\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\right\rangle=\exp(-\omega\left\|\mathbf{x}_i-\mathbf{x}_j\right\|_2^2),$$

and $\mathbf{w}=\sum_{i=1}^{l}\alpha_i y_i\phi(\mathbf{x}_i)$, it is difficult to interpret $f(\mathbf{x})=\langle\mathbf{w},\phi(\mathbf{x})\rangle+b$ since we do not know the explicit form of $\phi(\mathbf{x})$ and its dimension is infinite [19]. Because of these reasons, we propose to solve the primal problem to use kernel function implicitly without applying $\phi(\mathbf{x})$ with primal variables $\alpha_i$ in our problem.

### 5.3.3 PRIMAL PROBLEM

In this section, regularity conditions will be analyzed for the lower level of the primal problem given as follows:

$$\begin{aligned}&\max_{\theta,\beta}\ \theta \quad (\theta\in\mathbb{R},\ \beta \text{ is a positive measure on } \Omega)\\ &\text{subject to}\quad \theta-\int_\Omega T(\omega,\boldsymbol{\alpha})d\beta(\omega)\leqslant 0\ \ (\boldsymbol{\alpha}\in A),\\ &\qquad\qquad \int_\Omega d\beta(\omega)=1.\end{aligned} \tag{5.18}$$

The standard form of (5.18) can be easily written by

$$\begin{aligned} &\min_{\theta,\beta}\ (-\theta) \qquad (\theta \in \mathbb{R},\ \beta \text{ is a positive measure on } \Omega) \\ &\text{subject to} \qquad \int_\Omega T(\omega,\boldsymbol{\alpha})d\beta(\omega) - \theta \geqslant 0 \ \ (\boldsymbol{\alpha} \in A), \\ &\qquad\qquad \int_\Omega d\beta(\omega) = 1. \end{aligned} \tag{5.19}$$

Let us recall that the explicit statement of the equality constraint(s) $\int_\Omega d\beta(\omega) = 1$ (and $\int_A d\rho(\boldsymbol{\alpha})$) is not needed, when that $\beta$ (and $\rho$) are *probability* measures. Using the language of bilevel programming of SIP, we introduce the function $g((\theta,\beta),\boldsymbol{\alpha}) := \int_\Omega T(\omega,\boldsymbol{\alpha})d\beta(\omega) - \theta$ which is parametric in $(\theta,\beta)$. We state the

**Lower Level Problem (Primal)**: For a given $(\theta,\beta)$ we consider

$$\begin{aligned} &\min_{\boldsymbol{\alpha}}\ \ g((\theta,\beta),\boldsymbol{\alpha}) \\ &\text{subject to}\ \ \boldsymbol{\alpha} \in A. \end{aligned} \tag{5.20}$$

We write the defining inequality constraint functions of $A$ by $v_r((\theta,\beta),\boldsymbol{\alpha}) := \alpha_i$, $v_s((\theta,\beta),\boldsymbol{\alpha}) := -\alpha_{l-s} + C$, where $r \in \{1,2,\ldots,l\}$ and $s \in \{l+1, l+2,\ldots,2l\}$, and equality constraints by $u((\theta,\beta),\alpha) := \sum_{i=1}^{l} \alpha_i y_i$. Let us briefly denote $v_r((\theta,\beta),\boldsymbol{\alpha}) =: v_r(\boldsymbol{\alpha})$, $v_s((\theta,\beta),\boldsymbol{\alpha}) =: v_s(\boldsymbol{\alpha})$ and $u((\theta,\beta),\boldsymbol{\alpha}) =: u(\boldsymbol{\alpha})$, and $L_0(\bar{\boldsymbol{\alpha}}) := \{\ell \in L \,|\, v_\ell(\bar{\boldsymbol{\alpha}}) = 0\}$, where $L := \{1,2,\ldots,2l\}$. Consequently, for any critical point $\bar{\boldsymbol{\alpha}}$, the Lagrange function reads

$$\mathcal{L}^{\mathcal{P}}(\theta,\beta;\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma}) := g((\theta,\beta),\boldsymbol{\alpha}) - \zeta u(\boldsymbol{\alpha}) - \sum_{\ell \in L_0(\bar{\boldsymbol{\alpha}})} \gamma_l v_\ell(\boldsymbol{\alpha}).$$

Here, as in the dual case, we put $\boldsymbol{\gamma} := (\gamma_\ell)_{\ell \in L_0(\bar{\omega})}$. Let us shortly write $\mathcal{L}^p(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma}) := \mathcal{L}^p(\theta,\beta;\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})$. Since $A$ is compact and $g$ is continuous (since $g \in C^2$), for any local $(\theta,\beta)$, (global) minimizer(s) of (5.20) exists. We analyze the conditions of the nondegeneracy and *reduction ansatz* [32, 33, 34, 88, 90], at any such an $\boldsymbol{\alpha}$. For all $(\theta,\beta)$ and each candidate $\bar{\boldsymbol{\alpha}} \in A$, we evaluate:

1. *LICQ*: We have to check linear independence of $\nabla v_r(\boldsymbol{\alpha})$, $\nabla v_s(\boldsymbol{\alpha})$ and $\nabla u(\boldsymbol{\alpha})$, where the regarded $r \in \{1,2,\ldots,l\}$ and $s \in \{l+1, l+2,\ldots,2l\}$ are active. In other words, variables $\boldsymbol{\alpha} \in \mathbb{R}^l$ can satisfy either $v_r(\boldsymbol{\alpha}) = \alpha_r$ or $v_s(\boldsymbol{\alpha}) = -\alpha_{l-s} + C$. The Jacobian of the (active) inequalities can be calculated simply as follows: $\nabla v_r(\bar{\boldsymbol{\alpha}}) = (0,\ldots,0,1,0,\ldots,0)^T$ and $\nabla v_s(\bar{\boldsymbol{\alpha}}) = (0,\ldots,0,-1,0,\ldots,0)^T$. For simplicity, we introduce $\mathcal{A}(\boldsymbol{\alpha})$ as the vector of

all active constraints, the equality constraint included:

$$\mathcal{A}(\boldsymbol{\alpha}) = \begin{bmatrix} u(\boldsymbol{\alpha}) \\ v_{\ell_1}(\boldsymbol{\alpha}) \\ v_{\ell_2}(\boldsymbol{\alpha}) \\ \vdots \\ v_{\ell_k}(\boldsymbol{\alpha}) \end{bmatrix}, \text{ where } L_0(\bar{\boldsymbol{\alpha}}) = \{\ell_1, \ell_2, \ldots, \ell_k\},\ |L_0(\bar{\boldsymbol{\alpha}})| = k.$$

Then, the Jacobi matrix is a $(k+1) \times$ ł matrix and looks as follows:

$$D\mathcal{A}(\boldsymbol{\alpha}) = \begin{bmatrix} y_1 & y_2 & y_3 & \ldots & \ldots & \ldots & \ldots & y_l \\ 1 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \ldots & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & -1 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots & -1 \end{bmatrix},$$

where $y_i \in \{\pm 1\}$ $(i = 1, 2, \ldots, l)$ and $k = |L_0(\bar{\boldsymbol{\alpha}})|$. On the right-hand side, we took the example of some matrix for illustration. We directly understand that the last $k$ rows of $D\mathcal{A}(\boldsymbol{\alpha})$ constitute a linearly independent family of vectors.

We observe that $\text{rank}(D\mathcal{A}(\boldsymbol{\alpha})) = l$ if $l < k + 1$, which means then that the LICQ condition is violated since rank of $D\mathcal{A}$ is smaller than the number of rows (i.e., constraints involved). This shows linear dependence of the row vectors, i.e., linear dependence of gradients of (active) constraints. In fact, concerning linear independence (LICQ), the first row of $D\mathcal{A}(\boldsymbol{\alpha})$ $(\boldsymbol{\alpha} = \bar{\boldsymbol{\alpha}})$ is the crucial issue, i.e., its possible (non-) representation by the other $k$ rows.

To overcome these pathological situations, let us geometrically analyze this condition in 2 dimensions, i.e., $l = 2$. In Figure 5.3, two *different* examples of nondegeneracy cases are given such that at the origin and at the upper right corner, three active constraints meet and these points (corners) are degenerate because of the linear dependencies. At

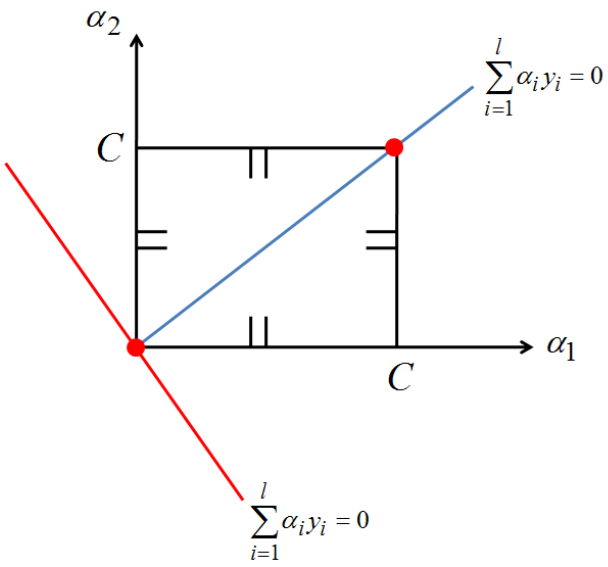


Figure 5.3: Active constraints, the red dots are degenerate points; two examples.

these points, we have three equations in two dimensions.

Let us introduce a sequence $\xi_\nu > 0 \quad (\nu \in \mathbb{N}_0)$ which is monotonically decreasing to zero such that the inequalities $-\xi_\nu \leqslant \sum_{i=1}^{l} \alpha_i y_i \leqslant \xi_\nu$ are requested. Regarding active inequality constraints as equality constraints will lead to lines which do not pass through the origin and cannot produce a corner with threefold activity. This is shown as two examples in Figure 5.3 and Figure 5.4. At the blue points which are feasible points for our perturbed problem, the gradients of all the active constraints are linearly independent. Thus, by decreasing $\xi_\nu$ to zero, for nondegeneracy, LICQ can be enforced by arbitrarily slight perturbations.

For the following points 2. and 3., we may assume now that LICQ is satisfied at the point $\bar{\alpha}$. This implies uniqueness of the Lagrange multipliers.

2. *Kuhn-Tucker condition with strictly positive Lagrange multipliers (for active inequalities):*

   There has to exists a multiplier vector $\bar{\gamma} \in \mathbb{R}^{|L_0(\bar{\alpha})|}$ such that

   $\nabla_\alpha \mathcal{L}^{\mathcal{P}}(\theta, \beta; \boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma}) = 0$ and $\bar{\gamma}_\ell > 0 \quad (\ell \in L_0(\bar{\alpha}))$.

   Let us consider all cases which make Lagrange multiplier strictly positive:

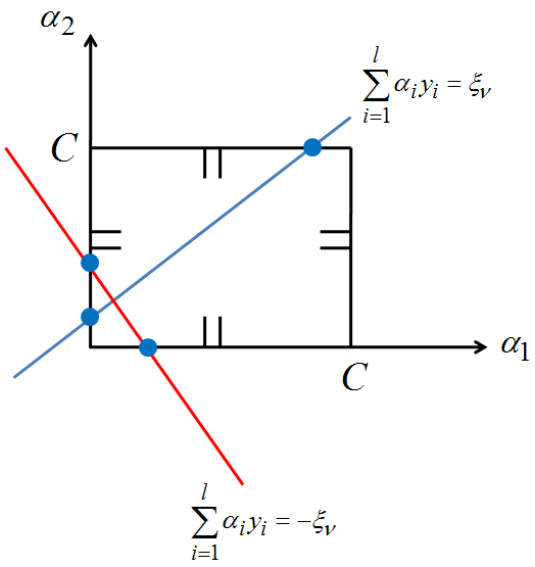


Figure 5.4: Active constraints with regular points in the perturbed problem; two examples.

**Case 1:** $L_0(\bar{\alpha}) \neq \emptyset$. Let us rewrite $g((\theta,\beta),\boldsymbol{\alpha})$, in the following form:

$$\begin{aligned} g((\theta,\beta),\boldsymbol{\alpha}) &= \int_\Omega T(\omega,\boldsymbol{\alpha}) d\beta(\omega) - \theta \\ &= \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\alpha_j y_i y_j \underbrace{\int_\Omega \kappa(\mathbf{x}_i,\mathbf{x}_j,\omega) d\beta(\omega)}_{=:M_{i,j}} - \sum_{i=1}^{l}\alpha_i - \theta \\ &= \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\alpha_j y_i y_j M_{i,j} - \sum_{i=1}^{l}\alpha_i - \theta. \end{aligned}$$

Here, we used a special condition of a probability measure $\beta$: $\int_\Omega d\beta(\omega) = 1$. Note that $M_{i,j}$ is independent of $\boldsymbol{\alpha}$ and $\theta$ but dependent on $\beta$. If we substitute $g((\theta,\beta),\boldsymbol{\alpha})$ into the Lagrange function, we will get the following representation:

$$\begin{aligned} \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma}) &= \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\alpha_j y_i y_j M_{i,j} - \sum_{i=1}^{l}\alpha_i - \theta - \zeta u(\boldsymbol{\alpha}) - \sum_{\ell\in L_0(\bar{\alpha})} \gamma_\ell v_\ell(\boldsymbol{\alpha}) \\ &= \frac{1}{2}\sum_{i=1}^{l} \alpha_i^2 y_i^2 M_{i,i} - \frac{1}{2}\sum_{\substack{i,j=1 \\ i\neq j}}^{l} \alpha_i\alpha_j y_i y_j M_{i,j} - \sum_{i=1}^{l}\alpha_i - \theta - \zeta u(\boldsymbol{\alpha}) \\ &- \sum_{\ell\in L_0(\bar{\alpha})} \gamma_\ell v_\ell(\boldsymbol{\alpha}). \end{aligned} \tag{5.21}$$

In the second line, we used the assumption that our kernel function is a Gaussian kernel, which is $\kappa(\mathbf{x}_i,\mathbf{x}_j,\omega) = \exp(-\omega\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2)$. In fact, for $i = j$, we get $\kappa(\mathbf{x}_i,\mathbf{x}_i,\omega) = 1$.

To find KKT points $(\bar{\boldsymbol{\alpha}}, \bar{\zeta}, \bar{\boldsymbol{\gamma}})$, we need to solve $\nabla_{\boldsymbol{\alpha}} \mathcal{L}^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma}) = \mathbf{0}$, which is a system of linear equations in $(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma})$ with

$$\nabla_{\boldsymbol{\alpha}} \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma}) = \left[ \begin{array}{cccc} \frac{\partial \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma})}{\partial \alpha_1}, & \frac{\partial \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma})}{\partial \alpha_2}, & \ldots, & \frac{\partial \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma})}{\partial \alpha_l} \end{array} \right]^T,$$

where for all $i = 1, 2, \ldots, l$,

$$\frac{\partial \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma})}{\partial \alpha_i} = \alpha_i y_i^2 M_{i,i} - \frac{1}{2} \sum_{\substack{i,j=1 \\ j \neq i}}^{l} \alpha_j y_i y_j M_{i,j} - 1 - \zeta \frac{\partial u(\boldsymbol{\alpha})}{\partial \alpha_i} - \sum_{\ell \in L_0(\bar{\boldsymbol{\alpha}})} \gamma_l \frac{\partial v_\ell(\boldsymbol{\alpha})}{\partial \alpha_i}.$$

Let us for the sake of simplicity assume that $L_0(\bar{\boldsymbol{\alpha}}) = \{1, 2, \ldots, k\}$ $(\boldsymbol{\alpha} = \bar{\boldsymbol{\alpha}})$, renumbering the active inequalities otherwise. Then, from $\nabla_{\boldsymbol{\alpha}} \mathcal{L}_1^p(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma}) = \mathbf{0}$ we get the following equations:

$$\begin{array}{rcl} \alpha_1 y_1^2 - \frac{1}{2} \sum_{\substack{i,j=1 \\ j \neq 1}}^{l} \alpha_j y_i y_j M_{i,j} - 1 & = & \zeta \frac{\partial u(\boldsymbol{\alpha})}{\partial \alpha_1} + \sum_{\ell=1}^{k} \gamma_\ell \frac{\partial v_\ell(\boldsymbol{\alpha})}{\partial \alpha_1}, \\ \alpha_2 y_2^2 - \frac{1}{2} \sum_{\substack{i,j=1 \\ j \neq 2}}^{l} \alpha_j y_i y_j M_{i,j} - 1 & = & \zeta \frac{\partial u(\boldsymbol{\alpha})}{\partial \alpha_2} + \sum_{\ell=1}^{k} \gamma_\ell \frac{\partial v_\ell(\boldsymbol{\alpha})}{\partial \alpha_2}, \\ & \vdots & \\ \alpha_l y_l^2 - \frac{1}{2} \sum_{\substack{i,j=1 \\ j \neq l}}^{l} \alpha_j y_i y_j M_{i,j} - 1 & = & \zeta \frac{\partial u(\boldsymbol{\alpha})}{\partial \alpha_l} + \sum_{\ell=1}^{k} \gamma_\ell \frac{\partial v_\ell(\boldsymbol{\alpha})}{\partial \alpha_\ell}. \end{array} \tag{5.22}$$

The systems of equations (5.22) can be written in the matrix-vector multiplication form as follows:

$$\boldsymbol{D} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_l \end{bmatrix} - \boldsymbol{B} \begin{bmatrix} \zeta \\ \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_k \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{5.23}$$

where

$$\boldsymbol{B} := \begin{bmatrix} \frac{\partial u}{\partial \alpha_1} & \frac{\partial v_1}{\partial \alpha_1} & \cdots & \frac{\partial v_k}{\partial \alpha_1} \\ \frac{\partial u}{\partial \alpha_2} & \frac{\partial v_1}{\partial \alpha_2} & \cdots & \frac{\partial v_k}{\partial \alpha_2} \\ \vdots & & \vdots & \vdots \\ \frac{\partial u}{\partial \alpha_l} & \frac{\partial v_1}{\partial \alpha_l} & \cdots & \frac{\partial v_k}{\partial \alpha_l} \end{bmatrix}_{|\boldsymbol{\alpha}}, \tag{5.24}$$

and

$$D := \begin{bmatrix} y_1^2 & -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq 1}} y_i y_j M_{i,j} & -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq 1}} y_i y_j M_{i,j} & \dots & -\frac{1}{2}\sum\limits_{\substack{j=1\\ j\neq 1}} y_i y_j M_{i,j} \\ -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq 2}} y_i y_j M_{i,j} & y_2^2 & -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq 2}} y_i y_j M_{i,j} & \dots & -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq 2}} y_i y_j M_{i,j} \\ -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq 3}} y_i y_j M_{i,j} & -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq 3}} y_i y_j M_{i,j} & y_3^2 & \dots & -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq 3}} y_i y_j M_{i,j} \\ \vdots & & & \ddots & \\ -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq l}} y_i y_j M_{i,j} & -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq l}} y_i y_j M_{i,j} & -\frac{1}{2}\sum\limits_{\substack{i,j=1\\ j\neq l}} y_i y_j M_{i,j} & \dots & y_l^2 \end{bmatrix}. \tag{5.25}$$

We can write (5.23) in a more closed form as

$$[\boldsymbol{D} \mid \boldsymbol{B}] \left( \begin{array}{c} \boldsymbol{\alpha} \\ \hline \zeta \\ \boldsymbol{\gamma} \end{array} \right) = \mathbf{1}. \tag{5.26}$$

If we solve (5.26) restricted to $\gamma_\ell > 0$ $(\ell = 1, 2, \dots, k)$, we can specify rank and conditioning properties for $\boldsymbol{\alpha} = \bar{\boldsymbol{\alpha}}$ being a candidate of a locally optimal solution. If the strict positivity of the corresponding Lagrange multiplier is satisfied, if second-order conditions is fulfilled and we guaranteed LICQ (employing our perturbational argument, if needed), such that we have nondegeneracy of the feasible point $\boldsymbol{\alpha} = \bar{\boldsymbol{\alpha}}$, then this point $\bar{\boldsymbol{\alpha}}$ is a candidate of a locally optimal solution on the lower level, provided a feasible pair $(\sigma, \rho)$ of the upper level problem.

We underline that the previous reflection holds true in the sense of two restrictions:

(i) The point $\boldsymbol{\alpha} = \bar{\boldsymbol{\alpha}}$ has to be ***feasible***, i.e., $\bar{\boldsymbol{\alpha}} \in A$, and

(ii) ***all*** the ***combinatorial*** possibilities of $L_0(\bar{\boldsymbol{\alpha}})$ have to be taken into account.

Here, we assumed $L_0(\bar{\boldsymbol{\alpha}}) = \{1, 2, \dots, k\}$ for the active index set without loss of generality. Note that the local optimality is provided by a use of Heine-Borel theorem. Since our infinite index set is compact, there exist finite indices which corresponds to active inequalities.

**Case 2:** $L_0(\bar{\boldsymbol{\alpha}}) = \emptyset$, i.e., the equality constraint is the only active constraint. Our

Lagrangian will take the following form:

$$\begin{aligned}\mathcal{L}_2^{\mathcal{P}}(\boldsymbol{\alpha},\zeta) &= \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j M_{i,j} - \sum_{i=1}^{l}\alpha_i - \theta - \zeta u(\boldsymbol{\alpha}) \\ &= \frac{1}{2}\sum_{i=1}^{l}\alpha_i^2 y_i^2 M_{i,i} - \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{l}\alpha_i\alpha_j y_i y_j M_{i,j} - \sum_{i=1}^{l}\alpha_i - \theta - \zeta u(\boldsymbol{\alpha}).\end{aligned}$$

Here, to illustrate our Case 2, we assume that we have a Gaussian kernel, as in Case 1. Let us find $(\boldsymbol{\alpha},\zeta)$ which satisfies $\nabla_{\boldsymbol{\alpha}}\mathcal{L}_2^{\mathcal{P}}(\boldsymbol{\alpha},\zeta) = 0$, i.e.,

$$\nabla_{\boldsymbol{\alpha}}\mathcal{L}_2^{\mathcal{P}}(\boldsymbol{\alpha},\zeta) = \left[\frac{\partial\mathcal{L}_2^{\mathcal{P}}(\boldsymbol{\alpha},\zeta)}{\partial\alpha_1}, \frac{\partial\mathcal{L}_2^{\mathcal{P}}(\boldsymbol{\alpha},\zeta)}{\partial\alpha_2}, \ldots, \frac{\partial\mathcal{L}_2^{\mathcal{P}}(\boldsymbol{\alpha},\zeta)}{\partial\alpha_l}\right]^T = 0, \tag{5.27}$$

where $\frac{\partial\mathcal{L}_2^{\mathcal{P}}(\boldsymbol{\alpha},\zeta)}{\partial\alpha_i} = \alpha_i y_i^2 M_{i,i} - \frac{1}{2}\sum_{i\neq j}\alpha_j y_i y_j M_{i,j} - 1 - \zeta\frac{\partial u(\boldsymbol{\alpha})}{\partial\alpha_i}$ and $M_{i,i} = 1$ as in previous case.

If we expand (5.27), we get the following system of equations:

$$\begin{aligned}\alpha_1 y_1^2 - \frac{1}{2}\sum_{\substack{i,j=1\\ j\neq 1}}^{l}\alpha_j y_i y_j M_{i,j} - 1 &= \zeta\frac{\partial u(\boldsymbol{\alpha})}{\partial\alpha_1}, \\ \alpha_2 y_2^2 - \frac{1}{2}\sum_{\substack{i,j=1\\ j\neq 2}}^{l}\alpha_j y_i y_j M_{i,j} - 1 &= \zeta\frac{\partial u(\boldsymbol{\alpha})}{\partial\alpha_2}, \\ &\vdots \\ \alpha_l y_l^2 - \frac{1}{2}\sum_{\substack{i,j=1\\ j\neq l}}^{l}\alpha_j y_i y_j M_{i,j} - 1 &= \zeta\frac{\partial u(\boldsymbol{\alpha})}{\partial\alpha_l}.\end{aligned} \tag{5.28}$$

The above system of equations (5.28) can be written in matrix-vector multiplication form as follows:

$$\begin{bmatrix}\frac{\partial u(\boldsymbol{\alpha})}{\partial\alpha_1}\\ \frac{\partial u(\boldsymbol{\alpha})}{\partial\alpha_2}\\ \vdots\\ \frac{\partial u(\boldsymbol{\alpha})}{\partial\alpha_l}\end{bmatrix}\zeta = \boldsymbol{D}\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_l\end{bmatrix} - \begin{bmatrix}1\\ 1\\ \vdots\\ 1\end{bmatrix}, \tag{5.29}$$

where $\frac{\partial u(\boldsymbol{\alpha})}{\partial \alpha_i} = y_i$ $(i = 1, 2, \ldots, l)$. Hence, (5.29) becomes the following linear system:

$$\boldsymbol{D}\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_l \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_l \end{bmatrix}\zeta = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{5.30}$$

where $\boldsymbol{D}$ is same with (5.25). We can rewrite (5.30) in the form of (5.26) where now, however, all terms related with $v$ and $\boldsymbol{\gamma}$ disappear. If we solve (5.30), we find the conditions for $\alpha = \bar{\alpha}$ to be the optimal solution. Let us recall that the feasibility condition (i) pointed out in Case 1 has to be fulfilled also here.

3. *Second Order Condition (SOC)*: With our value $\bar{\boldsymbol{\gamma}}$ introduced, it has to be fulfilled:

$$\boldsymbol{\eta}^T \nabla^2_{\boldsymbol{\alpha}} \mathcal{L}^{\mathcal{P}}(\bar{\boldsymbol{\alpha}}, \bar{\zeta}, \bar{\boldsymbol{\gamma}})\boldsymbol{\eta} > 0 \ \text{ for all } \ \boldsymbol{\eta} \in \mathcal{T}^{\mathcal{P}}(\bar{\boldsymbol{\alpha}}) \setminus \{0\},$$

where $\mathcal{T}^{\mathcal{P}}(\bar{\boldsymbol{\alpha}}) := \{\boldsymbol{\eta} \in \mathbb{R}^l \mid \nabla^T u(\bar{\boldsymbol{\alpha}})\boldsymbol{\eta} = 0,\ \nabla^T v_l(\bar{\boldsymbol{\alpha}})\boldsymbol{\eta} = \mathbf{0}\ \ (l \in L_0(\bar{\boldsymbol{\alpha}}))\}$.

Now, let us more explicitly find tangent space and conditions for ***(SOC)*** to be satisfied for all cases:

**Case 1**: If $L_0(\bar{\boldsymbol{\alpha}}) \neq 0$, then the tangent space of the form

$\mathcal{T}_1^{\mathcal{P}}(\bar{\boldsymbol{\alpha}}) = \{\boldsymbol{\eta} \in \mathbb{R}^l \,\big|\, \boldsymbol{D}\mathcal{A}(\bar{\boldsymbol{\alpha}})\boldsymbol{\eta} = 0\}$, with the condition (written a bit like an example again)

$$\begin{bmatrix} y_1 & y_2 & y_3 & \ldots & \ldots & \ldots & \ldots & y_l \\ 1 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \ldots & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & -1 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots & -1 \end{bmatrix}\begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \eta_l \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{5.31}$$

Here, as we know, $\nabla v_\ell(\boldsymbol{\alpha}) = (0, \ldots, \pm 1, 0, \ldots, 0)^T$ $(l \in L_0(\bar{\boldsymbol{\alpha}}))$ and $k = |L_0(\bar{\boldsymbol{\alpha}})|$. Equation (5.31) yields the following condition:

$$\eta_r = 0 \quad \forall\, r \in L_0(\bar{\boldsymbol{\alpha}}) \cap \{1, 2, \ldots, l\}, \tag{5.32}$$

$$\eta_s = 0 \quad \forall\, l+s \in L_0(\bar{\boldsymbol{\alpha}}) \cap \{l+1, l+2, \ldots, 2l\}, \tag{5.33}$$

$$\sum_{i=1}^{l} \eta_i y_i = 0. \tag{5.34}$$

From (5.32)-(5.34), it follows that $\sum_{\substack{i=1 \\ i,\ l+i \notin L_0(\bar{\alpha})}}^{l} y_i \eta_i = 0$.

Let us note the form of $\nabla^2_{\boldsymbol{\alpha}} \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma})$ explicitly:

$$\nabla^2_{\boldsymbol{\alpha}} \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma}) = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})}{\partial^2 \alpha_1} & \frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})}{\partial \alpha_1 \partial \alpha_2} & \cdots & \frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})}{\partial \alpha_1 \partial \alpha_l} \\ \frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})}{\partial \alpha_2 \partial \alpha_1} & \frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})}{\partial^2 \alpha_2} & \cdots & \frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})}{\partial \alpha_2 \partial \alpha_l} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})}{\partial \alpha_l \partial \alpha_1} & \frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})}{\partial \alpha_l \partial \alpha_2} & \cdots & \frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha},\zeta,\boldsymbol{\gamma})}{\partial^2 \alpha_l} \end{bmatrix},$$

with

$$\frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma})}{\partial^2 \alpha_i} = y_i^2 = 1 > 0, \tag{5.35}$$

and

$$\frac{\partial^2 \mathcal{L}_1^{\mathcal{P}}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\gamma})}{\partial \alpha_i \alpha_j} = -\frac{1}{2} y_i y_j M_{i,j} \ \ (i \neq j). \tag{5.36}$$

Furthermore, let us state:

**Theorem 5.3.4** *[15]. A symmetric $(n \times n)$ matrix $\boldsymbol{M}$ is positive definite (or positive semi-definite) if and only if any one of the following conditions holds:*

*(a) Every eigenvalue of $\boldsymbol{M}$ is positive (zero or positive, respectively).*

*(b) All the leading principal minors of $\boldsymbol{M}$ are positive definite (all the principal minors of M are positive semi-definite, respectively).*

*(c) There exists an $n \times n$ nonsingular matrix $\boldsymbol{N}$ (an $n \times n$ singular matrix $\boldsymbol{N}$ or an $(m \times n)$-matrix $\boldsymbol{N}$ with $m < n$, respectively) such that $\boldsymbol{M} = \boldsymbol{N}^T \boldsymbol{N}$.*

**Corollary 5.3.5** *[83]. If $\boldsymbol{A} = (a_{i,j})_{i,j=1,2,\ldots,n}$ is symmetric $(n \times n)$ strictly diagonally dominant matrix*[2] *with positive real diagonal entries, then $\boldsymbol{A}$ is positive definite.*

[2] Strictly diagonally dominant matrix, $\boldsymbol{A}$, has diagonal entries strictly bigger than the off diagonal ones, i.e., $a_{i,i} > a_{i,j}$ $(i, j = 1, 2, \ldots, n)$

In particular, if any of the corresponding conditions given in Theorem 5.3.4 and Corollary 5.3.5 is satisfied accordingly, then the Hessian is positive definite over the tangent space: $\boldsymbol{\eta}^T \nabla^2_{\alpha} \mathcal{L}_1^{\mathcal{P}}(\bar{\boldsymbol{\alpha}}, \bar{\zeta}, \bar{\boldsymbol{\gamma}})\boldsymbol{\eta} > 0$ for all $\boldsymbol{\eta} \in \mathcal{T}_1^{\mathcal{P}}(\bar{\boldsymbol{\alpha}}) \setminus \{\mathbf{0}\}$, so that *(SOC)* is satisfied. More explicitly, the diagonal entries given by (5.35) need only to be diagonally dominant, since by (5.35) the diagonal entries are strictly positive.

**Case 2:** $L_0(\bar{\boldsymbol{\alpha}}) = \emptyset$. Then,

$$\begin{aligned} \mathcal{T}_2^{\mathcal{P}}(\boldsymbol{\alpha}) &= \{\boldsymbol{\eta} \in \mathbb{R}^l \,\big|\, \nabla^T u(\bar{\boldsymbol{\alpha}})\boldsymbol{\eta} = 0\} \\ &= \left\{\boldsymbol{\eta} \in \mathbb{R}^l \,\middle|\, \sum_{i=1}^{l} y_i \eta_i = 0\right\}. \end{aligned}$$

The Hessian is the same as in Case 1, with the same entries as given in (5.35) and (5.36), because of the linearity of the constraints. Furthermore, there are the same (SOC) conditions, referring to $\mathcal{T}_2^{\mathcal{P}}(\bar{\boldsymbol{\alpha}})$ now.

Under these assumptions, the following theorem assures the optimal (local or global) solution of the primal problem in a neighbourhood of the regarded optimal (local or global) solution on the lower level. A corresponding theorem holds for the dual problem which will not be given repeatedly. Indeed, our extensions of the results given in [32, 33] hold true where, now, the parameter space is infinite dimensional.

**Theorem 5.3.6** *Let at some feasible point $(\bar{\theta}, \bar{\beta})$ of (5.10) the condition* reduction ansatz *be satisfied, and the inequality constraint function of the (upper level) problem (5.19) be of class $C^2$ with respect to $\boldsymbol{\alpha}$. Then, the following statement holds true:*
*(a) The set of local minimizers of lower level problem at feasible point $(\bar{\theta}, \bar{\beta})$ of (5.10) is finite and, hence, active index set at $(\bar{\theta}, \bar{\beta})$ is finite, in symbols: $A_0(\bar{\theta}, \bar{\beta}) = \{\bar{\boldsymbol{\alpha}}_{\mathbf{1}}, \bar{\boldsymbol{\alpha}}_{\mathbf{2}}, \ldots, \bar{\boldsymbol{\alpha}}_{\boldsymbol{\chi}}\}$ in the role of $Y_0(\cdot)$ in Subsection 2.4.6, and there exist unique Lagrange multipliers $\bar{\zeta}$ and unique Lagrange multiplier vectors $\bar{\boldsymbol{\gamma}}_{\boldsymbol{j}}$ $(j = 1, 2, \ldots, \chi)$, neighbourhoods $U_{(\bar{\theta},\bar{\beta})}$ of $(\bar{\theta}, \bar{\beta})$ and $V_{\bar{\alpha}_j}$ of $\bar{\boldsymbol{\alpha}}_{\boldsymbol{j}}$, and continuous mappings*

$$\alpha_j : U_{(\bar{\theta},\bar{\beta})} \to V_{\bar{\alpha}_j}, \text{ with } \alpha_j(\bar{\theta}, \bar{\beta}) = \bar{\boldsymbol{\alpha}}_{\boldsymbol{j}},$$

$$\zeta : U_{(\bar{\theta},\bar{\beta})} \to \mathbb{R} \text{ with } \zeta(\bar{\theta}, \bar{\beta}) = \bar{\zeta} \text{ and } \gamma_j : U_{(\bar{\theta},\bar{\beta})} \to \mathbb{R}^{|L_0(\bar{\boldsymbol{\alpha}})|}, \text{ with } \gamma_j(\bar{\theta}, \bar{\beta}) = \bar{\boldsymbol{\gamma}}_{\boldsymbol{j}}$$

*($j = 1, 2, \ldots, \chi$) such that for every $(\theta, \beta) \in U_{(\bar{\theta},\bar{\beta})}$ the value $\alpha_j(\theta, \beta)$ is the unique local minimizer of (5.20) in $V_{\bar{\alpha}_j}$, with corresponding unique Lagrange multipliers $\zeta(\theta, \beta)$ and unique Lagrange multiplier vectors $\boldsymbol{\gamma}_j(\theta, \beta)$ ($j = 1, 2, \ldots, \chi$).*
*(b) With the functions introduced in (a), the following finite reduction holds:*

*$(\theta, \beta) \in U_{(\bar{\theta},\bar{\beta})} \cap \mathcal{M}$, where $\mathcal{M}$ is the feasible set of the upper level problem (5.10), is a local solution of (5.10), if and only if $(\theta, \beta)$ is a local solution of the so-called* reduced problem

$$P_{red}(\theta, \beta): \quad \min_{(\theta,\beta)\in U_{(\bar{\theta},\bar{\beta})}} (-\theta) \quad \text{such that } G_j(\theta, \beta) := g((\theta, \beta), \boldsymbol{\alpha}_j(\theta, \beta)) \geq 0 \quad (j = 1, 2, \ldots, \chi). \tag{5.37}$$

We emphasize that this result is an extension to Theorem 2.4.20 into the presence of state variables in an infinite dimensional space. Moreover, we note that in our cases of kernels and of probability measures with density functions, the $C^2$-property of the inequality constraints are fulfilled.

**Remark 4** *An analogous theorem holds for the **dual** problem (5.11) under the reduction ansatz and with respect to the dual variables. We underline that by this theorem the reduced problem has (locally) finitely many constraints. Then, our task becomes a finitely constrained optimization problem, locally around an optimal solution. This insight is based on Implicit Function Theorem (IFT) and the neighbourhood notion defined by, e.g., the Prokhorov distance, introduced below.*

We note that in our application of the Implicit Function Theorem, the variable from an infinite dimensional space, $\beta$, is playing the role of a parameter, i.e., it is not involved into the differentiation which is needed in the assumptions of the reduction ansatz. For very general versions of Inverse and, hence, Implicit Function Theorem, we refer, e.g., to [29].

Let us start with some definitions which are necessary to define *neighbourhood* in terms of measures, including the *probability measures* of our study.

**Definition 5.3.7** *[56]. Let $(E, T)$ be a Hausdorff topological space and let $\Sigma$ be a $\sigma$-algebra on $E$ that contains the topology $T$ (so that every open set is a measurable set, and $\Sigma$ is at least as fine as the Borel $\sigma$-algebra on $E$). A measure $\mu$ defined on $\Sigma$ is called **locally finite** if, for every point $p$ of the space $E$, there is an open neighbourhood $N_p$ of $p$ such that the measure $\mu$ of $N_p$ is finite.*

*In more condensed notation, $\mu$ is locally finite if and only if*

$$\forall p \in E, \exists N_p \in T \text{ such that } p \in N_p \text{ and } \left|\mu(N_p)\right| < +\infty.$$

*With the same assumptions, a measure $\mu$ on the measurable space $(E, \Sigma)$ is called* ***inner regular*** *if for every set $A \in \Sigma$ it holds*

$$\mu(A) := \sup\{\mu(K) \mid K \subseteq A \ \ compact\}.$$

This property is sometimes referred to in words as *approximation from within by compact sets.*

In this study, we restrict ourselves to probability measures, which constitute our subspace of positive measures. It is clear that the probability measures satisfy the constraints $\int_\Omega d\beta(\omega) = 1$ and $\int_A d\rho(\alpha) = 1$. Hence, it is not necessary to write these constraints in our problem definitions. We note that the probability measures are inner regular and locally finite which satisfy the definition of Radon measure given by Definition 5.3.8. Next, we define a *Radon measure* and the distance metric needed for neighbourhoods in Theorem 5.3.6:

**Definition 5.3.8** *[43]. Let $(E, d)$ be the metric space. A* ***Radon measure*** *is a measure on the $\sigma$-algebra of Borel sets of E that is locally finite and inner regular.*

We denote the set of Radon measures on $E$ by $\mathcal{H}(E)$. In our problems, we look at the subspaces of all the probability measures $\rho$ for the dual problem (5.11), and $\beta$ for the primal problem (5.18).

**Definition 5.3.9** *[43]. Let $f_i : E \to \mathbb{R}$ $(i = 1, 2, \ldots, q)$ be continuous bounded functions and a metric space $(E, d)$), i.e., $f_i \in (\mathcal{H}(E))'$, where $(\mathcal{H}(E))'$ is the dual space of $\mathcal{H}(E)$. A* ***base of neighbourhood*** *of some Radon measure $\mu_0 \in \mathcal{H}(E)$ can be defined as*

$$\left\{\mu \in \mathcal{H}(E) \,\middle|\, \left|\int_E f_i d\mu - \int_E f_i d\mu_0\right| < \epsilon \quad (i = 1, 2, \ldots, q)\right\}.$$

In our problems, the elements in the dual space are probability density functions (pdfs). Now, to represent our neighbourhood notion by a metric, let us define *Prokhorov distance*:

**Definition 5.3.10** *[43]. Let $(E, d)$ be a metric space, where $d_0$ is a* ***Prokhorov distance*** *between any $\mu_2,\ \mu_1 \in \mathcal{H}(E)$ is defined by*

$$d_0(\mu_1, \mu_2) := \inf \{\epsilon \geqslant 0 \,|\mu_2(A) \leqslant \mu_1(A_\epsilon) + \epsilon \text{ and } \mu_1(A) \leqslant \mu_2(A_\epsilon) + \epsilon \ \ (A \subseteq E, \text{ closed})\} \tag{5.38}$$

*with $A_\epsilon := \{x \in E \,|d(x, A) < \epsilon\}$. Then, the open $\delta$-neighbourhood of $\mu_1$ is defined by $B_\delta(\mu_1) := \{\mu_2 \in \mathcal{H}(E) \,|\, d_0(\mu_1, \mu_2) < \delta\}$.*

**Remark 5** *Definition 5.3.10 allows to define a neighbourhood of $(\sigma, \rho)$ (or $(\theta, \beta)$) in an appropriate topological sense. By Theorem 5.3.6 and Definition 5.3.10 we specify the meaning of reduction ansatz and of a local optimal solution, namely, in one of these neighbourhoods.*

**Remark 6** *In the above definitions and theorems, the functions $f$ are in same role of* test functions*, and the mapping $(f, \mu) \mapsto \int_E f d\mu$ can be regarded as a* dual pairing*.*

*A counterpart to the Prokhorov metric $d_0$ is the* ***bounded Lipschitz metric*** *$\hbar$ (see, e.g., [24]). It is defined in terms of functions instead of sets (as the Prokhorov distance) and, hence, fits more consistently to the definition of the weak topology.*[3] *The bounded Lipschitz metric is of the form*

$$\hbar_F(\mu_1, \mu_2) := \sup_{f \in F} |\int_E f(x)(\mu_1 - \mu_2)(dx)|,$$

*where $F$ is a class of real-valued measurable functions defined on the metric space $E$. If $F$ corresponds to the unit ball in the Banach space of bounded and Lipschitz continuous functions, one arrives at $\hbar$. It holds $c\hbar(\mu_1, \mu_2) \leq d_0(\mu_1, \mu_2) \leq C\hbar(\mu_1, \mu_2)^{1/2}$ for some constants $c, C > 0$ and all probability measures $\mu_1, \mu_2$ on $E$. Hence, both metrics metricize the* ***weak topology****.*[4]

**Note 5.3.11** ***(Complexity)*** *Since our IP problem variables lie in an infinite dimensional space, minimizing (or maximizing) our objective function with respect to this variable from infinite dimension can cause a high model complexity. Thus, in the following section, we introduce a regularization term which flatten the model having not high energy functionals.*

[3] Weak topology $X$ is defined in its continuous dual space $X^*$ This dual space consists of all linear functions from $X$ into $\mathbb{R}$ or $\mathbb{C}$ which are continuous with respect to the strong topology.

[4] Discussion with Prof. Werner Römisch

## 5.4 REGULARIZATION OF INFINITE PROGRAMMING MODEL WITH RESPECT TO KERNEL COEFFICIENTS

In the previous section, our classification problem is modelled and analyzed with infinitely many kernels by infinite programming. "Infinity" of kernels may cause ill-posed problem, which is called *overfitting* in regression problems. Here, we consider classification problems which need to be *regularized* by penalizing overfitting caused by infinity in the model [30, 82]. Regularization is performed by adding penalization term to the objective function to reduce the complexity of the problem. This could be the case if any positive multiple of a kernel is also a kernel [5]. Argyriou et al. (2006) introduced a regularization term to prevent from overfitting of data by the objective function [5]:

$$Q(f) := \sum_{j=1}^{l} q(y_j, f(\mathbf{x}_j)) + \lambda \|f\|_k^2 , \tag{5.39}$$

where $q(\cdot,\cdot)$ is a loss function and $\|.\|_k$ is the norm induced by *reproducing Kernel Hilbert space*. Here, $f$ is represented by a combination of kernels as $f = \sum_{j=1}^{l} c_j k(\mathbf{x}_j, \cdot)$, which is known as *Representer Theorem* [66], and the parameters $c_j \geq 0$ $(j = 1, 2, \ldots, l)$ become optimized [5].

Unlikely to problem (5.39), in our infinite kernel representation with Riemann-Stieltjes integrals or positively defined measures, we need to find a penalization function in terms of measures $\beta(\omega)$ (or $\rho(\boldsymbol{\alpha})$) since they represent our continuous convex coefficients for infinite kernel combinations.

**Closer Explanation 5.4.1** *Since we have probability measures as state variables, we can hardly use the theory of regularization, e.g., Tikhonov regularization, directly. Instead, it is our proposal to measure the complexity of our model by "scanning" the integral terms via a running upper integration boundary, and to take partial derivatives of first and second order to record infinitesimal changes of those orders. By this and penalizing these kinds of change rates, we are looking for a "flat" model or, in particular, a one with a not too high an energy inscribed, respectively.*

*In fact, we aim at* ***stabilizing*** *of the model by penalizing high-second order derivatives since they lead to too high* ***energy****, and hence, sensitivity, inscribed. In addition, besides of this reference to (ν = ) 2nd order variations, we may also take into account (ν = ) 1st order infor-*

*mation. In that case, we record and penalize higher first-order derivatives, i.e., we can foster a **flat model**. We refer to [55, 80, 81] for closer information on these kinds of penalizations. In our research, we introduced the new idea of the scanning, of moving upper integration limits, herewith recording the "behaviour" of our measure $\beta$ (or $\sigma$) in first- and second-order senses.*

Motivated by the theory of *inverse problems* [6, 81], our Closer Explanation 5.4.1 can for the ***primal*** problem be elaborated as:

$$\begin{aligned} &\min_{\theta,\beta} \ (-\theta) + \lambda \sup_{t\in[0,1]} \left| \frac{d^\nu}{dt^\nu} \int_0^t d\beta(\omega) \right| \\ &\text{subject to } \int_\Omega T(\omega,\boldsymbol{\alpha}) d\beta(\omega) - \theta \geqslant 0 \ \ (\boldsymbol{\alpha} \in A), \end{aligned} \tag{5.40}$$

where the second term in the objective function is the regularization term, and $\lambda \geq 0$ is a regularization parameter. With $\nu = 1, 2$, we express that we take into account and penalize first- or second-order derivatives which we can interpret as *steepness* (or *flatness*) and *energy* within of our models, respectively [6, 55, 80, 81].

Another formulation can be done by including the kernel combination $k_\beta$ derived, e.g., by a homotopy as discussed in Section 3.1, as follows:

$$\begin{aligned} &\min_{\theta,\,\beta} (-\theta) + \lambda \sum_{i,j=1}^{l} \sup_{t\in[0,1]} \left| \frac{d^\nu}{dt^\nu} \int_0^t k(\mathbf{x}_i, \mathbf{x}_j, \omega) d\beta(\omega) \right| \\ &\text{subject to } \int_\Omega T(\omega,\boldsymbol{\alpha}) d\beta(\omega) - \theta \geqslant 0 \ \ (\boldsymbol{\alpha} \in A), \end{aligned} \tag{5.41}$$

where $\lambda$ is a regularization parameter again.
Let us observe that our regularization term, $t \mapsto \sum_{i,j=1}^{l} \sup_{t\in[0,1]} \left| \frac{d^\nu}{dt^\nu} \int_0^t k(\mathbf{x}_i, \mathbf{x}_j, \omega) d\beta(\omega) \right|$, highly depends on the parameter $\beta$ and it usually needs to be twice continuously differentiable to be well-defined. To weaken the need of differentiability, we replace the derivatives by *first-* and *second-order difference quotients*, as offered in the example below, where $0 = t_0 < t_1 < \ldots < t_m = 1$ is a discrete *subdivision* (one-dimensional *mesh*) of the interval $[0, 1]$.

Instead of introducing regularized problem in a measure theoretical setting, we look at shortly a special case in which parameters of both (5.40) and (5.41) can be pdf functions $f$ such that $d\beta(\omega) = f(\omega)d\omega$. If $f \in L_\infty[0, 1]$ then our functions of the form $t \mapsto \int_0^t \delta(\omega) f(\omega) d\omega$ are differentiable almost everywhere. But even if the derivatives exist everywhere, they may

be hard to compute. For these reasons, we prefer to "discretize" the differential quotients by difference quotients for primal and the dual problems as presented by the following examples.

**Example 5.4.2** ***(primal case):***

- ***First-order difference quotient:***

$$\begin{aligned}\frac{d}{dt}\int_0^t d\beta(\omega)\Big|_{t=t_\nu} &\approx \frac{\int_0^{t_{\nu+1}} d\beta(\omega) - \int_0^{t_\nu} d\beta(\omega)}{t_{\nu+1}-t_\nu} \\ &= \frac{1}{t_{\nu+1}-t_\nu}\int_{t_\nu}^{t_{\nu+1}} d\beta(\omega) \qquad (\nu \in \{0,1,\ldots,m-1\}).\end{aligned}$$

- ***Second-order difference quotient:***

$$\frac{d^2}{dt^2}\int_0^t d\beta(\omega)\Big|_{t=t_\nu} \approx \frac{\frac{1}{t_{\nu+2}-t_{\nu+1}}\int_{t_{\nu+1}}^{t_{\nu+2}} d\beta(\omega) - \frac{1}{t_{\nu+1}-t_\nu}\int_{t_\nu}^{t_{\nu+1}} d\beta(\omega)}{t_{\nu+1}-t_\nu}$$

($\nu \in \{0,1,\ldots,m-2\}$). *Now, let us analyze the regularization of the* ***dual*** *problem as we did in (5.40) and (5.41):*

$$\begin{aligned}&\min_{\sigma,\rho}\ \sigma + \lambda \cdot \sup_{\boldsymbol{\tau}\in[0,C]^l} \sum_{\ell=1}^{l}\left|\frac{d^\nu}{dt_\ell^\nu}\int_{\boldsymbol{Q}^{\boldsymbol{\tau}}} d\rho(\boldsymbol{\alpha})\right| \\ &\textit{subject to } \sigma - \textstyle\int_A T(\omega,\boldsymbol{\alpha})d\rho(\boldsymbol{\alpha}) \geqslant 0\ \ (\omega\in\Omega),\end{aligned} \tag{5.42}$$

*and*

$$\begin{aligned}&\min_{\sigma,\rho}\ \sigma + \lambda \cdot \sum_{i,j=1}^{l}\sup_{\boldsymbol{\tau}\in[0,C]^l} \sum_{\ell=1}^{l}\left|\frac{d^\nu}{dt_\ell^\nu}\int_{\boldsymbol{Q}^{\boldsymbol{\tau}}} \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j,\omega)d\rho(\boldsymbol{\alpha})\right| \\ &\textit{subject to } \sigma - \textstyle\int_A T(\omega,\boldsymbol{\alpha})d\rho(\boldsymbol{\alpha}) \geqslant 0\ \ (\omega\in\Omega),\end{aligned} \tag{5.43}$$

*where* $\boldsymbol{\tau} := (t^1, t^2, \ldots, t^l)^T$, $\boldsymbol{Q}^{\boldsymbol{\tau}} := \prod_{i=1}^{l}[0,t^i]$ *and* $\nu = 1, 2$, *respectively.*

**Note 5.4.3** *Alternatively, we might also integrate along the* ***line segment*** $[\mathbf{0},\boldsymbol{\tau}] \subset \mathbb{R}^l$. *By means of real analysis and the theory of functions, we could closer characterize, for which functions integrated and which probability measures and densities, the integration does not depend on the arc which may connect* $\mathbf{0}$ *and* $\boldsymbol{\tau}$.

As in the previous (primal) case, we replace the need of differentiability and of differentiation by first- and second-order quotients:

**Example 5.4.4** *(**dual case**): In the above problems given by (5.42) and (5.43), we now assume that the iterated integrals exist, e.g., by existence and continuity of density functions of $\rho$. Instead of using the integration over $[0,C]^l$ by iterated integration, we present the following approach and the integral evaluation with respect to a probability measure. Here, we refer to grid points $\boldsymbol{\tau_\nu} = (t_\nu^1, t_\nu^2, \ldots, t_\nu^l)^T$ at the place of the subdivision points $t_\nu$ of the one-dimensional (primal) case ($\nu \in \{0,1,\ldots,m\}$) with $t_0^\mu := 0$ and $t_m^\mu := C$ ($\mu \in \{1,2,\ldots,l\}$). For simplicity, we may assume that we have the same number $s$ of subintervals in all the dimensions i.e., $m = (s+1)^l$.*

- ***First-order difference quotient:***

$$\left[\frac{\partial}{\partial t_1}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}), \frac{\partial}{\partial t_2}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}), \ldots, \frac{\partial}{\partial t_l}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha})\right]^T_{|\boldsymbol{\tau}=(t_\nu^1,t_\nu^2,\ldots,t_\nu^l)^T}$$

$$\approx \left[\frac{\int_{\boldsymbol{Q}_1^\tau} d\rho(\boldsymbol{\alpha}) - \int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha})}{t^1{}_{\nu+1} - t^1{}_\nu}, \frac{\int_{\boldsymbol{Q}_2^\tau} d\rho(\boldsymbol{\alpha}) - \int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha})}{t^2{}_{\nu+1} - t^2{}_\nu}, \ldots, \frac{\int_{\boldsymbol{Q}_l^\tau} d\rho(\boldsymbol{\alpha}) - \int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha})}{t^l{}_{\nu+1} - t^l{}_\nu}\right]^T \tag{5.44}$$

$$= \left[\frac{1}{t^1{}_{\nu+1} - t_\nu^1}\int_{\boldsymbol{\Delta Q}_1^\tau} d\rho(\boldsymbol{\alpha}), \frac{1}{t^2{}_{\nu+1} - t_\nu^2}\int_{\boldsymbol{\Delta Q}_2^\tau} d\rho(\boldsymbol{\alpha}), \ldots, \frac{1}{t^l{}_{\nu+1} - t_\nu^l}\int_{\boldsymbol{\Delta Q}_l^\tau} d\rho(\boldsymbol{\alpha})\right]^T,$$

*where $\boldsymbol{Q}_\mu^{\boldsymbol{\tau_\nu}} := [0,t_\nu^1] \times [0,t_\nu^2] \times \ldots \times [0,t_\nu^{\mu-1}] \times [0,t_{\nu+\mu}^\mu] \times [0,t_\nu^{\mu+1}] \times \ldots \times [0,t_\nu^l]$ and $\boldsymbol{\Delta Q}_\mu^{\boldsymbol{\tau_\nu}} := [0,t_\nu^1] \times [0,t_\nu^2] \times \ldots \times [0,t_\nu^{\mu-1}] \times [t_\nu^\mu, t_{\nu+\mu}^\mu] \times [0,t_\nu^{\mu+1}] \times \ldots \times [0,t_\nu^l]$, ($\mu \in \{1,2,\ldots,l\}$), where $\mu$ and $\nu$ are elements in $\{1,2,\ldots,l\}$ and $\{0,1,\ldots,m\}$, respectively. Here, we denote the $l$ neighbours of $\boldsymbol{\tau_\nu}$ (as far as they are lying in $[0,C]^l$) according to the $l$ coordinates, where just one of them become increased respectively, by $\boldsymbol{\tau_{\nu+1}}, \boldsymbol{\tau_{\nu+2}}, \ldots, \boldsymbol{\tau_{\nu+l}}$ (renumerating if needed).*

- ***Second-order difference quotient:***

*The Hessian matrix*

$$\left[\begin{array}{cccc} \frac{\partial^2}{\partial t_1^2}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}) & \frac{\partial^2}{\partial t_2 \partial t_1}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}) & \ldots & \frac{\partial^2}{\partial t_l \partial t_1}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}) \\ \frac{\partial^2}{\partial t_1 \partial t_2}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}) & \frac{\partial^2}{\partial t_2^2}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}) & \ldots & \frac{\partial^2}{\partial t_l \partial t_2}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial t_1 \partial t_l}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}) & \frac{\partial^2}{\partial t_2 \partial t_l}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}) & \ldots & \frac{\partial^2}{\partial t_l^2}\int_{\boldsymbol{Q}^\tau} d\rho(\boldsymbol{\alpha}) \end{array}\right]_{|\boldsymbol{\tau}=(t_\nu^1,t_\nu^2,\ldots,t_\nu^l)^T} \tag{5.45}$$

*can be discretely approximated by employing the methodology presented for the first-order difference quotients on the components of the approximate gradients which we obtained there.*

In this section, we added a regularization term because of our state (decision) variable in infinite dimensional space, and in order to reduce the complexity of the model. In the following section, we propose various *probability density functions* (since we study *probability* measures now) for our SIP problem, instead of further addressing probability measures themselves. In the following, we will refer to our SIP problems, based on a parametrization by these pdfs, for the rest of the thesis.

## 5.5 DIFFERENT PARAMETRIZATION FUNCTIONS FOR INFINITE PROBLEM

Until now, we have assumed that parameters $(\theta, \beta)$ and $(\sigma, \rho)$ are given for both the primal problem and the dual problem which are from infinite dimensional spaces. For the remaining part of the thesis, we assume that we are given certain classes of probability measures and, in fact, we parametrize probability density functions (pdfs). In closer detail, we consider these positive measures $\beta$ and $\rho$ such that $\int_0^1 d\beta(\omega) = 1$ and $\int_A d\rho(\boldsymbol{\alpha}) = 1$, and we refer to probability density functions $f$ such that $f(\omega)d\omega$ and $f(\boldsymbol{\alpha})d\boldsymbol{\alpha}$ take the place of $d\beta(\omega)$ and $d\rho(\boldsymbol{\alpha})$, respectively. For example, there are the pdfs of a *normal, exponential, uniform, beta,* or a *Poisson distribution* [69, 30]. We note that we herewith reduce our IP problems into SIP problems, since now our state (or decision) variables are from finite dimensional spaces by the parametrization.

**Normal Distribution:** This distribution is also called *Gaussian distribution*; it is very appropriate for modelling of various continuous random variables. The sampling distribution of the sample mean is approximately normal, even if the distribution of the population from which the sample is taken is not normal [30, 69]. The pdf of a normal distribution is

$$f(x; (\mu, \sigma^2)) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(x-\mu)^2}{2\sigma^2}, \tag{5.46}$$

where $x,\ \mu,\ \sigma \in \mathbb{R}$. Here, $\mu$ is the *expected value* of the point $x$, $\sigma \geq 0$ is the *standard deviation* and $\sigma^2$ is the *variance*. For simplicity, we denote the variable $x := \omega$ for our dual problem, $\mathbf{x} := \boldsymbol{\alpha}$ for our primal problem. Since our primal variable $\boldsymbol{\alpha}$ is multidimensional, i.e., $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_l)^T$, next, we define our pdf for the multivariate case:

$$f(\mathbf{x}; (\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \frac{1}{(2\pi)^{l/2}\det(\boldsymbol{\Sigma})^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})). \tag{5.47}$$

Here, $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is a *covariance matrix* which is symmetric and requested to be positive definite, i.e., $\boldsymbol{\Sigma} > 0$. In fact, $\boldsymbol{\Sigma}$ is defined in the group $S^+$, which is the group of the symmetric and positive definite matrices.

In the special case where the (random) variables $\mathbf{x}_i$ ($i = 1, 2, \ldots, l$) are uncorrelated and of the same variance, we can take the covariance matrix as a scalar multiple of the identity matrix, i.e., $\boldsymbol{\Sigma} = \sigma^2 I$, where $\sigma$ is the standard deviation and $\sigma^2$ is the variance [6]. Then, pdf turns into a function having two parameter, i.e., $\sigma^2$ and $\boldsymbol{\mu}$:

$$f(\mathbf{x}; (\boldsymbol{\mu}, \sigma^2)) = \frac{1}{(2\pi)^{1/2}\sigma} \exp(-\frac{\|\mathbf{x} - \boldsymbol{\mu}\|_2^2}{2\sigma^2}). \tag{5.48}$$

We can use the above formulas (5.46), (5.47) and (5.48) in our SIP reformulation problem, with the parameters $\mu$ and $\sigma^2$, or $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively, as new state variables (see Chapter 6).

**Exponential Distribution:** This distribution is a class of continuous probability distributions which is useful for modelling time between independent events of constant average rate [69]. The pdf of an *exponential distribution* looks as follows:

$$f(x; \lambda) = \begin{cases} \lambda \exp(-\lambda x), & x \geqslant 0, \\ 0, & x < 0. \end{cases}$$

Since $x := \omega \in [0, 1]$ in our problem, $\beta(\omega) = \lambda \exp(-\lambda\omega)d\omega$, where $\lambda \in \mathbb{R}$ is a parameter of rate. Of course, translations of the origin 0, e.g., delay, are possible.

**Continuous Uniform Distribution:** This is a family of probability distributions such that for each member of the family, all intervals $[a, b]$ of the same length on the distribution's support are equally probable. The pdf of a *continuous uniform distribution* looks as follows:

$$f(x; (a, b)) = \begin{cases} \frac{1}{b-a}, & a \leqslant x \leqslant b, \\ 0, & x < a \text{ or } x > b. \end{cases}$$

Here, $x := \omega$, $a, b \in \mathbb{R}$, $a < b$. We can enforce $a \leq b - \epsilon$ with $\epsilon > 0$ sufficiently small, to insert it as a constraint into our SIP problem (see Chapter 6).

**Beta Distribution:** The *Beta distribution* is a family of continuous probability distributions defined on the interval [0,1] parameterized by two positive shape parameters, typically denoted by $\alpha$ and $\beta$. (No confusion with the meaning of $\alpha$ and $\beta$ in our paper needs to be

expected.) The pdf of a Beta distribution looks as follows:

$$f(x;(\alpha,\beta)) = \frac{x^{\alpha-1}(1-x^{\beta-1})}{\int_0^1 \omega^{\alpha-1}(1-\omega)^{\beta-1}d\omega}.$$

## 5.6 SUMMARY OF INFINITE KERNEL LEARNING AND ITS ADVANTAGES

In this section, we compare our *Infinite Kernel Learning (IKL)* problem with *Multiple Kernel Learning (MKL)* studied in [8, 71] and give advantages of our IP formulation both in theory and computation.

A *multiple* combination of kernels limits our choice of kernels to a discrete search space or it can lead to miss important kernel elements in this limited space. Making the search space infinite dimensional enables the formerly discretely many kernels and their coefficients to be in a continuous domain. We interpret this continuity by Riemann-Stieltjes integrals (see Section 2.3). Among the infinitesimal coefficients, some of them are point masses, i.e., finitely many of the inequality constraints are active (see Remark 5.3.3). We guarantee this by the reduction ansatz, by smoothness of the model functions, and by employing the Theorem of Heine-Borel. Then, since our infinite index sets of inequality constraints are compact, there are finitely many active constraints (point masses) in this compact domain (see Corollary 5.3.3).

When using a multiple kernel combination, practically, one needs to store all kernels in the computer memory. Thus, it can be intractable, but this drawback can be overcome by parallel algorithms or the *chunking algorithm* [71]. In our infinite kind of generalization, one may have doubts about any storing infinitely many kernels in the memory. However, we do not need to save infinitely many kernels, we only need to evaluate the Riemann-Stieltjes formulation and record just the result of the integral. We believe that this will save training time of the SVM. Continuity of the search space implies for us that we have one unique kernel by integration. Hence, the classification function of our Infinite Kernel Learning SVM looks as follows:

$$f^{IKL}(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i k_\beta(\mathbf{x}_i, \mathbf{x}) + b, \tag{5.49}$$

where $k_\beta(\mathbf{x}_i, \mathbf{x}) = \int_\Omega k(\mathbf{x}_i, \mathbf{x}, \omega) d\beta(\omega)$ and $b$ is the bias (see Section 2.2 and Section 5.3.1).

Our classifier of Infinite Kernel Learning (IKL), $f^{IKL}$, differs from the classifier of Multiple

Kernel Learning (MKL), $f^{MKL}$ (given by 5.50) only in kernel definitions (by integral implied):

$$f^{MKL}(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i k_{\boldsymbol{\beta}}(\mathbf{x}_i, \mathbf{x}) + b, \tag{5.50}$$

where $k_{\boldsymbol{\beta}}(\mathbf{x}_i, \mathbf{x}, \omega) = \sum_{\kappa=1}^{K} \boldsymbol{\beta}_\kappa k_\kappa(\mathbf{x}_i, \mathbf{x})$ and $b$ is the bias.

In order to find a combination of kernels in (5.50), one needs to store all kernels and also to update the combination and solve SILP.

Again, we point out that the continuity of the combination will save memory and use the benefit of all possibilities of kernels via the integration.

## 5.7 CONCLUSION

The method we proposed in this study leads to the selection of kernels from an infinite space which enables us to enrich the learning process SVM through the range interval $[0, 1]$ of the integration of a specific or homotopy parameter of $\omega$. Hence, we are not limited to choose kernel parameter(s), Gaussian kernels in our special case, as discrete values with a cross validation method, but that depending on the examples given beforehand we can learn from data through this infinite process. Hence, we avoid model selection of kernels. By reduction ansatz, an infinite problem is turned into a locally finitely constrained problem where, however, probability measures are our main state variables. By focusing on measures which possess a Radon-Nikodym density, we turn to a space of density functions [92]. By looking at parametric density functions, we get semi-infinite and, via reduction ansatz, a locally finitely constrained and finite dimensional program indeed. Besides of that analytical ansatz which is hard to implement because of IFT, discretization and exchange methods will be analyzed and developed in the next chapter.

In this thesis, the classification problem by SVM is modeled with infinitely many kernels by infinite programming. The proposed dimension is infinite, and our SVM has infinitely many constraints which may cause ill-posedness. To overcome this, we introduced some regularization terms into the objective function, where the derivative of the regularization term is approximated by first- and second-order difference quotients. This kind of problems can be useful for real-world data which are huge and heterogeneous, e.g., in bioinformatics and finan-

cial applications. The proposed method is novel by its kernel definitions in Riemann-Stieltjes or Lebesgue integral form. Our optimization problems become defined alternatively in probability measures as state variables, which are from an infinite dimensional space. Here, we offered a parametrization of the probability measures via their probability distribution functions. We gave some examples of pdfs to be applied. Another novelty of our approach is to use *Prokhorov distances* between *Radon measures* in the *weak topology* to define neighbourhoods in the state space. Furthermore, we defined the *bounded Lipschitz metric* which better fits to the definition of the weak topology. It enables us to define distance in terms of functions rather than sets, where the space of functions is a class of real-valued measurable functions defined on a metric space $E$. There, our probability measures, special Radon measures, are defined on.

# CHAPTER 6

# NUMERICAL TREATMENT OF INFINITE KERNEL LEARNING PROBLEM

## 6.1 INTRODUCTION

In this chapter, existing numerical methods of *SIP* will be adapted to our infinite kernel learning problem. Since there are an infinite number of constraints both in the primal and the dual problems, (5.18) and (5.11), and the variables $\beta$ in the primal and $\rho$ in the dual problem are from infinite dimensional spaces, the infinite problems are reduced into or approximated by semi-finite problems to be solvable in practice. Reducing the IP into SIP is an alternative way to solve the problem. In this thesis, we checked the regularity conditions of our IP and, hence, of our SIP problem on the lower level. On the other hand, we can check the optimality conditions of IP for our primal and the dual IP problem. We note that our primal and dual IP tasks, (5.18) and (5.11), respectively, are linear in the infinite dimensional variables $\beta$ and $\rho$. Thus Infinite Linear Programming (ILP) techniques and treatment by primal-dual methods [3] can also be searched alternatively which, however, is not the main scope of this thesis. In Section 5.5, the parameters, i.e., measures, $\beta$ in the primal and $\rho$ in dual problem, were considered by pdfs; different examples for pdfs have been given in order to reduce our *IP* to *SIP*. Thus, given a pdf, it remains to solve an SIP problem in some neighbourhood of these parametrized measures. In the following sections, given a pdf, different numerical methods will be adapted to our reduced IP problem. In this chapter, we denote the iteration steps as $k$; there should not be any confusion with the names of our kernel functions.

## 6.2 DISCRETIZATION

In this section, we will adapt fundamental numerical methods to our reduced *SIP* problems. In Section 5.5, we introduced different parametrization functions for the infinite dimensional variables $\beta$ and $\rho$ to reduce our *IP* problems into *SIP* problems. Throughout this chapter, we assume that we are given pdf functions $f^{\mathcal{P}}(\omega;\cdot)$ and $f^{\mathcal{D}}(\boldsymbol{\alpha};\cdot)$ for our primal and dual problems, respectively. In this chapter, we do not need to write the equality constraint $\int_\Omega d\beta(\omega) = 1$ (or $\int_A d\rho(\boldsymbol{\alpha}) = 1$), since we assume that our measures are probability measures. Then, we parametrize these measures via **pdfs** $f^{\mathcal{P}} = f^{\mathcal{P}}(\omega;\boldsymbol{\wp}^{\mathcal{P}})$ and $f^{\mathcal{D}} = f^{\mathcal{D}}(\boldsymbol{\alpha};\boldsymbol{\wp}^{\mathcal{D}})$, taking the place of positive measures $\beta$ and $\rho$. Let us denote the parameters of these pdfs by $\boldsymbol{\wp}^{\mathcal{P}} = (\wp_1^{\mathcal{P}}, \wp_2^{\mathcal{P}}, \ldots, \wp_{\iota^{\mathcal{P}}}^{\mathcal{P}})^T$ and $\boldsymbol{\wp}^{\mathcal{D}} = (\wp_1^{\mathcal{D}}, \wp_2^{\mathcal{D}}, \ldots, \wp_{\iota^{\mathcal{D}}}^{\mathcal{D}})^T$ for the primal and the dual SIP problems, respectively. They are constrained and elements of suitable sets:

$$P^{\mathcal{P}} := \{\boldsymbol{\wp}^{\mathcal{P}} \in \mathbb{R}^{\iota^{\mathcal{P}}} \;\Big|\; \mathrm{u}_i^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) = 0 \;\; (i \in I^{\mathcal{P}}),\; \mathrm{v}_j^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) \geq 0 \;\; (j \in J^{\mathcal{P}})\}$$

and

$$P^{\mathcal{D}} := \{\boldsymbol{\wp}^{\mathcal{D}} \in \mathbb{R}^{\iota^{\mathcal{D}}} \;\Big|\; \mathrm{u}_i^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) = 0 \;\; (i \in I^{\mathcal{D}}),\; \mathrm{v}_j^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) \geq 0 \;\; (j \in J^{\mathcal{D}})\}.$$

Then, instead of optimizing with respect to measure $\beta$ (or $\rho$), we minimize with respect to the pdf parameter vector $\boldsymbol{\wp}^{\mathcal{P}}$ (or $\boldsymbol{\wp}^{\mathcal{D}}$). Hence, our problems turn into the following *SIP* tasks with additional constraint functions $\mathrm{u}_i^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}})$, $\mathrm{v}_j^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}})$, $\mathrm{u}_i^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}})$ and $\mathrm{v}_j^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{P}})$, coming from the definition of the parameter sets related to the specific pdf functions of the primal and the dual problems, respectively:

$$\begin{aligned}
(\textit{Primal SIP}) \quad & \min_{\theta,\boldsymbol{\wp}^{\mathcal{P}}} \;\; (-\theta) \\
& \text{such that} \;\; \textstyle\int_\Omega T(\omega,\boldsymbol{\alpha}) f^{\mathcal{P}}(\omega;\boldsymbol{\wp}^{\mathcal{P}}) d(\omega) - \theta \geq 0 \;\; (\boldsymbol{\alpha} \in A), \\
& \mathrm{u}_i^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) = 0 \;\; (i \in I^{\mathcal{P}}), \\
& \mathrm{v}_j^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) \geq 0 \;\; (j \in J^{\mathcal{P}}),
\end{aligned} \tag{6.1}$$

and

$$\begin{aligned}
(\textit{Dual SIP}) \quad & \min_{\sigma,\boldsymbol{\wp}^{\mathcal{D}}} \;\; \sigma \\
& \text{such that} \;\; \sigma - \textstyle\int_A T(\omega,\boldsymbol{\alpha}) f^{\mathcal{D}}(\boldsymbol{\alpha};\boldsymbol{\wp}^{\mathcal{D}}) d(\boldsymbol{\alpha}) \leq 0 \;\; (\omega \in \Omega), \\
& \mathrm{u}_i^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) = 0 \;\; (i \in I^{\mathcal{D}}), \\
& \mathrm{v}_j^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) \geq 0 \;\; (j \in J^{\mathcal{D}}).
\end{aligned} \tag{6.2}$$

One of the early methods mostly used to solve SIP problems in practice, e.g., in engineering applications, is *discretization* [33]. It is based on a discretization of the infinite index set of

inequality constraints. In our study, these infinite index sets are $A$ and $\Omega$ for the primal and the dual problems, respectively.

The discretized primal SIP and the discretized dual SIP problems of (6.1) and (6.2) can be rewritten by the following formulations:

$$\begin{aligned} P(A_k) \quad & \min_{\theta, \boldsymbol{\wp}^{\mathcal{P}}} \; -\theta \\ & \text{subject to} \quad \int_{\Omega} T(\omega, \boldsymbol{\alpha}) f^{\mathcal{P}}(\omega; \boldsymbol{\wp}^{\mathcal{P}}) d\omega - \theta \geqslant 0 \;\; (\boldsymbol{\alpha} \in A_k), \\ & \mathrm{u}_i^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) = 0 \;\; (i \in I^{\mathcal{P}}), \\ & \mathrm{v}_j^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) \geq 0 \;\; (\mathrm{j} \in J^{\mathcal{P}}), \end{aligned} \tag{6.3}$$

and

$$\begin{aligned} D(\Omega_k) \quad & \min_{\sigma, \boldsymbol{\wp}^{\mathcal{D}}} \; \sigma \\ & \text{subject to} \quad \sigma - \int_{A} T(\omega, \boldsymbol{\alpha}) f^{\mathcal{D}}(\boldsymbol{\alpha}; \boldsymbol{\wp}^{\mathcal{D}}) d\boldsymbol{\alpha} \leqslant 0 \; (\omega \in \Omega_k), \\ & \mathrm{u}_i^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) = 0 \;\; (i \in I^{\mathcal{D}}), \\ & \mathrm{v}_j^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) \geq 0 \;\; (j \in J^{\mathcal{D}}). \end{aligned} \tag{6.4}$$

Here, by the symbol $P(\cdot)$ we denote the primal and by $D(\cdot)$ the dual problems, $k$ is the iteration step, and the discretized set $A_k$ will be discussed within Strategy I and Strategy II presented later in this section. It is obvious that $\Omega_k$ can be defined by a one-dimensional uniform grid[1].

Let $v^{\mathcal{P}}(A_k)$, $\mathcal{M}^{\mathcal{P}}(A_k)$ and $\mathcal{G}^{\mathcal{P}}(A_k)$ denote the minimal value, the feasible set and the set of (global) minimizers of our primal problem (6.1) with $A$ replaced by $A_k$. Furthermore, $v^{\mathcal{D}}(\Omega_k)$, $\mathcal{M}(\Omega_k)$ and $\mathcal{G}^{\mathcal{D}}(\Omega_k)$ be the corresponding ones for the dual problem, i.e., they denote the minimal value, the feasible set and the set of (global) minimizers of our dual problem (6.2) with $\Omega$ replaced by $\Omega_k$. As discussed in Chapter 5, under suitable regularity conditions (reduction ansatz), the optimal solutions of the lower level problems depend locally on the parameters, i.e., measures, the Prokhorov distance has been introduced in the space of Radon measures. Furthermore, the relation with the pdfs has been established with a dual pairing and by the pdfs as test kind of functions from the dual space.

Let $d_1$ be the *Hausdorff* distance $d_1(A_k, A)$ between $A$ and $A_k$, which is given by

$$d_1(A_k, A) := \max_{\mathbf{y} \in A} \; \min_{\mathbf{y}' \in A_k} \left\| \mathbf{y} - \mathbf{y}' \right\|_2 .$$

[1] A uniform grid is discretization of a considered set where all elements $\mathbf{x} = (x_1, x_2, \ldots, x_l)^T$ have same spacing with respect to their $i$-th coordinate ($i = 1, 2, \ldots, l$). For example in $\mathbb{R}^2$, all rows have the same spacing and all of the columns have the same spacing (but not necessarily the same as the row spacing).

Now, with the Hausdorff distance, we will introduce the discretizability notion for our problems based on the definitions in [78]. In our problems, $\mathbf{y} = \boldsymbol{\alpha}$ and $\mathbf{y}' = \boldsymbol{\alpha}'$ for the primal and $y = \omega,\ y' = \omega'$ for the dual case. In the following definitions, the distance to the solutions $(\theta^*, \boldsymbol{\wp}^{\mathcal{P}^*})$ and $(\sigma^*, \boldsymbol{\wp}^{\mathcal{D}^*})$ of the primal and the dual SIP, respectively, will be defined with the Hausdorff distance, too. We note that the optimal solution of the primal problem and of the dual problem exist because of the continuity of the objective functions and inequality constraints, and compactness of the feasible sets proposed subsequently in Closer Explanation 6.2.5. Here, we employ Theorem of Weierstrass . We denote the distance functions $d_1$ for the dual problem and the primal problem as $d_1^{\mathcal{D}}$ and $d_1^{\mathcal{P}}$, respectively.

**Definition 6.2.1** *The primal and the dual problems, (6.1), (6.2), respectively, are called* ***finitely reducible*** *if there are finite sets $A_{k^0} \subset A$, $\Omega_{k^0} \subset \Omega$ for some $k = k^0$, such that $v^{\mathcal{P}}(A_{k^0}) = v^{\mathcal{P}}(A)$, $v^{\mathcal{D}}(\Omega_{k^0}) = v^{\mathcal{D}}(\Omega)$, and $(A_k)_{k\in\mathbb{N}_0}$, $(\Omega_k)_{k\in\mathbb{N}_0}$ strictly isotonically increase[2] as $k \to \infty$.*

**Definition 6.2.2** *The primal and the dual problems, (6.1), (6.2), respectively, are called* ***weakly discretizable*** *if there exist sequences of discretizations $(A_k)_{k\in\mathbb{N}_0}$ and $(\Omega_k)_{k\in\mathbb{N}0}$ such that $v^{\mathcal{P}}(A_k) \to v^{\mathcal{P}}(A)$ and $v^{\mathcal{D}}(\Omega_k) \to v^{\mathcal{P}}(\Omega)$ $(k \to \infty)$.*

We note that we have $v^{\mathcal{P}}(A_{k^1}) \leq v^{\mathcal{P}}(A_{k^2})$ if $A_{k^1} \subset A_{k^2}$ for our primal problem, and $v^{\mathcal{D}}(\Omega_{k^1}) \leq v^{\mathcal{D}}(\Omega_{k^2})$ if $A_{k^1} \subset A_{k^2}$ for our dual problem. We recall that we consider the standard form of primal SIP problems given by (5.19), i.e., minimization problems. In closer explanation, as the infinite index set grows, the number of inequality constraints increases. This forces the feasible set to become smaller at each iteration $k$. Thus, the minimum of the objective function increases (see Figure 6.1). In Figure 6.1, $A_k \subset A_{k+1}$ is not the case, but obviously, $v^{\mathcal{P}}(A_{k^1}) \leq v^{\mathcal{P}}(A_{k^2})$.

**Definition 6.2.3** *The dual and the primal problems, (6.1), (6.2), respectively, are called* ***discretizable*** *if for each sequence of finite grids $A_k \subset A$ $(k \in \mathbb{N}_0)$ for the primal problem, and $\Omega_k \subset \Omega$ $(k \in \mathbb{N}_0)$ for the dual problem, satisfying $d_1^{\mathcal{P}}(A_k, A) \to 0$ $(k \to \infty)$ and $d_1^{\mathcal{D}}(\Omega_k, \Omega) \to 0$ $(k \to \infty)$, where $d_1^{\mathcal{P}}(A_k, A) =: \max\limits_{\boldsymbol{\alpha}\in A} \min\limits_{\boldsymbol{\alpha}'\in A_k} \left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right\|_2$, and $d_1^{\mathcal{D}}(\Omega_k, \Omega) =: \max\limits_{\omega\in\Omega} \min\limits_{\omega'\in\Omega_k} \left\|\omega - \omega'\right\|_2$, there exist solutions $(\bar{\theta}_k, \bar{\boldsymbol{\wp}}_{\boldsymbol{k}}^{\mathcal{P}})_{k\in\mathbb{N}_0}$ of the discretized primal*

[2] A sequence $(A_k)_{k\in\mathbb{N}_0}$ is strictly isotonically increasing if $A_k \subsetneq A_{k+1}$ $(k \in \mathbb{N}_0)$.

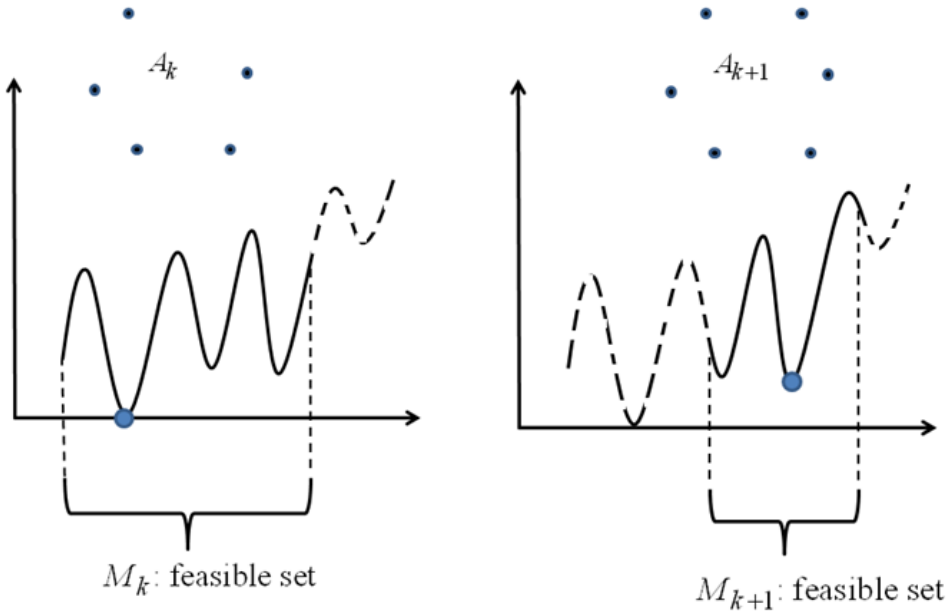


Figure 6.1: Illustration of the minimum values with respect to different feasible sets corresponding different discretizations; an example.

*problems (6.3) and* $(\bar{\sigma}_k, \bar{\boldsymbol{\varphi}}_{\boldsymbol{k}}^{\mathcal{D}})_{k\in\mathbb{N}_0}$ *of the discretized dual problems (6.4) such that the following relations*

$$\min_{(\theta,\boldsymbol{\varphi}^{\mathcal{P}})\in\mathcal{G}^{\mathcal{P}}(A)} \left\|(\bar{\theta}_k, \bar{\boldsymbol{\varphi}}_{\boldsymbol{k}}^{\mathcal{P}}) - (\theta, \boldsymbol{\varphi}^{\mathcal{P}})\right\|_2 \to 0 \tag{6.5}$$

$$\text{and} \quad v^{\mathcal{P}}(A_k) \to v^{\mathcal{P}}(A) \ \ (k \to \infty),$$

$$\min_{(\sigma,\boldsymbol{\varphi}^{\mathcal{D}})\in\mathcal{G}^{\mathcal{D}}(\Omega)} \left\|(\bar{\sigma}_k, \bar{\boldsymbol{\varphi}}_{\boldsymbol{k}}^{\mathcal{D}}) - (\sigma, \boldsymbol{\varphi}^{\mathcal{D}})\right\|_2 \to 0 \tag{6.6}$$

$$\text{and} \quad v^{\mathcal{D}}(\Omega_k) \to v^{\mathcal{D}}(\Omega) \ \ (k \to \infty),$$

*hold for each problem, respectively.*

**Corollary 6.2.4** *If the primal and the dual problems, (6.1) and (6.2), respectively, are finitely reducible, then both problems (6.1) and (6.2) are weakly discretizable.*

***Proof 2*** *Let us assume that (6.1) and (6.2) are finitely reducible. Then, by definition, there exist a* $k_0 \in \mathbb{N}_0$ *and finite sets* $A_{k^0} \subset A$ *for the primal and* $\Omega_{k^0} \subset \Omega$ *for the dual such that* $v^{\mathcal{P}}(A_{k^0}) = v^{\mathcal{P}}(A)$ *and* $v^{\mathcal{D}}(\Omega_{k^0}) = v^{\mathcal{D}}(\Omega)$*. Then, it is obvious that* $v^{\mathcal{P}}(A_k) \to v^{\mathcal{P}}(A)$ *and* $v^{\mathcal{D}}(\Omega_k) \to v^{\mathcal{D}}(\Omega)$ $(k \to \infty)$.

Under the discretizability notion established above, we introduce the conceptual discretization algorithm in the following subsection.

### 6.2.1 CONCEPTUAL DISCRETIZATION METHOD

The *conceptual discretization* method is based on an update of the discretization according to some stopping criterion for the convergence of the optimal solution. We adapt the conceptual discretization method [33, 34, 78] to our primal and the dual problem in Algorithm 1 and Algorithm 2.

**Algorithm 1 Primal Conceptual Discretization Method (PCDM)**

**Input:**

$\delta$ positive number, i.e., $\delta > 0$

$f^{\mathcal{P}}$ probability density function (pdf)

$P^{\mathcal{P}}$ the set where pdf parameters lie

**Output:**

$\theta$ unknown variable for minimization, to be evaluated

$\boldsymbol{\wp}^{\mathcal{P}}$ the parameter vector of the pdf

**PCDM**$\left(\theta, \boldsymbol{\wp}^{\mathcal{P}}, A, \delta, f^{\mathcal{P}}, P^{\mathcal{P}}\right)$

1: $k := 0$

2: Initialize a discretization $A_k \subset A$.

3: **DO** Compute a solution $(\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}})$ of

$$\begin{aligned} &\min_{\theta \in \mathbb{R}, \boldsymbol{\wp}^{\mathcal{P}}} (-\theta) \\ &\text{subject to} \quad g^{\mathcal{P}}((\theta, \boldsymbol{\wp}^{\mathcal{P}}), \boldsymbol{\alpha}) := \int_{\Omega} T(\omega, \boldsymbol{\alpha}) f^{\mathcal{P}}(\omega; \boldsymbol{\wp}^{\mathcal{P}}) d\omega - \theta \geqslant 0 \ \ (\boldsymbol{\alpha} \in A_k), \\ &\mathrm{u}_i^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) = 0 \ \ (i \in I^{\mathcal{P}}), \\ &\mathrm{v}_j^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) \geq 0 \ \ (j \in J^{\mathcal{P}}). \end{aligned}$$

4: **if** $g^{\mathcal{P}}((\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}}), \boldsymbol{\alpha}) \geq -\delta \ \ (\boldsymbol{\alpha} \in A)$ **then**

5: STOP

6: **else**

7: $A_{k+1} := A_k \cup \{\text{any new discretized points from } A\}$

8: $k := k + 1$

9: **end if**

10: **END DO**

**Algorithm 2 Dual Conceptual Discretization Method (DCDM)**

**Input:**

$\delta$ positive number, i.e., $\delta > 0$

$f^{\mathcal{D}}$ probability density function (pdf)

$P^{\mathcal{D}}$ the set where pdf parameters lie

**Output:**

$\sigma$ unknown variable for minimization, to be evaluated

$\wp^{\mathcal{D}}$ the parameter vector of the pdf

**DCDM**$\left(\sigma, \wp^{\mathcal{D}}, \Omega, \delta, f^{\mathcal{D}}, P^{\mathcal{D}}\right)$

1: $k := 0$

2: Initialize a discretization $\Omega_k \subset \Omega$.

3: **DO** Compute a solution $(\sigma_k, \wp_{\boldsymbol{k}}^{\mathcal{D}})$ of

$$\min_{\sigma \in \mathbb{R}; \wp^{\mathcal{D}}} \ \sigma$$

$$\text{subject to } \ g^{\mathcal{D}}((\sigma, \wp^{\mathcal{D}}), \omega) := \sigma - \int_A T(\omega, \boldsymbol{\alpha}) f^{\mathcal{D}}(\boldsymbol{\alpha}; \wp^{\mathcal{D}}) d\boldsymbol{\alpha} \geqslant 0 \ (\omega \in \Omega_k),$$

$$\mathrm{u}_i^{\mathcal{D}}(\wp^{\mathcal{D}}) = 0 \ \ (i \in I^{\mathcal{D}}),$$

$$\mathrm{v}_j^{\mathcal{D}}(\wp^{\mathcal{D}}) \geq 0 \ \ (j \in J^{\mathcal{D}}).$$

4: **if** $g^{\mathcal{D}}((\sigma_k, \wp_{\boldsymbol{k}}^{\mathcal{D}}), \omega) \geq -\delta \ \ (\omega \in \Omega)$ **then**

5: STOP

6: **else**

7: $\Omega_{k+1} := \Omega_k \cup \{\text{any new discretized points from } \Omega\}$

8: $k := k + 1$

9: **end if**

10: **END DO**

In Algorithm 1 and Algorithm 2, stopping criteria are *theoretically* established since one needs to check, e.g., $g((\sigma_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{D}}), \omega) \geq -\delta \ \ (\omega \in \Omega)$. Alternatively, we introduce some stopping criterion based on the idea of a *Cauchy* sequence.

Generally speaking, in our problem and many real-world situations, an optimal solution is not known. In order to stop at a sufficiently close approximately optimal solution, the increment between the steps have to be small enough, i.e., $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 < \epsilon_0$ for a fixed $\epsilon_0 > 0$ which comes from the definition of "Cauchy sequence" evaluated at the $k$-th iteration for a fixed $\epsilon_0 > 0$. A second alternative stopping criterion is based on the idea of a Cauchy sequence again, but on the value of the objective function, $F$; it is determined by looking at the decrement of the objective function at iterations by $(F(\mathbf{x}_k) - F(\mathbf{x}_{k+1})) < \epsilon_1$ for a fixed $\epsilon_1 > 0$. As a third alternative, the first and the second criteria are both integrated in a single criterion by $(F(\mathbf{x}_k) - F(\mathbf{x}_{k+1})) \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2^{-1} < \epsilon_2$ for a fixed $\epsilon_2 > 0$.

In our problems, the objective functions are $F^{\mathcal{P}}(\theta, \boldsymbol{\wp}^{\mathcal{P}}) := -\theta$ and $F^{\mathcal{D}}(\sigma, \boldsymbol{\wp}^{\mathcal{D}}) := \sigma$ for the primal and the dual problems, respectively. With this notion, we establish our stopping criteria in different forms. In the following, we refer to one of the stopping criteria for the primal and the dual problems, in the following ways:

**Stopping Criteria for the Primal Problem:**

$$\begin{aligned} \left\|(\theta_{k+1}, \boldsymbol{\wp}_{\boldsymbol{k+1}}^{\mathcal{P}}) - (\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{P}})\right\|_2 &< \epsilon_0 \text{ for a fixed } \epsilon_0 > 0, \\ \|-\theta_k - (-\theta_{k+1})\|_2 &< \epsilon_1 \text{ for a fixed } \epsilon_1 > 0, \\ (-\theta_k - (-\theta_{k+1})) \left\|(\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{P}}) - (\theta_{k+1}, \boldsymbol{\wp}_{\boldsymbol{k+1}}^{\mathcal{P}})\right\|_2^{-1} &< \epsilon_2 \text{ for a fixed } \epsilon_2 > 0. \end{aligned} \tag{6.7}$$

**Stopping Criteria for the Dual Problem:**

$$\begin{aligned} \left\|(\sigma_{k+1}, \boldsymbol{\wp}_{\boldsymbol{k+1}}^{\mathcal{D}}) - (\sigma_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{D}})\right\|_2 &< \epsilon_0 \text{ for a fixed } \epsilon_0 > 0, \\ \|\sigma_k - (\sigma_{k+1})\|_2 &< \epsilon_1 \text{ for a fixed } \epsilon_1 > 0, \\ (\sigma_k - (\sigma_{k+1})) \left\|(\sigma_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{D}}) - (\sigma_{k+1}, \boldsymbol{\wp}_{\boldsymbol{k+1}}^{\mathcal{D}})\right\|_2^{-1} &< \epsilon_2 \text{ for a fixed } \epsilon_2 > 0. \end{aligned} \tag{6.8}$$

Next, we will give an important assumption for the following Theorem 6.2.6.

**Assumption 1:** The feasible sets $\mathcal{M}^{\mathcal{P}}(A)$ and $\mathcal{M}^{\mathcal{D}}(\Omega)$ are compact.

**Closer Explanation 6.2.5** *In fact, our (feasible) sets satisfy compactness on the lower level but not on the upper level. Indeed, on the upper level, $\theta \in \mathbb{R}$ and $\sigma \in \mathbb{R}$ are unbounded for the primal problem (6.1) and the dual problem (6.2), respectively. Let us recall that we parametrized $\beta$ and $\rho$. We need* ***compact*** *feasible sets to have convergence of subsequences towards the optimal solution guaranteed, and also for the discretizability given in the following theorem. We encounter this problem by* ***transversally*** *intersecting the feasible set with sufficiently large transversal families of elementary geometrical sets (squares, boxes, cylinders or balls); this* ***compactification*** *is introduced in [64, 88].*

*In an implicitly defined way, this corresponds to the following feasible subset of the primal SIP with some nonnegative (semi-continuous) functions $\mathrm{G}^{\mathcal{P}}$:*

$$\begin{aligned}\mathcal{M}^{\mathcal{P}}_{comp}(A) := \{(\theta, \boldsymbol{\wp}^{\mathcal{P}}) \,\big|\, \theta \in \mathbb{R},\ g^{\mathcal{P}}((\theta, \boldsymbol{\wp}^{\mathcal{P}}), \boldsymbol{\alpha}) \geq 0\ \ (\boldsymbol{\alpha} \in A),\\ (g^{\mathcal{P}} - \mathrm{G}^{\mathcal{P}})((\theta, \boldsymbol{\wp}^{\mathcal{P}}), \boldsymbol{\alpha}) \leq 0\ \ (\boldsymbol{\alpha} \in A)\},\end{aligned} \tag{6.9}$$

*and to the following feasible subset of the dual SIP with some nonnegative semi-continuous function $\mathrm{G}^{\mathcal{D}}$:*

$$\begin{aligned}\mathcal{M}^{\mathcal{D}}_{comp}(\Omega) := \{(\sigma, \boldsymbol{\wp}^{\mathcal{D}}) \,\big|\, \sigma \in \mathbb{R},\ g^{\mathcal{D}}((\sigma, \boldsymbol{\wp}^{\mathcal{D}}), \omega) \geq 0\ \ (\omega \in \Omega),\\ (g^{\mathcal{D}} - \mathrm{G}^{\mathcal{D}})((\sigma, \boldsymbol{\wp}^{\mathcal{D}}), \omega) \leq 0\ \ (\omega \in \Omega)\},\end{aligned} \tag{6.10}$$

*where $g^{\mathcal{P}}((\theta, \boldsymbol{\wp}^{\mathcal{P}}), \boldsymbol{\alpha})$ and $g^{\mathcal{D}}((\sigma, \boldsymbol{\wp}^{\mathcal{D}}), \omega)$ denote the inequality constraint functions of the primal and the dual problems, respectively. We note that the latter functions may also be vector-valued.*

*Besides of this theoretical approach by functions $\mathrm{G}^{\mathcal{P}}$ and $\mathrm{G}^{\mathcal{D}}$, a more practical one consists of the idea of transversally cutting with a cube. This can be geometrically illustrated by the cube in Figure 6.2:*

**Remark 7** *When performing the transversal sections, it is important to take into account any given a priori information about where a possible global solution, minimizer or maximizer, of our regarded optimization problem is located. Let us recall that we look at the primal and dual problems after parametrization, such that the parameters themselves became new decision variables. So we would choose the intersecting parallelpipe large enough in order*

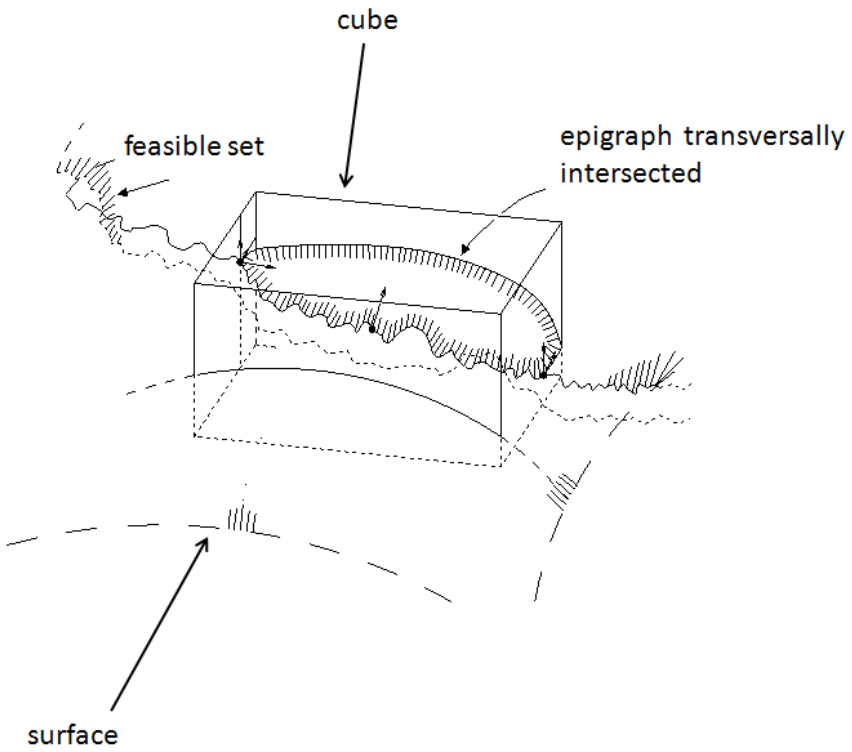


Figure 6.2: Transversal intersection (excision) of the feasible set with box; an example taken from [64, 88]. (The surface may come from an equality constraint; the figure implies perturbational arguments of [88].)

*to include such an expected global solution. Of course, to gain that* a priori *knowledge, a careful analytical investigation may be helpful and should be done, e.g., in terms of growth behaviour and convexity kind of properties. In fact, for the ease of exposition, we just think of minimization rather than both minimization and maximization.*

*As a first, simple but important class of problems we mention such ones with a strictly convex graph (given by the constraints), i.e., an epigraph with the form of a potential, e.g., a paraboloid. In any such a case, we know that the lower level set with respect some arbitrary and sufficiently large level is nonempty and compact. Then, we can choose and raise our transversally cutting parallelpipe so that, in a projective sense, the lower level set and, hence, as an element, the global minimizer is contained in the parallelpipe and, therefore, in the excised subset of the epigraph.*

*This treatment and careful geometrical arrangement guarantees the equality of set of minimizers of the original problem, $\mathcal{G}^{\mathcal{P}}(A)$, and the set of minimizers after compactification, $\mathcal{G}^{\mathcal{P}}_{comp}(A)$,*

*i.e.,* $\mathcal{G}^{\mathcal{P}}(A) = \mathcal{G}^{\mathcal{P}}_{comp}(A)$ *which is illustrated in Figure 6.3.*

*Let us underline that our strict convexity is not guaranteed in general. In fact, the fulfillment of this property on the one hand depends on how the kernel functions were chosen and how the kernel matrices, evaluated at the input data, are conditioned. On the other hand, it depends on how the parameters are involved into the density functions, how the possible nonlinearity can be characterized by convexity and growth kinds of conditions, e.g., in terms of Morse indices [37].*

We can adopt this idea to our problem to transversally cut around of our height function the boundary of the epigraph with a cube as shown by Figure 6.3.

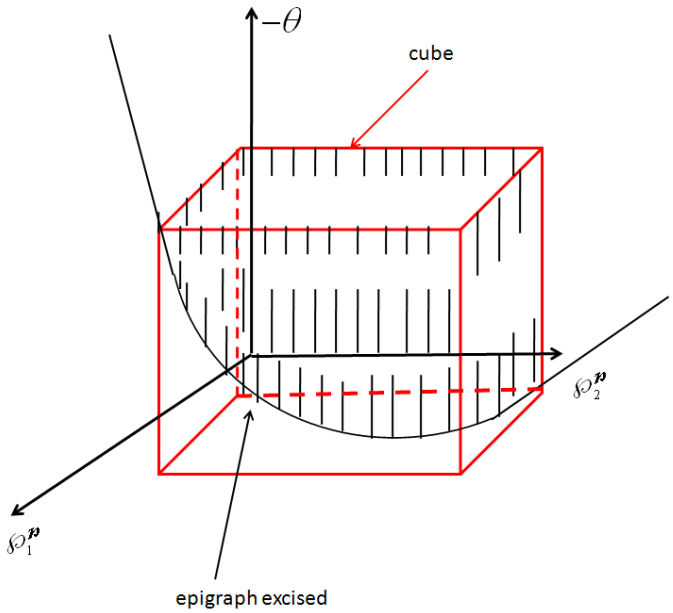


Figure 6.3: Illustration of the transversal cutting around the height function with a box; an example.

Under our Closer Explanation 6.2.5, we obtain a general convergence result for this method based on Theorem 13 in [78].

**Theorem 6.2.6** *Let Assumption 1, after the compactification introduced in Closer Explanation 6.2.5 be satisfied for the dual, and let the primal problems and the sequences of discretizations* $(A_k)_{k\in\mathbb{N}_0}$ *and* $(\Omega_k)_{k\in\mathbb{N}_0}$ *satisfy*

$$A_0 \subset A_k \;\; (k \in \mathbb{N}_0) \;\; \textit{and} \;\; d_1^{\mathcal{P}}(A_k, A) \to 0 \;\; \textit{for} \; k \to \infty$$

*and*

$$\Omega_0 \subset \Omega_k \ \ (k \in \mathbb{N}_0) \ \text{ and } \ d_1^{\mathcal{D}}(\Omega_k, \Omega) \to 0 \ \text{ for } k \to \infty.$$

*Based on possible compactifications, we may from now on suppose that $\mathcal{M}(A_0)$ and $\mathcal{M}(\Omega_0)$ are compact. Then, the primal and the dual problems, (5.18), (5.11), respectively, are* ***discretizable****, i.e., the problems $P(A_k)$ $(k \in \mathbb{N}_0)$ and $D(\Omega_k)$ $(k \in \mathbb{N}_0)$ have solutions $(\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{P}})$, $(\sigma_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{D}})$ $(k \in \mathbb{N}_0)$, respectively, and such sequences of iterative solutions satisfy*

$$\min_{(\theta^*, \boldsymbol{\wp}^{\mathcal{P}^*}) \in \mathcal{G}^{\mathcal{P}}(A)} \left\| (\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{P}}) - (\theta^*, \boldsymbol{\wp}^{\mathcal{P}^*}) \right\|_2 \to 0 \ \ (k \to \infty) \tag{6.11}$$

*and*

$$\min_{(\sigma^*, \boldsymbol{\wp}^{\mathcal{D}^*}) \in \mathcal{G}^{\mathcal{D}}(\Omega)} \left\| (\sigma_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{D}}) - (\sigma^*, \boldsymbol{\wp}^{\mathcal{D}^*}) \right\|_2 \to 0 \ \ (k \to \infty). \tag{6.12}$$

We refer to [78] for the proof of Theorem 6.2.6. By Theorem 6.2.6, we guarantee the convergence of approximate solutions to optimal solutions for sufficiently large $k$ with (6.11) and (6.12).

**Closer Explanation 6.2.7** *We note that the assumptions of the Theorem 6.2.6 must be satisfied before we discretize our infinite index set. We know that our index sets $A$ and $\Omega$ are compact, and we assume that our sequences of discretized sets $A_k$ and $\Omega_k$ $(k \in \mathbb{N}_0)$ converge to $A$ and $\Omega$. Then, our semi-infinite problems are discretizable.*

*We also note that the minima which are stated in the theorem exist since the Euclidean norm is continuous and bounded from below and, indeed, always nonnegative. Other properties used here are the existence of optimal solutions $(\theta^*, \boldsymbol{\wp}^{\mathcal{P}^*})$ and $(\sigma^*, \boldsymbol{\wp}^{\mathcal{D}^*})$, i.e., the sets of minimizers $\mathcal{G}^{\mathcal{P}}(A)$ and $\mathcal{G}^{\mathcal{D}}(\Omega)$ exist for the primal and the dual problems, respectively, since our feasible sets are compact and the objective functions are continuous, that the sets $\mathcal{G}^{\mathcal{P}}$ and $\mathcal{G}^{\mathcal{D}}$ are compact, too, and we use Theorem of Weierstrass (see [4]).*

Next, we give the definition for the *local* primal and the *local* dual problems which are defined around some open neighbourhoods of the local minimizers.

**Definition 6.2.8** *[78]. Given local minimizers $(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}})$, $(\bar{\sigma}, \bar{\boldsymbol{\wp}}^{\mathcal{D}})$ of the primal and the dual problems, (6.1) and (6.2), respectively, the primal and the dual SIP are called* ***locally discretizable*** *at $(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}})$ and $(\bar{\sigma}, \bar{\boldsymbol{\wp}}^{\mathcal{D}})$ if the discretizability relation holds locally, i.e., if there*

*exist neighbourhoods* $U_{(\bar{\theta},\bar{\boldsymbol{\wp}}^{\mathcal{P}})}$, $V_{(\bar{\theta},\bar{\boldsymbol{\wp}}^{\mathcal{D}})}$ *of* $(\bar{\theta},\bar{\boldsymbol{\wp}}^{\mathcal{P}})$ *and* $(\bar{\theta},\bar{\boldsymbol{\wp}}^{\mathcal{D}})$*, respectively, such that the locally discretized problems* $P^{loc}(A)$ *and* $D^{loc}(\Omega)$ *for the primal and the dual problem, respectively, namely,*

$$\begin{aligned} P^{loc}(A): \quad & \min_{(\theta,\boldsymbol{\wp}^{\mathcal{P}})\in U_{(\bar{\theta},\bar{\boldsymbol{\wp}}^{\mathcal{P}})}} -\theta \\ & \text{subject to} \quad \textstyle\int_\Omega T(\omega,\boldsymbol{\alpha}) f^{\mathcal{P}}(\omega,\boldsymbol{\wp}^{\mathcal{P}})d\omega - \theta \geqslant 0 \;\; (\boldsymbol{\alpha}\in A), \\ & \qquad\qquad \mathrm{u}_i^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) = 0 \;\; (i\in I^{\mathcal{P}}), \\ & \qquad\qquad \mathrm{v}_j^{\mathcal{P}}(\boldsymbol{\wp}^{\mathcal{P}}) \geq 0 \;\; (j\in J^{\mathcal{P}}), \end{aligned}$$

*and*

$$\begin{aligned} D^{loc}(\Omega): \quad & \min_{(\sigma,\boldsymbol{\wp}^{\mathcal{D}})\in V_{(\bar{\sigma};\bar{\boldsymbol{\wp}}^{\mathcal{D}})}} \sigma \\ & \text{subject to} \quad \sigma - \textstyle\int_A T(\omega,\boldsymbol{\alpha}) f^{\mathcal{D}}(\omega;\boldsymbol{\wp}^{\mathcal{D}})d\boldsymbol{\alpha} \leqslant 0 \;(\omega\in\Omega), \\ & \qquad\qquad \mathrm{u}_i^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) = 0 \;\; (i\in I^{\mathcal{D}}), \\ & \qquad\qquad \mathrm{v}_j^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) \geq 0 \;\; (j\in J^{\mathcal{D}}), \end{aligned} \tag{6.13}$$

*obtained as the restriction of* $P(A)$ *and* $D(\Omega)$ *to open neighborhoods* $U_{(\bar{\theta},\bar{\wp}^{\mathcal{P}})}$ *and* $V_{(\bar{\sigma},\bar{\wp}^{\mathcal{D}})}$*, respectively, are discretizable.*

The following Theorem 6.2.10 is based on [78], Theorem 15, and it gives a convergence result for the discretization method applied to our problems. Let us recall the definition of a local minimum of order $p$ before giving the result.

**Definition 6.2.9** *A feasible point* $\bar{\mathbf{x}}$ *is called a local minimizer of order* $p>0$ *of the problem to minimize* $f(\mathbf{x})$ *on a feasible set* $\mathcal{M}\subseteq\mathbb{R}^n$ *if with suitable constants* $\epsilon>0$, $M>0$ *the following relation holds:*

$$f(\mathbf{x}) - f(\bar{\mathbf{x}}) \geq M\,\|\mathbf{x}-\bar{\mathbf{x}}\|_2^p \quad \text{for all } \mathbf{x} \text{ with } \|\mathbf{x}-\bar{\mathbf{x}}\|_2 < \epsilon.$$

**Theorem 6.2.10** *Let* $(\bar{\theta},\bar{\boldsymbol{\wp}}^{\mathcal{P}})$ *and* $(\bar{\sigma},\bar{\boldsymbol{\wp}}^{\mathcal{D}})$ *be a local minimizer of the primal problem (6.1) and the dual problem (6.2), respectively, of order p, and let sets* $\mathcal{M}(A_k)$, $\mathcal{M}(A)$, $\mathcal{M}(\Omega_k)$ *and* $\mathcal{M}(\Omega)$ *be restricted to a compact subset* $K\subset\mathbb{R}^n$*. We further suppose that MFCQ (see Section 2.4.3) holds at* $(\bar{\theta},\bar{\boldsymbol{\wp}}^{\mathcal{P}})$ *and* $(\bar{\sigma},\bar{\boldsymbol{\wp}}^{\mathcal{D}})$*. Then (6.1) and (6.2) are locally discretizable at* $(\bar{\theta},\bar{\boldsymbol{\wp}}^{\mathcal{P}})$ *and* $(\bar{\sigma},\bar{\boldsymbol{\wp}}^{\mathcal{D}})$*, respectively. In closer detail: There are some* $\varsigma^{\mathcal{P}}>0$ *and* $\varsigma^{\mathcal{D}}>0$ *such that for any sequences of grids* $(A_k)_{k\in\mathbb{N}_0}\subset A$, $(\Omega_k)_{k\in\mathbb{N}_0}\subset\Omega$ *with* $d_1^{\mathcal{P}}(A_k,A)\to 0$ *and* $d_1^{\mathcal{D}}(\Omega_k,\Omega)\to$

$0$ $(k \to \infty)$, *respectively, and for any sequences of solutions* $(\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{P}})_{k\in\mathbb{N}_0}$ *and* $(\sigma_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{D}})_{k\in\mathbb{N}_0}$ *of the locally restricted problems* $P^{loc}(A)$ *and* $D^{loc}(\Omega)$*, the following relations hold:*

$$\left\|(\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{P}}) - (\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}})\right\|_2 \leq \varsigma^{\mathcal{P}} d_1^{\mathcal{P}}(A_k)^{1/p} \quad (k \to \infty)$$

*and*

$$\left\|(\sigma_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{D}}) - (\bar{\sigma}, \bar{\boldsymbol{\wp}}^{\mathcal{D}})\right\|_2 \leq \varsigma^{\mathcal{D}} d_1^{\mathcal{P}}(\Omega_k)^{1/p} \quad (k \to \infty).$$

**Closer Explanation 6.2.11** *The result of Theorem 6.2.10 is true for the global minimization problems (6.1) and (6.2) since the sets* $\mathcal{M}(A_k)$*,* $\mathcal{M}(\Omega_k)$*,* $\mathcal{M}(A)$ *and* $\mathcal{M}(\Omega)$ *are restricted to a compact subset [78]. We note that after compactification by transversally intersecting the feasible set (see Closer Explanation 6.2.5) with sufficiently large transversal elementary geometrical sets, we satisfy the compactness assumption for Theorem 6.2.10.*

*Let us observe that both sets A and* $\Omega$ *are compact. We recall that the discretization of* $\Omega$ *may simply be a one-dimensional grid, and the elements of the discretized set of A may consist of a combination of its corner points, which will be explained later in this section. All the discretized sets are further refined based on the previous sets, i.e.,* $A_k \subset A_{k+1}$ $(k \in \mathbb{N}_0)$*. The refinement of the following iterations depends on the type and the dimension of the set. For example, if the index set Y is an interval* $\Omega := [a, b]$*, then a one-dimensional grid* $\hat{Y}$ *can be chosen such that the distance between neighbouring grid points is defined by* $\Delta y_i := \frac{b-a}{k_0}$ $(i = 0, 1, \ldots, k_0)$ *for some* $k_0 \in \mathbb{N}$*, and with the grid* $\hat{Y} := \{y_i \in [a, b] \,|\, y_i = a + i\Delta y,\ i = 0, 1, \ldots, k_0\}$*. We can refine* $\hat{Y}$ *by updating* $k_0$ *such that* $k_1 = k_0 + 1$*.*

Until now, we have provided theorems which guarantee convergence of the discretization method under some assumptions. If the dimension of the continuous index variable is larger than 2, then the computational complexity of the discretization grows exponentially. In fact, we need an $(l-1)$-dimensional grid of the index set. For example, we use a grid of $[0, C]^l$ for the vector $\boldsymbol{\alpha}$ in our primal problem for the discretization of the index set $A$. The size of the mesh grows fastly as the dimension, $l$, increases. In closer detail: For our primal problem (6.1), the infinite index variable $\boldsymbol{\alpha}$ is lying in an $l$-dimensional underlying space. Moreover, the dimension of the elements in $A$ is the same as the number of the training points used in our SVM which forces the index variable to be in a high dimension as the number of the training points increases. This makes the discretization algorithmically more difficult. Let us observe

that the set $A$ is an $(l-1)$-dimensional *polytope*, indeed, it is the intersection of the hyperplane $\sum_{i=1}^{l} \alpha_i y_i = 0$ with the box constraints $0 \leq \alpha_i \leq C \quad (i = 1, 2, \ldots, l)$, as we learn from the definition of $A$.

We propose two strategies to find a discretization of the set $A$. The first Strategy I is based on an interpretation of the set $A$ by the combination of its corner points. In this way, we can discretize the standard simplex instead of the set $A$ directly. The second Strategy II is based on the linearization of the set $A$, which is established on theoretical foundations [88].

***Strategy I*** [51] (**Triangulation**):
In this first strategy, we use Lemma of ***Carathedory*** given to represent the elements of $A$ by its corner points. Furthermore, we apply a triangulation for some standard simplex $\Delta^N$ and, hence, a discretization of $A$ will be inherited via $\Delta^N$. To do this, we transform the polytope $A$ to the standard simplex and doing a normalization by representing the coordinates of $A$ with its barycentric coordinates. After Example 6.2.14, we will explain how the triangulation is refined stepwise in an algorithmic way. Let us define the standard simplex and the relation with barycentric coordinates:

**Definition 6.2.12** *For any $N \in \mathbb{N}_0$, let the **standard $N$-simplex** (or unit $N$-simplex) is the subset of $\mathbb{R}^{N+1}$ be given by*

$$\Delta^N := \left\{ \boldsymbol{a} \in \mathbb{R}^{N+1} \,\middle|\, a_i \geq 0 \;\; (i = 1, 2, \ldots, N+1), \; \sum_{i=1}^{N+1} a_i = 1 \right\}.$$

*The simplex $\Delta^N$ is lying in the affine hyperplane obtained by removing the restrictions $a_i \geq 0$ $(i = 1, 2, \ldots, N+1)$ in the above definition.*

*The vertices of the standard N-simplex are the standard unit-vectors (points)*

$$\begin{aligned} \boldsymbol{e_0} &= (1, 0, 0, \ldots, 0)^T, \\ \boldsymbol{e_1} &= (0, 1, 0, \ldots, 0)^T, \\ &\vdots \\ \boldsymbol{e_N} &= (0, 0, 0, \ldots, 1)^T. \end{aligned}$$

*There is a canonical map from the standard N-simplex to an arbitrary N-simplex (polytope)*

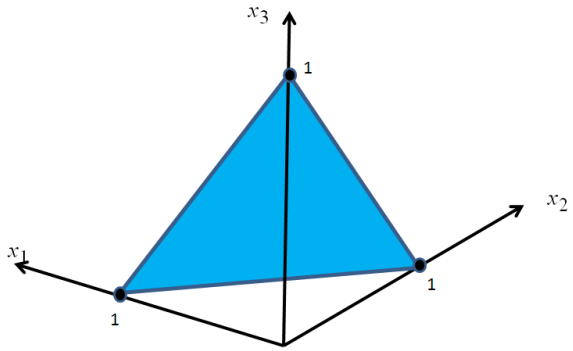


Figure 6.4: Illustration of the 2-simplex in $\mathbb{R}^3$.

$\hat{\Delta}^N$ *with vertices* $\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_N}$*, given by*

$$\boldsymbol{a} \mapsto \hat{\boldsymbol{a}} := \sum_{i=1}^{N+1} a_i \boldsymbol{v_i} \quad (\boldsymbol{a} = (a_1, a_2, \ldots, a_{N+1})^T \in \Delta^N).$$

*The coefficients* $a_i$ *are called the* ***barycentric coordinates*** *of a point* $\hat{\boldsymbol{a}}$ *in the* $N$*-simplex* $\hat{\Delta}^N$ $(i = 1, 2, \ldots, N+1)$*. The standard 2-simplex in* $\mathbb{R}^3$ *is illustrated in Figure 6.4.*

**Closer Explanation 6.2.13** *In order to apply the canonical mapping with barycentric coordinates, we assume* $A = \hat{\Delta}^N$*,* $N+1$ *is the number of vertices of* $A$ *and all vertices of* $A$ *have entries never different from 0 and 1. Then, we can benefit from representing the points* $\boldsymbol{\alpha} \in A$ *by its barycentric coordinates and by the vertices of standard simplex or, a bit weaker and as we will use below, from that we may assume all components* $\alpha_i$ $(i = 1, 2, \ldots, l)$ *to be* 0 *or* 1*, respectively.*

Let us fix $y_i \in \{\pm 1\}$ $(i = 1, 2, \ldots, l)$ being the output data (labels) and recall the index set $A = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^l \,\middle|\, 0 \leq \alpha_i \leq C \;\; (i = 1, 2, \ldots, l) \text{ and } \sum_{i=1}^{l} \alpha_i y_i = 0 \right\}$.

Without loss of generality, we assume that there is some $i_0 \in \{1, 2, \ldots, l-1\}$ such that $y_1 = \ldots = y_{i_0} = 1$ and $y_{i_0+1} = \ldots = y_l = -1$. Furthermore, as prepared in our Closer Explanation 6.2.13 for simplicity, we take $C = 1$ for this strategy. (We could also choose $C$ different than 1; in fact, we can apply the same procedure below.) Since $\sum_{i=1}^{l} \alpha_i y_i = 0$, we have the following equation from the definition of the set $A$:

$$\alpha_1 + \ldots + \alpha_{i_0} = \alpha_{i_{0+1}} + \ldots + \alpha_l, \tag{6.14}$$

where $\alpha_i \in \{0,1\}$ $(i \in \{1,2,\ldots,l\})$. Specifically, the trivial solution to the equation (6.14) is a vertex of our polytope $A$. By this intuition, we will consider the elements of polytope $A$ by the combination of its binary vertices.

**Remark 8** *The polytope $A$ has finitely many corner points. In particular, let $r := \min\{i_0, l - i_0\}$. Then, $A$ has $\sum_{i=0}^{r} \binom{i_0}{r}\binom{l-i_0}{r}$ corner points.*

**Example 6.2.14** *Let $l = 6,\ y_1 = y_2 = 1,\ y_3 = \ldots = y_6 = -1$. Then,*

$$\alpha_1 + \alpha_2 = \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6. \tag{6.15}$$

*There are 15 different corner points. The trivial one is $(0,0,\ldots,0)^T$, which corresponds to the number $\binom{2}{0}\binom{4}{0} = 1$.*

*We observe that we must have corner points with 2 nonzero elements or 4 nonzero elements to satisfy equation (6.15). Let us start with the corners having 2 nonzero elements:*

$$(1,0,1,0,0,0),\ (1,0,0,1,0,0), \tag{6.16}$$

$$(1,0,0,0,1,0),\ (1,0,0,0,0,1), \tag{6.17}$$

$$(0,1,1,0,0,0),\ldots,(0,1,0,0,0,1), \tag{6.18}$$

$$(1,1,1,1,0,0),\ldots,(1,1,0,0,1,1), \tag{6.19}$$

*where (6.16), (6.17) and (6.18) represent $\binom{2}{1}\binom{4}{1} = 8$ many points, and (6.19) corresponds to $\binom{2}{2}\binom{4}{2} = 6$ many ones. Then, the total number of corner points is $1 + 2 \cdot 4 + 1 \cdot 6 = 15$.*

**Algorithmic Way to Find all Vertices (or Corner Points) of $A$:**

Let $\boldsymbol{p} \in A$ be any point. Indeed, for the ease and completeness of explanation, we may assume that $\boldsymbol{p}$ is an interior point of $A$, especially, not a corner point. Now, we choose a line $d$ through $\boldsymbol{p}$ in $A$. We take two points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ on $d$ which lie on the opposite sides of $\boldsymbol{p}$ and maximize the distance to $\boldsymbol{p}$. Then, $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ must be on some hypersurfaces (hyperfaces) bounding the convex region $A$. Next, choose a line $d_2$ through $\boldsymbol{q_1}$ which lies in the hypersurface containing $\boldsymbol{q_1}$. This line will intersect this face into two parts. The face has one more *codimension* (one less dimension). The point $\boldsymbol{q_1}$ is a convex combination of the two new intersection points. Continuing this way finishes the construction principle.

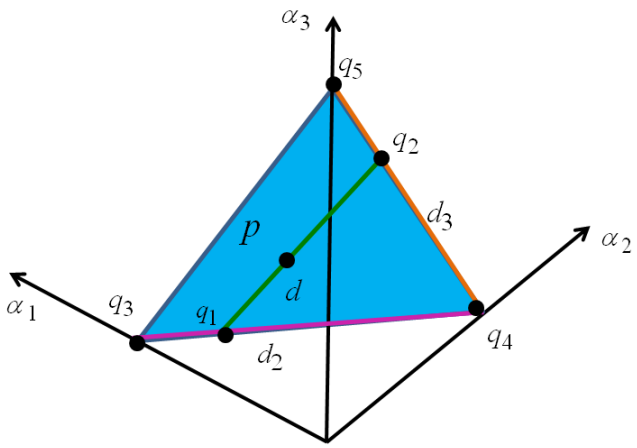


Figure 6.5: Illustration of the algorithmic way of finding corner points of $A$; an easy example for $l = 3$ (in $\mathbb{R}^3$).

We illustrate the intuition of this algorithmic way of finding corners of polytope $A$ with Figure 6.5. Obviously, $\boldsymbol{p}$ is a convex combination of $\boldsymbol{q_3}$, $\boldsymbol{q_4}$ and $\boldsymbol{q_5}$, the vertices of $A$.

Now, let $N := \sum_{i=0}^{r} \binom{i_0}{r}\binom{l-i_0}{r}$. Then, we can discretize the *standard simplex* in $\mathbb{R}^{N+1}$ and finally map it onto $A$ to discretize $A$. More formally, we firstly recall Definition 6.2.12,

$$\Delta^N = \left\{ \boldsymbol{a} \in \mathbb{R}^{N+1} \;\middle|\; a_i \geq 0 \;\; (i = 1, 2, \ldots, N+1), \; \sum_{i=1}^{N+1} a_i = 1 \right\}. \tag{6.20}$$

Let us define a mapping

$$T : \Delta^N \longrightarrow A \text{ with } \; T(\boldsymbol{a}) := \sum_{i=1}^{N+1} a_i \boldsymbol{v_i} \in A \;\; (\boldsymbol{a} = (a_1, a_2, \ldots, a_{N+1})^T \in \Delta^N),$$

where the set $\{\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_{N+1}}\}$ consists of the vertices of $A$. By this methodology, we can find the elements of this discretization $A_k$ of $A$ which are represented by a combination of vertices of the simplex. This can be mathematically formulated as follows. Any point $\boldsymbol{p} \in A$ can be represented by

$$\boldsymbol{p} = \sum_{i=1}^{N+1} a_i \boldsymbol{v_i}, \tag{6.21}$$

where the set $\{\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_{N+1}}\}$ is the set of vertices of $A$ and $a_i$ $(i = 1, 2, \ldots, N+1)$ are the barycentric coordinates for $A$ (see Definition 6.2.12). To be able to write a point $\boldsymbol{p}$ from $A$ as in (6.21), we need to find the coordinates $a_i$ $(i = 1, 2, \ldots, N+1)$ from the standard $N$-simplex. Hence, the simplex $\Delta^N$ has to be discretized.

One of the main advantages of this strategy consists in working with the standard simplex and

its vertices. However, the discretization of the simplex is **not** uniform because of the unsymmetries of the grid points. As it is clear from Figure 6.6, the distances of the neighbouring mesh points are nonuniform, i.e., $\Delta_1 \neq \Delta_2 \neq \Delta_3 \neq \Delta_4$.

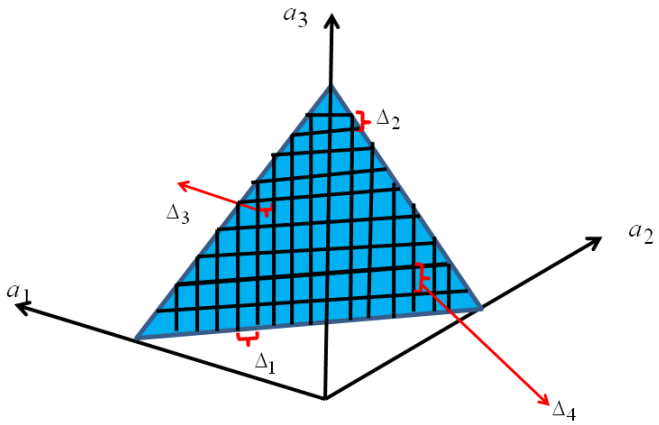


Figure 6.6: Nonuniform sampling of a standard simplex $\Delta^N$, an example in $\mathbb{R}^3$, $\Delta_1 \neq \Delta_2 \neq \Delta_3 \neq \Delta_4$.

In order to overcome nonuniformity, we propose a method which transforms the barycentric coordinates of polytopes to a sphere as shown by Figure 6.7 (for closer information, see [91]). Let us consider a particular face $F$ of some polytope and its corresponding spherical face $F'$ as shown in Figure 6.7. Each point in $F$ can be described by barycentric coordinate systems induced by vertices of $F$ after the triangulation as given above. Let us assume that we create a distribution of points inside $F$. We can obtain each of the points in this distribution by a linear interpolation between the vertices of our barycentric coordinates system. Similarly, the distribution on $F'$ can be obtained through the same steps of interpolation between the vertices of barycentric coordinate systems on the sphere [91]. Since we have a uniform sampling over a sphere (see Figure 6.8), we achieve a uniformly discretized points of our polytope $A$.

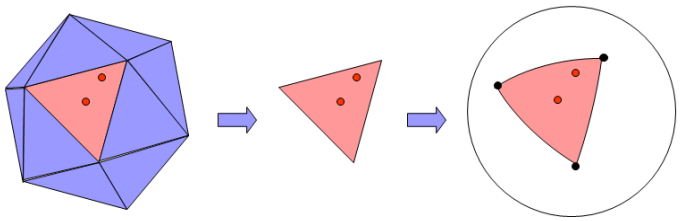

Figure 6.7: Transformation of the barycentric coordinates of a polytope to a sphere [91].

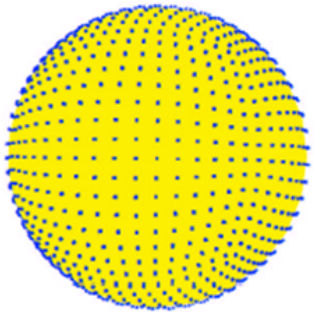

Figure 6.8: Discretization of the sphere; an example [91].

**Remark 9** *It is important to observe the computational intractability of Strategy I because of the exponential growth of the corner points, as the dimension of $\boldsymbol{\alpha}$, i.e., the number of data points, increases. It is clear from the example that the number of binary vectors grows exponentially, namely, in the way of $2^l$ which makes the algorithm impractical. We offer a theoretically prepared second strategy below which is based on a linearization procedure.*

Next, we propose a second strategy which is more theoretical.

***Strategy II*** (**Linearization**):
The second strategy is based on the linearization of $A$ in some neighbourhood $U_{(\bar{\theta},\bar{\boldsymbol{\varphi}}^{\mathcal{P}})}$, locally around a given point $\bar{\boldsymbol{\alpha}} \in A$, e.g., a vertex of $A$. [88]. Mathematically, we define $\boldsymbol{z} = \hat{T}(\boldsymbol{\alpha})$ as follows:

$$\hat{T} : \begin{cases} z_1 & := u(\boldsymbol{\alpha}), \\ z_2 & := v_{\ell_1}(\boldsymbol{\alpha}), \\ & \vdots \\ z_{k+1} & := v_{\ell_k}(\boldsymbol{\alpha}), \\ z_{k+2} & := \boldsymbol{\zeta_1}^T(\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}), \\ & \vdots \\ z_l & := \boldsymbol{\zeta_{l-1-k}}^T(\boldsymbol{\alpha} - \bar{\boldsymbol{\alpha}}), \end{cases} \tag{6.22}$$

where $k$ is the cardinality $|L_0(\bar{\boldsymbol{\alpha}})|$ of $L_0(\bar{\boldsymbol{\alpha}}) = \{\ell_1, \ell_2, \ldots, \ell_k\}$ and the vectors $\boldsymbol{\zeta_\nu} \in \mathbb{R}^l$ $(\nu = 1, 2, \ldots, l-1-k)$ complete the set $\{\nabla u(\bar{\boldsymbol{\alpha}})\} \cup \{\nabla v_\ell(\bar{\boldsymbol{\alpha}}) \,|\, \ell \in L_0(\bar{\boldsymbol{\alpha}})\}$ to a basis of $\mathbb{R}^l$.

Now, let us assume that the LICQ condition is satisfied for the lower level problem of (6.1). Here, we refer to our analysis from Subsection 5.3.3, including the perturbation theory (if

needed) as being presented there. Then, by means of Inverse Function Theorem applied at $\bar{\boldsymbol{\alpha}}$ on $\hat{T}$, we conclude that there exist open and bounded neighbourhoods $U^1 \subseteq \mathbb{R}^\iota$, $U^2 \subseteq \mathbb{R}^l$ around $((\bar{\theta}, \bar{\boldsymbol{\wp}}^{\boldsymbol{\mathcal{P}}}), \bar{\alpha})$ such that $T := \hat{T}_{|U^1 \times U^2} : U^1 \times U^2 \to \mathcal{W} := \hat{T}(U^1 \times U^2)$ is a $C^1$-diffeomorphism. Shrinking $U^1$, we can guarantee that $\mathcal{W}$ is an axis parallel open box around $((\bar{\theta}, \bar{\boldsymbol{\wp}}^{\boldsymbol{\mathcal{P}}}), \mathbf{0}_l) \in \mathbb{R}^\iota \times \mathbb{R}^l$. Then, for each $(\theta, \boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}}) \in U^1$, the mapping $\Phi_{(\theta, \boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}})} := \left(\hat{T}((\theta, \boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}}), \cdot)\right)_{|U^2} : U^2 \to S^2$ is a $C^1$-diffeomorphism which transforms the (relative) neighbourhood $A \cap U^2$ of $\bar{\boldsymbol{\alpha}}$ on the (relative) neighbourhood

$$(\{\mathbf{0}\} \times \mathbb{H}^k \times \mathbb{R}^{l-1-k}) \cap S^2 \subseteq \mathbb{R}^l$$

of $\mathbf{0}$, where $S^2 = S(\mathbf{0}, \delta)$ stands for the open square around $\mathbf{0} = \mathbf{0}_l$ with a half side of length $\delta$. Here, $\mathbb{H}^k$ denotes the nonnegative orthant of $\mathbb{R}^k$:

$$\mathbb{H}^k := \{z \in \mathbb{R}^k \mid \ z_\ell \geq 0 \ \ (\ell \in \{1, 2, \ldots, k)\}.$$

We call $\Phi_{(\theta, \boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}})}$ a *canonical local change of coordinates* of $\boldsymbol{\alpha}$. By this strategy, we transformed $A$ into a locally rectangular manifold with corners and edges where the discretization will takes place in. More generally, a discretization point $\boldsymbol{z}$ from the discretized set (*regular grid*) $\mathbb{H}^k$ corresponds to a discretization point

$$\boldsymbol{\alpha} = \hat{T}^{-1}(\boldsymbol{z}) \tag{6.23}$$

from the set $A$ by the back transformation $\hat{T}^{-1}$, implicitly represented in (6.24):

$$\hat{T}^{-1} : \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{k+1} \\ \alpha_{k+2} \\ \vdots \\ \alpha_l \end{bmatrix} := \hat{T}^{-1} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{k+1} \\ z_{k+2} \\ \vdots \\ z_l \end{bmatrix}. \tag{6.24}$$

The geometric illustration is shown in Figure 6.9. The details of this method can be found in [88].

In the case of our problem, $A$ is already given by linear equalities and inequalities. For this reason, we can perform the linearization more easily. Indeed, we go from any vertex $\bar{\boldsymbol{\alpha}}$ to

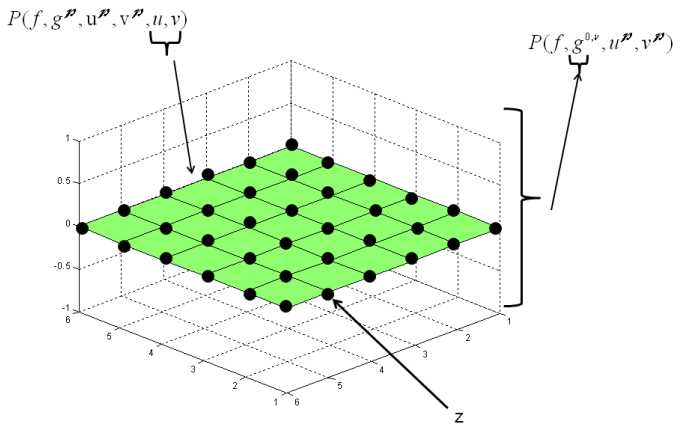


Figure 6.9: Illustration of the local discretization in $\mathbb{H}^k$, $P(f, g^{0,\nu}, u^{\mathcal{P}}, v^{\mathcal{P}})$ is the discretized problem and $P(f, g^{\mathcal{P}}, u^{\mathcal{P}}, v^{\mathcal{P}}, u, v)$ is the primal SIP problem, where $\nu$ is the number of grid points, $f$ is the objective function and $g^{\mathcal{P}}$ is the inequality constraint of the SIP problem; an example [88].

the neighbouring vertices and, by this, find a relative neighbourhood of $\boldsymbol{\alpha}$ in $A$ of "triangular" shape (cf. Figure 6.10). Herewith, we obtained a linearization, but we do not have guaranteed $90^0$ inscribed at $\bar{\boldsymbol{\alpha}}$ which, however, could be achieved by the transformation described above (if being wished).

**Note 6.2.15** *Strategy II is more theoretical, but we can perform it more practically: it aims at finding how to compute "local" (neighbourhoods). In our problems, u and v are linear, so that the transformation $\hat{T}$ is linear and that inverse transformation, $\hat{T}^{-1}$ is linear, too. However, since A has the special form of a polytope, one can use the neighbourhoods by the (relative) interiors of sub-polytopes (generated by neigbouring vertices), as being shown in Figure 6.10. If we do that for all vertices $\bar{\boldsymbol{\alpha}}$, then only* ***interior*** *points remain, which constitute an* ***(interior sub-) polyhedron*** *which is often relatively small, especially, if the number of vertices is not too high. This interior sub-polyhedron is shown by the shaded region in Figure 6.10. With this sub-polyhedron, we could proceed in our way again, and we continue, until the sub-polyhedron remaining is small enough, indeed. Now, all subdividing sets can be discretized by some scheme (e.g., by some canonical grids in them or by a uniform sampling on a sphere after transforming barycentric coordinates inside of the sub-polyhedron).*

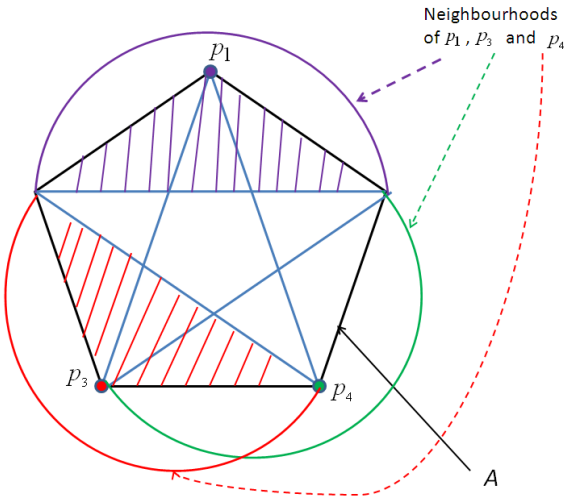


Figure 6.10: Illustration of the (local) linearization of $A$, with linear $u$ and $v$.

## 6.3 EXCHANGE METHOD

Another concept which is more powerful than discretization is the *exchange method*. It is, in terms of refinement and complexity of the algorithm, located between discretization and the reduction ansatz. Given a discretization, the reduced problem (5.20) of SIP is solved, and in a next iteration, discretization points become updated, until the algorithm terminates according to some stopping criterion. The adapted exchange algorithm to our primal problem is given by Algorithm 3.

As it is discussed in Subsection 6.2.1 by given stopping criteria (6.7) and (6.8) for the primal and the dual problems, respectively, we can use anyone of our alternative stopping criteria.

In this section, we apply an exchange algorithm to our SIP problem which is parametrized by *uniform continuous density* function. Before giving our algorithm "parametrized" by a uniform continuous density function, we analyze the continuity of uniform continuous density function in the following example.

**Example 6.3.1** *As it is assumed in the previous chapters (see Chapter 2), the objective and the constraint functions, $f, h, g, u, v$, respectively, are two-times continuously differentiable ($C^2$-) functions. Now, the global continuity can fail for our function $g$, depending on the parametrization of the corresponding pdf. As an example, we choose a uniform continuous*

**Algorithm 3 Primal Exchange Method (PEM)**

**Input:**

$\delta$ positive number, i.e., $\delta > 0$

$f^{\mathcal{P}}$ probability density function (pdf)

$P^{\mathcal{P}}$ the set where pdf parameters lie

**Output:**

$\theta_k$ unknown variable for minimization, to be evaluated

$\boldsymbol{\wp}_{\boldsymbol{k}}^{\boldsymbol{\mathcal{P}}}$ the parameter vector of the pdf

$\boldsymbol{\alpha_k}$ dual variable of SVM (support vectors)

**PEM**$\left(\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\boldsymbol{\mathcal{P}}}, \boldsymbol{\alpha_k}, A, \delta, f^{\mathcal{P}}, P^{\mathcal{P}}\right)$

1: $k := 0$

2: Initialize a discretization $A_k \subset \Omega$.

3: **DO** Compute a solution $(\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\boldsymbol{\mathcal{P}}})$ of

$$\begin{aligned}
&\min_{\theta\in\mathbb{R},\boldsymbol{\wp}^{\mathcal{P}}} \ -\theta \\
&\text{subject to} \quad g^{\mathcal{P}}((\theta,\boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}}),\boldsymbol{\alpha}) := \int_\Omega T(\omega,\boldsymbol{\alpha}) f^{\mathcal{P}}(\omega;\boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}})d\omega - \theta \geqslant 0 \ \ (\boldsymbol{\alpha}\in A_k), \\
&\mathrm{u}_i^{\mathcal{P}}(\boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}}) = 0 \ \ (i \in I^{\mathcal{P}}), \\
&\mathrm{v}_j^{\mathcal{P}}(\boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}}) \geq 0 \ \ (j \in J^{\mathcal{P}}).
\end{aligned}$$

4: Compute local solutions $\boldsymbol{\alpha}_{\boldsymbol{k}}^{\boldsymbol{i}}$ $(i = 1, 2, \ldots, i_k)$ of the reduced problem such that one of them, say $\boldsymbol{\alpha}_{\boldsymbol{k}}^{\boldsymbol{i_0}}$, is a global solution, i.e.,

$$g^{\mathcal{P}}((\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\boldsymbol{\mathcal{P}}}), \boldsymbol{\alpha}_{\boldsymbol{k}}^{\boldsymbol{i_0}}) = \min_{\boldsymbol{\alpha}\in A} \ g^{\mathcal{P}}((\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\boldsymbol{\mathcal{P}}}), \boldsymbol{\alpha}).$$

5: **if** $g^{\mathcal{P}}((\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\boldsymbol{\mathcal{P}}}), \boldsymbol{\alpha}_{\boldsymbol{k}}^{\boldsymbol{i_0}}) \geq -\delta$ with a solution $(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\boldsymbol{\mathcal{P}}}) \approx (\theta_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\boldsymbol{\mathcal{P}}})$, **then**

6: STOP

7: **else**

8: $A_{k+1} := A_k \cup \left\{\alpha_k^i \mid i = 1, 2, \ldots, i_k\right\}$

9: $k := k + 1$

10: **end if**

11: **END DO**

*density function with parameter vector* $\wp^{\mathcal{P}} = (a, b)$ $(a \leq b)$. *Let us recall that the pdf of the uniform continuous density is*

$$f^{\mathcal{P}}(\omega; (a, b)) = \begin{cases} \frac{1}{b-a}, & a \leqslant \omega \leqslant b, \\ 0, & \omega < a \text{ or } \omega > b. \end{cases}$$

*We observe that the term* $\frac{1}{b-a}$ *makes the function g (cf. (6.1)) discontinuous, actually, undefined at* $a = b$. *On the other hand, we need an inequality constraint, e.g., of the form "*$\leq$*", such as in* $a \leq b$. *To encounter this, let us introduce a sufficiently small positive number* $\epsilon > 0$ *such that the following relation is requested:*

$$a + \epsilon \leq b.$$

*Then, we prevent from equality of a and b with this small positive number and, hence, from discontinuity. In the following, the algorithm of exchange method (PEM) for solving our primal problem parametrized by a uniform continuous density function is presented.*

**Algorithm 4 Primal Exchange Method (PEM) for Uniform Continuous Probability function**

**Input:**

$A$ an infinite index set

$\delta$ positive number, i.e., $\delta > 0$

$\epsilon$ positive number, i.e., $\epsilon > 0$

**Output:**

$\theta_k$ unknown variable for minimization, to be evaluated

$a_k$ a parameter of the pdf

$b_k$ a parameter of the pdf

$\boldsymbol{\alpha_k}$ dual variable of SVM (support vectors)

**PEM**$(\theta_k, a_k, b_k, \boldsymbol{\alpha_k}, A, \delta, \epsilon)$

1: $k := 0$

2: Initialize a discretization $A_k \subset A$.

3: **DO** Compute a solution $(\theta_k, a_k, b_k)$ of

$$\begin{aligned} &\min_{\theta\in\mathbb{R}, a\in\mathbb{R}, b\in\mathbb{R}} \quad -\theta \\ &\text{subject to} \quad g^{\mathcal{P}}((\theta, a, b), \boldsymbol{\alpha}) := \int_{\Omega} T(\omega, \boldsymbol{\alpha}) f^{\mathcal{P}}(\omega; a, b) d\omega - \theta \geqslant 0 \;\; (\boldsymbol{\alpha} \in A_k), \\ &a + \epsilon \leq b. \end{aligned}$$

4: Compute local solutions $\boldsymbol{\alpha_k^i}$ $(i = 1, 2, \ldots, i_k)$ of the reduced problem such that one of them, say $\boldsymbol{\alpha_k^{i_0}}$, is global solution, i.e.,

$$g^{\mathcal{P}}((\theta_k, a_k, b_k), \boldsymbol{\alpha_k^{i_0}}) = \min_{\boldsymbol{\alpha}\in A} \; g^{\mathcal{P}}((\theta_k, a_k, b_k), \boldsymbol{\alpha}).$$

5: **if** $g^{\mathcal{P}}((\theta_k, a_k, b_k), \boldsymbol{\alpha_k^{i_0}}) \geq -\delta$ with a solution $(\bar{\theta}, \bar{a}, \bar{b}) \approx (\theta_k, a_k, b_k)$, **then**

6: STOP

7: **else**

8: $A_{k+1} := A_k \cup \left\{ \boldsymbol{\alpha_k^i} \mid i = 1, 2, \ldots, i_k \right\}$.

9: $k := k + 1$

10: **end if**

11: **END DO**

The convergence of the exchange method applied on our primal problem by Algorithm 4 is presented with the following theorem [78].

**Theorem 6.3.2** *[78]. We refer to $\mathcal{M}^{\mathcal{P}}_{comp}(A)$ which is obtained by the compactification of fea-*

*sible set $\mathcal{M}^{\mathcal{P}}(A)$, by transversally intersection of original feasible set with simple geometrical bodies (e.g., parallelpipes) provided by Closer Explanation 6.2.5. Then, the exchange method (with $\delta = 0$) either stops at some iteration $k_0 \in \mathbb{N}_0$ with a solution $(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}}) = (\theta_{k_0}, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k_0}})$ of (6.1) or the sequence $(\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}})_{k\in\mathbb{N}_0}$ of solutions of (6.3) satisfies*

$$\min_{(\theta,\boldsymbol{\wp}^{\mathcal{P}})\in\mathcal{G}^{\mathcal{P}}(A)} \left\|(\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}}) - (\theta, \boldsymbol{\wp}^{\mathcal{P}})\right\|_2 \to 0 \ \ (k \to \infty).$$

**Proof 3** *We prove the theorem by contradiction. Let us assume that the algorithm does not stop with a minimizer of (6.1). As in the proof of Theorem 6.2.6 given in [78], by our assumptions, a solution $(\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}})$ of (6.1) exists, $(\bar{\theta}_k, \bar{\boldsymbol{\wp}}^{\mathcal{P}}_{\boldsymbol{k}}) \in \mathcal{M}^{\mathcal{P}}_{comp}(A_0)$, and with a suitable, existing subsequence $(\theta_{k_\nu}, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k_\nu}})_{\nu\in\mathbb{N}_0}$ and a vector $(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}})$ such that $(\theta_{k_\nu}, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k_\nu}}) \to (\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}})$ $(\nu \to \infty)$, where the solution is in the compact elementary geometrical body (e.g., parallelpipe) $C$ (see Closer Explanation 6.2.5), $(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}}) \in C$ and $\bar{\boldsymbol{\wp}}^{\mathcal{P}} \in P^{\mathcal{P}}$, and we find*

$$-\bar{\theta} \le v(A).$$

*Again, we must show $(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}}) \in \mathcal{M}^{\mathcal{P}}_{comp}(A)$ or, equivalently, $\varphi(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}}) \ge 0$ $(\boldsymbol{\alpha} \in A)$ for the **value function** $\varphi(\theta, \boldsymbol{\wp}^{\mathcal{P}})$ of lower level problem, i.e., $\varphi(\theta, \boldsymbol{\wp}^{\mathcal{P}}) = \min_{\boldsymbol{\alpha}\in A} \ g((\theta, \boldsymbol{\wp}^{\mathcal{P}}), \boldsymbol{\alpha})$. In view of $\varphi(\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}}) = g((\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}}), \boldsymbol{\alpha}^{\mathbf{1}}_{\boldsymbol{k}})$, we can write*

$$\varphi(\bar{\theta}) = \varphi(\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}}) + \varphi(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}}) - \varphi(\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}}) = g((\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}}), \boldsymbol{\alpha}^{\mathbf{1}}_{\boldsymbol{k}}) + \varphi(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}}) - \varphi(\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}}).$$

*Since $\boldsymbol{\alpha}^{\mathbf{1}}_{\boldsymbol{k}} \in A_{k+1}$, we have $g((\theta_{k+1}, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k+1}}), \boldsymbol{\alpha}^{\mathbf{1}}_{\boldsymbol{k}}) \ge 0$ and by continuity of $g$ and $\varphi$, we find*

$$\varphi(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}}) \ge \left(g((\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}}), \boldsymbol{\alpha}^{\mathbf{1}}_{\boldsymbol{k}}) - g((\theta_{k+1}, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k+1}}), \boldsymbol{\alpha}^{\mathbf{1}}_{\boldsymbol{k+1}})\right) + \left(\varphi(\bar{\theta}, \bar{\boldsymbol{\wp}}^{\mathcal{P}}) - \varphi(\theta_k, \boldsymbol{\wp}^{\mathcal{P}}_{\boldsymbol{k}})\right) \to 0$$

*for $k \to \infty$, which concludes the proof.*

We refer to [33] for detailed explanation.

### 6.3.1 CONCEPTUAL REDUCTION METHOD

The *conceptual reduction method* is based on local reduction which starts with an arbitrary point, $\mathbf{x}^*$ (not necessarily feasible) for the SIP problem (2.11) and solves the lower level problem at that point, i.e., it solves $Q(\mathbf{x}^*)$ to find all the local minima $\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^r$ of $Q(\mathbf{x}^*)$ (see Section 2.4.6, equation (2.27)). It finds the optimal solution for the reduced finite problem

which has $r$ many constraints, and the iteration continues until the stopping criterion given by the Algorithm 5 in line 4 is fulfilled. Alternatively, one can choose one of the stopping criteria from (6.7) and (6.8) for the primal and the dual problems, respectively. In the following algorithms, we presented the *conceptual reduction* method, adapted to the primal and dual problems (6.1) and (6.2) based on [33].

**Algorithm 5 Primal Conceptual Reduction Method (PCRM)**

**Input:**

| | |
|---|---|
| $(\theta_0, \wp_{\mathbf{0}}^{\mathcal{P}})$ | initial guess for the optimal solution which is not necessarily feasible |
| $\epsilon$ | sufficiently small positive number to be used for one of the stopping criteria given by (6.7) |
| $f^{\mathcal{P}}$ | probability density function (pdf) |
| $P^{\mathcal{P}}$ | the set where the pdf parameters lie |

**Output:**

| | |
|---|---|
| $\theta_k$ | unknown variable for minimization, to be evaluated |
| $\wp_{\boldsymbol{k}}^{\mathcal{P}}$ | the parameter vector of the pdf |
| $\boldsymbol{\alpha_k}$ | dual variable of SVM (support vectors) $(i = 1, 2, \ldots, r)$ |

**PCRM**$\left(\theta_k, \wp_{\boldsymbol{k}}^{\mathcal{P}}, \boldsymbol{\alpha_k}, \theta_0, \wp_{\mathbf{0}}^{\mathcal{P}}, \delta, f^{\mathcal{P}}, P^{\mathcal{P}}\right)$

1: $k := 0$

2: Determine all *local minima* $\boldsymbol{\alpha}_{\boldsymbol{k}}^{\mathbf{1}}, \boldsymbol{\alpha}_{\boldsymbol{k}}^{\mathbf{2}}, \ldots, \boldsymbol{\alpha}_{\boldsymbol{k}}^{\boldsymbol{r}}$ of

$$\min_{\boldsymbol{\alpha} \in A} \; g^{\mathcal{P}}((\theta_k, \wp_{\boldsymbol{k}}^{\mathcal{P}}), \boldsymbol{\alpha})$$

3: **DO** Compute a solution $(\theta^*, \wp^{\mathcal{P}*})$ of

$$\begin{aligned} &\min_{\theta \in \mathbb{R}, \wp^{\mathcal{P}}} \; -\theta \\ &\text{subject to} \;\; g^{\mathcal{P}}((\theta, \wp^{\mathcal{P}}), \boldsymbol{\alpha}_{\boldsymbol{k}}^{\boldsymbol{l}}) := \int_{\Omega} T(\omega, \boldsymbol{\alpha}) f^{\mathcal{P}}(\omega; \wp^{\mathcal{P}}) d\omega - \theta \geqslant 0 \;\; (l = 1, 2, \ldots, r), \\ &\mathrm{u}_i^{\mathcal{P}}(\wp^{\mathcal{P}}) = 0 \;\; (i \in I^{\mathcal{P}}), \\ &\mathrm{v}_j^{\mathcal{P}}(\wp^{\mathcal{P}}) \geq 0 \;\; (j \in J^{\mathcal{P}}). \end{aligned}$$

4: **if** One of the stopping criteria given by (6.7) is satisfied, **then**

5: STOP

6: **else**

7: $(\theta_{k+1}, \wp_{\boldsymbol{k+1}}^{\mathcal{P}}) := (\theta^*, \wp^{\mathcal{P}*})$

8: $k := k + 1$

9: **end if**

10: **END DO**

**Algorithm 6 Dual Conceptual Reduction Method (DCRM)**

**Input:**

| | |
|---|---|
| $(\sigma_0, \boldsymbol{\wp}_{\mathbf{0}}^{\mathcal{D}})$ | initial guess for the optimal solution which is not necessarily feasible |
| $\epsilon$ | sufficiently small positive number to be used for one of the stopping criteria given by (6.8) |
| $f^{\mathcal{D}}$ | probability density function (pdf) |
| $P^{\mathcal{D}}$ | the set where probability density function (pdf) parameters lie |

**Output:**

| | |
|---|---|
| $\sigma_k$ | unknown variable for minimization, to be evaluated |
| $\boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{D}}$ | the parameter vector of the pdf |
| $\omega_k^i$ | primal variable of our (SIP) problem (Gaussian width in our case) $(i = 1, 2, \ldots, r)$ |

**DCRM**$\left(\sigma_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\mathcal{D}}, \omega_k^i, \sigma_0, \boldsymbol{\wp}_{\mathbf{0}}^{\mathcal{D}}, \delta, f^{\mathcal{D}}, P^{\mathcal{D}}\right)$

1: $k := 0$

2: Determine all *local minima* $\omega_k^1, \omega_k^2, \ldots, \omega_k^r$ of

$$\min_{\omega \in \Omega} \; g^{\mathcal{D}}((\sigma_k, \boldsymbol{\wp}_{\boldsymbol{k}}^{\boldsymbol{\mathcal{P}}}), \omega).$$

3: **DO** Compute a solution $(\sigma^*, \boldsymbol{\wp}^{\mathcal{D}*})$ of

$$\begin{aligned}
&\min_{\sigma \in \mathbb{R}, \boldsymbol{\wp}^{\mathcal{D}}} \; \sigma \\
&\text{subject to} \;\; g^{\mathcal{D}}((\theta, \boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}}), \omega_k^l) := \sigma - \int_{\Omega} T(\omega, \boldsymbol{\alpha}) f^{\mathcal{P}}(\boldsymbol{\alpha}; \boldsymbol{\wp}^{\boldsymbol{\mathcal{P}}}) d\boldsymbol{\alpha} \geqslant 0 \quad (l = 1, 2, \ldots, r), \\
&\mathrm{u}_i^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) = 0 \;\; (i \in I^{\mathcal{D}}), \\
&\mathrm{v}_j^{\mathcal{D}}(\boldsymbol{\wp}^{\mathcal{D}}) \geq 0 \;\; (j \in J^{\mathcal{D}}).
\end{aligned}$$

4: **if** One of the stopping criteria is satisfied given by (6.8), **then**

5: STOP

6: **else**

7: $(\sigma_{k+1}, \boldsymbol{\wp}_{\boldsymbol{k+1}}^{\mathcal{D}}) := (\sigma^*, \boldsymbol{\wp}^{\mathcal{D}*})$.

8: $k := k + 1$

9: **end if**

10: **END DO**

We observe that the substeps 2 in Algorithm 5 and Algorithm 6 are very costly as they require a global search for minima of $g((\theta_k, \wp_k^{\mathcal{P}}), \alpha)$ or $g((\sigma_k, \wp_k^{\mathcal{D}}), \omega)$ on $A$ or $\Omega$, respectively. We must avoid an execution of this step in the overall process as much as possible. Substep 2 assumes that there are only finitely many minima of the lower level problem for the primal and the dual case. If it does not hold, another method, e.g., discretization, should be used. Let us note that substep 3 solves a finite constrained optimization problem which requires only local searches and can be efficiently performed, e.g., by Newton's method.

**Remark 10** *The only difference between the exchange method and the conceptual reduction method is the starting point of the iteration. In the exchange method, we start with an initial feasible (discretized) index set. But, on the other hand, in the conceptual reduction method, we do not need to find a discretized set but an initial guess of the optimal solution of the upper level problem which does not need to be feasible. In our primal problem, as it is discussed in Section 6.2, we have difficulties in computing the discretization of the set A. We proposed different strategies to discretize the set A. Alternatively, to solve our primal and the dual problems, we can use the conceptual reduction method without any need of a discretization step.*

# CHAPTER 7

# CONCLUSION

We have developed three new methods of classification and we analyzed our mathematical model under regularity conditions of continuous optimization. The first two methods represent contributions to **model selection**. The first method is a contribution in Bioinformatics, specifically to biological sequence analysis. The second one targets *general model selection* in classification and includes modifications in the confidence level approach (in comparison to the first method) which has different kinds of applications. The third method focuses on *infinite kernel learning*, using infinite and semi-infinite programming and represents a contribution in *Applied Mathematics* and *Continuous Optimization*. These three methods also represent contributions to the general field of *Machine Learning*. Next, we summarize the contributions of each of the developed methods.

1. **Bioinformatics problem:** We developed a new model selection method which is based on the notion of a *confidence interval* [53]. We applied this new methodology to solve a bioinformatics problem. The biological problem we solved was to find the pro-peptide cleavage site of the proteins in fungi. Recognizing the critical positions of protein sequences consisting of a large amounts of amino acid sequences is a very exhaustive and time consuming process. Machine learning methods have been used to solve such a kind of problems. However, the model selection phase of the machine learning methods can also be time consuming. In this thesis, we established new methods and models for Support Vector Machines.

   We collected our data from NCBI database[1] specifically for fungi proteins. We applied some data mining algorithms, e.g., phylogenetic tree analysis, and used a bioinformat-

[1] http://www.ncbi.nlm.nih.gov/

ics tool such as ClustalW[2] to the raw data before carrying out classification. Since the amino acid sequences are consisted of letters, we transformed these letters using PAM and binary matrices with accordance to the evolutionary criteria of the proteins. Window based scanning of the amino acids is used for the input and output pairs for the classifier. We tested different window sizes as a parameter and chose the one having the best error rate.

We established our classification model by defining a new kernel function which is based on the occurrence of the same letters of amino acid sequences. The occurrence of the amino acid sequences are determined by scanning the windows through the protein sequence. This new kernel function is then proved to be the Gaussian kernel with a special Gaussian width [53]. One of the main contributions of this study is the selection of the classifiers (model selection) by "confidence intervals", which is based on functional margins on the test window sequences. Our new model selection method is compared with one of the well known model selection methods known as called cross validation. We pointed out the accuracy rate and training time of our new model selection algorithm in Chapter 3. We achieved a faster method and highly comparable accuracy rate when compared with cross validation.

2. **Development in Model Selection:** Our second contribution is in the field of machine learning (support vector machines). We generalize our confidence interval approach by a new model selection method, based on the observed margin [65]. In this context, three different norms are defined for the test point margins [52]. We tested our new method on different kinds of data collected from UCI machine learning repository[3]. We succeded to reduce the training time of the SVM with our new model selection method based on "observed margin" and "maximum margin principle" [52]. Furthermore, AUC results showed that our model selection method is also successfull in the class imbalance case.

3. **Development of Kernel Learning by Semi-infinite and Infinite Programming:** We also introduce the use of infinite and semi-infinite programming in order to model our new classifier with infinitely many kernels. Since the real world data can be heterogeneous and large scale, combinations of multiple kernels are helpful to classify such data, e.g. splice cite recognition [71]. Multiple kernel learning has been developed and

[2] http://www.ebi.ac.uk/Tools/clustalw2/index.html
[3] http://archive.ics.uci.edu/ml/

tackled in [71] by using semi-infinite programming. Convex combinations of the kernels are fitted to the model using positive kernel coefficients. Note that choosing finitely many kernels from the kernel space is parametric on the kernel choice and highly dependent on our selection of these finitely many kernels. This dependence can limit our selection and may lead to the choice of a poor kernel. We improved the idea of multiple kernel learning by enlarging the selection to infinitely many and calculating our new classifier via infinite programming [54]. This allows us to smooth the discrete set of kernels to a continuous range of kernels in an infinite-dimensional space. We define the combination of infinitely many kernels by Riemann-Stieltjes integrals and the monotonic increasing function as positive measures. After some assumptions and by using Theorem 5.3.6, we defined neighborhoods of these measures on Radon measures by Prokhorov distances, allowing us to define our neighborhoods of implicit functions.

The constitution by infinitely many kernels allows us to check all possibilities of kernels in a continuous domain and it also avoids the model selection for kernel in cross validation. In other words, we reduce the search domain by including kernel parameter, e.g., $\omega$, into our infinite programming model, and hence, only the error constant term, $C$, is left as a parameter. We model our classification problem by infinite programming since we have infinite dimensions in the kernel coefficients and infinitely many constraints. The regularity conditions are analyzed on our lower level problem and optimality conditions are discussed by theorems. One of the important theoretical problem is to find conditions which makes monotonic function (kernel coefficient function, $\beta$) point masses. By these conditions, we can guarantee that there are finitely many active points, i.e., finitely many kernel coefficients. In other words, we ensure that searching through the infinite dimensional space allows us to find the active ones under some assumptions and regularity conditions.

One of the advantage of infinite kernel learning is saving memory when using our method as compared to multiple kernel learning. In multiple kernel learning, in order to compute a convex combination, one needs to save $n$-kernels in computer memory, however, in our case, we need to compute only the integration result as a new kernel and we do not need to save infinitely many kernels.

Another development in the model is the reduction of infinite programming into semi-infinite programming by parametrization using probability density functions. Note that,

probability measures are the subspace of positive measures so that the construction of infinitely kernel coefficients by positive measures are still hold.

Infinite dimensions can lead the problem of curse of dimensionality (increase the model complexity) and it may result in ill-posedness. In order to overcome this discrepancy, we propose a regularization term to our objective function. Since we have probability measures as state variables, we can not use the theory of regularization, e.g., Tikhonov regularization. Instead, it is our proposal to measure the complexity of our model by "scanning" the integral terms via a running upper integration boundary, and taking partial derivatives of first and second order to record infinitesimal changes of these orders. By this and penalizing these kinds of change rates, we are looking for a "flat" model or one with not too high energy, respectively. We refer to [55, 80, 81] for more information on these kinds of penalizations. In our research, we introduced the new idea of the *scanning*, of moving upper integration limits.

Finally, by means of new ideas, we developed well-known numerical methods of semi-infinite programming for our new kernel machine in Chapter 6. We improved the discretization method for our specific model and proposed two new algorithms (see Strategy I and Strategy II in Chapter 6). The advantages of these methods are discussed and the intuition behind these algorithms are visualized by figures and examples. We stated convergence of the numerical methods with theorems and we analyzed the conditions and assumptions of these theorems such as optimality and convergence.

As a future study, we will apply and compare with other numerical methods and illustrate these methods on real-world data. In addition, we intend to study infinite programming and investigate primal-dual methods instead of reducing the infinite problem into semi-infinite programming. Furthermore, we will study our works with MFCQ and strong stability of all KKT points will be studied further instead of nondegeneracy.

# REFERENCES


[1] S.F. Altschul. Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology*, 219:555–665, 1991.

[2] J. Amaya and J.A. Gómez. Strong duality for inexact linear programming problems. *Optimization*, 49 (3):243 – 269, 2001.

[3] E.J. Anderson and P. Nash. *Linear Programming in Infinite-Dimensional Spaces*. John Wiley and Sons Ltd, 1987.

[4] T.M. Apostol. *Mathematical Analysis: A Modern Approach to Advanced Calculus*. Addison Wesley, 1974.

[5] A. Argyriou, R. Hauser, C. Micchelli, and M. Pontil. A dc-programming algorithm for kernel selection, 2006.

[6] A. Aster, B. Borchers, and C. Thurber. *Parameter Estimation and Inverse Problems*. Academic Press, 2004.

[7] V. Atalay and R. Cetin-Atalay. Implicit motif distribution based hybrid computational kernel for sequence classification. *Bioinformatics*, 21 (8):1429–1436, 2005.

[8] F.R. Bach and G.R.G. Lanckriet. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[9] D. Baker, A. K. Shiau, and D.A. Agard. The role of pro regions in protein folding. *Curr. Opin. Cell Biol.*, 5 (6):966–970, 1993.

[10] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25:1–13, 1999.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[12] Y.-D. Cai, S.-L. Lin, and K.-C. Chou. Support vector machines for prediction of protein signal sequences and their cleavage sites. *Peptides*, 24:159–161, 2003.

[13] Y.-D. Cai, X.-J. Liub, X.-B. Xu, and K.-C. Chou. Prediction of protein structural classes by support vector machines. *Computers and Chemistry*, 26:293–296, 2002.

[14] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm*, 2001.

[15] C.-T. Chen. *Linear System Theory and Design*. Oxford University Press, 1999.

[16] D. Cheng, R.Wang, F. Zhang, and B. Silbermann. Rna translation. *http : //library.thinkquest.org/C004535/rna − translation.html*.

[17] K.-C. Chou. Prediction of signal peptides using scaled window. *Peptides*, 22:1973–1979, 2001.

[18] M.G. Claros, S. Brunak, and G. von Heijne. Prediction of n-terminal protein sorting signals. *Current Opinion in Structural Biology*, 7:394–398, 1997.

[19] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[20] M. Dambrine and M. Pierre. About stability of equilibrium shapes. *Marhem. Modelling and Num. Analysis*, 34 (4):811, 2000.

[21] M.O. Dayhoff, R.M. Schwartz, and B.C. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure, Dayhoff, M.O. eds., National Biomedical Research Foundation, Washington*, 5 (3):345–352, 1978.

[22] A. Dubey, M.J. Realff, J.H. Lee, and A.S. Bommarius. Support vector machines for learning to identify the critical positions of a protein. *Journal of Theoretical Biology*, 234:351–361, 2005.

[23] P. Duckert, S. Brunak, and N. Blom. Prediction of proprotein convertase cleavage sites. *Protein Engineering, Design and Selection*, 17 (1):107–112, 2004.

[24] R.M. Dudley. *Real Analysis and Probability*. Chapman and Hall, 1989.

[25] G. Evin, A. Zhu, R.M.D. Holsinger, C.L. Masters, and Q-X. Li. Proteolytic processing of the alzheimers's disease amyloid precursor protein in brain and platelets. *Journal of Neuroscience Research*, 74:386–392, 2003.

[26] P.A. Flach. The many faces of roc analysis in machine learning. In *The Twenty-First International Conference on Machine Learning*, 2004.

[27] M. A. Goberna and M. A. Lopez. *Linear Semi-Infinite Optimization*. John Wiley and Sons Ltd, 1998.

[28] E. Haaren-Retagne. *A semi-infinite programming algorithm for robot trajectory planning*. PhD thesis, University Trier, 1992.

[29] R.S. Hamilton. The inverse function theorem of nash and moser. *Bulletin (New Series) of American Mathematical Society*.

[30] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning - Data Mining, Inference and Prediction*. Springer Series in Statistics, 2001.

[31] G. Von Heijne. A new method for predicting signal sequence cleavage sites. *Nucleic Acids Research*, 14 (1):4683–4690, 1986.

[32] R. Hettich and H.Th. Jongen. Semi-infinite programming: conditions of optimality and applications. In J. Stoer, editor, *Optimization Techniques 2, Lecture notes in Control and Information Sci.* Springer, Berlin, Heidelberg, /New York, 1978.

[33] R. Hettich and O. Kortanek. Semi-infinite programming: Theory, methods and applications. *SIAM Review*, 35, 3:380–429, 1993.

[34] R. Hettich and P. Zencke. *Numerische Methoden der Approximation und semi-infiniten Optimierung*. Tuebner, Stuttgart, 1982.

[35] R. Jalving. *Proteolytic processing in the secretory pathways of Aspergillus niger*. Phd Thesis, Wageningen University, 2005.

[36] D.J. Jeenes, D.A. Mackenzie, I.N. Roberts, and D.B. Archer. Heterologous protein production by filamentous fungi. *Biotechnol. Genet. Eng. Rev.*, 9:327–67, 1991.

[37] H.Th. Jongen, P. Jonker, and F. Twilt. *Nonlinear Optimization in Finite Dimensions - Morse Theory, Chebyshev Approximation, Transversality, Flows, Parametric Aspects*. Springer Verlag, 2000.

[38] S. Kawashima, H. Ogata, and M. Kanehisa. Aaindex: amino acid index database. *Nucleic Acids Res.*, 27:368–369, 1999.

[39] H.J. Keisler. *Elementary Calculus*. Prindle, Weber and Schmidt, 1986.

[40] G.R.G. Lanckriet, N. Cristianini, L. E. Ghaoui, P. Bartlett, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Machine Learning Research*, 5:27–72, 2004.

[41] J. Langford and J. Shawe-Taylor. PAC bayes and margins. In *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2003. MIT Press.

[42] T. Len and E. Vercher. Optimization under uncertainty and linear semi-infinite programming: a survey. In 1999) Goberna/López, (Alicante, editor, *Nonconvex Optim. Appl.* Kluwer Acad. Publ., Dordrecht, 2001.

[43] W. Linde. *Probability in Banach Spaces - Stable and Infinitely Divisible Distributions*. John Wiley and Sons, Chichester-New York, 1983.

[44] D.A. McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

[45] S. G. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill in January, 1996.

[46] K.M.H. Nevalainen, V.S.J. Te'o, and P.L. Bergquist. Heterologous protein expression in filamentous fungi. *Trends in Biotechnology*, 23 (9):468–474, 2005.

[47] H. Nicholas and A. Ropelewski. Sequence analysis: Which scoring method should i use? *http://www.psc.edu/research/biomed/homologous/scoring_primer.html.*

[48] H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering*, 10:1–6, 1997.

[49] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2006.

[50] Z.B. Ögel. *Molecular Analysis of a fungal galactose oxidase gene*. Univ. of Leeds, Leeds, U.K., 1993.

[51] Y. Ozan. Private communication with Yildiray Ozan, Department of Mathemetics, METU, Turkey, 2008.

[52] S. Özöğür Akyüz, Z. Hussain, and J. Shawe-Taylor. Prediction with the svm using test point margins. In S. Lessmann, editor, *Annals of Information Systems (to appear)*. Springer, 2009.

[53] S. Özöğür-Akyüz, J. Shawe-Taylor, G.-W. Weber, and Z.B. Ögel. Pattern analysis for the prediction of eukoryatic pro-peptide cleavage sites. *Discrete Applied Mathematics*, Special issue on Networks in Computational Biology, doi:10.1016/j.dam.2008.06.043, 2008.

[54] S. Özöğür-Akyüz and G.-W. Weber. Learning with infinitely many kernels via semi-infinite programming. *Optimization Methods and Software (submitted)*, 2008.

[55] G.-W. Weber P. Taylan and A. Beck. New approaches to regression by generalized additive models and continuous optimization for modern applications in finance, science and technology. In B. Burachik and X. Yang, editors, *in honour of Prof. Dr. Alexander Rubinov*, volume 56 5-6.

[56] K. R. Parthasarathy. *Probability Measures on Metric Spaces*. AMS Chelsea Publishing, Providence, RI, 2005.

[57] E. Polak. On the mathematical foundation of nondifferentiable optimization in engineering design. *SIAM Rev.*, 29, 1997.

[58] P.J. Punt, A. Drint-Kuijvenhoven, B.C. Lokman, J.A. Spencer, D. Jeenes, D.A. Archer, and C.A. van den Hondel. The role of the aspergillus niger furin-type protease gene in processing of fungal proproteins and fusion proteins. evidence for alternative processing of recombinant (fusion-) proteins. *J. Biotechnol.*, 5 106 (1):23–32, 2003.

[59] P.J. Punt, N. van Biezen, A. Conesa, A. Albers, J. Mangnus, and C. van den Hondel. Filamentous fungi as cell factories for heterologous protein production. *Trends in Biotechnology*, 20 (5):200–206, 2002.

[60] A.F.R. Rahman, H. Alam, and M.C. Fairhurst. Multiple classifier combination for character recognition: Revisiting the majority voting system and its variations. In D. Lopresti, J. Hu, and R. Kashi, editors, *ALecture Notes in Computer Science*, volume 2423. Springer Berlin / Heidelberg.

[61] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning, 2007.

[62] R.T. Rockafellar. *Convex Anlaysis*. Princeton University Press, 1997.

[63] R.Tichatschke, R. Hettich, and G. Still. Connections between generalized inexact and semi-infinite linear programming. *ZOR-Methods and Models of OR*, 33:367–382, 1989.

[64] J.-J Rückmann and G.-W Weber. Semi-infinite optimization: Excisional stability of noncompact feasible sets. *Sibirskij Mathematicheskij Zhurnal*, 39:129–145, 1995.

[65] J. Shawe-Taylor. Classification accuracy based on observed margin. *Algorithmica*, 22:157–172, 1998.

[66] J. Shawe-Taylor. *Review of Anthony, Martin; Bartlett, Peter L., Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2001.

[67] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[68] U.P. Shinde, J.J. Liu, and M. Inouye. Protein memory through altered folding mediated by intramolecular chaperones. *Nature*, 389 (6650):520–2, 1997.

[69] A. N. Shiryaev. *Probability*. Springer, 1995.

[70] E.M. Silvia. Riemann-stieltjes integration. Technical report, UC Davis, One Shields Avenue University of California Davis, 1999.

[71] S. Sonnenburg, G. Raetsch, C. Schafer, and B. Schoelkopf. Large scale multiple kernel learning. *J. Machine Learning Research*, 7:1531–1565, 2006.

[72] A.L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21:1154 – 1157, 1973.

[73] R.G. Spiro. Protein glycosylation: nature, distribution, enzymatic formation, and disease implications of glycopeptide bbonds. *Glycobiology*, 12(4):43R–56R, 2002.

[74] J.Y. Springael, E. Nikko, B. Andr*é*, and A.M. Marini. Yeast npi3/bro1 is involved in ubiquitin-dependent control of permease trafficking. *FEBS Letters*, 517:103–109, 2002.

[75] C. De Stefano, A. Della Cioppa, and A. Marcelli. An adaptive weighted majority vote rule for combining multiple classifiers. In *16th International Conference on Pattern Recognition (ICPR'02)*, volume 2, 2002.

[76] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

[77] G. Still. Generalized semi-infinite programming: Theory and methods. *European Journal of Operational Research*, 119:301–313, 1999.

[78] G. Still. Semi-infinite programming: An introduction, preliminary version. Technical report, University of Twente Department of Applied Mathematics, P.O.Box 217 7500 AE Enschede, The Netherlands, 2004.

[79] K.D. Stroyan. *Mathematical Background: Infinitesimal Calculus*. Academic Press, Inc., 1997.

[80] P. Taylan and G.-W. Weber. New approaches to regression in financial mathematics by additive models. *Journal of Computational Technologies*, 12 (2):3–22, 2007.

[81] P. Taylan, G.-W. Weber, and F. Yerlikaya. Continuous optimization applied in mars for modern application in finance, science and technology. In *Continuous Optimization and Knowledge Based Technologies, 20th EURO Mini conference*, 2008.

[82] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[83] R.S. Varga. *Matrix Iterative Analysis*. Springer, 2000.

[84] G.-W. Weber. Charakterisierung struktureller stabilit at in der nichtlinearen optimierung. In H.Th. Jongen H.H. Bock and W. Plesken, editors, *Aachener Beitraege zur Mathematik 5*. Augustinus publishing house (now: Mainz publishing house), Aachen, 1992.

[85] G.-W. Weber. Minimization of a max-type function: Characterization of structural stability. In B. Kummer J. Guddat, H.Th. Jongen and F. Nozicka, editors, *Parametric Optimization and Related Topics III*. Peter Lang publishing house, Frankfurt a.M., Bern, 1993.

[86] G.-W. Weber, P. Taylan, S.Z. Alparslan Gök, S. Özöğür, and B. Akteke-Öztürk. Optimization of gene-environment networks in the presence of errors and uncertainty with chebychev approximation. *to appear in TOP, the Operational Research journal of SEIO (Spanish Statistics and Operations Research Society)*, 2008.

[87] G.-W. Weber, P. Taylan, S. Ozogur, and B. Akteke-Ozturk. Statistical learning and optimization methods in data mining. In H.O. Ayhan and I. Batmaz, editors, *Recent Advances in Statistics*. Turkish Statistical Institute Press, 2007.

[88] G.W. Weber. *Generalized Semi-Infinite Optimization and Related Topics*, volume 29 of *Research and Exposition in Mathematics*. Heldermann Verlag, Germany, 2003.

[89] A. Weidemann, K. Paliga, U. Dürrwang, F.B.M. Reinhard, O. Schuckert, G. Evin, and C.L. Masters. Proteolytic processing of the alzheimer's disease amyloid precursor protein within its cytoplasmic domain by caspase-like proteases. *The Journal of Biological Chemistry*, 274 9 (26):5823–5829, 1999.

[90] W.W.E. Wetterling. Definitheitsbedingungen für relative extrema bei optimierungsund approximationsaufgaben. *Numer. Math.*, 12:122–136, 1970.

[91] A. Yershova and S.M. LaVelle. Deterministic sampling methods for spheres and *so*(3). In *IEEE International Conference on Robotics and Automation (ICRA 2004)*, 2004.

[92] Z.Wan, S.Y. Wu, and K.L. Teo. Some properties on quadratic infinite programs of integral type. *Applied Mathematics Letters*, 20:676–680, 2007.

# VITA

## PERSONAL INFORMATION


Surname, Name: Özöğür- Akyüz, Süreyya

Nationality: Turkish (TC)

Date and Place of Birth: 25 January 1982, Ankara

Marital Status: Married to Akın Akyüz

Phone: +90 216 483 9000 Internal: 2117

Fax: +90 216 483 9559

email: sozogur@metu.edu.tr


## EDUCATION

Ph.D. Department of Scientific Computing, February 2009
Institute of Applied Mathematics
Middle East Technical University-Ankara
Advisor: Prof. Dr. Gerhard Wilhelm Weber
Co-advisor: Prof. Dr. John Shawe-Taylor
Thesis Title: A Mathematical Contribution of Statistical Learning and Continuous Optimization Using Infinite and Semi-Infinite Programming To Computational Statistics

Msc. Department of Computer Science, January 2005
Middle East Technical University-Ankara
Advisor: Prof. Dr. Bülent Karasözen
Co-advisor: Prof. Dr. Gerhard Wilhelm Weber
Thesis Title: " Mathematical Modelling of Enzymatic Reactions, Simulation and Parameter Estimation, Middle East Technical University, Institute of Applied Mathemetics, January, 2005.

B.S. Department of Mathematics, June 2002
Middle East Technical University-Ankara

High School Yükseliş Koleji
June 1998, Ankara

## WORK EXPERIENCE

2007-Present Research Fellow
Department of Electronics Engineering at the Center of Computer Vision and Pattern Analysis Labaratory (VPA Lab), Faculty of Engineering and Natural Sciences,
Sabancı University- Istanbul

2002-2007 Research Assistant
Department of Computer Science, Institute of Applied Mathematics
Middle East Technical University-Ankara

## FOREIGN LANGUAGES

Turkish (native), English (High level),

## COMPUTER ABILITIES

Microsoft Office, Latex, Matlab

## HOBBIES

tracking, swimming, cinema, theatre, playing classical guitar, drawing

## PUBLICATIONS

(i) S. Özöğür-Akyüz and G.W.-Weber "On Numerical Optimization Theory of Infinite Kernel Learning", *preprint no.* ***129****, Institute of Applied Mathematics, METU*, submitted to Journal of Global Optimization, January 21, 2009.

(i) S. Özöğür-Akyüz and G.W.-Weber "Modelling of Kernel Machines by Infinite and Semi-Infinite Programming", *preprint no.* ***128****, Institute of Applied Mathematics, METU*, to appear in Global Conference on Power Control and Optimization (PCO'2009), Bali, Indonesia, 1-3 June 2009.

(ii) S. Özöğür-Akyüz and G.W.-Weber "Learning with Infinitely Many Kernels via Infinite and Semi-Infinite Programming", submitted to *special issue of Optimization Methods and Software on Engineering Optimization*, *guest ed:* Klaus Schittkowski, September 2008.

(iii) S. Özöğür-Akyüz, G. W.-Weber, "Learning with Infinitely Many Kernels via Semi- Infinite Programming", *in the ISI Proceedings of 20th Mini - EURO conference, Continuous Optimization and Knowledge-Based Technologies, Neringa, Lithuania* May 20-23, 2008.

(iv) S. Özöğür-Akyüz, B. Akteke-Öztürk, T. Tchemisova and G.-W. Weber, "New optimization methods in data mining", to appear in the *ISI proceedings of International Conference Operations Research* (OR 2008; Augsburg, Germany, September 3-5, 2008), Springer Verlag.

(v) S. Özöğür-Akyüz, J. Shawe-Taylor, G.-W. Weber, Z. B. Ögel "Pattern Analysis for the Prediction of Eukoryatic Pro-peptide Cleavage Sites", *Article In Press in Discrete Applied Mathematics*, doi:10.1016/j.dam.2008.06.043, 2008.

(vi) G. W.-Weber, P. Taylan, S.Z. Alparslan Gök, S. Özöğür Akyüz and B. Akteke Öztürk, "Optimization of gene-environment networks in the presence of errors and uncertainty with Chebychev approximation", *TOP, the Operational Research journal of SEIO (Spanish Statistics and Operations Research Society*, Vol. **16**, no. **2** (2008) pp. 284-318.

(vii) S. Özöğür-Akyüz, Z. Hussain, J. Shawe-Taylor, "Model Selection via Test Margin", Model Selection via Test Margin, *article in press in the Special Issue on Data Mining of journal of Annals of Informations Systems (AoIS)*, Springer Book Series, 2008.

(viii) S. Özöğür-Akyüz, G. W.-Weber, "Learning with Infinitely Many Kernels via Semi- Infinite Programming", *preprint no. **86**, Institute of Applied Mathematics, METU* (2007).

(ix) G.-W. Weber, P. Taylan, S. Özöğür, B. Akteke-Öztü rk, "Statistical Learning and Optimization Methods in Data Mining", in the *book of the Turkish Association of Statiticians at the Occasion of "Graduate Summer School On New Advances in Statistics*, Middle East Technical University, 2007.

(x) A. Akyüz, T. Yurdun, Z. Pelin, S. Özöğür, "Serum Copper and Zinc Status in Obstructive Sleep Apnea Patients", J*ournal of Sleep Research* Published on behalf of the European Sleep Research Society September 2006 - Vol. 15 Issue s1 Page v-281.

(xi) S. Özöğür, A.G. Sağdıçoğlu Celep, B. Karasözen, N. Yıldırım, G.-W. Weber "Dynamical modelling of enzymatic reactions, simulation and parameter estimation with genetic algorithms", *In HIBIT-proceedings of international symposium on health informatics and bioinformatics*, Turkey'05, Antalya, Turkey, November 2005, pp 78-84.

(xii) S. Özöğür-Akyüz, B. Karasözen, and G.-W. Weber "Challenges in the Optimization of Biosytems I: Parameter Estimation of Enzymatic Reactions with Genetic Algorithm", *preprint no. **41**, Institute of Applied Mathematics, METU* (2005).

## PRESENTATIONS IN INTERNATIONAL SCIENTIFIC MEETINGS

(i) "Learning with Infinitely Many Kernels via Semi-Infinite Programming", *EURO mini conference on Continuous Optimization and Knowlegde Based Technologies*, Neringa, Lithunia, May 20-23, 2008.

(ii) "Maximizing the Margin on a Test Set", *EURO XXII, 22nd European Conference on Operational Research in the session of "Data Mining and Knowledge Discovery*, Prague, Czech Republic, July, 2007.

(iii) "Biological Data Mining by Using SVM and Pattern Analysis", *EURO XXI, 21st European Conference on Operational Research*, Iceland, July, 2006.

(iv) "Dynamical Modeling of Enzymatic Reactions, Simulation and Parameter Estimation with Genetic Algorithm", *International Symposium on Health and Bioinformatics (HIBIT)*, Antalya, November, 2005.

(v) Various talks in Seminar of Applied Mathematics in Life and Human Sciences and Economy group,
`http://www.iam.metu.edu.tr/research/groups/compbio/seminars.html`).

(vi) Various joint talks with Prof. Dr. Gerhard Wilhelm Weber in Ballarat, Australia (2006), Alberta, Canada (2007), Kiev, Ukraine (2006, 2007, 2008), Rio de Janeiro, Brazil (2008), Braga, Portugal (2008) and Haifa, Israel (2008).

## RESEARCH VISITS

May 2006 - August 2006, Southampton University, Department of Electronics and Computer Science, Southampton, UK, supervised by Prof. John Shawe- Taylor (funded by PASCAL).

August 2006 - February 2007 University College London, Department of Computer Science, London, UK, supervised by Prof. John Shawe-Taylor (funded by PASCAL).

July 1- July 14 2007 University College London, Department of Computer Science, London, UK, supervised by Prof. John Shawe-Taylor (funded by PASCAL).

## PARTICIPATION IN INTERNATIONAL SCIENTIFIC MEETINGS

EURO mini conference on Continuous Optimization and Knowledge Based Technologies, Neringa, Lithuania, May 20-23, 2008.

EURO XXII, European Conference on Operational Research, Prague, Czech Republic, July 8-11, 2007.

Patenting in Molecular Biology, London, UK, January 30, 2007.

Introduction to Bioinformatics Course, London, UK, December 4-8, 2006.

EURO XXI, 21st European Conference on Operational Research, Reykjavik, Iceland, July 2-5, 2006.

Mathematical Foundations of Learning Theory (II) Conference, Paris, France, 31 May-3 June, 2006.

Machine Learning, Support Vector Machines and Large Scale Optimization, Wissenschaftszentrum Schloß Thurnau, Germany, March 16-18, 2005.

HIBIT, International Symposium on Health Informatics and Bioinformatics, Antalya, Turkey, November 9-12, 2005.

EURO Summer Institute "Optimization and Data Mining", Ankara Turkey, July 9-25, 2004.

Bioinformatics Summer Institute, Istanbul, Turkey, August 15-21, 2004.

Workshop on Advances in Continuous Optimization, Istanbul, Turkey, July 4-5, 2003.

Workshop on Advances in Continuous Optimization, Istanbul, Turkey,
July 4-5, 2003.

## MEMBERSHIPS

EUROPT - The Continuous Optimization Working Group of EURO,
http://www.iam.metu.edu.tr/EUROPT/.

EURO - Association of European Operational Research Societies

SIAM - Society of Industrial and Applied Mathematics

PASCAL - Pattern Anaysis, Statistical Modelling and Computational Learning
http://www.pascal-network.org/

Applied Mathematics in Life and Human Sciences and Economy, Institute of Applied Mathematics, METU,
http://www.iam.metu.edu.tr/research/groups/compbio/index.html

## ORGANIZATION OF SCIENTIFIC EVENTS

Member of organizing commitee (stream organizer) of *EURO XXIII conference*, Bonn, Germany, July 5-8, 2009.

Member of organizing commitee of *Applied Mathematics in Life and Human Sciences and Economy group*, Institute of Applied Mathematics, METU, Ankara.

Member of organizing commitee (stream organizer) of *EURO mini conference on Continuos Optimization and Knowledge Based Technologies*, Neringa, Lithunia, May 20-23, 2008.

## REFEREE ACTIVITIES

Discrete Applied Mathematics (DAM).

European Journals of Operations Research (EJOR).

IEEE Transactions on Information Technology in Biomedicine.

IEEE, ICMLA '08: The Seventh International Conference on Machine Learning and Applications, 2008.

Journal of Machine Learning (JMLR)