Support Vector Machines for Survival Analysis: Methods and Variable Relevance

Màquines de Suport Vectorial per Anàlisi de la Supervivència: Mètodes i Rellevància de Variables

Héctor Sanz Ródenas

ADVERTIMENT. La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX (**www.tdx.cat**) i a través del Dipòsit Digital de la UB (**diposit.ub.edu**) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX ni al Dipòsit Digital de la UB. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX o al Dipòsit Digital de la UB (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR (**www.tdx.cat**) y a través del Repositorio Digital de la UB (**diposit.ub.edu**) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR o al Repositorio Digital de la UB. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR o al Repositorio Digital de la UB (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING. On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX (**www.tdx.cat**) service and by the UB Digital Repository (**diposit.ub.edu**) has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized nor its spreading and availability from a site foreign to the TDX service or to the UB Digital Repository. Introducing its content in a window or frame foreign to the TDX service or to the UB Digital Repository is not authorized (framing). Those rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

UNIVERSITAT DE
BARCELONA

Support Vector Machines for Survival Analysis: Methods and Variable Relevance

Màquines de Suport Vectorial per Anàlisi de la Supervivència:
Mètodes i Rellevància de Variables

Memòria presentada per Héctor Sanz Ródenas
per optar al grau de doctor per la Universitat de Barcelona

El Doctorand:
Héctor Sanz Ródenas

El Director de la Tesi:
Dr. Ferran Reverter Comes

Programa de doctorat en estadística
Departament d'Estadística, Facultat de Biologia
Universitat de Barcelona

Barcelona, Maig 2017

Als meus pares, la Sònia i l'Aniol

Acknowledgements

En primer lloc vull agrair al meu director, Ferran Reverter, el suport i optimisme que ha mostrat envers la meva feina durant els darrers cinc anys. He après molt d'un tema totalment desconegut per mi com és el relacionat amb els mètodes *kernel* en general i el de màquines de suport vectorial en particular. Un món enorme i apassionant que sense l'ajuda d'en Ferran no hauria arribat a aprofundir-hi mai.

Special thanks to Clarissa Valim for supporting me and taking time to revise my work. Her words and encouraging enthusiasm has been specially helpful these last months.

Thanks to Mal067 principal investigator, Carlota Dobaño, for allowing me to use the Mal067 data.

Thanks to Xinhua Zhang for gently clarifying our doubts related to his local invariances work.

Gràcies a la Roser Maldonado per ajudar tant, sempre amb paraules amables, a un procés moltes vegades frustrant com és el de les gestions administratives.

Moltes gràcies a la María Assumpció Farreras per tot el suport, ajuda i comprensió mostrada en aquests darrers 5 anys, especialment l'últim que ha estat el més difícil.

No puc agrair prou tota la paciència i comprensió mostrada per la Sònia i l'Aniol tots aquests anys. Gràcies per donar-me suport en tot moment de manera incondicional.

Contents

List of Tables

List of Figures

Notation and Symbols

i.i.d.	independent and identically distributed
$\mathcal{X}$	the input domain
$\mathbb{R}$	set or real numbers
y_i	class of observation $i,$ that is $y_i \in \{\pm 1\}$ (in binary classification) or $y_i \in \mathbb{R}$ (in regression setting)
$\boldsymbol{x}$	a vector of all observations
$\boldsymbol{x}_i$	a vector of the observation i
x_i^a	component a of the $\boldsymbol{x}_i$ observation vector
b	bias term in support vector machine methodology
$\mathcal{H}$	feature space
$\boldsymbol{\phi}$	feature map, $\boldsymbol{\phi} : \mathcal{X} \mapsto \mathcal{H}$
$\boldsymbol{w}$	weight vector in feature space
$\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$	dot product between $\boldsymbol{x}$ and $\boldsymbol{x}'$
$k(\boldsymbol{x}, \boldsymbol{x}')$	kernel function between $\boldsymbol{x}$ and $\boldsymbol{x}'$
$\|\boldsymbol{w}\|$	2 norm, $\sqrt{\langle \boldsymbol{x}, \boldsymbol{x}' \rangle}$
$\mathcal{L}$	Likelihood function
L	Lagrangian
$\boldsymbol{\alpha}$	vector of Lagrange multipliers
α_i	Lagrange multiplier
$\log(\cdot)$	base e logarithm
$\log_{10}(\cdot)$	base 10 logarithm
$\exp(\cdot)$	exponential function $(x \mapsto e^x)$
f	a function $\mathcal{X} \mapsto \mathbb{R}$ or $\mathcal{X} \mapsto \{\pm 1\}$
Q^{-1}	inverse matrix
$Q^{\top}$	transposed matrix or vector
Δ	small positive increment or positive value

SVM	Support Vector Machines
SVR	Support Vector Regression
ϵ-SVR	Support Vector Regression specifically based on ϵ-insensitive-loss
LUPI	Support Vector Machines Learning Using Privileged Information
pSVM	Support Vector Machines with uncertain classes
inSVM	Semi-Supervised Support Vector Machines with local invariances
wSVM	Weighted Support Vector Machines
RKHS	Reproducing kernel Hilbert space

Introduction

This PhD thesis has been motivated by a study aiming to identify immune responses correlated with protection from malaria that were elicited by the malaria RTS,S vaccine[1] and by natural immunity, so called Mal067. More specifically, this thesis has been motivated by the study component focused on cellular immune responses that has been measured in a multiplex bead-based array assay of 30 cytokines, chemokines and growth factors. Results of this study could inform development of future malaria vaccines.

The process of creating an efficacious malaria vaccine is complex due to the characteristics of the diseases that are directly related to the responsible parasite. In the disease-vaccine interaction several aspects need to be taken into account to improve and understand the vaccine. For that reason different types of data need to be analyzed. In this context, finding new ways to predict the disease, model it and identify the most important markers associated with the disease will improve the understanding and as a consequence, the efficacy of the vaccine.

Luminex® technology allows analyzing several proteins simultaneously. The combination of the medium throughput dataset and the small sample size of some malaria studies may hinder the use of classical statistical methods. In the context of low number of observations and medium or high number of variables the support vector machines (SVM) models are a powerful tool to analyze sparse data, i.e., data in which the number of predictors is larger or approximately equal to the number of observations, especially when handling binary outcomes. However, biomedical research often involves analysis of time-to-event outcomes. Therefore, further research is needed to fully extend SVM to survival data with censored outcomes. Moreover, further work is needed in the visualization and measurement of variable relevance when using SVM. The primary objective of this thesis has been to propose new SVM for survival data approaches by extending the binary outcome approach to time-to-event, and proposing new methods to visualize and rank the relevance of the predictors involved in these survival-SVM methods. Therefore, applying these methods to the Mal067 data to help understanding the immune responses induced by the RTS,S vaccine and by natural exposure to the parasite that are associated with malaria risk or protection.

[1] RTS,S/AS01 (trade name Mosquirix) is a recombinant protein-based malaria vaccine. Approved for use by European regulators in July 2015. The vaccine was conceived of and created by GlaxoSmithKline laboratories in the late 1980s, being further developed through a collaboration between GSK and Walter Reed Army Institute of Research.

Terminology

This thesis is focused on SVM which is one of the main methods from the machine learning field. The terminology used in machine learning sometimes is different than the one in the classical statistical literature. The *predictors* or *independent variables* are called *inputs* in machine learning. The *outputs* are known as *responses* or *dependent variables* in classical statistical field. An *obervation* in statistics (or *case* or *subject*) is referred as *instance* in machine learning theory.

Another distinction is done related to the prediction tasks: *regression* for quantitative responses and *classification* for qualitative ones. When dealing with binary classification problems: the two groups of observations are classified as -1 and $+1$, and sometimes are referred to as *targets* or *classes*. Finally, an interesting distinction must be clarified related to *variable* and *feature*. This thesis follows the distinction made by Guyon and Elisseeff (2003):

> “ We call *variable* the *raw* input variables and *features* variables constructed for the input variables. We use without distinction the terms *variable* and *feature* when there is no impact on the selection algorithms, e.g., when features resulting from a preprocessing of input variables are explicitly computed. The distinction is necessary in the case of kernel methods for which features are not explicitly computed.”

The terminology used along this thesis is mainly the classical statistical one. However, when referring to the groups in which observations are classified the machine learning term *class* is used.

Software

The software used to implement and analyze the real and simulated data along this thesis has been the R software. This software has become the main software in biomedical research area. Due to this, a brief overview, in specific chapters, on the implemented packages in the R software is described. Besides that, several functions have been developed to implement all proposed SVM methods and variable relevance algorithms.

Structure of the thesis

The thesis consists of 5 parts that in turn are subdivided into 14 chapters. The first part reviews some general background and presents the main objectives of the thesis. In Chapter 1, we briefly review malaria and the multiplex bead array assay. Additionally, we describe the Mal067 project and the importance of the naturally acquired immunity. In Chapter 2, we review the main concepts related to survival data and time-to-event variables, specifically, the non-parametric Kaplan-Meier estimator of survival probability and the semi-parametric Cox proportional hazards model are described. Both chapters 1 and 2 are short introductions but given their importance they have been structured into two independent chapters. Chapter 3 describes the main concepts related to SVM models that are used along the thesis. In Chapter 4, the state of the art for SVM dealing with censored data and the state of the art methodologies for the identification of relevant variables are presented. Chapter 5 lays out the specific aims of the thesis. In Part II, four survival-SVM methods are presented, in Chapter 6, and a simulation study is performed comparing these methods under several scenarios in Chapter 7. Part II, ends in Chapter 8 with a discussion and the main points for future research. Part III, Chapter 9, presents three alternative methods to deal with visualization and ranking of relevant variables. To test the performance of the proposed methods a simulation study is conducted in Chapter 10, the discussion and future research are presented at the end of the same part, in Chapter 11. In Part IV, data from the Mal67 study is analyzed, summarizing the importance of each antigen in Chapter 12. In Chapter 13, a discussion and future research work related to the Mal067 study are presented. Finally, this PhD thesis concludes in Part V, Chapter 14, summarizing the main conclusions and findings of the SVM for survival analysis and relevance of predictor variables.

Part I

General Background and Objectives

Chapter 1

Malaria and Quantification of Cellular Immunity

In this chapter we present the main stages of malaria transmission from mosquitoes to humans and the importance of a vaccine to prevent deaths associated to malaria (Section 1.1). In the next, Section 1.2, we describe the cytokines and chemokines data in which the cellular component of the Mal067 project is based. In the subsequent section, Section 1.3, the Mal067 study design and the stimulations used to measure the concentration of the antigens are described. Finally, Section 1.4 address the importance of the AMA1 stimulation in the Mal067 study and the association to the naturally acquired immunity (NAI).

1.1 Malaria infection and vaccines

Malaria is an infection transmitted through the bites of infected female *Anopheles* mosquitoes. When biting an infected person, a mosquito takes a small amount of blood containing the malaria parasite. The parasite goes through some transformations and approximately one week later is transmissible. Therefore, in next blood meal, the mosquito injects the saliva with the parasite and transmits malaria to a new person. *Plasmodium falciparum*, one of the five types of malaria parasites, causes the most severe malaria. It is estimated that during 2015 there were 214 million cases and 438,000 deaths attributable to malaria (WHO, 2015), mostly in sub-Sahara African children, in whom *P.falciparum* is the predominant specie. Co-morbidities associated with malaria include anaemia in children and pregnant women, low birth weight, premature birth, and neurological diseases.

Figure 1.1 represents the life cycle of the malaria parasite once a person has been infected. The parasite changes through the life stages and at each stage, presents numerous antigens to a person. Briefly, the life cycle of the parasite goes through the following stages (Crawley et al., 2010):

1. Malaria infection begins when an infected mosquito bite injects the parasite in the form

of a *sporozoite*[1] into the bloodstream.

2. These sporozoites quickly invade the human liver.

3. The sporozoites multiply asexually in the liver cells over the next 7 to 10 days, so called incubation period, causing no symptoms.

4. After the incubation period, the parasites, in the form of *merozoites*[2], are released from the liver cells in the blood circulation, packed in vesicles. The vesicles eventually disintegrate, freeing the merozoites to enter the blood phase of their development.

5. In the bloodstream, the merozoites invade red blood cells (erythrocytes) and multiply again until the cells burst to then invade new erythrocytes. This cycle is called asexual parasite cycle and is, repeated, causing fever each time parasites break free and invade red blood cells.

6. Some of the merozoites in infected red blood cells leave asexual multiplication cycle and develop into sexual forms of the parasite, called *gametocytes*, that circulate in the blood stream.

7. During a bite, mosquitoes ingest the gametocytes, which further develop in the mosquito into mature sex cells called *gametes*.

8. The fertilized female gametes develop into actively moving ookinetes that burrow through the mosquito's midgut wall and form *oocysts*[3] on the exterior surface.

9. Inside the oocyst, thousands of active sporozoites develop. The oocyst eventually bursts, releasing sporozoites into the body cavity that travel to the mosquito's salivary glands.

10. The cycle of human infection begins again when the mosquito bites another person.

There are several vaccines candidates each targeting a different stage of infection. The RTS,S is a pre-erythrocytic vaccine and is the vaccine in the most advanced stage of clinical development. A Phase III randomized multisite trial has been recently completed and vaccine efficacy against clinical malaria within one year of follow-up is around 56% in 5 to 17 months old children and around 31% in 6 to 12 weeks infants (RTS,S Clinical Trials Partnership, 2015). The vaccine confers protection of limited duration: understanding the immunity conferred by the vaccine may help to further develop the vaccine and improve the efficacy and the duration of vaccine protection.

[1] The infectious and immature form of the malaria parasite.

[2] The form of the malaria parasite that invades human red blood cells.

[3] A parasite stage within the mosquito, produced by the union of male and female gametes.

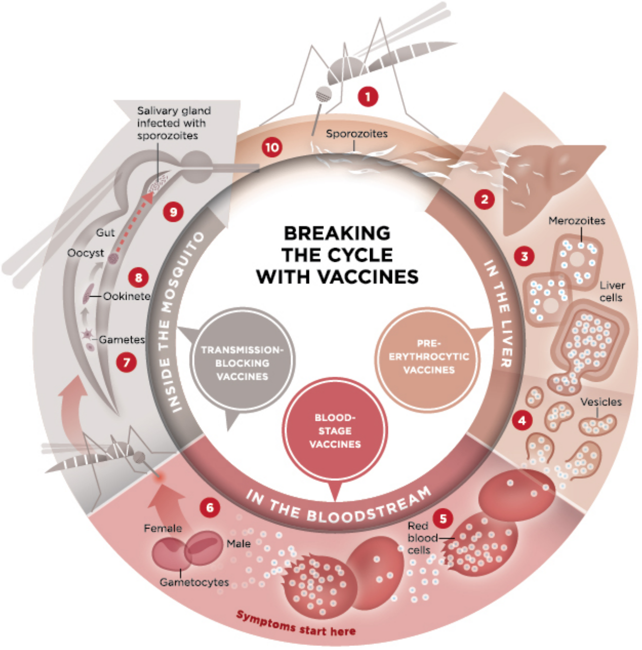


Figure 1.1: Life cycle of the malaria parasite representing the ten stages of infection. Reproduced from the The Lancet (2010) 375:1468-81.

1.2 Cytokines, chemokines and growth factors data

Cellular immunity[4] is complex and involves different cell types producing cytokines, chemokines and growth factors as immune mediators[5]. These cell products can be quantified through multiplex bead arrays that are commonly carried out using the Luminex® platform. These assays allows measuring several cytokines, chemokines and other immune cell products simultaneously using small volumes of blood.

Generally, in these assays, sets of magnetic beads individually labeled for the analyte quantified are coated with specific antibodies to capture the analytes of interest. The different sets of beads, targeting each analyte of interest, are mixed with subject liquid samples (serum, plasma or cells culture supernatant, for instance) in plastic plates that are often in the 96-well format. In these plates, the beads are allowed to capture all existing analytes in the samples. A second antibody is then added to the plate that binds to the analyte captured by the antibodies adhered to the beads, making a sandwich with the analyte to be quantitated. This second antibody is then conjugated with streptavidin R-phycoerithrin which gives the signal to quantify the analyte. After several washing steps to remove fluorescent antibodies and analytes that were not captured in the sandwich, the plate is analyzed by a reader endowed with two different lasers. Heuristically, one laser bin *recognizes* the bead label through the color of the emitted fluorescence, and the second laser bin quantitates the amount of analyte through the intensity

[4] Branch of the immune system which targets cells infected with microorganisms such as viruses, fungi, and certain bacteria.

[5] Antigens and analytes are terms used indistinctly to refer cytokines, chemokines and growth factors.

of the emitted fluorescence when the light interacts with the fluorescent antibody (Figure 1.2). The assay final raw output is the median fluorescence intensity (MFI) over all beads containing the corresponding analyte that are captured by the reader. The individual MFI is transformed through a standard curve in concentration. This standard curve represents the assay output of a reference standard with known concentration. In each plate wells are included containing increasing dilutions of the reference standard sample for each marker in the assay. With the observed MFIs for this sample and the known concentrations, these wells are used to fit a function of the association between MFI and concentration through a sigmoidal curve. To calibrate each subject's response, the MFI of the subject is converted in analyte concentration using the standard curve, and that is the measured outcome of the participant sample.

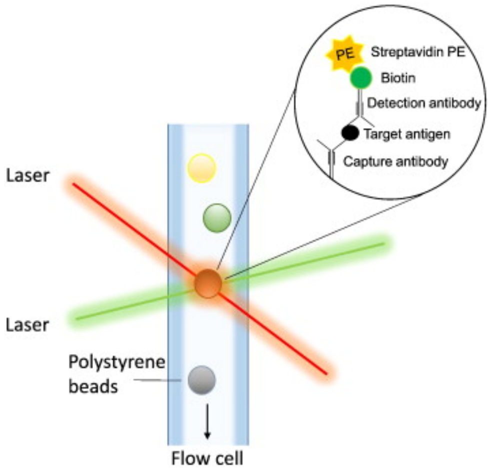


Figure 1.2: Representation of the Luminex® multiplex bead-array technology.

After evaluation of different commercial kits (Moncunill et al., 2013), the Cytokine Human Magnetic 30-Plex Panel from Life Technologies® was chosen to analyze the cellular component of the Mal067 study. This assay includes the following cytokines, chemokines and growth factors: epidermal growth factor (EGF), Eotaxin, fibroblast growth factor (FGF), granulocyte colony-stimulating factor (G-CSF), granulocyte-macrophage colony-stimulating factor (GM-CSF), hepatocyte growth factor (HGF), interferon (IFN)-α, IFN-γ, interleukin (IL)-1RA, IL-1β, IL-2, IL-2R, IL-4, IL-5, IL-6, IL-7, IL-8, IL-10, IL-12(p40/p70), IL-13, IL-15, IL-17, IFN-γ induced protein (IP-10), monocyte chemoattractant protein (MCP-1), monokine induced by IFN-γ (MIG), macrophage inflammatory protein (MIP)-1α, MIP-1β, regulated on activation, normal T cell expressed and secreted (RANTES), tumor necrosis factor (TNF), and vascular endothelial growth factor (VEGF). Some of these cytokines, chemokines and growth factors are expected to be highly, medium and low correlated given the nature of the immune response.

1.3 Mal067 study

The study that has motivated the present PhD thesis is the Mal067 immune correlates of protection. The Mal067 study is nested within the Mal055 vaccine trial, a multicenter, double blind, Phase III, randomized controlled trial aiming to evaluate the efficacy, safety and immunogenicity of the RTS,S malaria vaccine candidate. In Figure 1.3 there is a schematic representation of the Mal067 study. The participants of the study were vaccinated with either RTS,S or a comparator vaccine, i.e., rabies vaccine for the children (5-17 months old) or meningococcal C conjugate vaccine for infants (6-12 weeks old), at four timepoints during the study's follow-up time. In this thesis, we have used information based on samples collected 30 days after the third dose of the vaccine (M3). Moreover, we focus on samples collected from a subset of children who had a sample available at M3, and were followed for up to one year after M3.

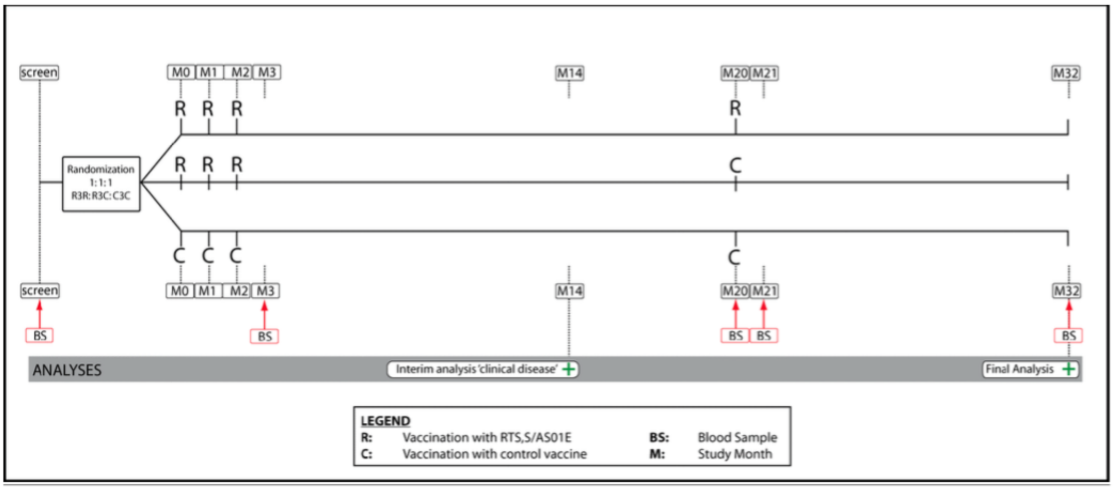


Figure 1.3: Schematic representation of sample collection and study subject follow-up in the Mal055 study.

Each obtained sample's participant was stimulated with different antigens to evaluate different components of the immune response. In this analysis, we focus on the stimulation using the Apical Membrane Antigen 1 (AMA1) recombinant protein, resuspended in culture medium with dimethyl sulfoxide (DMSO). The concentration of markers in supernatant of immune cells of each sample after stimulation with the AMA1 antigen and with DMSO (a negative control) were measured. Therefore, each participant has quantified levels of marker after AMA1 and DMSO and the ratio AMA1/DMSO that is used to evaluate responses specific to AMA1 after "controlling" for non-specific responses. The data is highly skewed, and thus, logarithm transformed.

All malaria cases detected 1-12 months post-vaccination (M3) that met the according to protocol criteria were studied. Malaria cases were defined as subjects who sought care at a health facility and had any *P.falciparum* asexual parasitaemia detected by blood smear. Investigators conducted all assays blinded to vaccine status.

1.4 Naturally acquired immunity

The importance of AMA1 stimulation is related to what is known as *naturally acquired immunity.* Across sub-Saharan Africa where the disease is holoendemic, most people are almost continuously infected by *P.falciparum*, and the majority of infected adults rarely experiment overt disease. The levels of parasitemia would be lethal to malaria-naive visitor but for them does not affect their daily routines. Studies have shown that anti-AMA1 antibodies tend to be present in those who have acquired natural immunity to malaria (Courtin et al., 2009, Udhayakumar et al., 2001). Moreover, repeated natural exposure to the malaria parasite often leads to high titres[6] of IgG (the most common type of antibody) to AMA1.

During malaria infection, the innate immune system contributes by limiting growth of the parasite (Stevenson and Riley, 2004, Inoue et al., 2013) and can influence development of adaptative immunity. The innate response involves a complex interplay between different cell types that include antigen presenting cells, such as dendritic cells[7], natural killer cells[8], monocytes/-macrophages[9], natural killer T cells, and $\gamma\delta$ T cells[10]. If unregulated, this inflammatory response can also contribute to the pathology that is associated with malaria infection (Stevenson et al., 2011, Butler et al., 2013, Perez-Mazliah and Langhorne, 2015). It is generally accepted that IFN-γ, TNF and IL-10 antigens play a fundamental role in the response to the parasite and control of clinical disease (Stanisic and Good, 2016).

The mechanisms of the RTS,S-induced protection and the reasons for the moderate efficacy remain unclear. Identifying immune correlates of protection and understanding the NAI-RTS,S interaction as some authors suggest (Sutherland et al., 2007, Wipasa and Riley, 2007, Doolan et al., 2009) can help improve RTS,S and rationally design the next generation of malaria vaccines.

[6] A titer is a way of expressing concentration and corresponds to the highest dilution factor that still yields a positive reading.

[7] Are a heterogeneous cell type and are considered a link between innate and adaptative immune responses. They can also enhance the function of other innate immune cells.

[8] Play an important role in the innate immune response against parasitized erythrocytes via interaction with dendritic cells and secretion of cytokines.

[9] Small inflammatory molecules produced by these cell types may also contribute directly killing the parasite.

[10] Play an essential role in the clearance of the blood stage parasites in rodent models.

Chapter 2

Survival Analysis

This chapter presents the main concepts, functions, and models classically used in survival analysis. Section 2.1 describes the basic functions in survival analysis and Section 2.2 introduces the concept of censoring. Three of the most used parametric models for the time-to-event data in the biomedical field are presented in Section 2.3. Section 2.4 reviews non-parametric Kaplan-Meier estimator and semi-parametric Cox proportional hazards model.

2.1 Basic concepts in survival analysis

This section describes the basic aspects of univariate survival data. In survival analysis we consider a random variable X, non-negative, usually representing the time for an event of interest, be the subject death or the subject presenting with malaria. We have restricted to the case of continuous X, since it is the most commonly used. All functions presented are defined over the interval $[0, \infty)$. The probability density function of X is denoted by f_X. The distribution of a random variable is completely and uniquely determined by its probability density function but there are many other notions which are useful in the survival analysis context. An important one is the cumulative distribution function of X, which is defined by

$$F_X(x) = \text{Prob}(X < x) = \int_0^x f_X(s)ds$$

Nonetheless, in survival analysis, one usually is interested in the probability of an individual to survive to time x, which is given by the survival function

$$S_X(x) = 1 - F(X) = \text{Prob}(X \geq x) = \int_x^\infty f_X(s)ds$$

The major notion in survival analysis is the hazard function λ_X which is defined by

$$\lambda_X(x) = \lim_{\vartriangle \to 0} \frac{\text{Prob}(x \leq X < x + \vartriangle |X \geq x)}{\vartriangle} = \frac{f_X(x)}{1 - F_X(x)}$$

The hazard function characterizes the risk of dying changing over time. It specifies the instantaneous failure rate at time x, given that an individual survives until x. Sometimes is useful to deal with the cumulative (or integrated) hazard function

$$\Lambda_X(x) = \int_0^x \lambda_X(s)ds$$

Applying the notion of the Laplace transformation of a random variable X is helpful to derive to equivalent expressions. So other useful equivalences are

$$\Lambda_X(x) = \int_0^x \lambda_X(s)ds = \int_0^x \frac{f_X(s)}{1 - F_X(s)} ds = -\log(1 - F_X(x))$$

for the cumulative hazard function and

$$S_X(x) = 1 - F_X(x) = \exp\left(-\int_0^x \lambda_X(s)ds\right) = \exp(-\Lambda_X(x))$$

for the survival function. Another important concept derived from the survival function is the conditional survival function

$$S_{X,z}(x) = \frac{S(x+z)}{S(x)} \tag{2.1}$$

which denotes the conditional probability of surviving $x + z$ years, given that the participant is still alive at x.

2.2 Censoring

Usually it is unfeasible to follow a participant over a long period of time either because the study ends before the event has occurred or the participant is lost to follow-up. This partially observed information is known in the survival framework as censoring. A censored observation contains only partial information about the variable of interest. There are different types of censoring, here we consider type I right censoring[1] (Figure 2.1).

Let $X_1, X_2, \ldots, X_n$ be i.i.d. survival times with cumulative distribution function F_X and let $Y_1, Y_2, \ldots, Y_n$ be i.i.d, censoring times with cumulative distribution function G_Y. We assume that F_X and G_Y are absolutely continuous. Furthermore, let f_X and g_Y be the probability functions with respect F_X and G_Y. We only observe the pair $(T_1, \delta_1), (T_2, \delta_2), \ldots, (T_n, \delta_n)$ where $T_i = \min(X_i, Y_i)$ is the follow-up time of the individual i and the censoring indicator is defined as

$$\delta_i = \begin{cases} 1 & \text{if} \quad X_i \leq Y_i \\ 0 & \text{if} \quad X_i > Y_i \end{cases}$$

Therefore, for each observation i the (T_i, δ_i) pair is observed and informs whether time-to-censoring or time-to-event is observed. Generally, it is assumed that Y is independent from X, i.e., the censoring mechanism is independent of the survival time and individuals censored at any time t are representative of all individuals at that same time. It must be remarked that the cumulative distribution function of the non-censored observations (while discarding the censored observations of the sample) is not F_X:

$$\begin{aligned} \text{Prob}(T < t, \delta = 1) = \text{Prob}(X < t, X \leq Y) &= \int_{x<t} \int_{x \leq y} f_X(x) g_Y(y) dx dy = \\ = \int_{x<t} f_X(x) \left(\int_{x \leq y} g_Y(y) dy \right) dx &= \int_{x<t} f_X(x)(1 - G_Y(x))dx \end{aligned} \tag{2.2}$$

[1] Right censoring occurs when an observation leaves the study before an event occurs, or the study ends before the event has occurred. Type I censoring occurs if given a set number of observations, the study stops at a predetermined time, at which point any observations remaining are right-censored.

As can be seen (2.2) is different from $F_X(t)$.

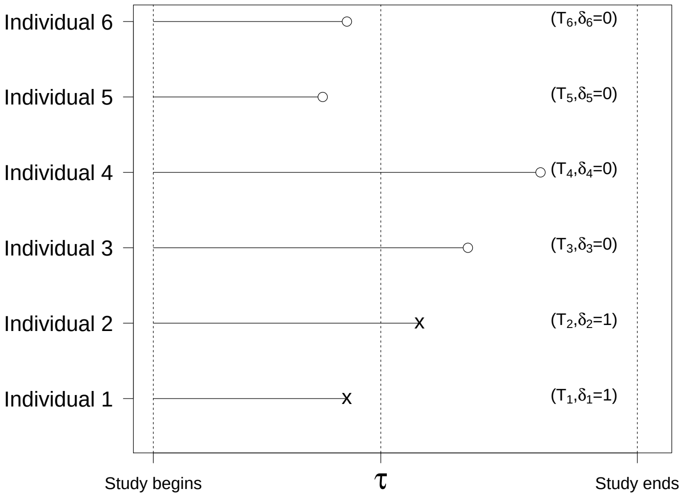


Figure 2.1: Example of right censoring patterns; τ: time in which prediction and estimation of parameters are performed; $\circ$: right censored data; $\times$: event; T: observed time; δ: censoring indicator.

2.3 Parametric models

Any distribution of non-negative random variables could be used to describe time-to-event data. The exponential and Weibull models have been commonly used in the survival literature, but other parametric models can also be used, including the log-normal, gamma and Gompertz distribution. The main advantage of parametric models, such as the mentioned, exponential and Weibull, is that they have closed form expressions for survival and hazards functions. We describe some of the standard failure time models and their main properties.

2.3.1 Exponential distribution

The exponential model,

$$X \sim Exp(\lambda)$$

is the simplest parametric model. Assumes a constant hazard over time, thus, this model assumes memoryless time-to-event: the probability of dying within a particular time interval depends only on the length of the interval but not on the location on this interval. This assumption means that the distribution of $X - x$ conditional on $X \geq x$ is the same as the original distribution. It holds that

$$\text{Prob}(x \leq X < x + \triangle \,|X \geq x) = \text{Prob}(X < \triangle) \tag{2.3}$$

for any positive $\triangle$. As a result, the exponential distribution is the sole distribution that is not influenced by the definition of time zero. The parameter λ attains all positive values; the distribution with $\lambda = 1$ is called the unit or standard exponential. The exponential distribution has density, survival and hazard function as follows:

$$\begin{aligned} \text{Probability density function:} &\quad \lambda \exp(-\lambda x) \\ \text{Survival function:} &\quad \exp(-\lambda x) \\ \text{Hazard function:} &\quad \lambda \\ \text{Mean:} &\quad \frac{1}{\lambda} \\ \text{Variance:} &\quad \frac{1}{\lambda^2} \end{aligned} \tag{2.4}$$

Exponential models are very sensitive to minor violations of the assumption of constant hazard over time and memoryless time-to-event. As a characteristic, both the mean and standard deviation are based on a single parameter used to describe the function of time.

2.3.2 Weibull distribution

The Weibull model is an important generalization of the exponential model with two positive parameters. The second parameter in the model allows flexibility of the model and different shapes of the hazard function.

$$X \sim Weibull(\lambda, \gamma)$$

The main functions related to the Weibull distribution are

$$\begin{aligned} \text{Probability density function:} &\quad \lambda\gamma x^{\gamma-1} \exp(-\lambda x^{\gamma}) \\ \text{Survival function:} &\quad \exp(-\lambda x^{\gamma}) \\ \text{Hazard function:} &\quad \lambda\gamma x^{\gamma-1} \\ \text{Mean:} &\quad \lambda^{-\frac{1}{\gamma}\Gamma(1+\frac{1}{\gamma})} \\ \text{Variance:} &\quad \lambda^{-\frac{2}{\gamma}} \left(\Gamma(1+\frac{2}{\gamma}) - \Gamma(1+\frac{1}{\gamma})^2 \right) \end{aligned} \tag{2.5}$$

where $\lambda > 0$, $\gamma > 0$ and Γ denotes the Gamma function with $\Gamma(r) = \int_0^\infty s^{r-1} \exp(-s) ds$ $(r > 0)$. As a property the hazard function decreases monotonously for $\gamma < 1$, is constant for $\gamma = 1$ (the exponential distribution is obtained) and it monotonously increases for $\gamma > 1$.

2.3.3 Gompertz distribution

The Gompertz distribution has been widely used in biology and in demography. A random variable follows a Gompertz distribution,

$$X \sim Gompertz(\lambda, v)$$

with parameters $\upsilon > 0$ and $\lambda > 0$, if the following relations hold:

$$\begin{aligned} \text{Probability density function:} \quad & \upsilon \exp(\lambda x) \exp\left(\frac{1}{\lambda}(1-\exp(\lambda x))\right) \\ \text{Survival function:} \quad & \exp\left(\frac{\upsilon}{\lambda}(1-\exp(\lambda x)\right) \\ \text{Hazard function:} \quad & \upsilon \exp(\lambda x) \\ \text{Mean:} \quad & \frac{1}{\upsilon} G\left(\frac{\upsilon}{\lambda}\right) \end{aligned} \tag{2.6}$$

where $G(r) = \int_r^\infty \frac{1}{y}\exp(-y)dy$. The hazard function is increasing from υ at time zero to ∞ at time ∞.

2.4 Non-parametric and semi-parametric models

For parametric inference, it is necessary to make assumptions about the distribution of times of event whereas non-parametric models avoid these assumptions. The simplest non-parametric estimate of a distribution function is the empirical distribution function. Two major developments in survival analysis (with censored observations) were the Kaplan-Meier estimator (Kaplan and Meier, 1958) and the proportional hazards model (Cox, 1972).

2.4.1 Kaplan-Meier estimator

A useful way of characterizing the survival function is to compute the empirical survival function. In the absence of censoring, the empirical survival function at time t is the ratio of survivors at time t and sample size n. This step function decreases by $\frac{1}{n}$ just after each observed event (assuming no ties). When dealing with censored data it is convenient to account for this partial information. Let $T_{(1)} < T_{(2)} < \ldots < T_{(n)}$ be the order statistics of the observed time of $T_1, T_2, \ldots, T_n$ and defining $\delta_{(i)}$ to be the censoring indicator value associated to $T_{(i)}$, that is, $\delta_{(i)} = \delta_j$ if $T_{(i)} = T_j$. Note that $\delta_{(1)}, \delta_{(2)}, \ldots, \delta_{(n)}$ are not ordered. The Kaplan-Meier estimator of the survival function can be obtained applying the product:

$$\hat{S}(t) = \prod_{i:T_{(i)}<t} \left(1 - \frac{\delta_{(i)}}{n-i+1}\right) \tag{2.7}$$

where n is the total number of individuals. This is a decreasing step function that changes only at event time. $\hat{S}$ never reduces to zero if the largest observation is censored and, as a result, $\hat{S}$ is usually left unspecified for $t > t_{(n)}$.

2.4.2 Proportional hazards model

The Cox proportional hazards model is the most popular model to analyze survival data. It is defined in terms of the hazard function:

$$\lambda(t|\boldsymbol{x}_i) = \lambda_0(t)\exp(\langle \boldsymbol{x}_i, \boldsymbol{\beta} \rangle) \tag{2.8}$$

where $\lambda(t|\boldsymbol{x}_i)$ is the hazard at time t of an observation i with covariates vector $\boldsymbol{x}_i$, $\lambda_0(t)$ is the baseline hazard function and $\boldsymbol{\beta}$ is the vector of coefficients of the model. The model assumes a baseline hazard that is common to all observations in the study population. In this model, the hazard of a subject increases multiplicatively with covariates.

The baseline hazard can be modelled in parametric or in semi-parametric models, as in Cox proportional hazards model. In the semi-parametric model, the baseline hazard does not need to be specified and the optimization function is based on a partial likelihood. Proportional hazards model is more robust to outliers than other models because the model uses only the rank ordering of the failure and censoring times.

To define the partial likelihood let $t_{(1)} < t_{(2)} < t_{(3)} < \ldots < t_{(k)}$ be the unique ordered times when an event is presented in a sample of n subjects, assuming that there are no repeated time events. Considering that the set of individuals at risk of having an event at time $t_{(j)}$ is defined by $R_j = R(t_{(j)}) = \{i : T_i \geq t_{(j)}\}$ and is comprised by all observations censored or with an event by time $t_{(j)}$. The conditional probability that individual i present an event at $t_{(i)}$, given that the subject is in the risk set R_i, and given that exactly one failure occurs at $t_{(i)}$ is

$$\begin{aligned}\text{Prob}(i \text{ have and event at } t_{(i)}|R_i \text{ and one event at } t_{(i)}) &= \frac{\lambda_0(t_{(i)})\exp(\langle \boldsymbol{x}_{(i)}, \boldsymbol{\beta}\rangle)}{\sum_{j\in R_i}\lambda_0(t_{(i)})\exp(\langle \boldsymbol{x}_j, \boldsymbol{\beta}\rangle)} \\ &= \frac{\exp(\langle x_{(i)}, \boldsymbol{\beta}\rangle)}{\sum_{j\in R_i}\exp(\langle \boldsymbol{x}_j, \boldsymbol{\beta}\rangle)}\end{aligned}$$

The likelihood function can be computed multiplying these individual likelihoods over all event times. This is the known as the *partial likelihood* for parameters $\boldsymbol{\beta}$, taking into account the censored observations, is expressed as:

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^{n}\left(\frac{\exp(\langle \boldsymbol{x}_i, \boldsymbol{\beta}\rangle)}{\sum_{j\in R_i}\exp(\langle \boldsymbol{x}_j, \boldsymbol{\beta}\rangle)}\right)^{\delta_i} \tag{2.9}$$

where δ_i is the censoring indicator variable of observation i. Applying the logarithm transformation to equation (2.9) we obtain the log partial likelihood:

$$\log(\mathcal{L}(\boldsymbol{\beta})) = \sum_{i=1}^{n}\delta_i\left(\langle \boldsymbol{\beta}, \boldsymbol{x}_i\rangle - \log\sum_{j\in R_i}\exp(\langle \boldsymbol{x}_j, \boldsymbol{\beta}\rangle)\right) \tag{2.10}$$

Maximizing the expression (2.10), with Newton-Raphson algorithm, we obtain the maximum partial likelihood estimators $\hat{\boldsymbol{\beta}}$ that are asymptotically non biased, efficient and normally distributed.

Probability of having an event

Sometimes the information of interest is the probability of having an event during an interval of time. The estimators of the hazard function, cumulative hazard function and survival function for an observation i, with a vector of covariates $\boldsymbol{x}_i$ are, respectively:

$$\hat{\lambda}(t|\boldsymbol{x}_i) = \hat{\lambda}_0(t)\exp(\langle \hat{\boldsymbol{\beta}}, \boldsymbol{x}_i\rangle)$$

$$\begin{aligned}\hat{\Lambda}(t|\boldsymbol{x}_i) &= \hat{\Lambda}_0(t)\exp(\langle\hat{\boldsymbol{\beta}},\boldsymbol{x}_i\rangle)\\ \hat{S}(t|\boldsymbol{x}_i) &= (\hat{S}_0(t))^{\exp(\langle\hat{\boldsymbol{\beta}},\boldsymbol{x}_i\rangle)}\end{aligned} \tag{2.11}$$

where $\hat{\lambda}_0(t), \hat{\Lambda}_0(t)$ and $\hat{S}_0(t)$ can be obtained through the Kalbfleisch-Prentice (Kalbfleisch and Prentice, 1973) or the Breslow (Breslow, 1972) estimators. From (2.11) we can derive the probability of having an event before t as:

$$\hat{F}(t|\boldsymbol{x}_i) = 1 - \left((\hat{S}_0(t))^{\exp(\langle\hat{\boldsymbol{\beta}},\boldsymbol{x}_i\rangle)}\right)$$

Main assumptions

The Cox proportional hazards model is more flexible than parametric models such as the accelerated failure time models: it does not rely on any assumption about the distribution of the time-to-event variable and makes no assumption on the shape of the underlying hazard function. The assumptions of the Cox proportional hazards model are the following (Therneau, TM. and Grambsch, PM., 2000):

- The time-to-event variable is a continuous random variable. Oftentimes this assumption is violated and for these cases, there are four approaches to handle ties in the Cox model: Breslow, Efron, exact partial likelihood and averaged likelihood method.
- Linearity and additivity of the predictors with respect to log-hazard or log-cumulative hazard.
- Proportionality of the hazards over time (or constancy of the hazard ratio).
- The data is i.i.d., however there are extensions of the model allowing for non-independent data.

Another important requirement to obtain unbiased estimations is related to the minimum number of events per variable (EPV). As it has been shown in the expression (2.10) the partial likelihood depends on the number of events and not on the total sample size. Simulations conducted by Concato et al. (1995) and by Peduzzi et al. (1995) suggest that the minimum EPV for a proportional hazards model is 10. Vittinghoff and McCulloch (2007) suggest that results including 5 to 9 EPV are comparable to those including 10 to 16 EPV. However, the three studies agree that results should be interpreted with caution and further analyses are necessary to understand the obtained results, in situations where confounding cannot be addressed without violating the minimum EPV.

Proportional hazards assumption

A key assumption of the Cox model is proportional hazards. That is, with time-fixed covariates, the relative hazard for any two subjects i and j obey the relationship

$$\frac{\lambda(t|\boldsymbol{x}_i)}{\lambda(t|\boldsymbol{x}_j)} = \frac{\lambda_0(t)\exp(\langle\boldsymbol{x}_i,\boldsymbol{\beta}\rangle)}{\lambda_0(t)\exp(\langle\boldsymbol{x}_j,\boldsymbol{\beta}\rangle)} = \frac{\exp(\langle\boldsymbol{x}_i,\boldsymbol{\beta}\rangle)}{\exp(\langle\boldsymbol{x}_j,\boldsymbol{\beta}\rangle)} \tag{2.12}$$

which is independent of time. Furthermore, the relationship holds individually for each variable in the model, as can be seen by choosing hypothetical subjects such that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ differ in a single variable.

We can check the proportional hazards assumption by testing the scaled Schoenfeld residuals. Extending expression (2.8) to a time-dependent coefficient model we obtain

$$\lambda(t|\boldsymbol{x}_i) = \lambda_0(t)\exp(\langle \boldsymbol{x}_i, \boldsymbol{\beta}(t)\rangle) \tag{2.13}$$

From expression (2.13) we can conclude that proportionality of hazards holds when $\boldsymbol{\beta}(t) = \boldsymbol{\beta}$ for any time t. Therefore, one way to test if a variable violates the proportional hazards assumption is to plot $\boldsymbol{\beta}(t)$ as a function of t. Therneau, TM. and Grambsch, PM. (2000) show that if $\hat{\beta}_k$ is the estimated coefficient by the Cox model for covariate k and assuming no tied events, then

$$E(\hat{r}^*_{ik}) + \hat{\beta}_k \approx \beta_k(t_i) \tag{2.14}$$

where $\hat{r}^*_{ik}$ is the scaled Schoenfeld residual for the ith observation and covariate k. The Schoenfeld residual for ith observation on the kth covariate using Hosmer et al. (2011) notation, is expressed as

$$\hat{r}_{ik} = \delta_i(x^k_i - \hat{\bar{x}}^k_{w_i}) \tag{2.15}$$

where δ_i is the censoring indicator for observation i and

$$\hat{\bar{x}}^k_{w_i} = \frac{\sum^n_{j\in R_i} x^k_j \exp(\langle \hat{\boldsymbol{\beta}}, \boldsymbol{x}_j\rangle)}{\sum^n_{j\in R_i} \exp(\langle \hat{\boldsymbol{\beta}}, \boldsymbol{x}_j\rangle)}$$

is the estimator of the risk set conditional mean of the covariate.

Therneau, TM. and Grambsch, PM. (2000) suggest that scaling the Schoenfeld residuals by an estimator of its variance yields a residual with greater diagnostic power than the unscaled residuals; this scaled version is computed easily by an approximation suggested by the same authors that is expressed as

$$\hat{\boldsymbol{r}}^*_i = m\widehat{Var}(\hat{\boldsymbol{\beta}})\hat{\boldsymbol{r}}_i$$

where m is the total number of events, $\widehat{Var}(\hat{\boldsymbol{\beta}})$ is the covariance matrix of the estimated coefficients and $\hat{\boldsymbol{r}}_i$ is the vector of Schoenfeld residuals with as many components as covariates for the ith observation.

For a given covariate k we can plot the scaled Schoenfeld residuals as a function of time. In a proportional hazards scenario we would expect all residuals to fall around the estimated coefficient value $\hat{\beta}_k$. Usually a line is fit and a test for zero slope is performed, being a nonzero slope evidence against proportional hazards.

Strategies for non-proportionality

As stated by Therneau, TM. and Grambsch, PM. (2000) there are several possible model failures that may appear as time-varying coefficients but would be dealt better by another approach. These include the omission of an important covariate and incorrect functional form for a covariate. There are two main approaches to deal with non-proportionality:

- Time-dependent covariates: to model a time-varying effect, one can always create a time-dependent covariate so that
$$\langle \boldsymbol{\beta}(t), \boldsymbol{x} \rangle = \langle \boldsymbol{\beta}, \boldsymbol{x}^*(t) \rangle$$
where $\boldsymbol{x}^*(t)$ could be based on theoretical considerations or could be a function inspired by the smoothed residual plot.

- Stratification: the second method that enables to handle non-proportional hazards is stratification. The main idea is to split the whole sample into subgroups on the basis of categorical variable which is called stratification variable and re-estimate the model, letting the baseline hazard function differ between these subgroups.
$$\lambda_s(t|\boldsymbol{x}_i) = \lambda_{0s}(t) \exp(\langle \boldsymbol{x}_i, \boldsymbol{\beta} \rangle)$$
where $s = 1, 2, \dots, S$ and S is the total number of subgroups created on the basis of stratification variable.

Chapter 3

Support Vector Machines

The SVM methodology was developed by Cortes and Vapnik (1995). Intuitively, these models are based on discriminating two classes of observations by a linear decision surface (hyperplane) maximizing the distance between the hyperplane and the individual observations. If the classes are not separable by a linear surface, a non-linear transformation can be obtained through mapping the data on a different dimension space (feature space), usually a much higher dimensional. This yields a non-linear decision boundary in the original space (input space). By the use of a kernel function, it is possible to construct the separating hyperplane without explicitly carrying out the map into feature space.

To fully understand the rational behind the SVM some basic concepts in terms of geometry are described in Appendix A.1. Some preliminary concepts related to the optimization procedures used in SVM are reviewed in Section 3.1. From Section 3.2 to 3.5 the soft margin SVM and all main related concepts (kernels, parameters and assumptions) are described. Finally, main SVM extensions unrelated to survival-SVM, are briefly described in Section 3.6.

3.1 Optimization preliminaries

All concepts presented in this section are based on the book by Scholkopf and Smola (2001) about kernel methods. The optimization function of the original SVM and most of the SVM extensions is convex quadratic. The properties of convex optimization functions are reviewed in the following.

3.1.1 Convex optimization concepts

Minimization of a convex objective function on a convex set exhibits one global minimum.

Definition 1 (Convex set). *A set X in vector space is called convex if for any $x, x' \in X$ and any $\lambda \in [0,1]$ we have*

$$\lambda x + (1-\lambda)x' \in X \tag{3.1}$$

Definition 2 (Convex function)**.** *A function f defined on a set X (not need to be convex itself) is called convex if, for any $x, x' \in X$ and any $\lambda \in [0,1]$ such that $\lambda x + (1 - \lambda x') \in X$, we have*

$$f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda)f(x') \tag{3.2}$$

A function f is called strictly convex if for $x \neq x'$ and $\lambda \in (0,1)$ is a strict inequality.

A convenient definition of a convex set is based on the definition of the *below-sets* of convex functions, such as the sets for which $f(x) \leq c$, for instance.

Lemma 1 (Convex sets as below-sets)**.** *Denote by $f : \mathcal{X} \mapsto \mathbb{R}$ a convex function on a convex set $\mathcal{X}$. Then the set*

$$X := \{x | x \in \mathcal{X} \text{ and } f(x) \leq c\}, \text{ for all } c \in \mathbb{R}, \tag{3.3}$$

is convex.

Theorem 1 (Minima on convex sets)**.** *If the convex function $f : \mathcal{X} \mapsto \mathbb{R}$ has a minimum on a convex set $X \subset \mathcal{X}$, then its arguments $x \in \mathcal{X}$, for which the minimum value is attained, form a convex set. Moreover, if f is strictly convex, then this set will contain only one element.*

3.1.2 Convex quadratic optimization problems concepts

The general expression of a quadratic problem is given by

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{minimize}} \quad & \frac{1}{2}\boldsymbol{z}^\top Q \boldsymbol{z} + \boldsymbol{c}^\top \boldsymbol{z}, \\ \text{subject to} \quad & A\boldsymbol{z} + \boldsymbol{d} \leq 0 \end{aligned} \tag{3.4}$$

where Q is a strictly positive definite matrix, $\boldsymbol{z}$, $\boldsymbol{c} \in \mathbb{R}^m, A \in \mathbb{R}^{n \times m}$, and $\boldsymbol{d} \in \mathbb{R}^n$. Note that this is a differentiable convex optimization problem. To address the problem (3.4) we have to construct the Lagrangian with the corresponding multipliers $\boldsymbol{\alpha} \in \mathbb{R}^n$ with $\boldsymbol{\alpha} \geq 0$, expressed as

$$L(\boldsymbol{z}, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{z}^\top Q \boldsymbol{z} + \boldsymbol{c}^\top \boldsymbol{z} + \boldsymbol{\alpha}^\top (A\boldsymbol{z} + \boldsymbol{d}) \tag{3.5}$$

3.1.3 Karush-Kuhn-Tucker conditions

Once we have constructed the Lagrangian we have to obtain the Karush-Kuhn-Tucker (KKT) conditions (Boser et al., 1992, Cortes and Vapnik, 1995, Karush, 1939, Kuhn and Tucker, 1951). These conditions are necessary for an optimal solution to any non-linear programming problem. They are sufficient conditions when the primal objective and inequality constraints are convex and continuously differentiable, and each equality constraint is an affine function. These conditions hold for the SVM optimization problem and assure an optimal SVM solution

$$\frac{\partial L(\boldsymbol{z}, \boldsymbol{\alpha})}{\partial \boldsymbol{z}} = Q\boldsymbol{z} + A^\top \boldsymbol{\alpha} + \boldsymbol{c} = 0, \text{ (Saddle point in } \boldsymbol{z}) \tag{3.6}$$

$$\frac{\partial L(\boldsymbol{z}, \boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = A\boldsymbol{z} + \boldsymbol{d} \leq 0, \text{ (Saddle point in } \boldsymbol{\alpha}) \tag{3.7}$$

$$\boldsymbol{\alpha}^\top (A\boldsymbol{z} + \boldsymbol{d}) = 0, \text{ (Vanishing KKT-Gap)} \tag{3.8}$$

In order to compute the dual of (3.4) we have to eliminate $\boldsymbol{z}$ from (3.5) and write it as function of $\boldsymbol{\alpha}$.This gives the following expression:

$$\begin{aligned} L(\boldsymbol{z}, \boldsymbol{\alpha}) &= -\frac{1}{2}\boldsymbol{z}^\top Q \boldsymbol{z} + \boldsymbol{\alpha}^\top \boldsymbol{d} & (3.9)\\ &= -\frac{1}{2}\boldsymbol{\alpha}^\top A Q^{-1} A^\top \boldsymbol{\alpha} + \left[\boldsymbol{d} - \boldsymbol{c}^\top Q^{-1} A^\top\right] \boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{c}^\top Q^{-1} \boldsymbol{c} & (3.10) \end{aligned}$$

In (3.9) we have used (3.6) and (3.8) directly, whereas in order to eliminate $\boldsymbol{z}$ completely from (3.10) we solved (3.6) for $\boldsymbol{z} = -Q^{-1}(\boldsymbol{c} + A^\top \boldsymbol{\alpha})$. Ignoring the constant term leads to the dual quadratic optimization problem

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & -\frac{1}{2}\boldsymbol{\alpha}^\top A Q^{-1} A^\top \boldsymbol{\alpha} + \left[\boldsymbol{d} - \boldsymbol{c}^\top Q^{-1} A^\top\right] \boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{\alpha} \geq 0 \end{aligned} \tag{3.11}$$

The constraints in the dual problem (3.11) are simpler than in the primal (3.4) and, thus, this is the problem that it is solved.

3.1.4 Kernels preliminary concepts

One of the advantages of the SVM models, as other kernel-based methods, is the possibility of efficiently applying non-linear methods using kernels. The kernels that are used across this thesis are the class of kernels k that correspond to dot products in feature space $\mathcal{H}$ via a map ϕ

$$\begin{aligned} \phi : & \mathcal{X} \to \mathcal{H} \\ & \boldsymbol{x} \mapsto \phi(\boldsymbol{x}), \end{aligned} \tag{3.12}$$

that is

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle \tag{3.13}$$

Some other concepts and definitions need to be established.

Kernel and kernel matrix definitions

Definition 3 (Gram matrix)**.** *Given a function $k : \mathcal{X}^2 \to \mathbb{K}$ (where $\mathbb{K} = \mathbb{C}$ or $\mathbb{K} = \mathbb{R}$) and patterns $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$, the $n \times n$ matrix K with elements*

$$K_{ij} := k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

is called Gram matrix (or kernel matrix) of k with respect to $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$

Definition 4 (Positive definite matrix.)**.** *A complex $n \times n$ matrix K satisfying:*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i \bar{c}_j K_{ij} \geq 0 \tag{3.14}$$

for all $c_i \in \mathbb{C}$ is called positive definite ($\bar{c}$ denotes complex conjugation, for real numbers it has no effect). Similarly, a real symmetric $n \times n$ matrix K satisfying (3.14) for all $c_i \in \mathbb{R}$ is called positive definite.

Definition 5 (Positive definite kernel.)**.** *Let $\mathcal{X}$ be a non empty set. A function k on $\mathcal{X} \times \mathcal{X}$ which for all $n \in \mathbb{N}$ and all $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ gives rise to a positive definite Gram matrix is called a positive definite kernel.*

Feature map

Assume that k is a real-valued positive definite kernel, and $\mathcal{X}$ a non empty set. We define a map from $\mathcal{X}$ into the space of functions mapping $\mathcal{X}$ into $\mathbb{R}$, denoted as $\mathbb{R}^{\mathcal{X}} := \{f : \mathcal{X} \to \mathbb{R}\}$, via

$$\begin{aligned} \phi :& \mathcal{X} \to \mathbb{R}^{\mathcal{X}} \\ & \boldsymbol{x} \mapsto k(\cdot, \boldsymbol{x}). \end{aligned}$$

Here, $\phi(\boldsymbol{x})$ denotes the function that assigns the value $k(\boldsymbol{x}, \boldsymbol{x}')$ to $\boldsymbol{x}' \in \mathcal{X}$, i.e., $\phi(\boldsymbol{x})(\cdot) = k(\cdot, \boldsymbol{x})$. This means, that each observation is turned into a function on the domain $\mathcal{X}$. In this sense, an observation is now represented by its similarity to all other points in the input domain $\mathcal{X}$.

Reproducing kernel Hilbert space

Given a positive definite kernel, we can use the functions $k(\boldsymbol{x}, \cdot)$, $\boldsymbol{x} \in \mathcal{X}$ to construct a normed space by defining an appropriate inner product. We define the vector space by taking linear combinations of the functions

$$f(\cdot) = \sum_{i=1}^{l} \alpha_i k(\boldsymbol{x}_i, \cdot)$$

for arbitrary $l \in \mathbb{N}$, $\alpha_i \in \mathbb{R}$ and $\boldsymbol{x}_i, \ldots, \boldsymbol{x}_l \in \mathbb{R}$. The inner product between f and $g = \sum_{j=1}^{r} \beta_j k(\boldsymbol{x}_j^{'}, \cdot)$ is defined as

$$\langle f, g \rangle = \sum_{i=1}^{l} \sum_{j=1}^{r} \alpha_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j^{'}).$$

This definition can be shown to satisfy the properties of an inner product (Scholkopf and Smola, 2001), namely:

$$\begin{aligned} \text{Symmetry} &: \langle f, g \rangle = \langle g, f \rangle \\ \text{Linearity} &: \langle af + bg, h \rangle = a\langle f, h \rangle + b\langle g, h \rangle \\ \text{Positive definiteness} &: \begin{cases} \langle f, f \rangle \geq 0 \\ \langle f, f \rangle = 0 \end{cases} \Rightarrow f = 0 \end{aligned}$$

With this definition, we have

$$\|f\|^2 = \langle f, f \rangle = \sum_{j=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$$

Another useful property of this space is the fact that k is a *reproducing kernel*, that is

$$f(\boldsymbol{x}) = \langle k(\boldsymbol{x}, \cdot), f \rangle$$

which follows from the definition of the inner product.

With the inner product on the vector space, we have obtained a *pre-Hilbert* space. We complete the space by adding the limit points of convergent sequences to form a Hilbert space, usually called a *reproducing kernel Hilbert space* (RKHS).

3.2 Soft margin support vector machines

Originally, SVM was defined based on a *hard margin* approach in which the observations could not be misclassified across classes. Later the hard margin approach was replaced by the *soft margin* approach allowing for a certain degree of misclassification. The soft margin SVM expression corresponds to

$$\begin{aligned} \underset{\boldsymbol{w},\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & \xi_i \geq 0, && i = 1,\dots,n \\ & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i, && i = 1,\dots,n \end{aligned} \tag{3.15}$$

where $\boldsymbol{w}$ is the weight vector of the hyperplane, $\boldsymbol{x}_i$ is the covariates vector of the observation i, $y_i \in \{\pm 1\}$ is the class label[1] associated to the $\boldsymbol{x}_i$, C is the regularization constant trading off the violations of the constraints and maximizing the overall margin and b is the *threshold* (*bias* term or *intercept*). The optimization problem (3.15) is quadratic thus with a minimum always observable and it is the primal optimization problem; to solve the dual problem we need to construct the Lagrange function

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\eta}) \;= \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1 - \xi_i - y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b)) - \sum_{i=1}\eta_i\xi_i \tag{3.16}$$

where $\alpha_i \geq 0$ and $\eta_i \geq 0$ for all $i = 1,\dots,n$, are the Lagrange multipliers. The Lagrangian has to be minimized with respect to the primal variables and maximized with respect the dual variables (a saddle point has to be found). Hence, at the saddle point, the derivatives of L with respect the primal variables must vanish:

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{w}} = 0 \quad & \Longrightarrow \quad \boldsymbol{w} - \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i = 0 \\ \frac{\partial L}{\partial b} = 0 \quad & \Longrightarrow \quad -\sum_{i=1}^{n}\alpha_i y_i = 0 \\ \frac{\partial L}{\partial \xi_i} = 0 \quad & \Longrightarrow \quad C - \alpha_i + \eta_i = 0 \end{aligned} \tag{3.17}$$

This translates into $\boldsymbol{w} = \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i$, the linear constraint $\sum_{i=1}^{n}\alpha_i y_i = 0$, and the box constraint $\alpha_i \in [0, C]$ arising from $\eta_i \geq 0$. Applying the derivatives (3.17) into (3.16) it yields the

[1] In the SVM for binary classification framework, classes are identified by +1 or -1 because of mathematical convenience.

dual optimization problem,

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle - \sum_{i=1}^{n}\alpha_i \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \qquad\qquad i = 1, \ldots, n \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \qquad\qquad i = 1, \ldots, n \end{aligned} \tag{3.18}$$

This optimization problem can be solved using several approaches since it has one equality constraint and one single vector. After finding the optimal $\boldsymbol{\alpha}$ vector it can be shown that the weights of the marginal hyperplane are

$$\boldsymbol{w} = \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i \tag{3.19}$$

and the decision function[2] is given by

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{n}\alpha_i\langle \boldsymbol{x}_i, \boldsymbol{x}\rangle y_i + b\right) \tag{3.20}$$

that allows predicting the class of a new observation $\boldsymbol{x}$. After estimating the $\boldsymbol{\alpha}$ vector and given any observation j, the b term can be computed as

$$b = y_j - \sum_{i=1}^{n}\alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \tag{3.21}$$

It must be noted that $\boldsymbol{w}$ vector lies in the span of the $\boldsymbol{x}_i$ and $f(\boldsymbol{x})$ with the form of equation (3.20) is an instance of the representer theorem (Kimeldorf and Wahba, 1971).

Theorem 2 (Representer theorem)**.** *Denote by $\Omega : [0, \infty) \to \mathbb{R}$ a strictly monotonic increasing function, by $\mathcal{X}$ a set, and by $c : (\mathcal{X} \times \mathbb{R}^2)^n \to \mathbb{R} \cup \{\infty\}$ an arbitrary loss function. Then each minimizer $f \in \mathcal{H}$ of the regularized risk functional:*

$$c((x_1, y_1, f(x_1)), \ldots, (x_n, y_n, f(x_n))) + \Omega(\|f\|^2_{\mathcal{H}})$$

admits a representation of the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{n}\alpha_i k(\boldsymbol{x}_i, \boldsymbol{x})$$

where $\mathcal{H}$ is the Reproducing Kernel Hilbert Space associated to the kernel k.

This theorem shows that solutions of a large class of optimization problems can be expressed as kernel expansions over the sample points. The significance of the representer theorem is that although we might be trying to solve an optimization problem in an infinite-dimensional space $\mathcal{H}$, containing linear combinations of kernels centered on arbitrary points of $\mathcal{X}$, it states that the solution lies in the span of n particular kernels, those centered on the training points.

[2] Not all α_i contributes to the decision function of a new given observation. Only those $\alpha_i > 0$ contribute to equation (3.20), the $\boldsymbol{x}_i$ with associated $\alpha_i > 0$ are known as *support vectors*.

3.3 Kernelizing the optimal margin hyperplane

Definitions presented in Section 3.2 are based on a SVM with linear classification borders. More general decision surfaces can be allowed by using non-linear kernels to perform a non-linearity transformation of the input data, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, into a high-dimensional feature space using a map $\phi : \boldsymbol{x}_i \mapsto \phi(\boldsymbol{x}_i)$ and then creating a linear separation in the newly defined space. This procedure can be justified using Cover's theorem (Cover, 1965). The theorem formalizes the intuition that the number of separations increases with the dimensionality and states that given a set of training data that is not linearly separable, one can with high probability transform it into a training set that is linearly separable by projecting it into a higher-dimensional space via some non-linear transformation. This can be done through application of the kernel trick that, basically, is the substitution of the described dot products in the expressions (3.18), (3.20) and (3.21) by

$$\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

where k is a positive definite kernel. The kernel trick states that, given an algorithm which is formulated in terms of a positive definite kernel k, one can construct an alternative algorithm by replacing k by another positive definite kernel k'; the dot product is a linear positive definite kernel and hence we can apply the kernel trick.

Everything that has been described for the linear case, i.e., the original input space, is also applicable to the non-linear case. Some of the most used positive definite kernels are:

- Linear:

  $$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle$$

  this kernel is the usual dot product.

- Polynomial:

  $$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle^d$$

  where $d \in \mathbb{N}$. The polynomial kernel of degree d computes a dot product in the space spanned by all monomials of degree d in the input coordinates.

- Inhomogeneous Polynomial:

  $$k(\boldsymbol{x}, \boldsymbol{x}') = (\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + c)^d$$

  similar to polynomial kernel where $d \in \mathbb{N}$ and $c \geq 0$.

- Gaussian:

  $$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\sigma \|\boldsymbol{x} - \boldsymbol{x}'\|^2)$$

  or

  $$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2})$$

  where $\sigma > 0$. One of its main characteristics is that the $k(\boldsymbol{x}, \boldsymbol{x}) = 1$.

Apart from the Gaussian, the linear kernel is also one of the kernels most used. There is no formal theory to guide the choice of the kernel to be used. Advantages and disadvantages of the two mentioned kernels are:

Linear

- There are no parameters in the linear kernel so there is no need to tune an extra parameter and, depending on the data and algorithm, using linear kernels can save computational cost and time.
- The feature space is in the same space of the original input space, making results more interpretable.
- When analyzing numerous variables, as compared to the number of observations, the linear kernels work well because the chances that the data are linearly separable increase, therefore there is no need to apply a non-linear kernel.
- The linear kernel is less prone to overfit the data.

Additionally the linear kernel should always be used when the data is linearly separable.

Gaussian

- The feature space induced by a Gaussian kernel is infinite dimensional (Appendix A.2). Therefore, it yields more flexible solutions than other kernels.
- The prediction error of non-linear kernel is better than or as good as the prediction error of a linear kernel (Keerthi and Lin, 2003). Therefore, in an uncertain scenario it is more conservative to use non-linear kernel.

Additionally the non-linear kernel should always be used when the data is not linearly separable.

Cross-validation and bootstrap approaches can also support decisions about the specific kernel to be used.

3.4 Tuning parameters

Some parameters included in the optimization problems are not known a priori and are not estimable through the SVM optimization process. Therefore, these parameters need to be tuned, i.e., go through a different estimation procedure. When using SVM with a linear kernel, there is only one parameter to be tuned: the misclassification parameter C. When using a Gaussian kernel other parameters need to be tuned increasing the complexity of the SVM model. The best value for tuning a parameter for a given problem is unknown but the criteria to choose it is based on the prediction accuracy for any new observation. Assessment of the prediction accuracy for a new data can be done through k-fold cross-validation over a grid of alternative combinations of parameter values (Table 3.1 shows a simple example of this approach).

There are several variants of the simple cross-validation design including nested cross-validation, coarse-fine grid search, and bootstrap-based resampling. Dealing with two parameters is relatively straightforward and fast, but the computational time can be substantially increased with more than two tuning parameters. Some work has been done to optimize the estimation of tuning parameters. Caputo et al. (2002) suggest that optimal values of the σ parameter for the Gaussian kernel range from 0.1 to 0.9 quantile of the $\|\boldsymbol{x}-\boldsymbol{x}'\|^2$ statistic based on a sample of the training set. Chapelle and Zien (2005) suggest a similar approach, i.e., to fix the default value of σ parameter and to assign to the C parameter the inverse of the empirical variance S^2 of the data in the feature space, defined as $S^2 = \sum_{i=1}^{n} K_{ii} - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} K_{ij}$, from a $n \times n$ kernel matrix K. Afterwards, they use multiples of the default value as search range in a grid-search using cross-validation.

Table 3.1: Example of using 3-fold cross-validation over a grid of 2 values for the C tuning parameter (1 or 2). Column (A) shows the training set partitions used to fit the models; column (B) shows the test set partitions in which accuracy is evaluated; column (C) shows the best parameters values, i.e., those with highest accuracy of the tuning parameter with highest accuracy (column [D]). Finally, in column (E), the average accuracy is displayed.

(A) Training set	(B) Testing set	(C) Best parameter	(D) Accuracy	(E) Average accuracy
P_1, P_2	P_3	1	89%	
P_1, P_3	P_2	2	84%	83%
P_2, P_3	P_1	1	75%	

3.5 Assumptions

SVM methods are mainly non-parametric, in the sense that no statistical distributions are assumed for neither parameters nor data. The main assumption related to the data is to be i.i.d.. The tuning parameters affect accuracy of the prediction of SVM models but these are not assumptions, just parameters. Previous work has tried to link the SVM models with classical probabilistic theory. Platt et al. (1999) suggested a method to fit a logit function to map SVM output to probabilities, this method is also known as *Platt's scaling*. Grandvalet et al. (2005) approximate the SVM objective function to a negative log-likelihood so the SVM output is translated into probability intervals for unbalanced classification problems. Franc et al. (2011) show that the SVM can be formulated as a maximum likelihood approach and the class can be estimated using probabilistic models, however this conclusion is based on linear SVM models without bias.

Other aspects that could affect the accuracy of the prediction are:

- Imbalance of classes: SVM models are sensitive to the data imbalance between classes. Veropoulos et al. (1999) show that the separating hyperplane of a SVM with an imbalanced dataset can be skewed towards the minority class, and this skewness can degrade the performance of that model with respect to the minority class. Some strategies to deal with imbalance of classes described in the literature are:

- Boundary movement: it is a post-processing method in which the b term in the SVM decision function is changed to trade false positives and false negatives.
- Biased penalties: Veropoulos et al. (1999) suggest using different penalty factors C^+ and C^- for positive and negative classes, reflecting their importance during training.
- Class boundary alignment: Wu and Chang (2003) propose the kernel boundary alignment that adjusts the boundary toward the majority class by modifying the kernel matrix.
- SMOTE algorithm: Akbani et al. (2004) suggest using Chawla et al. (2002) SMOTE algorithm. Basically, the authors, using the SMOTE technique (Synthetic Minority Oversampling Technique), make the distribution of positive/negative observations denser. The SMOTE approach generates new observations between two existing positives/negatives observations which helps in making their distribution more well-defined.
- Granular SVM-repetitive: Tang et al. (2009) suggest a guided repetitive undersampling strategy to *rebalance* the data at hand. The two main characteristics of the method are i) extract informative samples that are essential for classification and ii) elimination of redundant samples.

Although there are several ways to deal with this issue the level of *tolerable* imbalance remains an unresolved problem.

- Violation of the assumption that training and test data are samples from the same population.

The number of observations per variable can affect other classical statistical models but not SVM models because they work with dot products and kernels. In SVM models, the hyperplane can be calculated when the number of variables is much higher than the number of observations.

3.6 Support vector machines extensions

There are extensions, or alternative approaches, to the SVM for binary classification methodology that has been described so far. Briefly, the main SVM extensions, excluding the time-to-event ones that are presented in Section 4.1, are:

Multi-class classification. This extension deals with more than 2 classes. The three main approaches are:

- One versus the rest: in this framework, a set of binary classifiers are trained to separate one class from the rest, and then combine them by doing the multi-class classification according to the maximal output before applying the sign function.

- Pairwise classification: the classifier is trained for each possible pair of classes. The number of trained classifiers is usually larger than the one versus the rest approach, nonetheless, the individual problems are significantly smaller.
- Multi-class objective functions: in this framework the SVM objective function is modified in such a way that it simultaneously allows the computation of a multi-class classifier.

Single-class problem. This is an unsupervised problem, i.e., there are no specific classes, that addresses the problem in which we have a dataset drawn from an underlying probability distribution R, and one want to estimate a *simple* subset S of input space, such that the probability that a test point drawn from R lies outside of S equals some a priori specified value between 0 and 1.

Support vector regression. This is a generalization to the case of regression estimation, i.e., to the estimation of real-valued functions, rather than just two values $\{\pm 1\}$ representing the two classes. The first approach to the regression estimation was described by Cortes and Vapnik (1995), devising the so called ϵ-insensitive loss function (see Appendix A.3.1), and implementing the ϵ-support vector regression (ϵ-SVR). The method seeks to estimate linear functions of the form

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$$

The main goal of the method is to find the function $f(\boldsymbol{x})$ that has at most ϵ deviation from the response quantitative values, $y_i \in \mathbb{R}$, for all the training data and, at the same time, be as flat as possible. The primal objective function of the ϵ-SVR is:

$$\begin{aligned}
\underset{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\xi}^*,b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) & \\
\text{subject to} \quad & ((\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - y_i \leq \epsilon + \xi_i, & i = 1, \dots, n, \\
& y_i - ((\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \leq \epsilon + \xi_i^*, & i = 1, \dots, n, \\
& \xi_i \geq 0, & i = 1, \dots, n, \\
& \xi_i^* \geq 0, & i = 1, \dots, n
\end{aligned} \tag{3.22}$$

where the constant C determines the trade off between the flatness of f and the amount up to which deviations larger than ϵ are tolerated.

Chapter 4

State of the Art on Survival-Support Vector Machines Methods and Variable Relevance

This chapter presents the state of the art for analyzing censored data and for selecting relevant variables into the SVM framework. Section 4.1 presents the work that has been done related to SVM for survival analysis, that basically focus on the treatment of the censoring data, and Section 4.2 presents the state of the art of visualization and variable selection developed in SVM, focusing mainly on non-linear kernels.

4.1 State of the art on support vector machines for survival analysis

The basic SVM classifies subjects into two classes. It does not include any parameter for incomplete information, such as censoring. Two alternative approaches could be used to incorporate censoring into SVM:

- Treat the censoring time as the actual event time. This approach underestimates the actual event time.

- Ignore the censored data and estimate a binary classifier using only observations with all complete data, i.e., observed time-to-event. This approach yields suboptimal models as we ignore the information available in the censored data.

Therefore, to model survival data it is important to incorporate information about follow-up time of subjects with censored event time. Several methods have been recently developed for treating time-to-event data that are described in the subsequent sections.

4.1.1 Kernel Cox regression

The kernel Cox regression is not, strictly speaking, a SVM extension for survival data, but since Cox model is one of the most used classical statistical models it is worth to understand how this extension is addressed. Moreover, kernel Cox regression is one of the methods used as gold standard to compare new survival-SVM approaches. Li and Luan (2003) present a kernelized version of the Cox model. The method is a penalized version of the Cox model, in which a kernel is added to model the hazard as a function of the covariates. For the general Cox model assuming an arbitrary function for the variables, for observation i, at time t, with a vector of covariates $\boldsymbol{x}_i$, the hazard can be expressed as

$$\lambda(t|\boldsymbol{x}_i) = \lambda_0(t)\exp(f(\boldsymbol{x}_i)) \tag{4.1}$$

where $\lambda_0(t)$ is the unspecified baseline hazard function and $f(\boldsymbol{x}_i)$ is an arbitrary function. The authors propose to use the log partial likelihood as a loss function and reformulate the problem as finding the function f in the penalized log-likelihood such that

$$\log(\mathcal{L}(f)) = \sum_{i=1}^{n} \delta_i \left(f(\boldsymbol{x}_i) - \log \sum_{j \in R_i} \exp(f(\boldsymbol{x}_j)) \right) - \xi \|f\|^2_{\mathcal{H}} \tag{4.2}$$

where f is assumed to be from a RKHS, $\mathcal{H}$, defined by a kernel function k. The solution to this problem is given by the representer theorem where the optimal $f(\boldsymbol{x})$ has the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{4.3}$$

The optimal $\boldsymbol{\alpha}$ vector can be found by plugging (4.3) into (4.2) and solving the resulting optimization problem. This method is also proposed and modified by Evers and Messow (2008) who numerically approximate (4.2) with only dependence of a subset of training data, containing only, what they call the *important* cases. These authors fit the model by generalizing the import vector machine (Zhu and Hastie, 2005) to the proportional hazards model. The basic idea of the survival import vector machine is to keep the loss function of the penalized partial log-likelihood, but to set most of α_i values to 0. The observations with non-zero α_i are the called *import vectors.*

Specific remarks

The main issue of kernel Cox regression is that, in contrast to SVM, it does neither perform margin maximization, nor does it yield a sparse solution only depending on a subset of the input data. The latter is an important property as it makes computing future predictions cheaper and helps obtaining good generalization performance when using non-linear kernels. Another issue is that, since is not focused explicitly in the prediction of the classes (alive versus death given an specific period of time) it predicts an score of risk. The interpretation of this score is not as clear as the classical SVM. Therefore, the authors suggest to classify data based on positive and negative scores.

4.1.2 Support vector regression approach

Taking as reference the ϵ-SVR, Shivaswamy et al. (2007) present a method mainly focused on prediction of survival time using an optimization function comparable to the one in equation (3.22) but including a censoring indicator with the form

$$\begin{aligned}
\underset{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\xi}^*,b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\epsilon_i + \epsilon_i^*) & \\
\text{subject to} \quad & (\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq y_i - \epsilon_i, & i = 1, \dots, n, \\
& -\delta_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq -\delta_i y_i - \epsilon_i^*, & i = 1, \dots, n, \\
& \epsilon_i \geq 0, & i = 1, \dots, n, \\
& \epsilon_i^* \geq 0, & i = 1, \dots, n
\end{aligned}$$

where δ_i is the censoring indicator for observation i and y_i is the observed survival time for that same observation. Khan and Zubek (2008), following a similar approach, apply different penalties depending on the predicted-observed survival case.

Several other authors consider methods within the SVM regression framework. Shim and Hwang (2009) propose an approach based on a support vector censored quantile regression approach instead regular regression, in which the parameters of the optimization problem are estimated using iterative re-weighted least squares procedure. To rank new observations with respect to their survival time, Van Belle et al. (2010) use a least squares SVM (Suykens and Vandewalle, 1999) for kernel-based modelling of failure time data accounting for censored data. The goal of the model is to order new data points correctly with respect to their survival time. The proposal of Kim and Jeong (2006) is to use a weighted-least-squares SVM optimization problem in which the weights are calculated using the Kaplan-Meier estimator, resulting in the following expression

$$\begin{aligned}
\underset{\boldsymbol{w},b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}(\zeta_i + e_i^2) & \\
\text{subject to} \quad & y_i - \langle \boldsymbol{w}, \boldsymbol{x}_i\rangle - b = e_i, & i = 1, \dots, n,
\end{aligned}$$

where ζ_i is the vector of weights, calculated using the survival Kaplan-Meier estimator, and being the regression function $f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \langle \boldsymbol{x}, \boldsymbol{x}_i\rangle + b$. Goldberg and Kosorok (2012) propose a similar version, in which they construct a data-dependent version of the loss function using inverse-probability-of-censoring weighting.

Specific remarks

These methods are interesting and maintain the flexibility and most of the SVM properties because they are directly based on SVM theory as opposite to kernel Cox regression. Computations of the rank-based methods are complex and computer intensive because they are based on the comparison of pairwise observations. Van Belle et al. (2011) compared the performance of SVM-based models using a constrained regression and a ranking approach and concluded that the regression approach outperformed the ranking approach. Although they didn't obtain significant differences between SVR-based models and the Cox proportional hazards model.

4.1.3 Support vector ranking-ordinal classification approach

The approached proposed by Van Belle et al. (2007), Van Belle et al. (2008) and Evers and Messow (2008) is based on a ranking (ordinal) SVM maximizing the C-statistic (Harrell et al., 1984) and the goal of the model is to predict a prognostic index. According to Evers and Messow (2008) results this method performs better than kernel Cox model and is also based on the proportional hazards assumption. The primal model is expressed as

$$\begin{aligned} \underset{\boldsymbol{w},\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \sum_{\substack{j:y_i>y_j \\ comp_{ij}=1}} \xi_{ij} \\ \text{subject to} \quad & \langle \boldsymbol{w}, \boldsymbol{x}_i\rangle - \langle \boldsymbol{w}, \boldsymbol{x}_j\rangle \geq 1 - \xi_{ij}, \quad i = 1, \ldots, n; j : y_i > y_j \text{ and } comp_{ij} = 1, \\ & \xi_{ij} \geq 0, \qquad\qquad\qquad\qquad\quad i = 1, \ldots, n; j : y_i > y_j \text{ and } comp_{ij} = 1 \end{aligned} \tag{4.4}$$

where C is a positive real constant and $comp_{ij}$ is the comparability indicator function for a pair of observations $(\boldsymbol{x}_i, y_i, \delta_i)$ and $(\boldsymbol{x}_j, y_j, \delta_j)$, defined as

$$comp_{ij} = \begin{cases} 1 \text{ if } (\delta_i = 1 \text{ and } \delta_j = 1) \text{ or } (\delta_i = 1 \text{ and } \delta_j = 0 \text{ and } y_i \leq y_j) \\ 0 \text{ else} \end{cases}$$

and $\boldsymbol{x}_i, y_i$ and δ_i define the covariates vector, observed time and censoring indicator of any observation i, respectively. There is no need to add a constant term b in the optimization problem (4.4) because only differences in prognostic indices are considered.

Specific remarks

Basically, at every event time t, a hyperplane is constructed separating the individual(s) event at time t from the individuals censored at time t. In contrast to the proportional hazards model, to find the hyperplane, the margin, as in classical SVM is maximized. Therefore, for different event times the hyperplanes are just translated. This is in analogy to the proportional hazards model where the same estimated coefficient is used for all events as well. Although Evers and Messow (2008) report that their results are robust to mild violations of the proportional hazards assumption, they also conclude that the performance is very similar to kernel Cox regression, no obtaining significant differences.

4.1.4 Support vector machines for classification approach

Little work has been done to extend SVM for binary classification to survival data compared to extending SVM regression to analysis of survival data. Shiao and Cherkassky (2013) propose the SVM-learning using privileged information (LUPI) approach developed by Vapnik and Vashist (2009), which uses the censoring information as privileged information (only available for the training data). As the name of the method suggests this approach is based on including additional information into the training process to enrich the learning process. Two different spaces are described, the decision space and the correcting space (the one with the censoring information). The optimization problem to be minimized is

$$\begin{aligned} \underset{\boldsymbol{w},\boldsymbol{w}^*,b,b^*}{\text{minimize}} \quad & \frac{1}{2}(\|\boldsymbol{w}\|^2 + \gamma\|\boldsymbol{w}^*\|^2) + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & \xi_i = (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*), && i = 1,\dots,n, \\ & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*), && i = 1,\dots,n, \\ & (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \geq 0, && i = 1,\dots,n \end{aligned}$$

where $(\boldsymbol{x}, \boldsymbol{y})$ is the usual training data for binary classification. The $\boldsymbol{w}^*, \boldsymbol{x}^*$ and $\boldsymbol{b}^*$ are the equivalent in correcting space to the ones in decision space, i.e., defines the privileged information only present in the training data.

The other approach that the authors propose is the SVM with uncertain classes (Niaf et al., 2011). This method allows to not perfectly definite the belonging class of observations, i.e., it allows some degree of confidence regarding the class. The basic optimization problem is expressed as

$$\begin{aligned} \underset{\boldsymbol{w},b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1, && i = 1,\dots,n, \\ & z_i^- \leq \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \leq z_i^+, && i = n+1,\dots,m \end{aligned}$$

where z_i^- and z_i^+ are boundaries depending on p_i (the probability of belonging to the specific class), $(\boldsymbol{x}, \boldsymbol{y})$ is the training data defining the covariates vector and class, n is the number of definite observations and $(m - n - 1)$ is the number of observations with uncertain classes.

The authors suggest to measure privileged information for LUPI and the uncertainty using the proportion of follow-up time as it is given by

$$\begin{cases} 0, & \text{if observation is an event} \\ \frac{T_i}{\tau}, & \text{if observation } i \text{ is censored} \end{cases}$$

being T_i and τ the observed follow-up time for observation i and the maximum follow-up time established, respectively.

Specific remarks

Depending on the scenario (in terms of proportion of censoring and linearity of the data) to each these two approaches are applied, they outperform the proportional hazards model. The main characteristic of Shiao and Cherkassky (2013) survival-SVM approach, is that is addressed from a SVM for binary classification perspective. The authors remark that using regression setting is intrinsically more difficult than classification and further research needs to be done in this specific framework. The other characteristic is that, in both LUPI and SVM with uncertain classes methods, they construct a *weight* or probability based on the proportional time of follow-up for the censored observations.

4.1.5 Implementation in R

The first implementation of SVM in R was introduced in the e1071 package (Meyer et al., 2015). The implemented functions provide visualization and parameter tuning methods. Package kernlab (Karatzoglou et al., 2004) features a variety of kernel-based methods including a SVM. It aims to provide a flexible and extensible SVM implementation. Package klaR (Weihs et al., 2005) includes an interface to SVMlight, a SVM implementation that additionally offers classification tools such as Regularized Discriminant Analysis. Package svmpath (Hastie, 2016) provides an algorithm that fits the entire path of the SVM solution, i.e., for any value of the cost parameter. Another package with implemented SVM methods is caret (Kuhn, 2016), but all SVM-related methods are wrappers of the kernlab one. All packages mentioned so far are available at the *The Comprehensive R Archive Network* repository (CRAN).

Regarding survival-SVM methods, no packages or specific functions have been found except survpack package (Evers, 2009), which implements kernel Cox regression and survival import vector machines, however this package is not available in CRAN.

4.1.6 Overall remarks and gaps in the literature addressed by this thesis

Overall remarks

Most of the work done in survival-SVM is based on extensions of the SVR approach. These approaches seem to perform as well as the Cox proportional hazards model or the kernel Cox regression and, thus, may not provide any improvements in accuracy of predictions of these other traditional regression methods. Moreover, they may yield biased estimates in the same situations that proportional hazards models do, e.g., when the assumption of proportional hazards is violated.

Using SVM extensions of SVM for binary classification to analyze survival data may result in improved prediction accuracy. However, limited research has been done in this direction. Therefore, there is an opportunity to investigate new approaches based on extensions of SVM for binary classification and compare them to the gold standard Cox proportional hazards and kernel Cox regression models. Having a clear picture of all alternatives to survival-SVM will help to identify the most appropriate model for biomedical problem with alternative data generating mechanisms, for instance with substantial censoring or with non-proportional hazards.

The only two approaches proposed to extend SVM for binary classification to survival data and handle censoring are LUPI and uncertain classes. Shiao and Cherkassky (2013) weight the censored data using the follow-up time. This implies that a censored observation at the beginning of the study will have always the same *weight*, no matter the survival probability at the end of the follow-up period. This seems a strong assumption because: i) it assumes a linear function of the follow-up period, i.e., it is not taking into account the shape of the survival curve and ii) it assumes that the risk of event is the same for any survival probability at the end of the follow-up. Therefore, an alternative to this approach is needed.

Implementation through R-functions and evaluation of additional methods could help choosing the right approach to analyze the data at hand.

Gaps addressed

The present thesis addresses the gaps in the literature by:

- Proposing an alternative approach for *weighting* censored observations when fitting LUPI and SVM with uncertain classes, that is a conditional survival approach as described in expression (2.1) (models are described in Section 6.1 and Section 6.2).
- Proposing a semi-supervised SVM with local invariances approach to deal with censored data (described in Section 6.3).
- Evaluating a weighted SVM for binary classification approach weighting the censored observations (described in Section 6.4).
- Implementing R functions to fit LUPI, uncertain classes SVM, semi-supervised with local invariances SVM and weighted SVM (Appendix F).

4.2 State of the art on variable relevance

Oftentimes investigators are interested in learning about the relative importance of the analyzed variable to predict the response variable. The methodology used to find the relevance of a variable can be classified into three main categories: filter, wrapper and embedded methods. In this section, the three groups of methods for estimate the relative importance of variables and conduct variable selection in SVM context is reviewed.

4.2.1 Filter methods

Those methods asses the relevance of variables by looking only at the intrinsic properties of the data without taking into account any information provided by the classification algorithm. In other words, they perform variable selection before conducting the learning method, including SVM. In most cases a variable relevance score is calculated, and low-scoring variables are removed, since they are supposed to be less important than the ones with highest values on the metric used. Afterwards, the subset variables are presented as input to the classification algorithm.

Chen and Lin (2006) propose some filter methods combined with the SVM. One of them is the F-score, which is also proposed as reference for filter methods by other authors like Maldonado and Weber (2009). Given an observation $\boldsymbol{x}_i$, where $i = 1, \ldots, n$, and the number of positives and negatives instances are n_+ and n_- respectively, the F-score of the pth variable is defined as

$$F(p) = \frac{\left(\overline{\boldsymbol{x}}^{p(+)} - \overline{\boldsymbol{x}}^{p}\right)^2 + \left(\overline{\boldsymbol{x}}^{p(-)} - \overline{\boldsymbol{x}}^{p}\right)^2}{\frac{1}{n_+-1}\sum_{i=1}^{n_+}\left(\boldsymbol{x}_i^{p(+)} - \overline{\boldsymbol{x}}^{p(+)}\right)^2 + \frac{1}{n_--1}\sum_{i=1}^{n_-}\left(\boldsymbol{x}_i^{p(-)} - \overline{\boldsymbol{x}}^{p(-)}\right)^2}$$

where $\overline{\boldsymbol{x}}^p$, $\overline{\boldsymbol{x}}^{p(+)}$, $\overline{\boldsymbol{x}}^{p(-)}$ are the average of the pth variable of the whole, positive, and negative datasets, respectively; $\boldsymbol{x}_i^{p(+)}$ is the pth variable of the ith positive observation, and $\boldsymbol{x}_i^{p(-)}$ is the pth variable of the ith negative observation. The larger the F-score the more likely the variable is more discriminative.

Another proposal from the same authors is to use the F-score and Random Forest. In a first step, the F-score is used for selecting a subset of variables, then the Random Forest is used to obtain a rank of the variables.

Specific remarks

The advantages of filter techniques are that they easily scale to very high-dimensional datasets, they are usually computationally simple and fast, and they are independent of the classification algorithm so they are transversal methods. The main disadvantage of these approaches is that they ignore the interaction with the classifier.

4.2.2 Wrapper methods

The evaluation of a specific subset of variables is obtained by training and testing a specific classification model, rendering this approach tailored to a specific classification algorithm. To search the space of all variable subsets, a search algorithm is then *wrapped* around the classification model. However, as the space of variables subset grows exponentially with the number of variables, heuristic search methods are used to guide the search for an optimal subset.

Guyon et al. (2002) propose one of the most used methods for variable selection into the SVM context. The method is known as *SVM-Recursive Feature Elimination* (SVM-RFE) and the algorithm consists, using the linear kernel, in the steps shown in Algorithm 1.

The final output of this algorithm is a ranked list with the variables ordered according to their relevance. The method allows removing more than one variable at each iteration.

In the same paper the authors propose an approximation for non-linear kernels. The idea is based on measuring the smallest change in the cost function by assuming no change in the value of the estimated $\boldsymbol{\alpha}$ parameters in optimization problem (3.18). Thus, one avoids to retrain a classifier for every candidate variable to be eliminated. The function to be minimized of the soft margin SVM is given by

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{n}\alpha_i \\ \text{subject to} \quad & 0 \le \alpha_i \le C, & i = 1, \ldots, n, \\ & \sum_{i=1}^{n}\alpha_i y_i = 0, & i = 1, \ldots, n \end{aligned}$$

```
Data   : Dataset with $p^*$ variables and binary outcome.
Output: Ranked list of variables according to their relevance.

Find the optimal values for the tuning parameters of the SVM model;
Train the SVM model;
$p \leftarrow p^*$;
while $p \geq 2$ do
    $SVM_p \leftarrow$ SVM with the optimized tuning parameters for the $p$ variables and observations in Data;
    $w_p \leftarrow$ calculate weight vector of the $SVM_p$ $(w_{p1}, \ldots, w_{pp})$;
    $rank.criteria \leftarrow (w_{p1}^2, \ldots, w_{pp}^2)$;
    $min.rank.criteria \leftarrow$ variable with lowest value in $rank.criteria$ vector;
    Remove $min.rank.criteria$ from Data;
    $Rank_p \leftarrow min.rank.criteria$;
    p ← p - 1 ;
end
$Rank_1 \leftarrow$ variable in Data $\notin (Rank_2, \ldots, Rank_{p^*})$;
return $(Rank_1, \ldots, Rank_{p^*})$
```

Algorithm 1: Pseudo-code of the SVM-RFE algorithm (Guyon et al., 2002) using the linear kernel in a SVM model for binary classification.

To compute the change in the cost function caused by removing the input variable p, one leaves the $\boldsymbol{\alpha}$ unchanged and re-calculates the kernel values from the kernel matrix. This corresponds to compute $k(\boldsymbol{x}_i^{-p}, \boldsymbol{x}_j^{-p})$, meaning calculate the kernel values by removing the variable p. The resulting ranking coefficient is

$$\Delta(p) = \left(\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \right) - \left(\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i^{-p}, \boldsymbol{x}_j^{-p}) \right)$$

where $\Delta(p)$ is the difference in cost functions due to the exclusion of variable p. The variable corresponding to the smallest difference $\Delta(p)$ is removed. The process is iterated to carry out the Recursive Feature Elimination (RFE).

Liu et al. (2011) extent the SVM-RFE to the radial basis Gaussian kernel. The method expands the Gaussian kernel into its Maclaurin series and compute the weight vector from the series according to the contribution made to the classification by each variable. Then, using w_i^2 as ranking criterion, the algorithm starts with all the variables, and eliminates one variable with the least squared weight in the weight vector at each step. This process is repeated until all variables are ranked. The main characteristic of this method is that numeric variables must be discretize into $\{0, 1\}$. So, from the decision function (note that the sign function is not included)

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

and expressing the Gaussian kernel as

$$k(\boldsymbol{U}, \boldsymbol{V}) = \exp(-\sigma \langle \phi(\boldsymbol{U}), \tau(\boldsymbol{V}) \rangle)$$

where $\boldsymbol{U} \in \{0,1\}^n$, $\boldsymbol{V} \in \{0,1\}^n$, $\phi(\boldsymbol{U}) = (U_1, \dots, U_n, 1 - U_1, \dots, 1 - U_n)$ and $\tau(\boldsymbol{V}) = (1 - V_1, \dots, 1 - V_n, V_1, \dots, V_n)$. Assuming that

$$\lambda_i = \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) = \alpha_i y_i \exp(-\sigma \langle \phi(\boldsymbol{x}_i), \tau(\boldsymbol{x}) \rangle), \tag{4.5}$$

the Maclaurin series can be computed as

$$\lambda_i = \alpha_i y_i \left(1 + \sum_{t=1}^{\infty} \frac{(-1)^t \sigma^t}{t} \langle \phi(\boldsymbol{x}_i), \tau(\boldsymbol{x}) \rangle^t \right) \tag{4.6}$$

Equation (4.6) is the main expression in which the weight vector is based applying this approach.

Brank et al. (2002) propose a method in which a SVM is trained with a subset of the training data using linear kernel. Then, variables with low weights of the $\boldsymbol{w}$ vector are eliminated, the lower variables can be selected using cross-validation. The last step is, using only variables retained, train the SVM in all data.

Alonso-Atienza et al. (2012) suggest to compare the classification error of the SVM model including all variables to the classification error of the SVM model without the variable p. This procedure is applied for all variables and repeated for a B number of bootstrap samples. In this way the authors calculate what they called *paired confidence intervals*, so that the variables whose interval include the 0 are not relevant and the ones whose interval do not contain 0 are relevant.

Maldonado and Weber (2009) approach is very similar to the SVM-RFE but instead of basing the ranking criteria into the weights, the criteria of removing a variable is based on the classification error of the validation data once the variable p is removed

$$E_{(-p)} = \sum_{i=1}^{m} \left| y_i^v - \operatorname{sign}\left(\sum_{j=1}^{n} \alpha_i y_i k\left(\boldsymbol{x}_j^{-p}, \boldsymbol{x}_i^{v(-p)} \right) + b \right) \right|$$

where y_i^v and $\boldsymbol{x}_i^{v(-p)}$ are the class value and covariates vector of the validation set with m observations removing information of variable p; and y_i and $\boldsymbol{x}_i^{-p}$ are the class value and covariates vector of the training set with n observations removing information of variable p.

Weston et al. (2000) suggest upon finding those variables which minimize bounds on the leave-one-out error via gradient descent. The main idea is that given that the optimal hyperplane is the one with the maximal distance M, in feature space, to the closest image $\phi(\boldsymbol{x}_i)$ from the training data and that the images of all training data are within a sphere of radius R then the bound of the expectation of the error probability has the bound

$$EP_{err} \leq \frac{1}{n} E \left\{ \frac{R^2}{M^2} \right\} = \frac{1}{n} E \left\{ R^2 W^2(\alpha^0) \right\}$$

where $W(\alpha^0)$ is the solution of the optimization problem (3.18). From this main concept, the authors derived the optimization problem which in order to be optimized requires searching all over possible subsets of variables, penalizing large numbers of variables.

Krooshof et al. (2010) approach is slightly different to the others because is mainly focused on the visualization and it can be used with any kernel. The method consists in plotting the score components (pairwise) of a kernel PCA[1] (KPCA), using the tuned values of the kernel parameters found in SVM, and then projecting the scores of the data into them and projecting scores of *pseudo-samples* (an observation with 0 in all variables except in the tested one) with all range of the tested variable. The method doesn't rank variables but allows visualizing the importance of variables in specific KPCA components.

Specific remarks

Advantages of wrapper approaches include the interaction between variable subset search and model selection as well as the ability to take into account variable dependencies. A common drawback of these techniques is that they have a higher risk of overfitting than filter methods and are computationally intensive, especially if building the classifier has a high computational cost (Saeys et al., 2007).

4.2.3 Embedded methods

The search for an optimal subset of variables is built into the classifier construction. Just like wrapper approaches, embedded approaches are, thus, specific to a given learning algorithm. In the SVM framework, all methods found in the literature are limited to linear kernels and most of them are based on a variation or addition of a penalization term, i.e., variables are penalized depending on their values.

Aytug (2015) suggests two versions of the *feature selection problem* formulated previously by Weston et al. (2000); one that explicitly constrains the number of variables, and one that penalizes the number of variables. The author also developed an exact algorithm to solve it in the SVM classification context using the Benders decomposition algorithm (Benders, 1962).

Becker et al. (2009) and Becker et al. (2011) suggest a penalized version of the SVM with different penalization terms[2]

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(\boldsymbol{x}_i)) + pen_\lambda(\boldsymbol{w})$$

where l is the loss term described as a sum of the hinge loss function (Appendix A.3.2) and the penalty term expressed as $pen_\lambda(\boldsymbol{w})$. Depending on the form of the penalty the authors propose several versions of the SVM: Ridge SVM, LASSO SVM, Elastic net SVM, Smoothly Clipped Absolute Deviation (SCAD) SVM and Elastic-SCAD SVM. These methods can only be applied when using linear kernel.

A similar methodology is described in Bradley and Mangasarian (1998) and Bi et al. (2003) focusing on the l_1-norm into the linear kernel and also emphasizing plotting the weight vectors

[1] See Section 9.2.1 for description of the method.

[2] The soft margin SVM optimization problem sometimes is expressed as a loss function (the hinge loss) and a penalization term. See Friedman et al. (2001) for more details.

of the SVM using starplots.

Specific remarks

Embedded methods have the advantage that include the interaction with the classification model and they are far less computationally intensive than wrapper methods (Saeys et al., 2007). The main issue of these methods is that for non-linear kernels it is complicated to penalize the weight vector of the SVM due to the fact that it is in a different space than the original one.

4.2.4 Implementation in R

The package `penalizedSVM` implements the methods described in Becker et al. (2009) and Becker et al. (2011). The RFE algorithm is implemented and available for SVM in the `caret` package (Kuhn, 2016), although the metric for the importance of the variables is not based on the weight vector but on Root Mean Square Error, R^2, Accuracy or Kappa depending on the classification or regression method used.

4.2.5 Overall remarks and gaps in the literature addressed by this thesis

Overall remarks

The main advantage of SVM is the possibility of using it with non-linear kernels. Therefore, a variable selection algorithm should be applicable to this type of kernels. The best approaches to work with non-linear kernels are wrapper methods: i) filter methods are less efficient than wrapper methods and ii) embedded methods are focused on linear kernels. The gold standard of wrapper methods is the RFE algorithm proposed by Guyon et al. (2002). Although wrapper methods outweigh other procedures, there is not approach implemented to visualize and interpret results obtained during RFE iterations. The RFE algorithm for non-linear kernels allows ranking variables but not comparing the performance of all variables in a specific iteration, i.e., interpreting results in terms of: association with the response variable, association with the other variables and magnitude of this association.

Gaps addressed

The present thesis addresses the gaps in the literature by:

- Proposing a RFE-based algorithm that allows visualizing the relevance of the variables by plotting the predictions of the SVM model (described in Section 9.1).

- Proposing two RFE-algorithms based on a representation of variables into KPCA space (described in Section 9.2).

Recursive feature elimination

SVM-RFE, as it has been explained, is the most well known method when ranking variables in SVM. This method is basically a backward elimination procedure. What it is obtained at the end of this algorithm is a variable ranking. However, the variables that are top ranked (eliminated last) are not necessarily the ones that are individually most relevant. Only taken together the variables of a subset are optimal in some sense. So for instance, if we are focusing on a variable that is p ranked we know that in the model with the 1 to p ranked variables, p is the variable least relevant.

We have used this approach as basis for our proposed methods. Although automated stepwise selection methods (backward, forward and backward-forward) are not optimal, they have some advantages, specially backward compared to forward. A backward selection approach is generally preferred if automatic stepwise selection is attempted (Steyerberg, 2008):

- The modeller is forced to consider the full model with a backward approach, and can judge the effects of all candidate predictors simultaneously (Harrell, 2015).
- Correlated variables may remain in the model, while none of them might enter the model with a forward approach (Derksen and Keselman, 1992).

Guyon et al. (2002) points out the distinction between the RFE algorithm and the naive ranking[3]. The naive ranking and RFE are qualitatively different, the former orders variables according to their individual relevance whereas the latter is a variable subset ranking. The nested variable subset contain complementary variables not necessarily individually most relevant. This is related to the relevance vs. usefulness distinction by Kohavi and John (1997).

[3] The ranking performed based on the first iteration of the RFE-agorithm, i.e., one SVM model with all variables.

Chapter 5

Objectives of the Thesis

The context of this thesis is focused on the SVM methods for survival analysis from a binary classification perspective and variable selection, with special emphasis on the application on real data.

5.1 Main objectives

This thesis has three main objectives:

1. To develop extensions of the SVM binary data to predict time-to-event response variables.

2. To develop methods for visualization and ranking relevance of variables into the survival-SVM framework.

3. To identify the relevant cytokines and chemokines that correlate with protection induced by the RTS,S vaccine.

5.2 Specific objectives associated with the SVM for survival analysis

The specific objectives associated with the proposed survival-SVM methods are:

1. To evaluate the performance of the conditional survival approach when compared to the proportional time approach.

2. To evaluate the overall performance of the proposed survival-SVM alternatives compared to the Cox proportional hazards model and kernel Cox regression in realistic scenarios about the data generating mechanism that are a function of: proportion and distribution of censoring, sample size, and violation of the proportional hazards assumption.

3. To implement code in R software for the compared survival-SVM methods.

5.3 Specific objectives for the visualization and relevance of variables

The specific objectives associated with the visualization and ranking of variables are:

1. To evaluate the performance of the proposed methods in terms of ranking relevance compared to the RFE approach for non-linear kernels, under the same data generating mechanism scenarios used to compare survival-SVM methods.

2. To interpret and visualize figures obtained for the proposed methods.

5.4 Specific objectives for the analysis of Mal067 data

The specific objectives associated with the analysis of correlates of protection are:

1. To identify the optimal method to analyze RTS,S and comparator cohorts, including: the proposed survival-SVM methods, Cox proportional hazards and kernel Cox regression.

2. To visualize and rank the relevance of cytokines, chemokines, and growth factors for the best model found in each vaccine cohort, and for the best RFE-algorithm found in the simulation study.

Part II

Support Vector Machines for Survival Analysis

Chapter 6

Material and Methods

In the following sections the proposed methods for dealing with censored observations into the SVM for binary classification framework are presented. In Section 6.1 the SVM extension using what is called privileged information is presented. The SVM extension that deals with uncertain classes is presented in Section 6.2. In Section 6.3, a semi-supervised version of the classical SVM, based on the hinge-loss function and local invariances is proposed as a potential method to be used with censored data. Finally, in Section 6.4, a weighted version of the classical SVM weighting for the observation class is described.

6.1 Support vector machines learning using privileged information

The SVM learning using privileged information approach, known as *SVM+* and *LUPI*[1], was described by Vapnik (Vapnik and Vashist, 2009). As the name of the method suggests it is based on including additional information into the training process to enrich the learning process.

The LUPI approach is based in a triplet $(\boldsymbol{x}_i, \boldsymbol{x}_i^*, y_i)$, for $i = 1, \ldots, n$ observations, where $\boldsymbol{x}_i \in \mathbb{R}^d$, $\boldsymbol{x}_i^* \in \mathbb{R}^k$ and $y_i \in \{\pm 1\}$. The $(\boldsymbol{x}, \boldsymbol{y})$ is the usual training data and $\boldsymbol{x}_i^*$ defines the privileged information only present in the training data, i.e., the information (variables) only present when modelling the data. This information is not available when predicting the class of a new observation. Therefore, it can be seen that the privileged information is in a different feature space than the classical SVM. In the LUPI approach two different spaces are described:

- The space related to $\boldsymbol{x}$, known as *decision* space, which is the same feature space used in standard SVM.
- The space related to $\boldsymbol{x}^*$, known as *correcting* space, which reflects the privileged information about the training data and not available for predictions of future observations.

The main objective of the LUPI is to estimate the usual decision function by correcting it using the correcting function via privileged information.

[1] Hereinafter will be referred as LUPI.

6.1.1 Optimization problem

The primal optimization problem to be minimized in the LUPI paradigm is

$$\begin{aligned}
\underset{\boldsymbol{w},\boldsymbol{w}^*,b,b^*}{\text{minimize}} \quad & \frac{1}{2}(\|\boldsymbol{w}\|^2 + \gamma\|\boldsymbol{w}^*\|^2) + C\sum_{i=1}^{n}\xi_i \\
\text{subject to} \quad & \xi_i = (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*), && i = 1, \ldots, n \\
& y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*), && i = 1, \ldots, n \\
& (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \geq 0, && i = 1, \ldots, n
\end{aligned} \tag{6.1}$$

To solve this problem the Lagrangian is constructed

$$\begin{aligned}
L(\boldsymbol{w}, b, \boldsymbol{w}^*, b^*, \boldsymbol{\alpha}, \boldsymbol{\beta}) = & \frac{1}{2}(\|\boldsymbol{w}\|^2 + \gamma\|\boldsymbol{w}^*\|^2) \\
& + C\sum_{i=1}^{n}(\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) - \sum_{i=1}^{n}\alpha_i(y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \\
& - 1 + (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*)) - \sum_{i=1}^{n}\beta_i(\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*)
\end{aligned}$$

The Lagrange function must be minimized with respect $\boldsymbol{w}, b, \boldsymbol{w^*}, b^*$ and maximized it with respect to Lagrange multipliers $\boldsymbol{\alpha} \geq 0$ and $\boldsymbol{\beta} \geq 0$. The dual space solutions are defined by the decision function in which the kernel trick can be applied obtaining

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{i=1}^{n} y_i\alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b\right) \tag{6.2}$$

The corresponding correcting function is given by

$$f^*(\boldsymbol{x^*}) = (\langle \boldsymbol{w}^*, \boldsymbol{x}^* \rangle + b^*) = \frac{1}{\gamma}\sum_{i=1}^{n}(\alpha_i + \beta_i - C)k^*(\boldsymbol{x}_i^*, \boldsymbol{x}^*) + b^*$$

where $k(\boldsymbol{x}_i, \boldsymbol{x})$ and $k^*(\boldsymbol{x}_i^*, \boldsymbol{x}^*)$ are kernels that define the inner products in the decision and correcting spaces respectively, and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the solution of the optimization problem to be maximized (Appendix B.1). Therefore, the dual problem to be maximized is expresses as

$$\begin{aligned}
\underset{\boldsymbol{\alpha},\boldsymbol{\beta}}{\text{maximize}} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
& - \frac{1}{2\gamma}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C)k^*(\boldsymbol{x}_i^*, \boldsymbol{x}_j^*) \\
\text{subject to} \quad & \sum_{i=1}^{n}(\alpha_i + \beta_i - C) = 0, && i = 1, \ldots, n \\
& \sum_{i=1}^{n} y_i\alpha_i = 0, && i = 1, \ldots, n \\
& \alpha_i \geq 0, && i = 1, \ldots, n \\
& \beta_i \geq 0, && i = 1, \ldots, n
\end{aligned} \tag{6.3}$$

The intuitive idea behind the LUPI is very similar to the classical SVM extrapolating the idea to two spaces instead of one. The decision function and the correcting functions depend on the decision and correcting space respectively. Although, decision function has the same expression as usual SVM, the LUPI decision function coefficients depend on kernels in both spaces. The SVM solution and the LUPI can be exactly the same when the correcting space is not appropriate and the privileged information is rejected (when γ tends to 0 in expression [6.1]).

From a mathematical point of view the LUPI algorithm, that takes into account both privileged and unprivileged information, is very similar to SVM algorithms for finding solutions in the classical binary classification framework. It requires solving a quadratic optimization problem under constraints that are similar to constraints in the classical SVM. However, the LUPI algorithm is computationally costlier than classical SVM because it requires optimizing two parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and tuning the cost parameter C.

6.1.2 Application to survival data

In the training data the time to follow-up and time-to-event is known but it is unknown in the test setting, so the censoring information at the training setting can be used as privileged information. According to Shiao and Cherkassky (2013) the privileged information can be formulated as

$$\boldsymbol{x}_i^* = (T_i, p_i)$$

where T_i is the observed time (event or censoring) and p_i is a certainty measure of the non-event, proportional to the observed time. Therefore, in the events is 0 and in censored observations is calculated as T_i/τ, being τ the maximum follow-up time established.

Our proposal is to define the privileged information based on the conditional probability of having the event using the Kaplan-Meier estimator defined in (2.1). In this way we take into account the *real* probability of having an event and the characteristics of the survival curve.

6.1.3 Parameters and kernels

Shiao and Cherkassky (2013) compared the Gaussian kernel and linear kernel for the correcting space and obtained similar results. Based on this we have used linear kernel for correcting space. For the decision space we have applied the Gaussian kernel. Regarding the censoring approach, both proportional time to follow-up and conditional survival approach are compared.

6.2 Support vector machines with uncertain classes

The SVM with uncertain classes was developed by Niaf et al. (2011). This method allows to not perfectly definite some observations and give them an uncertainty in their class. For these uncertainties a confidence level or probability regarding the class is provided. The method will be referred from now on to *pSVM* (*probabilistic* SVM).

6.2.1 Optimization problem

The pSVM takes into account belonging a class through a hinge loss as well as probability estimates using the ϵ-insensitive cost function (Appendix A.3). Given an observation i we define the pair $(\boldsymbol{x}_i, l_i)$ as the learning dataset of input vectors $\boldsymbol{x} \in \mathcal{X}$ along with their corresponding classes group. The classes can be defined as

$$\begin{aligned} l_i &= y_i \in \{\pm 1\} \quad \text{for } i = 1, \ldots, n \\ l_i &= p_i \in [0,1] \quad \text{for } i = n+1, \ldots, m \end{aligned}$$

where n is the number of observations with known classes (perfectly definite), $(m - n - 1)$ the number of observations with uncertain classes and p_i is the associated uncertainty about $\boldsymbol{x}_i$ in a regression setting. The posterior probability for class 1 is given by

$$p_i = \text{Prob}(Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i)$$

The associated optimization problem is

$$\begin{aligned} \underset{\boldsymbol{w},b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1, \qquad i = 1, \ldots, n, \\ & z_i^- \leq \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \leq z_i^+, \quad i = n+1, \ldots, m \end{aligned} \tag{6.4}$$

where z_i^- and z_i^+ are boundaries depending on p_i. If $n = m$ the problem is exactly the same to hard margin SVM. Allowing to misclassification in classes, slack variables ξ_i are introduced. Then the expression (6.4) can be rewritten as

$$\begin{aligned} \underset{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\xi}^-,\boldsymbol{\xi}^+,b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \xi_i + \widetilde{C} \sum_{i=n+1}^{m} (\xi_i^- + \xi_i^+) \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i, \qquad i = 1, \ldots, n, \\ & z_i^- - \xi_i^- \leq \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \leq z_i^+ + \xi_i^+, \qquad i = n+1, \ldots, m, \\ & \xi_i \geq 0, \qquad i = 1, \ldots, n, \\ & \xi_i^- \geq 0, \qquad i = n+1, \ldots, m, \\ & \xi_i^+ \geq 0, \qquad i = n+1, \ldots, m \end{aligned} \tag{6.5}$$

Parameters C and $\widetilde{C}$ are controlling the weighting of the certain classes and uncertain classes, respectively.

To define the probability boundaries, let ϵ be the class precision and $\triangle$ the confidence we have in that class. Defining $\eta = \epsilon + \triangle$. Then the regression problem consists in finding the optimal parameters $\boldsymbol{w}$ and b such that

$$\left| \frac{1}{1 + \exp(-a(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b))} - p_i \right| < \eta$$

Thus constraining the probability prediction for point $\boldsymbol{x}_i$ to remain around

$$\frac{1}{1 + \exp(-a(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b))}$$

within distance η. The boundaries where $(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)$ is $+1$ or -1 define parameter a as

$$a = \log\left(\frac{1}{\eta} - 1\right)$$

Finally,

$$\begin{aligned} \max(0, p_i - \eta) \leq & \quad \frac{1}{1 + \exp(-a(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b))} < \min(p_i + \eta, 1) \\ z_i^- \leq & \quad \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \quad < z_i^+ \end{aligned}$$

where

$$\begin{aligned} z_i^- &= -\frac{1}{a}\log\left(\frac{1}{p_i - \eta} - 1\right) \\ z_i^+ &= -\frac{1}{a}\log\left(\frac{1}{p_i + \eta} - 1\right) \end{aligned}$$

The optimization problem (6.5) can be rewritten into the Lagrangian by introducing the Lagrange multipliers:

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^-, \boldsymbol{\xi}^+, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}^+, \boldsymbol{\mu}^-, \boldsymbol{\gamma}^+, \boldsymbol{\gamma}^-) = & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \widetilde{C}\sum_{i=n+1}^{m}(\xi_i^- + \xi_i^+) \\ & - \sum_{i=1}^{n}\alpha_i(y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - (1 - \xi_i)) - \sum_{i=1}^{n}\beta_i\xi_i \\ & - \sum_{i=n+1}^{m}\mu_i^-((\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - (z_i^- - \xi_i^-)) - \sum_{i=n+1}^{m}\gamma_i^-\xi_i^- \\ & - \sum_{i=n+1}^{m}\mu_i^+((z_i^+ + \xi_i^+) - (\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)) - \sum_{i=n+1}^{m}\gamma_i^+\xi_i^+ \end{aligned}$$

where $\boldsymbol{\alpha} \geq 0$, $\boldsymbol{\beta} \geq 0$, $\boldsymbol{\mu}^+ \geq 0$, $\boldsymbol{\mu}^- \geq 0$, $\boldsymbol{\gamma}^+ \geq 0$ and $\boldsymbol{\gamma}^- \geq 0$.

After computing the derivatives with respect to $\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^-$ and $\boldsymbol{\xi}^+$, and after applying simplifications (see Appendix [B.2]) and the kernel trick, the dual formulation of (6.5) to be maximized is given by

$$\begin{aligned} \underset{\boldsymbol{\alpha}, \boldsymbol{\mu}^+, \boldsymbol{\mu}^-}{\text{maximize}} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=n+1}^{m}\sum_{j=1}^{n}k(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_j y_j(\mu_i^+ - \mu_i^-) \\ & -\frac{1}{2}\sum_{i=n+1}^{m}\sum_{j=n+1}^{m}(\mu_i^+ - \mu_i^-)(\mu_j^+ - \mu_j^-)k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{n}\alpha_i - \sum_{i=n+1}^{m}\mu_i^+ z_i^+ + \sum_{i=n+1}^{m}\mu_i^- z_i^- \\ \text{subject to} \quad & \sum_{i=n}^{n}\alpha_i y_i - \sum_{i=n+1}^{m}\mu_i^+ + \sum_{i=n+1}^{m}\mu_i^- = 0 \\ & 0 \leq \alpha_i \leq C, \qquad i = 1, \dots, n \\ & 0 \leq \mu_i^+ \leq \widetilde{C}, \qquad i = n+1, \dots, m \\ & 0 \leq \mu_i^- \leq \widetilde{C}, \qquad i = n+1, \dots, m \end{aligned} \tag{6.6}$$

Being the decision function

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{n}\alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) - \sum_{i=n+1}^{m}(\mu_i^+ - \mu_i^-)k(\boldsymbol{x}_i, \boldsymbol{x}) + b\right) \tag{6.7}$$

The pSVM method allows including both qualitative and quantitative information into the class of each observation combining the SVM for binary classification approach and a regression loss. The uncertainty of a non exact observation has to be specified into the $(0, 1)$ interval, which can be difficult depending on the data characteristics.

The decision function is similar to the one in the classical SVM, except that in pSVM the parameters associated to the non-exact observations are included.

6.2.2 Application to survival data

Shiao and Cherkassky (2013), following the same idea that in the LUPI approach, suggest to use as a probability for censored observations the proportional time of follow-up, as described in Section (6.1.2), i.e.,

$$\begin{cases} 0, & \text{if observation is an event} \\ \frac{T_i}{\tau}, & \text{if observation } i \text{ is censored} \end{cases} \tag{6.8}$$

being T and τ the observed follow-up time and the maximum follow-up time established, respectively.

Our proposal, following the same idea, is to use the conditional survival Kaplan-Meier probability for the censored observations.

6.2.3 Parameters and kernels

The pSVM model has two tuning parameters C and $\widetilde{C}$, that control the relative weighting of classification and regression performances, respectively. Besides that, there are parameters associated with the kernel. Our proposal, to test the performance, is to use the linear and Gaussian kernel. Regarding the treatment of the censored data both proportional time to follow-up and conditional survival approach are tested.

6.3 Semi-supervised support vector machines using local invariances

A different perspective to treat the censoring data and not considered in the SVM literature is as a semi-supervised problem. In the semi-supervised setting there are known classes and unknown classes and the main goal is to learn from both types of training data to find the decision surface that separates both classes from known data. Some recent approaches have been done in a non-SVM context like in Liang et al. (2016), in which the authors proposed the $\mathrm{L}_{1/2}$ regularization approach in the semi-supervised learning method in a cancer survival analysis context.

The first approach of Semi-Supervised SVM (S^3VM) was implemented by Joachims (1999). After that other approaches have been introduced, most of them are based on solving the standard SVM while treating the unknown classes as additional optimization variables. By maximizing the margin in the presence of unknown data, one learns a decision boundary that traverses thought low data-density regions while respecting classes in the input space. This approach assumes that points in a data cluster have similar classes. The main issue with S^{3}VM approach is that is related to a non-convex optimization problem. Most of the different approaches try to solve this issue (Chapelle et al., 2008).

In a non-SVM specific context, Lee et al. (2006) propose as framework of semi-supervised learning in RKHS using local invariances[2] that explicitly characterize the behaviour of the target function around both known and unknown data. The authors propose three types of invariances:

1. Invariance to small changes to the observations: restricting the gradient of the function to be small at the observed data.

2. Invariance to averaging across small neighbourhood around observations: restricting the function value at each observation to be similar to the average value across a small neighbourhood of that observation.

3. Invariances to local transformation, like rotational and translational invariance: this invariance is specially focused in problems such handwritten digit recognition and vision problems.

In this thesis, first and second invariances are described and compared. The third invariance is not considered due to it's focused in a different field than the survival analysis. These three local invariances can be summarized by two local invariances assumptions:

- The target function does not change much in the neighbourhood of each observed data; this is the case when observations from the same class are clustered together and away from observations from the other class. Therefore, the decision function is encouraged to fall in regions of low data density.

- Invariance to certain local transformations.

Before going to the optimization problem itself (Section 6.3.2) defined by Lee et al. (2006), some preliminary concepts need to be addressed.

6.3.1 Preliminaries

Definition 6 (Linear Operator and Functional)**.** *A linear operator T is a mapping from a vector space X to a vector space Y, such for all $x, y \in X$ and scalar α,*

$$T(x+y) = Tx + Ty$$
$$T(\alpha x) = \alpha T x$$

If the range $Y \subseteq \mathbb{R}$, the operator is called a functional.

[2] Prior knowledge in machine learning, and specifically in SVM for classification is known as *class-invariance*. A very common type of prior knowledge in binary classification is the invariance of the class to a transformation of the input variables. This type of knowledge is referred as *transformation-invariance*, but there are others type of invariances related to the prior knowledge.

Definition 7 (Bounded Linear Operator)**.** *Let $T : X \to Y$ be a linear operator on a normed spaces X and Y. The operator T is said to be bounded if there exists some $c > 0$ such that for all $x \in X$*

$$\|Tx\| \leq c\|x\|$$

The smallest value of c such that the inequality holds for all nonzero $x \in X$ is called the norm of the operator and denoted $\|T\|$.

Theorem 3 (Riesz)**.** *Every bounded linear functional* L *on a Hilbert space H can be represented in terms of an inner product*

$$\mathrm{L}(x) = \langle x, z \rangle, \qquad \forall x \in H$$

where the representer of the functional, z, has norm

$$\|z\| = \|\mathrm{L}\|$$

and is uniquely determined by L*.*

When H is a RKHS the representer of the functional has the form

$$z(x) = \langle z, k(x, \cdot) \rangle = \mathrm{L}(k(x, \cdot))$$

Riesz's theorem allows to represent functionals related to local invariances as elements of the RKHS.

6.3.2 Optimization problem

According to Lee et al. (2006), let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_l, y_l)$ be the usual training data, and $l_2(y, g(\boldsymbol{x}))$ be the loss function on g when training input $\boldsymbol{x}$ belong to class y. The method measure deviations from local invariances around each known or unknown input observation and express these as bounded linear functional $\mathrm{L}_{l+1}(g), \ldots, \mathrm{L}_n(g)$ on the RKHS H. The linear functionals are associated with another loss function $l_1(\mathrm{L}_i(g))$ penalizing violations of the local invariances. After including a squared loss function as a regularization term and putting these loss functions together the function to be minimized is

$$\rho_1\|g\|^2 + \rho_2 \sum_{i=l+1}^{n} l_1(\mathrm{L}_i(g)) + \sum_{i=1}^{l} l_2(y_i, g(\boldsymbol{x}_i)) \tag{6.9}$$

where $\rho_1 > 0$ and $\rho_2 > 0$ are the relative strengths of the loss functions, l is the number of known classes and $(n - l - 1)$ is the number of unknown classes. As stated by the authors reasonable examples of l_2 include logistic loss, hinge loss and squared loss. Examples of l_1 include the squared loss, absolute loss and ϵ-insensitive loss. The main reason for these loss functions is that all of them are convex and result in convex optimization problems finding the optimal g[3].

The loss function that we propose is the hinge loss for the l_2: the classical SVM formulation described in (3.15) is equivalent to an optimization problem using the hinge loss (Friedman et al.,

[3] The convexity property ensure that any local minimum must be a global minimum.

2001), so the formulation is equivalent to SVM. With regards the l_1 function, we propose the ϵ-insensitive loss function. It is important to show that a representer theorem can be derived showing that the solution of the optimization problem lies in the span of a finite number of functions associated with the known classes and the functionals.

Theorem 4. *Let* $\mathrm{L}_i, i = l+1, \ldots, n$, *be bounded linear functionals in the reproducing kernel Hilbert space* H *defined by the kernel* k*. The solution of the optimization problem*

$$g^* = \arg\min_{g \in H} \rho_1 \|g\|^2 + \rho_2 \sum_{i=l+1}^{n} l_1(\mathrm{L}_i(g)) + \sum_{i=1}^{l} l_2(y_i, g(\boldsymbol{x}_i))$$

for ρ_1 *and* $\rho_2, > 0$ *can be expressed as*

$$g^*(\cdot) = \sum_{i=1}^{l} \alpha_i k(\boldsymbol{x}_i, \cdot) + \sum_{i=l+1}^{n} \alpha_i z_i(\cdot)$$

where z_i *is the representer of* L_i*.*

After introducing the hinge loss and ϵ-insensitive loss into expression (6.9), we obtain the following optimization problem to minimize

$$\begin{aligned}
\underset{g,b}{\text{minimize}} \quad & \rho_1 \|g\|^2 + \rho_2 \sum_{i=l+1}^{n} (\xi_i + \xi_i^*) + \sum_{i=1}^{l} \gamma_i \\
\text{subject to} \quad & -\langle g, z_i \rangle - b \leq \epsilon + \xi_i, && i = l+1, \ldots, n, \\
& \langle g, z_i \rangle + b \leq \epsilon + \xi_i^*, && i = l+1, \ldots, n, \\
& \xi_i \geq 0, && i = l+1, \ldots, n, \\
& \xi_i^* \geq 0, && i = l+1, \ldots, n, \\
& y_i(\langle g, \phi(\boldsymbol{x}_i) \rangle + b) \geq 1 - \gamma_i, && i = 1, \ldots, l, \\
& \gamma_i \geq 0, && i = 1, \ldots, l
\end{aligned} \tag{6.10}$$

Adding the Lagrange multipliers we construct the Lagrangian

$$\begin{aligned}
L(g, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\eta}, \boldsymbol{\eta}^*) = {} & \rho_1 \|g\|^2 + \rho_2 \sum_{i=l+1}^{n} (\xi_i + \xi_i^*) + \sum_{i=1}^{l} \gamma_i - \sum_{i=l+1}^{n} (\eta_i \xi_i + \eta_i^* \xi_i^*) \\
& - \sum_{i=l+1}^{n} \alpha_i (\epsilon + \xi_i + \langle g, z_i \rangle + b) \\
& - \sum_{i=l+1}^{n} \alpha_i^* (\epsilon + \xi_i^* - \langle g, z_i \rangle - b) \\
& + \sum_{i=1}^{l} \beta_i^* \left[1 - y_i(\langle g, \phi(\boldsymbol{x}_i) \rangle + b) - \gamma_i \right] - \sum_{i=1}^{l} \beta_i \gamma_i
\end{aligned} \tag{6.11}$$

After computing the corresponding derivatives with respect the primal variables and after applying simplifications (see Appendix [B.3]). The dual formulation to be maximized becomes:

$$\begin{aligned}
\underset{\boldsymbol{\alpha},\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*}{\text{maximize}} \quad & -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\beta_i^*\beta_j^* y_i y_j k(\boldsymbol{x}_i,\boldsymbol{x}_j) - \frac{1}{2}\sum_{i=l+1}^{n}\sum_{j=l+1}^{n}\langle z_i, z_j\rangle(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*) \\
& -\sum_{i=1}^{l}\sum_{j=l+1}^{n}\beta_i^* y_i z_j(\boldsymbol{x}_i)(\alpha_j-\alpha_j^*) - \sum_{i=l+1}^{n}\epsilon(\alpha_i+\alpha_i^*) + \sum_{i=1}^{l}\beta_i^* \qquad (6.12)\\
\text{subject to} \quad & 0 \le \alpha_i \le \rho_2, \qquad i = l+1,\dots,n \\
& 0 \le \alpha_i^* \le \rho_2, \qquad i = l+1,\dots,n \\
& 0 \le \beta_i^* \le 1, \qquad i = 1,\dots,l
\end{aligned}$$

and the decision function is given by

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{i=1}^{l}\beta_i^* y_i k(\boldsymbol{x},\boldsymbol{x}_i) + \sum_{i=l+1}^{n} z_i(\boldsymbol{x})(\alpha_i-\alpha_i^*)\right) \tag{6.13}$$

The term z_i in equation (6.13) allows applying the desired invariance, i.e., the representer of the specific functional. The decision function has two parts, one for the known classes similar to the classical SVM, and the other related to the unknown classes.

This SVM approach will be referred from now on as *inSVM* (*invariances* SVM).

6.3.3 Derivative invariance

Given the Gaussian kernel expressed as $\exp(-\frac{1}{2\sigma^2}\|\boldsymbol{x}-\boldsymbol{x}'\|^2)$, Lee et al. (2006) show that the evaluation of the representer of the derivative functional for the Gaussian kernel is

$$z_{\boldsymbol{x}_{i,j}}(\boldsymbol{x}) = \frac{1}{\sigma^2}(\boldsymbol{x}^j - \boldsymbol{x}_i^j)\exp(-\frac{1}{2\sigma^2}\|\boldsymbol{x}-\boldsymbol{x}_i\|^2) \tag{6.14}$$

and the dot product is expressed as

$$\langle z_{\boldsymbol{x}_{i,j}}, z_{\boldsymbol{x}_{p,q}}\rangle = \begin{cases} -\frac{1}{\sigma^4}(\boldsymbol{x}_i^j - \boldsymbol{x}_p^j)(\boldsymbol{x}_i^q - \boldsymbol{x}_p^q)\exp(-\frac{1}{2\sigma^2}\|\boldsymbol{x}_i-\boldsymbol{x}_p\|^2) & \text{if } j \ne q \\ \frac{1}{\sigma^4}(\sigma^2 - (\boldsymbol{x}_i^j - \boldsymbol{x}_p^j)^2)\exp(-\frac{1}{2\sigma^2}\|\boldsymbol{x}_i-\boldsymbol{x}_p\|^2) & \text{if } j = q \end{cases} \tag{6.15}$$

where σ is the parameter of the Gaussian kernel, i and p represents the observation indices and j and q are the indices of the specific variable of the specific $\boldsymbol{x}$ vector. Taking into account this, in (6.12) we have to calculate the corresponding $z_{\boldsymbol{x}_{i,j}}(\boldsymbol{x})$ and $\langle z_{\boldsymbol{x}_{i,j}}, z_{\boldsymbol{x}_{p,q}}\rangle$ for all covariates in every sum they appear.

6.3.4 Local averaging invariance

Another type of local invariance is the local averaging one. Lee et al. (2006) show that the Gaussian kernel together with the Gaussian density function satisfies that $\mathrm{L}_{\boldsymbol{x}_i}(g)$ is a bounded linear functional and allows to be implemented efficiently. So, being the Gaussian kernel

$$k(\boldsymbol{x}_1,\boldsymbol{x}_2) = \exp(-\frac{1}{2\sigma_k^2}\|\boldsymbol{x}_1-\boldsymbol{x}_2\|^2)$$

and the Gaussian density

$$p(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma_p^d} \exp\left(-\frac{1}{2\sigma_p^2}\|\boldsymbol{x}\|^2\right)$$

and given that the convolution of a Gaussian density with another Gaussian density is a Gaussian density, the representer of the local average functional is expressed as

$$z_{\boldsymbol{x}_i}(\boldsymbol{x}) = \frac{\sigma_k^d}{(\sigma_k + \sigma_p)^d} \exp\left(-\frac{1}{2(\sigma_k + \sigma_p)^2}\|\boldsymbol{x}_i - \boldsymbol{x}\|^2\right) - \exp\left(-\frac{1}{2\sigma_k^2}\|\boldsymbol{x}_i - \boldsymbol{x}\|^2\right) \tag{6.16}$$

and the dot product

$$\begin{aligned} \langle z_{\boldsymbol{x}_i}, z_{\boldsymbol{x}_j} \rangle = \mathrm{L}_{\boldsymbol{x}_i}(z_{\boldsymbol{x}_j}) &= \frac{\sigma_k^d}{(\sigma_k + 2\sigma_p)^2} \exp\left(-\frac{1}{2(\sigma_k + 2\sigma_p)^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right) \\ &- \frac{\sigma_k^d}{(\sigma_k + \sigma_p)^d} \exp\left(-\frac{1}{2(\sigma_k + \sigma_p)^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right) - z_{\boldsymbol{x}_j}(\boldsymbol{x}_i) \end{aligned} \tag{6.17}$$

where σ_k and σ_p are the sigma values specified for the Gaussian kernel and Gaussian density respectively, d the number of covariates or directions and $z_{\boldsymbol{x}_j}(\boldsymbol{x}_i)$ is defined as (6.16).

6.3.5 Application to survival data

Our proposal is to treat censored observations, during the follow-up period, as unknown classes and events and non-events at the end of follow-up as known classes.

6.3.6 Parameters and kernels

The number of parameters and kernels to be tuned depends on the local invariances approach. For the derivative invariance, that is based on the Gaussian kernel, this one is used. Similarly, for the averaging invariance the Gaussian kernel is used as well as the Gaussian function that defines the width of averaging.

6.4 Weighted support vector machines

Another approach that has not been tested in the literature is to address the survival-SVM as a weighted SVM problem. The basic idea of weighted support vector machines (wSVM) is to assign to each observation a different weight according to its relative importance in the class such that different data points contributed differently to the learning of the decision surface (Yang et al., 2007). This methodology is focused mainly on the treatment of outliers, since if there is a way to detect outliers we can diminish their effect in the estimation of the separating hyperplane. Given the known weights $\boldsymbol{W}$, the training data set becomes

$$(\boldsymbol{x}_i, y_i, W_i), \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{\pm 1\}, W_i \in [0, 1]$$

for all $i = 1, \ldots, n$.

6.4.1 Optimization problem

The optimization problem is nearly identical to the classical SVM expression, except by the weight vector $\boldsymbol{W}$ that multiplies the slack variable $\boldsymbol{\xi}$,

$$\begin{aligned} \underset{\boldsymbol{w},\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} W_i\xi_i & \\ \text{subject to} \quad & \xi_i \geq 0, & i = 1,\dots,n \\ & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i, & i = 1,\dots,n \end{aligned} \tag{6.18}$$

The dual formulation becomes

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{n}\alpha_i & \\ \text{subject to} \quad & 0 \leq \alpha_i \leq CW_i, & i = 1,\dots,n \\ & \sum_{i=1}^{n}\alpha_i y_i = 0, & i = 1,\dots,n \end{aligned} \tag{6.19}$$

being the decision function the same as in usual SVM

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{n}\alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) y_i + b\right) \tag{6.20}$$

The only difference between SVM and wSVM is the upper bound of the Lagrange multiplier $\boldsymbol{\alpha}$ in the dual problem, that instead of being C is CW_i (equation [6.19]). This implies that, depending on the weight assigned to an observation that becomes a support vector, the effect of this support vector to the training procedure is reduced compared to the standard SVM training. This does not apply when $C \to \infty$ and $C \to 0$, then the performance of wSVM and classical SVM are identical.

6.4.2 Application to survival data

The censored observation can be seen as a partial or weighted observation, since for example an observation censored just at the beginning of the study is adding no information to the data, i.e., the weight of this observation should be close to 0. A censored observation just before the end of the follow-up period should be treated *almost* as complete observation (a weight close to 1). Applying this approach the censored observation can be treated as a weighting problem. The issue is which weight use. Our proposal, following the same idea than the previous methods is to compare the performance of a proportional time to follow-up approach as suggested by Shiao and Cherkassky (2013) and our proposal using a conditional survival approach, based on the Kaplan-Meier estimator as defined in (2.1).

6.4.3 Parameters and kernels

Our proposal is to evaluate the performance of this method using only the Gaussian kernel.

Chapter 7

Simulation Study

In this chapter we evaluate the performance of the proposed methodology described in Chapter 6 by means of a simulations study. The main model we have taken as reference is the proportional hazards model

$$\lambda(t|\boldsymbol{x}_i) = \lambda_0(t)\exp(\langle \boldsymbol{x}_i, \boldsymbol{\beta}\rangle)$$

The simulated scenarios take as reference the characteristics of the malaria parasite (specifically the *Plasmodium Falciparum* parasite) and an usual cytokine and chemokine panel trying to reproduce, specifically, Mal067 study and malaria-cytokine based projects. A total of 24 scenarios are considered and compared.

Within this chapter, we first present in Section 7.1 the scenarios that are compared and the overall simulation approach. The tuning parameters and testing models scheme are presented in Section 7.2. In Section 7.3 the metrics used to test the performance of the methods by simulated scenario are described. Finally, the simulations results are summarized in Section 7.4.

7.1 Simulation of scenarios and data generation

7.1.1 Generation of covariates

The covariates are created following a similar pattern to the Mal067 cytokines and chemokines data. In each simulated dataset 30 variables are generated following a multivariate normal distribution:

$$\boldsymbol{x} \sim N_{30}(\mu, \Sigma) \tag{7.1}$$

Being the mean of each variable a realization of a Uniform distribution $U(0.03, 0.06)$ and the covariance matrix Σ:

$$\Sigma = \begin{cases} 0.7, & \text{if } i = j. \\ \operatorname{Cov}(\boldsymbol{x}_i,\, \boldsymbol{x}_j), & \text{otherwise.} \end{cases} \tag{7.2}$$

for $i = 1, \ldots, 30$ and $j = 1, \ldots, 30$. The Σ matrix for all $i \neq j$ is calculated so that there are 4 blocks of correlated variables with the following Pearson's correlation:

- No correlation (around 0).

- Low correlation (around 0.2).
- Medium correlation (around 0.5).
- High correlation (around 0.8).

An example of the distribution of a randomly selected dataset and the correlation matrix is shown in appendix Figure C.1 and appendix Figure C.2, respectively.

7.1.2 Generation of time-to-malaria

Two time-to-malaria scenarios, within the context of Cox proportional hazards model, are compared. According to Sama et al. (2006) and Bejon et al. (2013) the parametric distribution that fits better time-to-malaria for the *P.falciparum* parasite is a Gompertz distribution. Following Bender et al. (2005), in the proportional hazards framework the time-to-event variable can be generated, based on the Gompertz distribution as

$$T = \frac{1}{\alpha} \log \left(1 - \frac{\alpha \log(U)}{\gamma \exp(\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle)} \right) \tag{7.3}$$

being the hazard function expressed as

$$\lambda(t|\boldsymbol{x}_i) = \gamma \exp(\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle) \exp(\alpha t) \tag{7.4}$$

where U is a variable following a Uniform (0,1) distribution, $\boldsymbol{\beta}$ is the coefficients variable vector, $\alpha \in (-\infty, \infty)$ and $\gamma > 0$ are the scale and shape parameters of the Gompertz distribution[1] (Figure 7.1).

Modifying the method proposed in Bender et al. (2005) to violate the proportional hazards assumption, a noise has been added into the $\exp(\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle)$ term in (7.3) and (7.4), forcing the hazard to be a shared frailty model (Duchateau and Janssen, 2007). Therefore, a common variability, called frailty, following a Gaussian distribution has been added:

$$\varpi_j \sim N(0, 5) \tag{7.5}$$

so that the proportionality of hazards only holds conditional to the frailty ϖ_j

$$\lambda(t|\boldsymbol{x}_{ij}) = \varpi_j \gamma \exp(\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle) \exp(\alpha t) \tag{7.6}$$

The frailty has been simulated so that there are 5 groups of observations with same size that share a common frailty $\varpi_j, j = 1, \ldots, 5$.

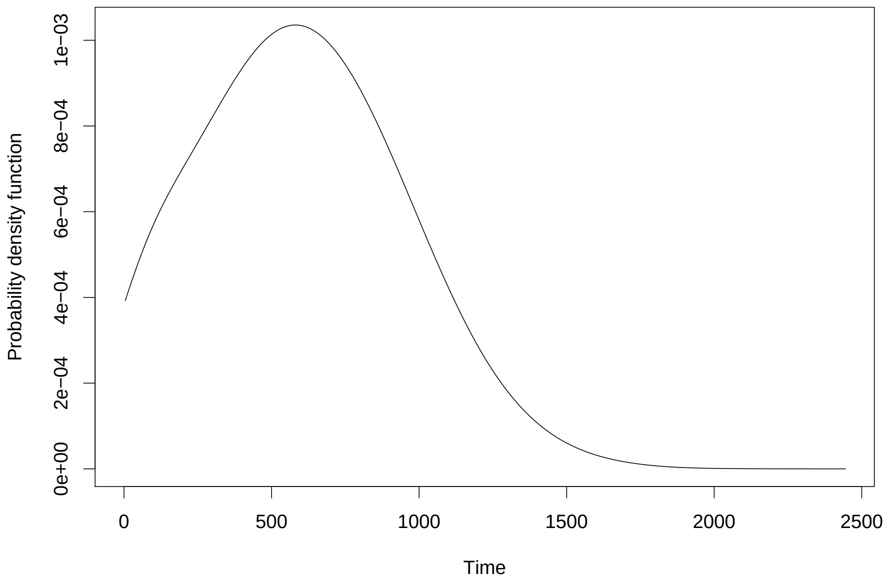
For both scenarios, proportional and non-proportional hazards, the α and γ parameters are selected so that the malaria at 18 months is around 0.6 (Agnandji et al., 2011).

7.1.3 Generation of censoring

Two types of censoring scenarios are compared, one scenario related to the proportion of censoring and the other based on the distribution along follow-up time.

[1] The parametrization used in Bender et al. (2005) is slightly different than the classical one showed in Section 2.3.3.

Figure 7.1: Distribution of time-to-malaria following the Cox-Gompertz distribution described in equation (7.3). In this example parameters selected are $\alpha = 0.013$ and $\gamma = 3.3 \times 10^{-6}$.



Proportion of censoring

Based on the proportion of censored data during the follow-up two opposites scenarios are chosen:

- 10% of censoring, since this is an scenario with a small amount of censored observations should not affect estimations.
- 30% of censoring, since this is an scenario with a big amount of censoring and methods should be affected.

Distribution of censoring

Based on the distribution along time 3 different distributions have been simulated:

- Uniform distribution (zero skew): the censoring is distributed along the time uniformly. To guarantee a fixed percentage c of censoring we have to use a $U(0, C_{\max})$ where the $C_{\max}$ term is calculated as follows

$$C_{\max} = P(C < T) = \int_0^\infty dy \int_0^y \frac{1}{C_{\max}} f(y) du = \frac{1}{C_{\max}} \int_0^\infty y f(y) dy = \frac{1}{C_{\max}} E(T)$$

  Being T the simulated time-to-malaria variable. So the Uniform distribution will follow $C \sim U\left(0, \frac{E(T)}{C_{\max}}\right)$

- Positively skewed distribution: the censoring is based on a exponential distribution whose probability density function, as shown in (2.3.1), is

$$f(x) = \lambda \exp(-\lambda x)$$

  where $\lambda > 0$ is the rate parameter. For each scenario the rate parameter was found so that the percentage of censoring achieves the desired censoring.

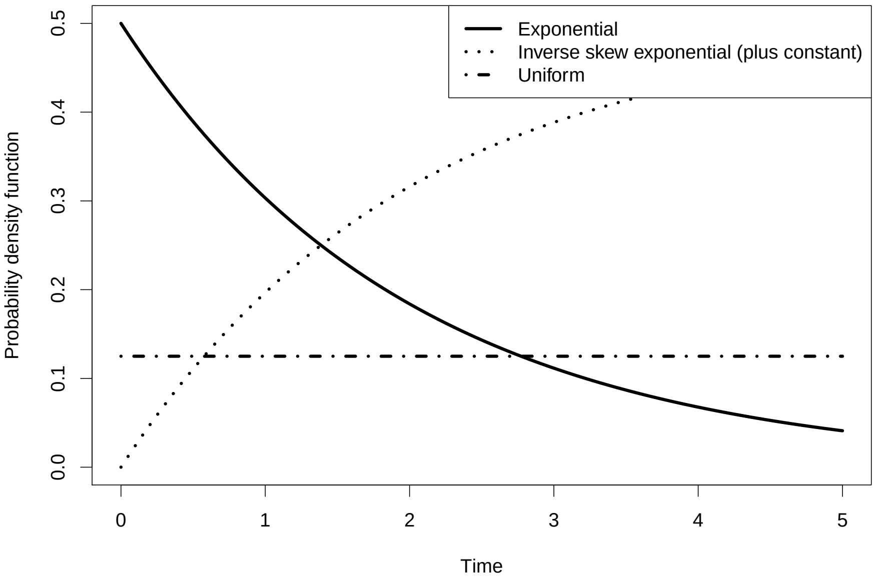
- Negatively skewed distribution: in order to have an exponential distribution but negatively skewed we can invert the skewness distribution by replacing the cumulative distribution F by its complement $F' = 1 - F$, obtaining the complementary distribution function that gives a mirror image. We can add a constant c so that we obtain non-negative values, then probability distribution is given by

$$f(x) = -\lambda \exp(-\lambda x) + c$$

  where $\lambda > 0$ is the rate parameter. For each scenario the rate parameter have been found so that the percentage of censoring achieves the desired censoring. The constant c have been fixed to be the minimum observed value so that all censoring times are non-negative.

An example, of the three censoring distributions used in the simulated scenarios is shown in Figure 7.2.

Figure 7.2: The three distributions selected for simulating the censoring patterns. In this representation $\lambda = 0.5$ is used, for both exponential distributions and Uniform(0,5) is shown.



7.1.4 Number of observations

Two scenarios are compared related to the number of observations, one with 300 observations and another one with 50.

7.1.5 Comparison of methods

The methods to be compared are the ones described in Chapter 6. Specifically, the methods, parameters and values evaluated are:

- Cox proportional hazards model.
- Kernel Cox regression: the values used for tuning the Gaussian kernel are 0.25, 0.5, 1, 2 and 4.
- Weighted support vector machines: two approaches are used; one defining the weights of the censored observations as proportional to the follow-up time, and another one using conditional survival. The values used for tuning the parameter C are 0.1, 1, 10 and 100, and for the kernel 0.25, 0.5, 1, 2 and 4.
- Support vector machines with uncertain classes: two kernels have been tested (linear and Gaussian kernels). The probabilities of the censored observations have been estimated using the proportional and the conditional survival methods. The grid values search for the Gaussian kernel are 0.25, 0.5, 1, 2 and 4. The C and $\widetilde{C}$ tested values are 0.1, 1, 10 and 100. The η value has been fixed to 0.0001.
- Support vector machines learning using privileged information: Gaussian kernel and linear kernel are used for the decision and correcting space respectively. For the latter two approaches based on proportionality of follow-up time and conditional survival have been compared. For the Gaussian kernel the values used in the grid search are 0.25, 0.5, 1, 2 and 4. The cost values of the misclassification parameter C are 0.1, 1, 10 and 100. The weight used for the correcting space, γ, has been tested to 0.1, 1, 10 and 100.
- Semi-supervised support vector machines using invariances: both derivative and gradient invariances have been implemented. The parameters tested have been 0.25, 0.5, 1, 2 and 4 for both Gaussian kernel and Gaussian density (only for averaging invariances). The values considered for ρ_2 are 0.1, 1, 10 and 100 (for both gradient and averaging approaches).

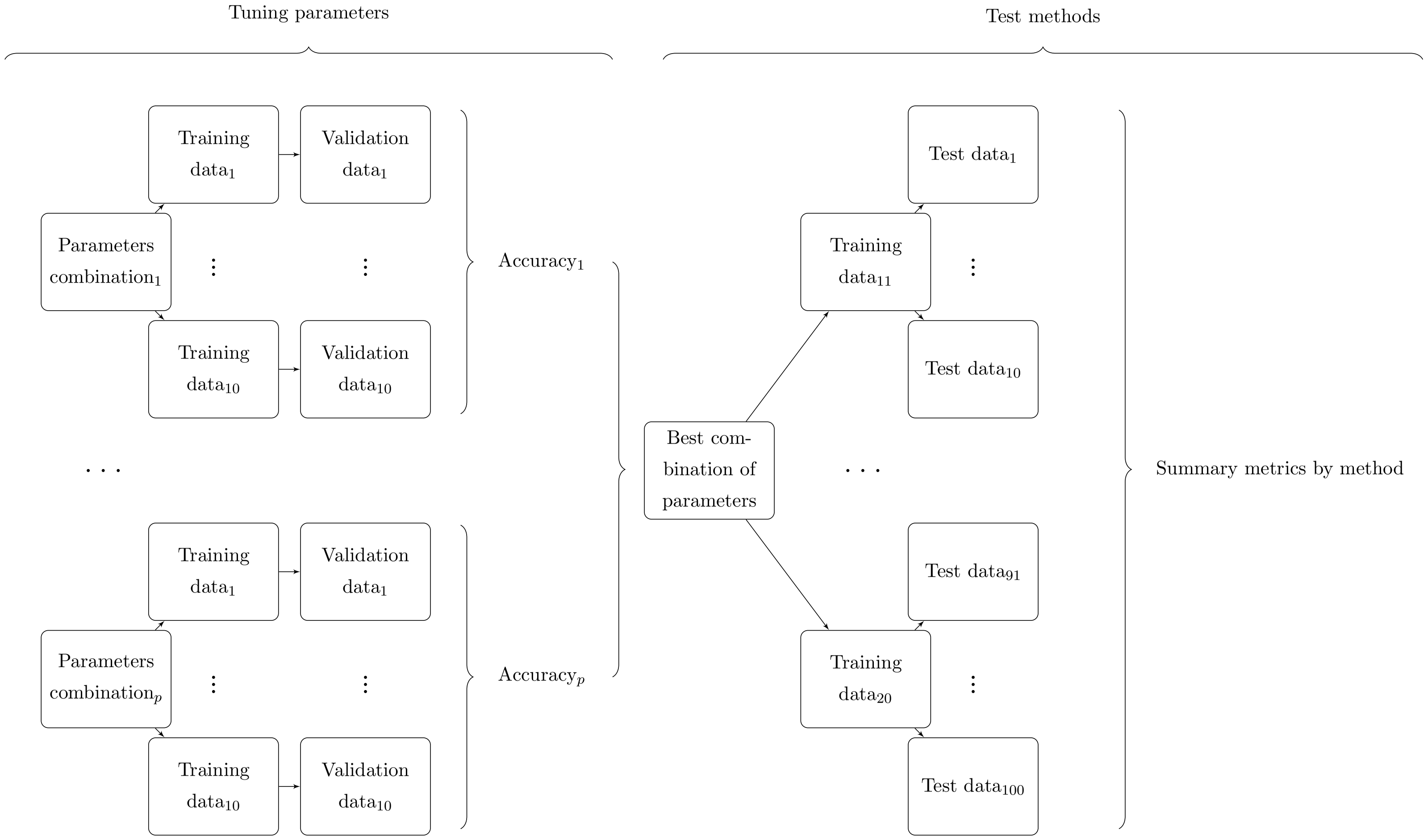
7.2 Evaluation criteria and tuning parameters

To tune parameters and evaluate the performance of each method and scenario a training-validation-test approach has been applied. Figure 7.3 illustrates the steps involved in the simulation approach by each one of the scenarios and methods. The simulation study is based on two steps:

1. Tuning parameters: for each combination of parameters 10 training datasets are fitted and validated using 10 different validation datasets. The accuracy, for each parameter combination, is measured 10 times and averaged. The parameters combination with highest accuracy is the one selected and used to test the models.

2. Test methods: 10 training datasets, different from all datasets used in the tuning parameters step, are simulated and fitted with the *best combination* found in tuning parameters step. For each one, 10 testing datasets are used to assess prediction and four metrics have been measured in order to evaluate the performance. Therefore, 100 datasets, have been tested. The mean and the standard deviation of the metrics is used as a summary performance of each method.

Figure 7.3: Tuning parameters and testing models scheme.



7.3 Metrics to evaluate model performance

As SVM is a binary classification method, most of the metrics used are based in a 2 × 2 contingency table as shown in Table 7.1. In the process of tuning parameters we have used the training datasets and the validation dataset. Although, we know exactly who have an event during the follow-up period, in real data, due to the censoring, this information is unknown. For this reason, to tune the parameters the accuracy is used only for events and non-events at the end of follow-up, i.e., censoring during follow-up time is not used. For testing the methods, all available information is used. In the Cox model there are no tuning parameters, thus the model has not been trained with the training data and has been tested directly. The metrics used are: accuracy, Matthews correlation coefficient, normalized mutual information and area under the ROC curve.

Table 7.1: Predicted malaria and true malaria contingency table for all combinations. The notation inside the table stands for: tp: true positives; fn: false negatives; fp: false positives; and tn: true negatives.

		Predicted malaria	
		+	-
True malaria	+	tp	fn
	-	fp	tn

Accuracy

The accuracy measures the proportion of correct classified observations over the total predicted observations.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \tag{7.7}$$

Matthews correlation coefficient

This correlation coefficient is expressed as

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fn)(tp + fp)(tn + fp)(tn + fn)}} \tag{7.8}$$

The MCC coefficients lies between -1 and $+1$. A value of -1 indicates total disagreement between predicted and true malaria, a value of 0 completely random predictions and $+1$ total agreement between prediction and reality.

Normalized mutual information

This approach measures a normalized version of the mutual information metric which represents the reduction in uncertainty of one variable when the other is observed. The expression is as follows:

$$NMI = \frac{MI}{H} \tag{7.9}$$

where MI is the mutual information and H is the maximum value that can be achieved by the MI. Being,

$$MI = -E\left(\frac{tp}{n}, \frac{tn}{n}, \frac{fp}{n}, \frac{fn}{n}\right) - \frac{tp}{n}\log\left(\frac{tp+fp}{n}\frac{tp+fn}{n}\right) - \frac{fn}{n}\log\left(\frac{tp+fn}{n}\frac{tn+fn}{n}\right) - \frac{fp}{n}\log\left(\frac{tp+fp}{n}\frac{tn+fp}{n}\right) - \frac{tn}{n}\log\left(\frac{tn+fn}{n}\frac{tn+fp}{n}\right) \tag{7.10}$$

where n is the total number of observations and

$$E\left(\frac{tp}{n}, \frac{tn}{n}, \frac{fp}{n}, \frac{fn}{n}\right) = -\frac{tp}{n}\log\left(\frac{tp}{n}\right) - \frac{tn}{n}\log\left(\frac{tn}{n}\right) - \frac{fp}{n}\log\left(\frac{fp}{n}\right) - \frac{fn}{n}\log\left(\frac{fn}{n}\right) \tag{7.11}$$

is the usual entropy. The normalized version is achieved by the boundary H expressed as

$$H = -\frac{tp+fn}{n}\log\left(\frac{tp+fn}{n}\right) - \frac{tn+fp}{n}\log\left(\frac{tn+fp}{n}\right) \tag{7.12}$$

As it has been explained the NMI is a normalized version of the mutual information, this means that when the NMI is 0 the prediction is totally random and when is 1 the prediction is perfect.

Area under the ROC curve

The area under the ROC curve, sometimes referred also as C-statistic (Harrell et al., 1984) or AUC, is a way of summarizing the discrimination ability of a model. The AUC is the probability in a random pair of observations, one with malaria and one with no malaria, the malaria observation has a higher probability than the other. The AUC thus gives the probability that the model correctly ranks such pairs of observations. An AUC of 0.5 is randomly discrimination and 1 perfect discrimination of malaria/non-malaria observations. For the Cox proportional hazards model the linear predictor obtained from the trained model and the true status of the test data have been used. For the SVM models the decision value has been used together with the true status of the test data.

For all models the described metrics are summarized using the mean and the standard deviation.

7.4 Simulation results

In this section main results are described by scenario and method. Differences in performance by proportionality of hazards, proportion of censoring, distribution of censoring and sample size are addressed from Section 7.4.2 to 7.4.5. Herein, for the purpose of illustration, the uniformly skewed censoring distribution results are described, all other tables summarizing the results are presented in Appendix C.2. Simulated datasets summary results are shown in appendix, from Table C.1 to Table C.4 and from Figure C.3 to Figure C.6.

Table 7.2: Proportional hazards, zero skew, 10% and 30% censoring and 300 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	Accuracy	Matthews	NMI	AUC	Accuracy	Matthews	NMI	AUC
Cox model	0.89 (0.02)	0.78 (0.03)	0.50 (0.05)	0.96 (0.01)	0.89 (0.02)	0.79 (0.04)	0.51 (0.06)	0.96 (0.01)
Kernel Cox	0.81 (0.02)	0.62 (0.05)	0.30 (0.05)	0.88 (0.02)	0.80 (0.02)	0.59 (0.04)	0.26 (0.05)	0.86 (0.02)
wSVM-KM	0.75 (0.03)	0.50 (0.06)	0.19 (0.05)	0.87 (0.02)	0.68 (0.02)	0.39 (0.05)	0.11 (0.03)	0.86 (0.03)
wSVM-Prop	0.75 (0.03)	0.50 (0.06)	0.18 (0.05)	0.87 (0.02)	0.68 (0.02)	0.38 (0.05)	0.11 (0.03)	0.85 (0.02)
pSVM-linear-KM	0.88 (0.02)	0.73 (0.04)	0.46 (0.05)	0.95 (0.01)	0.88 (0.02)	0.72 (0.05)	0.43 (0.07)	0.95 (0.02)
pSVM-linear-prop	0.87 (0.02)	0.73 (0.04)	0.45 (0.05)	0.95 (0.01)	0.86 (0.02)	0.72 (0.05)	0.42 (0.07)	0.94 (0.02)
pSVM-radial-KM	0.79 (0.02)	0.57 (0.05)	0.25 (0.05)	0.88 (0.02)	0.79 (0.02)	0.58 (0.04)	0.27 (0.04)	0.86 (0.02)
pSVM-radial-prop	0.77 (0.02)	0.57 (0.05)	0.24 (0.05)	0.88 (0.02)	0.77 (0.02)	0.58 (0.04)	0.27 (0.04)	0.86 (0.02)
LUPI-linear-KM	0.78 (0.03)	0.56 (0.05)	0.28 (0.05)	0.84 (0.03)	0.77 (0.02)	0.55 (0.05)	0.27 (0.06)	0.84 (0.03)
LUPI-linear-prop	0.77 (0.03)	0.55 (0.05)	0.28 (0.05)	0.84 (0.03)	0.77 (0.02)	0.55 (0.05)	0.27 (0.06)	0.84 (0.03)
inSVM-gradient	0.84 (0.02)	0.68 (0.05)	0.37 (0.06)	0.92 (0.02)	0.80 (0.02)	0.60 (0.05)	0.28 (0.05)	0.89 (0.02)
inSVM-averaging	0.83 (0.02)	0.66 (0.05)	0.35 (0.06)	0.92 (0.02)	0.83 (0.02)	0.66 (0.05)	0.35 (0.06)	0.92 (0.02)

Table 7.3: Non-proportional hazards, zero skew, 10% and 30% censoring and 300 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	Accuracy	Matthews	NMI	AUC	Accuracy	Matthews	NMI	AUC
Cox model	0.71 (0.02)	0.39 (0.05)	0.10 (0.03)	0.77 (0.03)	0.70 (0.03)	0.39 (0.06)	0.10 (0.04)	0.77 (0.03)
Kernel Cox	0.67 (0.02)	0.33 (0.05)	0.10 (0.04)	0.71 (0.03)	0.67 (0.03)	0.32 (0.06)	0.08 (0.04)	0.70 (0.03)
wSVM-KM	0.65 (0.02)	0.24 (0.05)	0.01 (0.02)	0.71 (0.03)	0.61 (0.02)	0.16 (0.06)	0.01 (0.02)	0.71 (0.03)
wSVM-Prop	0.64 (0.02)	0.24 (0.05)	0.01 (0.02)	0.70 (0.03)	0.61 (0.02)	0.17 (0.07)	0.01 (0.02)	0.70 (0.03)
pSVM-linear-KM	0.72 (0.03)	0.39 (0.05)	0.13 (0.04)	0.77 (0.03)	0.69 (0.03)	0.37 (0.05)	0.13 (0.03)	0.75 (0.03)
pSVM-linear-prop	0.70 (0.03)	0.38 (0.05)	0.13 (0.03)	0.76 (0.03)	0.69 (0.03)	0.37 (0.05)	0.13 (0.03)	0.75 (0.03)
pSVM-radial-KM	0.66 (0.02)	0.28 (0.05)	0.03 (0.03)	0.70 (0.03)	0.66 (0.03)	0.31 (0.07)	0.10 (0.04)	0.70 (0.03)
pSVM-radial-prop	0.66 (0.02)	0.28 (0.05)	0.03 (0.03)	0.70 (0.03)	0.66 (0.03)	0.31 (0.07)	0.08 (0.05)	0.70 (0.03)
LUPI-linear-KM	0.65 (0.02)	0.27 (0.05)	0.03 (0.03)	0.70 (0.03)	0.65 (0.03)	0.31 (0.05)	0.13 (0.05)	0.70 (0.03)
LUPI-linear-prop	0.65 (0.02)	0.27 (0.05)	0.03 (0.03)	0.70 (0.03)	0.65 (0.03)	0.31 (0.05)	0.13 (0.05)	0.70 (0.03)
inSVM-gradient	0.70 (0.02)	0.38 (0.05)	0.11 (0.03)	0.76 (0.02)	0.67 (0.03)	0.33 (0.06)	0.11 (0.03)	0.72 (0.03)
inSVM-averaging	0.70 (0.02)	0.38 (0.05)	0.11 (0.03)	0.76 (0.02)	0.69 (0.03)	0.37 (0.05)	0.13 (0.03)	0.76 (0.03)

7.4.1 General results

For all scenarios, within the 300 observations and proportional hazard scenario, the Cox model and pSVM (linear kernel) perform similar to inSVM (gradient and averaging) with metrics closest to the two best methods described. Specifically, the accuracy is around 0.89 for the Cox model, 0.87 for the linear pSVM and 0.84 for inSVM (Table 7.2). For the AUC metric the Cox model perform 0.96, pSVM 0.95 and inSVM 0.92. The distribution and proportion of censoring do not affect results in general, although from the three methods described inSVM-gradient seems most affected by the proportion of censoring.

Considering the 50 observations scenario and proportionality of hazards, results are the opposite, being the best performance the 10% censoring scenario for pSVM, inSVM and kernel Cox regression with an accuracy around 0.75. The worst model in this scenario is Cox model, wSVM and pSVM-radial with an accuracy of 0.67, 0.62 and 0.65, respectively. Increasing the proportion of censoring to 30% does not affect the overall performance of the SVM methods, with the exception of wSVM. The Cox model is clearly affected.

The non-proportionality scenarios metrics are lower than the proportional ones. The largest difference between proportionality compared to non-proportionality is in the 300 observations

Table 7.4: Proportional hazards, zero skew, 10% and 30% censoring and 50 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.67 (0.11)	0.42 (0.19)	0.23 (0.11)	0.66 (0.14)	0.54 (0.11)	0.34 (0.18)	0.11 (0.11)	0.56 (0.1)
Kernel Cox	0.74 (0.07)	0.47 (0.13)	0.20 (0.12)	0.78 (0.07)	0.72 (0.08)	0.46 (0.15)	0.17 (0.12)	0.77 (0.09)
wSVM-KM	0.62 (0.04)	0.21 (0.16)	0.05 (0.10)	0.77 (0.08)	0.54 (0.05)	0.16 (0.10)	0.01 (0.02)	0.76 (0.09)
wSVM-Prop	0.61 (0.04)	0.21 (0.16)	0.05 (0.10)	0.77 (0.08)	0.53 (0.05)	0.15 (0.11)	0.01 (0.02)	0.75 (0.08)
pSVM-linear-KM	0.77 (0.09)	0.54 (0.17)	0.26 (0.14)	0.86 (0.08)	0.75 (0.07)	0.50 (0.15)	0.22 (0.12)	0.83 (0.07)
pSVM-linear-prop	0.75 (0.07)	0.50 (0.14)	0.25 (0.12)	0.84 (0.07)	0.75 (0.07)	0.49 (0.15)	0.21 (0.13)	0.83 (0.07)
pSVM-radial-KM	0.65 (0.05)	0.26 (0.16)	0.07 (0.09)	0.77 (0.07)	0.66 (0.07)	0.33 (0.16)	0.36 (0.27)	0.77 (0.08)
pSVM-radial-prop	0.64 (0.05)	0.23 (0.17)	0.06 (0.10)	0.77 (0.07)	0.64 (0.07)	0.31 (0.16)	0.29 (0.34)	0.77 (0.08)
LUPI-linear-KM	0.70 (0.08)	0.42 (0.13)	0.26 (0.15)	0.76 (0.08)	0.71 (0.08)	0.40 (0.16)	0.18 (0.12)	0.74 (0.09)
LUPI-linear-prop	0.70 (0.08)	0.42 (0.13)	0.26 (0.15)	0.76 (0.08)	0.70 (0.08)	0.40 (0.16)	0.18 (0.12)	0.74 (0.09)
inSVM-gradient	0.76 (0.07)	0.52 (0.15)	0.23 (0.13)	0.84 (0.07)	0.74 (0.07)	0.47 (0.14)	0.20 (0.11)	0.82 (0.07)
inSVM-averaging	0.77 (0.07)	0.52 (0.15)	0.24 (0.13)	0.85 (0.07)	0.75 (0.06)	0.49 (0.13)	0.21 (0.11)	0.83 (0.07)

Table 7.5: Non-Proportional hazards, zero skew, 10% and 30% censoring and 50 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.59 (0.05)	0.14 (0.15)	0.04 (0.07)	0.55 (0.07)	0.58 (0.06)	0.11 (0.19)	0.07 (0.10)	0.53 (0.07)
Kernel Cox	0.61 (0.08)	0.22 (0.15)	0.15 (0.17)	0.64 (0.07)	0.63 (0.07)	0.24 (0.15)	0.07 (0.09)	0.64 (0.08)
wSVM-KM	0.62 (0.04)	0.08 (0.14)	0.02 (0.03)	0.64 (0.07)	0.59 (0.03)	0.05 (0.15)	0.02 (0.01)	0.64 (0.08)
wSVM-Prop	0.60 (0.04)	0.09 (0.14)	0.02 (0.03)	0.64 (0.07)	0.59 (0.03)	0.05 (0.15)	0.02 (0.01)	0.64 (0.08)
pSVM-linear-KM	0.63 (0.07)	0.23 (0.14)	0.08 (0.08)	0.66 (0.09)	0.61 (0.08)	0.22 (0.17)	0.11 (0.09)	0.66 (0.09)
pSVM-linear-prop	0.61 (0.07)	0.21 (0.14)	0.07 (0.07)	0.65 (0.09)	0.59 (0.09)	0.17 (0.18)	0.11 (0.09)	0.63 (0.09)
pSVM-radial-KM	0.63 (0.04)	0.14 (0.14)	0.04 (0.09)	0.63 (0.08)	0.61 (0.06)	0.17 (0.16)	0.10 (0.21)	0.63 (0.09)
pSVM-radial-prop	0.61 (0.04)	0.14 (0.14)	0.10 (0.21)	0.63 (0.08)	0.56 (0.07)	0.12 (0.14)	0.31 (0.36)	0.63 (0.09)
LUPI-linear-KM	0.62 (0.07)	0.22 (0.15)	0.07 (0.08)	0.63 (0.08)	0.62 (0.07)	0.18 (0.16)	0.03 (0.07)	0.63 (0.09)
LUPI-linear-prop	0.61 (0.07)	0.20 (0.15)	0.07 (0.08)	0.63 (0.08)	0.62 (0.07)	0.18 (0.16)	0.03 (0.07)	0.63 (0.09)
inSVM-gradient	0.66 (0.07)	0.27 (0.14)	0.06 (0.07)	0.67 (0.08)	0.64 (0.07)	0.27 (0.15)	0.10 (0.10)	0.69 (0.09)
inSVM-averaging	0.66 (0.07)	0.28 (0.14)	0.07 (0.07)	0.67 (0.09)	0.65 (0.07)	0.28 (0.16)	0.11 (0.11)	0.68 (0.09)

scenario (Table 7.3) compared to the 50 observations one (Table 7.5).

7.4.2 Proportionality of data

The proportionality of data influences the performance of all methods tested. In the 300 observations scenarios the decrease in accuracy of the pSVM (linear kernel) and the Cox model is similar, around 0.2 units, in Matthews correlation around 0.4 and around 0.2 in the AUC (Table 7.2 and 7.3). For the NMI the decrease in Cox model is around 0.4 and in pSVM around 0.3. The third method with more observed differences in the performance is inSVM and kernel Cox regression with a decrease of around 0.14 units in accuracy, both of them. The decrease in the performance of the other methods (wSVM, pSVM-radial and LUPI) due to the proportionality of hazards is similar: around 0.11, 0.22, 0.13 and 0.16 in accuracy, Matthews correlation, NMI and AUC, respectively.

The performance of the conditional survival methods compared to the proportional ones is similar within each one of the two proportionality of hazards scenarios, not showing a clear difference between both approaches.

In the 50 observations scenarios the major differences are observed in SVM methods, specially in the 10% censoring scenario, and kernel Cox regression. For the inSVM model the decrease due to the proportionality of hazards in accuracy and AUC is 0.10 units and 0.08, respectively. Must be noted that the observed values in the 50 observations scenario is lower than the 300, a 0.10 units difference is more relevant than in the 300 observations scenario.

In the scenario with lower number of observations, the performance of the conditional survival methods is less influenced by the proportionality of hazards.

7.4.3 Proportion of censoring

In the 300 observations scenario for proportionality of hazards (Table 7.2) the two methods most affected by modifying the proportion of censored data are the inSVM-gradient approach and the wSVM. The differences in the inSVM are around 0.03, 0.07, 0.07 and 0.04 for accuracy, Matthews correlation, NMI and AUC, respectively. For the wSVM method the differences are larger, being around 0.07 in accuracy and around 0.02 units in AUC, taking into account that the global performance is worst in wSVM. In the non-proportionality of hazards scenarios differences between proportion of censoring scenarios are similar.

In the 50 observations scenario the major differences are observed in Cox model, whose accuracy goes from around 0.67 to approximately 0.54, and the AUC from 0.66 to 0.56 in the proportionality of hazards scenario (Table 7.4). For the other methods, results are pretty consistent across amount of censoring except by wSVM in which there is a decrease in all metrics except in AUC which remains pretty constant, around 0.75. In the non-proportionality of hazards scenarios (Table 7.5) results for all methods are in general pretty similar.

7.4.4 Distribution of censoring

In the 300 observations scenario and proportionality of hazards all results are pretty similar, being equal in most of metrics, within each method by distribution of censoring. The best performance is achieved by the Cox model with a 0.89 accuracy, 0.77 in Matthews correlation, 0.49 NMI and 0.96 AUC. With regards the approach to treat the censoring (conditional survival or proportionality of time), there are no clear differences between the approach used and the distribution of the censoring.

In the 50 observations scenario the performance of the methods is worst compared to the 300 observations. Within the proportionality of hazards approach, in uniform and positive skew censoring, performance is slightly better for all methods than the negative skew distribution, except by inSVM-averaging model whose performance for all metrics is pretty similar in all censoring distributions.

7.4.5 Sample size

In all compared scenarios the performance is worst when decreasing the sample size. The maximum difference in metrics is observed in the proportional data for the Cox model: a decrease of approximately 0.32 units in accuracy, 0.44 in Matthews correlation, 0.35 in NMI and 0.4 in AUC. In the same scenario the SVM methods present lower values being the maximum difference in accuracy of around 0.14 units in wSVM model and the other methods around 0.1 difference. For the AUC the average difference due to the number of observation in the SVM methods is around 0.11 units. Kernel Cox regression approach performs similar to pSVM with 0.07 units difference in accuracy and 0.1 units in AUC.

7.4.6 Conditional survival and proportionality of time

In all 300 observations scenarios tested the conditional survival approach performs slightly better than the proportional of time. This difference is larger in the 50 observations scenario, specially in the non-proportionality of hazards one. Overall differences between both methods are around 0.02 units in accuracy and around 0.02 units in AUC.

7.4.7 Semi-supervised support vector machines approach

The inSVM method, both gradient and averaging approach, performs in all scenarios tested closest to the best method within each scenario. The performance does not present clearly differences between gradient and averaging in all scenarios and metrics, although averaging one performs slightly better, being more robust to modifications in proportion of censoring.

Chapter 8

Discussion and Future Research

8.1 Discussion

This part of the thesis has focused on the exploration of SVM for binary classification methods to deal with censored data. The research has tried to show the performance of this approach under several scenarios, on the one hand, and, on the other hand, to develop R functions implementing all proposed methods. Analysis of time-to-event data is of importance in any scientific area, but specially in biomedical field. One important issue to deal with is when the number of predictors is larger or approximately equal to the number of observations. SVM models are flexible enough to deal with this issue but only the regression approach have been investigated, the classification methodology have received little attention.

The proportional hazards model has been considered the gold standard. The Cox model is a robust model since it is, basically, a rank model that does not assume a distribution for the time-to-event variable and, the number of necessary events per variable it can be around 3 in *controlled* situations. That is the main reason the performance of this model is so good in the 300 observations, proportional hazards scenario. But for the same reason, the performance is worst in 50 observations, non-proportional hazards scenarios.

The wSVM approach is a naive approach because it's weighting the observations. As expected, the obtained results are highly dependent on the proportion of censoring because of the simplicity of the method. Surprisingly, however, results are influenced by the censoring distribution in a similar way to the other methods, suggesting that this approach is mainly affected by the proportion of censoring. Another important aspect to comment is that wSVM results are pretty similar to the LUPI ones as pointed out by Lapin et al. (2014). Both methods are using, basically, the same unique information (censored data), with the advantage for the wSVM method which is much less time consuming. However, the LUPI approach is, in general, more robust than the wSVM.

With respect to the LUPI approach, we have included the censoring data as a privileged information in the correcting space. Our results are consistent with the ones by Shiao and Cherkassky (2013) in which LUPI performs worst than Cox model and pSVM in all compared scenarios (being some of them comparable to the ones presented in this thesis). The correcting space is used as a complementary information to be combined with the decision space. Therefore, is not directly used to define the class of the observations, as it is in pSVM for example or wSVM. Related to this, we agree with Serra-Toro et al. (2014) conclusions that further work is needed to fully understand the LUPI approach and how the correcting space *interacts* with the decision one.

The performance of the pSVM has been similar to the Cox model in the 300 observations scenarios and better than kernel Cox regression, being the linear kernel slightly better than the radial one, as observed by Shiao and Cherkassky (2013). These results suggest that perhaps a more fine grid search could have benefit the overall performance of the non-linear approach. Besides that, in Shiao and Cherkassky (2013), results are consistent with ours considering that the authors conclude that linear pSVM performs better than pSVM using Gaussian kernel.

Our conditional survival proposed approach performs better than proportionality of time in all compared scenarios. The conditional method takes into account the events and follow-up period including more information and being more accurate in the *weighting* estimation than the proportionality of time approach. The latter is assuming linearity and does not take into account specificities of the data, for instance, variability in survival due to intrinsic data such the site or the vaccine efficacy. However, one aspect to be remarked is that the conditional approach is assuming that the survival probability of the test data is *similar* to the training one. This is a reasonable assumption, but depending on the difference between survival probabilities results, such as accuracy, may be affected.

With respect inSVM approach (both gradient and averaging), in the 300 observations scenarios, results are pretty similar to Cox, kernel Cox regression and pSVM. Although, being a semi-supervised approach, in which the censoring is not addressed using any weight, the performance is similar to the other methods in which censoring is taken into account. That could be explained because we are assuming that censoring is independent from the events and representative of the data, which means that patterns in observed data should be valid for censored observations. Therefore, the local invariances assumptions should be valid. On one hand, the main drawback of the local invariances is that its computational time is high, specially the gradient one. Must be remarked that the derivative invariance described in Section 6.3.3 calculates the gradient for all pair of variables and observations, as expected this increases the computational time. On the other hand, the main advantage is that no extra assumptions related to the censoring distribution are made.

During the development of the R functions several optimization and mathematical programming packages implemented in R have been tested in order to reduce computational time and find the optimal package. The optimization problems we have presented in this thesis are convex quadratic but some of them present equality constraints, inequality constraints and box constraints; for each type of constraint there are several R packages with different syntaxes structures making difficult the code standardization. This is an issue that have been, recently addressed, by optimization theory experts (Nash et al., 2011, 2014). Although this is beyond the scope of this thesis, this is an aspect that needs to be considered for future work.

8.2 Future research

One point of interest for future research is the fact of adding variability to the probability given in conditional survival approach. Since the probability is based on the Kaplan-Meier estimation, we can obtain a confidence interval for this specific estimation. Thus, the performance using the two bounds could be evaluated and the robustness of the prediction assessed. Specifically for the LUPI approach as there are no weights, including the variability in the correcting space perhaps could add more information and the method would be more generalizable. In a similar way, the wSVM and pSVM could implement a variability of the weights, or perhaps computing the predictions on the two confidence interval boundaries and then applying some kind of averaging, as it is done in multiple imputation methods when treating missing values (Little and Rubin, 1987).

Regarding the inSVM model it does not take into account any information related to the survival probability of the data. As a consequence, we are not using all available data. An approach for future research to be evaluated is the use of different weights for the censored observations as it is done in wSVM and assess the performance. This research might be divided in two main lines. On the one hand, the study of the method using an overall and unique weight for the censored observations estimated by the Kaplan-Meier survival probability at the end of follow-up, for instance. On the other hand the study of specific weights for each observation as in the presented wSVM method. We have conducted the local invariances to the unknown classes but as Lee et al. (2006) suggest, they can be applied to all observations (known and unknown). Given the observed results, we think that it would be interesting to study the performance of this alternative approach.

When applying the conditional survival approach we assume that the test and training data have a *similar* survival curve, i.e., both datasets are samples from the same population. It would interesting to study the performance and robustness of the proposed methods under violations of this assumption and determine the circumstances in which the predictions of events are overestimated and underestimated by increasing (decreasing) the survival probability at the end of follow-up in the test data, for instance.

In this thesis we have based the non-proportionality scenario in a shared frailty context. An alternative perspective for non-proportionality is due to the time-varying coefficient in which, for example, the efficacy of the vaccine is not constant along the follow-up period. Therefore, we are in a situation of non-proportionality due to an specific variable. It would be interesting and useful to study the behaviour of the proposed methods under this scenario and to adapt them into the time-varying coefficient context.

Part III

Relevance of Variables in Support Vector Machines for Survival Analysis

Chapter 9

Material and Methods

In the following sections of the present chapter the proposed methods related to relevance of variables in support vector machines are presented. Section 9.1 presents the proposed RFE-pseudo-samples algorithm. The two proposed approaches based on KPCA are described in Section 9.2.

9.1 RFE-pseudo-samples

Krooshof et al. (2010) and Postma et al. (2011) propose a method to visualize the importance of variables using *pseudo-samples* in kernel partial least square and support vector regression context respectively. The method they propose for visualizing variable importance is summarized in the following steps:

1. Optimize the kernel function and parameter settings for the SVM model used.

2. From the covariates data $\boldsymbol{x}_i$, for all $i = 1, \ldots, n$, observations compute the kernel matrix K, by using the optimized kernel function, and then center the matrix, obtaining K^c.

3. Apply singular value decomposition on K^c and construct several score plots (for the various combinations of principal components) for the n samples in K^c. Inspect these score plots to find the direction(s) in which maximum separation is obtained.

4. Construct a matrix P_j for the jth variable which contains p pseudo-samples. The range of the pseudo-samples should vary from the minimum value to the maximum of the original variable, and the other variables are constant with a value of 0.

5. Apply the kernel function K to the pseudo-samples data to obtain the kernel matrix of the pseudo-samples C.

6. Project the rows of the centered pseudo-samples kernel matrix C^c in the score plot that was found in step 3.

7. Repeat steps 4-6 for each variable j.

The resulting plot contains a trajectory of pseudo-samples for each original variable in the KPCA space. These variables yield information about the relative contribution of the variables to the SVM model. The other idea that the authors suggested, but based on the SVR approach is plotting the prediction of the pseudo-samples for the response $y \in \mathbb{R}$ for a given model, and based on that visualize the relevance of the variables.

Following these two main ideas we propose a method for visualizing the relevance of variables into the SVM for binary classification framework. Briefly, the main steps look as follows:

1. Optimize the SVM method and tune the parameters.

2. Create a pseudo-samples matrix with equally distanced values from the original variable, maintaining the other variables as the mean or median. As the data is usually normalized, we assume that the mean is 0,

$$\begin{array}{cl}
\begin{array}{ccccc} \text{Var.1} & \text{Var.2} & \text{Var.3} & & \text{Var.p} \end{array} & \\
\left(\begin{array}{ccccc}
z_1 & 0 & 0 & \dots & 0 \\
z_2 & 0 & 0 & \dots & 0 \\
z_3 & 0 & 0 & \dots & 0 \\
 & & \vdots & & \\
z_p & 0 & 0 & \dots & 0
\end{array}\right) &
\begin{array}{l}
\text{pseudo-sample}_1 \\
\text{pseudo-sample}_2 \\
\text{pseudo-sample}_3 \\
\\
\text{pseudo-sample}_p
\end{array}
\end{array} \tag{9.1}$$

3. Predict the decision value for each pseudo-sample (not the predicted class) using the SVM model fitted in step 1.

4. Plot in the X-axis the range of the variable and in the Y-axis the decision value. In order to plot all variables in the same figure it is necessary to scale and center all variables if their distribution is different.

The rationale of the proposed method is that if the response is associated with the variable, modifications in the variable will affect predictions of the class. On the contrary, if a variable is not associated with the response it does not matter which value one choose for that variable since the prediction will be approximately constant. Therefore, since the decision value can be used as a score that measure distance to the hyperplane, the larger the absolute value the more confident we are that the observation belong to the predicted class remarked by the sign.

9.1.1 Visualization and interpretation of variable relevance

Once we have plotted the decision values and the range of all variables, we have to take into account several points:

- Values associated with the response: since we are plotting the range of the variable and the decision value, we are able to detect whether larger values of the variable are protective or risk factors.

- The proposed method fix the values of the non-evaluated variables to 0 but this can be modified to evaluate the performance of the desired variables fixing the values to any other biologically meaningful value.
- The shape of the curve can be indicative of the type of association of each variable with respect the response.
- The variability on the decision values can be indicative of the relevance of the variable with the response.

9.1.2 Ranking variables

We are interested in identifying which variable presents less variability on the prediction and automatically remove it and re-iterate the process, applying the RFE algorithm. This can be done visualizing the plots described in Section 9.1.1 but becomes infeasible for a medium to large number of variables. Therefore, a metric should be used to measure the variability and rank the variables based on that. One univariate robust metric is the median absolute deviation (MAD) which is expressed as

$$\mathrm{MAD}_j = \mathrm{median}\left(|D_{ij} - \mathrm{median}(D_j)|\right) c$$

being D_{ij} the decision value of the variable j for the pseudo-sample i and being $\mathrm{median}(D_j)$ the median of all decision values for the evaluated variable j. The constant c is equal to 1.4826, and it is incorporated in the expression to ensure consistency in terms of expectation so that

$$E(\mathrm{MAD}(D_1, \ldots, D_n)) = \sigma$$

for D_i distributed as $N(\mu, \sigma^2)$ and large n. Our proposal is to apply the RFE approach to pseudo-samples prediction being the algorithm as it is described in Algorithm 2.

```
Data   : Dataset with $p^*$ variables, time-to-event and status.
Input  : Number of equidistant cutoff points $c^*$.
Output: Ranked list of variables according to their relevance.

Find the optimal values for the tuning parameters of the SVM model;
$p \leftarrow p^*$;
while $p \geq 2$ do
    $SVM_p \leftarrow$ SVM with the optimized tuning parameters for the $p$ variables and
    observations in Data;
    for $i = 1$ to $p$ do
        $pseudo_i \leftarrow$ prediction vector of $c^*$ pseudo-samples for variable $i$;
        $rank.criteria_i \leftarrow$ Median Absolute Deviation of $pseudo_i$ vector;
    end
    $min.rank.criteria \leftarrow$ variable with lowest value in
    $(rank.criteria_1, \ldots, rank.criteria_p)$;
    Remove $min.rank.criteria$ from Data;
    $Rank_p \leftarrow min.rank.criteria$ ;
    p ← p - 1 ;
end
$Rank_1 \leftarrow$ variable in Data $\notin (Rank_2, \ldots, Rank_{p^*})$;
return $(Rank_1, \ldots, Rank_{p^*})$
```

Algorithm 2: Pseudo-code of the RFE-pseudo-samples algorithm.

9.2 RFE-kernel principal components input variables

Reverter et al. (2014) propose a method, in a different context than the SVM, using KPCA to represent, for each variable, the direction of maximum growth locally. So, given two components the maximum growth for each variable is indicated in a plot in which each axis is one of the components. After representing all observations in the new space, if a variable is relevant under this context will show a clear direction across all samples and if it's not the sample's direction will be random. In the same work the authors suggest to incorporate functions of the original variables into the KPCA space, so it's possible to plot not only growth of individual variables but combination of them if biologically makes sense. Our proposed method, referred as *RFE-KPCA-maxgrowth*, consists of the following steps:

1. Fit the SVM.
2. Create the KPCA space using the tune parameters found in the SVM process with all variables.
3. Represent the two first components of the KPCA.
4. Compute and represent the input variables and the decision function of the SVM into the KPCA space.
5. Compute the average angle of each variable-observation with the decision function into the KPCA space.
6. Based on a metric calculate which variable is less relevant.
7. Remove the least relevant variable.
8. Repeat all the process until there is only one variable left.

9.2.1 Kernel principal component analysis

Given a feature space $\mathcal{H}$ related to the input domain by a map

$$\begin{aligned} \phi :& \mathcal{X} \to \mathcal{H} \\ & \boldsymbol{x} \mapsto \phi(\boldsymbol{x}) \end{aligned}$$

which is possible non-linear. Let define

$$\overline{\phi} := \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{x}_i)$$

then the points

$$\tilde{\phi}(\boldsymbol{x}_i) = \phi(\boldsymbol{x}_i) - \overline{\phi} \tag{9.2}$$

are centered. Let $\tilde{K}$ denote the kernel matrix of centered points $\tilde{K}_{ij} = \langle \tilde{\phi}(\boldsymbol{x}_i), \tilde{\phi}(\boldsymbol{x}_j) \rangle$. Because we do not have centered data, we cannot compute $\tilde{K}$ explicitly, however, according to Scholkopf and Smola (2001), it can be expressed in terms of its noncentered counterpart K. Using the vector $\mathbf{1}_n = (1, \ldots, 1)^\top$, we can get the expression

$$\tilde{K} = K - \frac{1}{n} K \mathbf{1}_n \mathbf{1}_n^\top - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top K + \frac{1}{n^2} (\mathbf{1}_n^\top K \mathbf{1}_n) \mathbf{1}_n \mathbf{1}_n^\top$$

In $\mathcal{H}$ the covariance matrix takes the form

$$\tilde{\boldsymbol{C}} = \frac{1}{n} \sum_{j=1}^{n} \tilde{\phi}(\boldsymbol{x}_j) \tilde{\phi}(\boldsymbol{x}_j)^\top$$

If $\mathcal{H}$ is infinite-dimensional, we think of $\tilde{\phi}(\boldsymbol{x}_j)\tilde{\phi}(\boldsymbol{x}_j)^\top$ as a linear operator on $\mathcal{H}$, mapping

$$\boldsymbol{x} \mapsto \phi(\boldsymbol{x}_j) \langle \phi(\boldsymbol{x}_j), \boldsymbol{x} \rangle.$$

We have to find eigenvalues $\tilde{\lambda} \geq 0$ and nonzero eigenvectors $\tilde{\boldsymbol{V}} \in \mathcal{H} \setminus \{0\}$ satisfying

$$\tilde{\boldsymbol{C}} \tilde{\boldsymbol{V}} = \tilde{\lambda} \tilde{\boldsymbol{V}} \tag{9.3}$$

To find the solution of (9.3) we solve the dual eigenvalue problem

$$\tilde{K} \tilde{\boldsymbol{\alpha}} = n \tilde{\lambda} \tilde{\boldsymbol{\alpha}} \tag{9.4}$$

with $\tilde{\boldsymbol{\alpha}}$ being the expansion coefficients of an eigenvector in terms of the centered points in (9.2)

$$\tilde{\boldsymbol{V}} = \sum_{i=1}^{n} \tilde{\alpha}_i \tilde{\phi}(\boldsymbol{x}_i) \tag{9.5}$$

The solution $\tilde{\boldsymbol{\alpha}}^k, k = 1, \ldots, r$, is normalized by normalizing the corresponding vector $\tilde{\boldsymbol{V}}^k$ in $\mathcal{H}$, which translates into

$$\tilde{\lambda}_k \langle \tilde{\boldsymbol{\alpha}}^k, \tilde{\boldsymbol{\alpha}}^k \rangle = 1 \tag{9.6}$$

To find the coordinates of a test point $\boldsymbol{s}$, with an image $\phi(\boldsymbol{s})$ in $\mathcal{H}$, we need to compute projections onto the eigenvectors $\tilde{\boldsymbol{V}}^k$ in $\mathcal{H}$

$$(\langle \tilde{\phi}(\boldsymbol{s}), \tilde{\boldsymbol{V}}^k \rangle)_{1 \times r} = \left(\boldsymbol{Z}^\top - \frac{1}{n} \mathbf{1}_n^\top K \right) \left(\mathbf{1}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top \right) \tilde{\boldsymbol{V}} \tag{9.7}$$

where $\tilde{\boldsymbol{V}}$ is a $n \times r$ matrix whose columns are the eigenvectors $\tilde{V}^1, \ldots, \tilde{V}^r$, and

$$\boldsymbol{Z} = (K(\boldsymbol{s}, \boldsymbol{x}_i))_{n \times 1} \tag{9.8}$$

9.2.2 Representation of input variables

We approach the problem of the interpretability of kernel methods by mapping simultaneously data points and relevant variables in a low dimensional linear manifold immersed in the feature space $\mathcal{H}$. Such linear manifold, usually a plane, can be determined according to some statistical requirement, for instance, we shall require that the final Euclidean interdistances between points in the plot have to be, as far as possible, similar to the interdistances in the feature space, which shall lead us to the KPCA. We have to distinguish between the feature space $\mathcal{H}$ and the surface in that space to which points in input space $\mathbb{R}^n$ actually map, which we denote by $\phi(\mathcal{X})$. In general is an n dimensional manifold embedded in $\mathcal{H}$. We assume here that $\phi(\mathcal{X})$ is sufficiently smooth that a Riemannian metric can be defined on it (Scholkopf et al., 1999).

The intrinsic geometrical properties of $\phi(\mathcal{X})$ can be derived once we know the Riemannian metric induced by the embedding of $\phi(\mathcal{X})$ in $\mathcal{H}$. The Riemannian metric can be defined by a symmetric metric tensor g_{ab}. It is interesting to note that we do not need to know the explicit mapping ϕ to construct g_{ab}; it can be written solely in terms of the kernel.

Any relevant variable can be described by a real valued function f defined on the input space $\mathbb{R}^n$. Since we assume that the feature map ϕ is one-to-one, we can identify f with $\hat{f} \equiv f \circ \phi^{-1}$ defined on $\phi(\mathcal{X})$. We aim to represent the gradient of $\hat{f}$. The gradient of $\hat{f}$ is a vector field defined on $\phi(\mathcal{X})$ through its components under the coordinates $\boldsymbol{x} = (x^1, ..., x^p)$ as

$$\operatorname{grad}(\hat{f})^a = \sum_{b=1}^{p} g^{ab}(\boldsymbol{x}) D_b f(\boldsymbol{x}) \qquad a = 1, \ldots, p \tag{9.9}$$

where g^{ab} is the inverse of the metric matrix $G = (g_{ab})$, and D_b denotes the partial derivative with respect the b variable.

The curves v corresponding to the integral flow of the gradient, i.e., the curves whose tangent vectors a t are $v'(t) = \operatorname{grad}(\hat{f})$, indicate, locally, the maximum variation directions of $\hat{f}$. Under the coordinates $\boldsymbol{x} = (x^1, \ldots, x^p)$ the integral flow is the general solution of the first order differential equation system

$$\frac{dx^a}{dt} = \sum_{b=1}^{p} g^{ab}(\boldsymbol{x}) D_b f(\boldsymbol{x}) \qquad a = 1, \ldots, p \tag{9.10}$$

which has always local solution given initial conditions $v(t_0) = \boldsymbol{w}$.

In order to enrich the interpretability of KPCA output plot we can represent in it the curves $v(t)$ solution of (9.10) which indicates, locally, the maximum variation directions of $\hat{f}$, or also, the corresponding gradient vector given in (9.9).

Let $v(t) = k(\cdot, \boldsymbol{x}(t))$, where $\boldsymbol{x}(t)$ are the solutions of (9.10). If we define

$$\boldsymbol{Z}_t = \Big(k(\boldsymbol{x}(t), \boldsymbol{x}_i)\Big)_{n\times 1}. \tag{9.11}$$

Taking into account the equation (9.7) the induced curve, $\tilde{v}(t)$, expressed in matrix form, is given by the row vector:

$$\tilde{v}(t)^q_{1\times r} = \left(\boldsymbol{Z}_t^\top - \frac{1}{n}\mathbf{1}_n^\top K\right)\left(\boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right)\tilde{\boldsymbol{V}} \tag{9.12}$$

where $\boldsymbol{Z}_t$ has the form (9.11).

We can also represent the gradient vector field of $\hat{f}$, that is, the tangent vector field corresponding to curve $v(t)$ through its projection into the KPCA subspace. The tangent vector at $t = t_0$, if $\boldsymbol{x}_0 = \boldsymbol{\phi}^{-1} \circ v(t_0)$ is given by $\frac{dv}{dt}|_{t=t_0}$, and its projection, in matrix form, is given by the row vector

$$\left(\frac{d\tilde{v}}{dt}\Big|_{t=t_0}\right)_{1\times r} = \frac{d\boldsymbol{Z}_t^\top}{dt}\Big|_{t=t_0}\left(\boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right)\tilde{\boldsymbol{V}}, \tag{9.13}$$

with

$$\frac{d\boldsymbol{Z}_t^\top}{dt}\Big|_{t=t_0} = \Big(\frac{d\boldsymbol{Z}_t^1}{dt}\Big|_{t=t_0}, \ldots, \frac{d\boldsymbol{Z}_t^n}{dt}\Big|_{t=t_0}\Big)^\top,$$

and,

$$\begin{aligned}
\frac{d\boldsymbol{Z}_t^i}{dt}\Big|_{t=t_0} &= \frac{dk(\boldsymbol{x}(t), \boldsymbol{x}_i)}{dt}\Big|_{t=t_0} \\
&= \sum_{a=1}^{p} D_a k(\boldsymbol{x}_0, \mathrm{x}_i)\frac{dx^a}{dt}\Big|_{t=t_0} \\
&= \sum_{a=1}^{p} D_a k(\boldsymbol{x}_0, \mathrm{x}_i)\sum_{b=1}^{p} g^{ab}(\boldsymbol{x}_0) D f_b(\boldsymbol{x}_0) \\
&= \sum_{a=1}^{p}\sum_{b=1}^{p} D_{p+a} k(\boldsymbol{x}_i, \boldsymbol{x}_0) g^{ab}(\boldsymbol{x}_0) D f_b(\boldsymbol{x}_0).
\end{aligned}$$

Representing input variables

For instance, input variables x^b, $b = 1, \ldots, p$, are primary relevant variables. In that case, we would like to represent the function $f(\boldsymbol{x}) = x^b$. We consider the Gaussian kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp -\frac{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2}{\sigma}, \qquad \forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^n$$

where $\sigma > 0$. Then, using the above notation, since

$$D_{p+a} k(\boldsymbol{x}_i, \boldsymbol{x}_j) = -\frac{2(x_j^a - x_i^a)}{\sigma} \exp\left(-\frac{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2}{\sigma}\right) = \frac{2(x_i^a - x_j^a)}{\sigma} k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

and taking into account the Riemannian metric corresponding to a Gaussian kernel, the contravariant components of the metric tensor are

$$g^{ab}(\boldsymbol{x}_0) = \frac{\sigma}{2}\delta^{ab}$$

where δ^{ab} are the contravariant Kronecker's deltas, therefore, we obtain

$$\frac{d\boldsymbol{Z}_t^i}{dt}\Big|_{t=t_0} = k(\boldsymbol{x}_i, \boldsymbol{x}_0)\sum_{a=1}^{p}(\boldsymbol{x}_i^a - \boldsymbol{x}_0^a)$$

9.2.3 Visualization and interpretation of importance of variables

As described by Reverter et al. (2014) we can represent for each test sample the variables as vectors (with a pre-specified length), which indicate the direction of maximum growth in each variable or a function of them. When two variables are positively correlated, the directions of maximum growth for all samples should go in the same direction. When two variables are negatively correlated the direction should be overall opposite, and if they are no correlated, directions should be random.

9.2.4 Ranking of variables

Our proposal is to take advantage of the representation of direction of input variables applying two alternative approaches:

- To include the SVM predicted decision values for each training sample as an extra variable, what we call *reference variable.* Then, compare directions of each one of the variables with the reference.

- To include the direction of the SVM decision function and use it as the *reference direction.* Since it is as a real-valued function of the original variables we can represent the direction of this expression. Specifically, the decision function removing the sign function of the expression of SVM is given by

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{9.14}$$

  we can reformulate (9.14) to

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \varrho_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{9.15}$$

  where $\varrho_i = \alpha_i y_i$. Applying the methodology defined in Section 9.2.2 to the (9.15) function we obtain

$$\left.\frac{d\boldsymbol{Z}_t^i}{dt}\right|_{t=0} = k(\boldsymbol{x}_i, \boldsymbol{x}) \sum_{a=1}^{p} (x_i^a - x^a) \left[\sum_{j=1}^{n} \varrho_j \sigma (x_j^a - x^a) k(\boldsymbol{x}_j, \boldsymbol{x}) \right] \tag{9.16}$$

In both alternatives and following the interpretation described in Section (9.2.3), we can calculate the overall similarity of one variable with respect the reference (either the prediction or the decision function) by averaging the angle of the maximum growth vector for all training points with the reference. So, if, for a given training point, the angle of the direction of maximum growth of variable p with the reference is 0° (0 radians) would mean that the vector of directions overlap and they are perfectly positively associated. If the angle is 180° (π radians) they go in opposite direction, indicating that they are perfectly negatively associated. By averaging the angle of all training points we obtain a summary of the similarity of each variable with the reference and consequently whether is relevant or not. Assuming that there is noise in real data a variable is classified as relevant or not compared to the others: the variable closest to the overall angle taking into account all variables is assumed to be the least relevant. Based on this, we can apply a RFE-KPCA-maximum-growth approach for prediction and for decision function as defined by Algorithm 3.

```
Data   : Dataset with $p^*$ variables, time-to-event and status.
Input  : Method of KPCA-maxgrowth to be applied (Prediction or Function).
Output: Ranked list of variables according to their relevance.

Find the optimal values for the tuning parameters of the SVM model.;
$p \leftarrow p^*$;
while $p \geq 2$ do
    Fit SVM with the optimized tuning parameters for the $p$ variables and observations in Data.;
    if $Method = Prediction$ then
        Calculate the decision value for each training point;
        Incorporate that decision value vector as a variable $k$;
        Calculate the KPCA space with all $(p+1)$ variables (including $k$);
        Project all variables into KPCA space for the first two components;
    end
    if $Method = Function$ then
        Calculate the KPCA space with all $p$ variables;
        Project all variables into KPCA space for the first two components;
        Project the SVM decision function, $k$, into the KPCA space;
    end
    $angle_{ij} \leftarrow$ calculate the angle of each training point $i$ and variable $j$ with respect $k$;
    $angle.mean_j \leftarrow$ average $angle_{ij}$ values by each variable $j$ obtaining a vector of $p$ components;
    $med_p \leftarrow$ overall median of all $angle.mean_1, \ldots, angle.mean_p$;
    if $p \geq 3$ then
        $rank.criteria_p \leftarrow (angle.mean_j - med_p)^2$;
    else
        $rank.criteria_p \leftarrow (angle.mean_j - 90°)^2$;
    end
    $min.rank.criteria \leftarrow$ variable with lowest value in $(rank.criteria_1, \ldots, rank.criteria_p)$;
    Remove $min.rank.criteria$ from Data;
    $Rank_p \leftarrow min.rank.criteria$ ;
    p ← p - 1 ;
end
$Rank_1 \leftarrow$ variable in Data $\notin (Rank_2, \ldots, Rank_{p^*})$;
return $(Rank_1, \ldots, Rank_{p^*})$
```

Algorithm 3: Pseudo-code of the RFE-KPCA-maximum-growth algorithm for both function and prediction approach.

Chapter 10

Simulation Study

In this chapter we evaluate the performance of the methodology proposed in Chapter 9 by means of a simulation study. The simulated scenarios and the data generation process is described in Section 10.1. The metrics used to evaluate the performance of the proposed algorithms are presented in Section 10.2. Finally, simulation results are shown in Section 10.3.

10.1 Simulation of scenarios and data generation

The simulated scenarios are based on covariates pattern as described in Section 7.1.1. Therefore, for each simulated dataset 30 variables normally distributed, structured in 4 different blocks attending to their correlation (high, medium, low and none) have been simulated.

The time-to-malaria event variable is based on a proportional hazards model as described in Section 7.1.2. The number of observations by dataset is 50. The censoring distribution follows a Uniform distribution (10% censoring) as described in Section 7.1.3.

10.1.1 Relevance of variables scenarios

To evaluate the performance of the methods, several scenarios regarding the variables associated with time-to-malaria have been tested. Scenarios are generated to test the performance in situations in which the relevant variables are: i) highly correlated (with other relevant and non-relevant variables) and non-correlated, ii) present positive (or negative) association with the response, and iii) present linearities, non-linearities and interactions among them. The relevant variables for each one of the 6 simulated scenarios are:

1. Variable 1

2. -Variable 29 + Variable 30

3. -Variable 1 + Variable 8 + Variable 20 + Variable 29 - Variable 30

4. Variable 1 + Variable 2 + Variable 1 × Variable 2

5. Variable 1 + Variable 30 + Variable 1 × Variable 30 + Variable 20 + Variable 20^2

6. Variable 1 + Variable 1^2 + exp (Variable 30)

10.1.2 Comparison of methods

The SVM model used to fit the data is the pSVM (Gaussian kernel) with conditional estimation of the uncertainties (based on Kaplan-Meier) as described in Section 6.2. The reason being, it is one of the SVM models with highest accuracy. The parameters selected to perform the grid-search are the same as in Section 7.1.5. For Gaussian kernel are 0.25, 0.5, 1, 2 and 4. The C and $\widetilde{C}$ values are 0.1, 1, 10 and 100. The ρ value has been fixed to be 0.0001. The tuning parameters are found as described in Section 7.2 and have been fixed for each RFE iteration, i.e., are not estimated at each iteration. Once the optimal parameters for the pSVM are found the methods compared are:

- RFE-Guyon for non-linear data: this method is taken as the gold standard.

- RFE-KPCA-maxgrowth-prediction: the KPCA is based on Gaussian kernel whose parameter is the one obtained in the pSVM model.

- RFE-KPCA-maxgrowth-decision: the KPCA is based on Gaussian kernel whose parameter is the one obtained in the pSVM model.

- RFE-pseudo-samples: the range of the data, to create the pseudo-samples is created splitting data into 50 equidistant points. The range of the pseudo-samples goes from -2 to 2, since variables are normally distributed around 0 approximately.

10.2 Metrics to evaluate algorithm performance

The mean and standard deviation of the position obtained in 100 simulated datasets have been used to summarize the performance by method and scenario. For the RFE-pseudo-samples algorithm the first iteration figure with all 100 datasets is created summarizing the information by variable. For the RFE-maxgrowth approach, as example, one of the datasets is presented in order to interpret the method, since is not possible to summarize all 100 principal components plots in one figure.

10.3 Simulation results

In this section main results are described by algorithm and scenario. Results are structured according to overall ranking of variables and visualization and interpretation of two scenarios for illustrative purposes. All other tables and figures summarizing the results are presented in Appendix D.

10.3.1 Overall ranking comparison

Scenario 1 results are shown in Figure 10.1. All 4 methods identify the relevant variable being the RFE-maxgrowth-prediction the one with the lowest average rank, followed by the RFE-maxgrowth-function, RFE-pseudo-samples and RFE-Guyon. For all methods, except the RFE-Guyon, a set of variables is closest to the Variable 1 rank (variables 2 to 8). These variables are highly correlated with Variable 1.

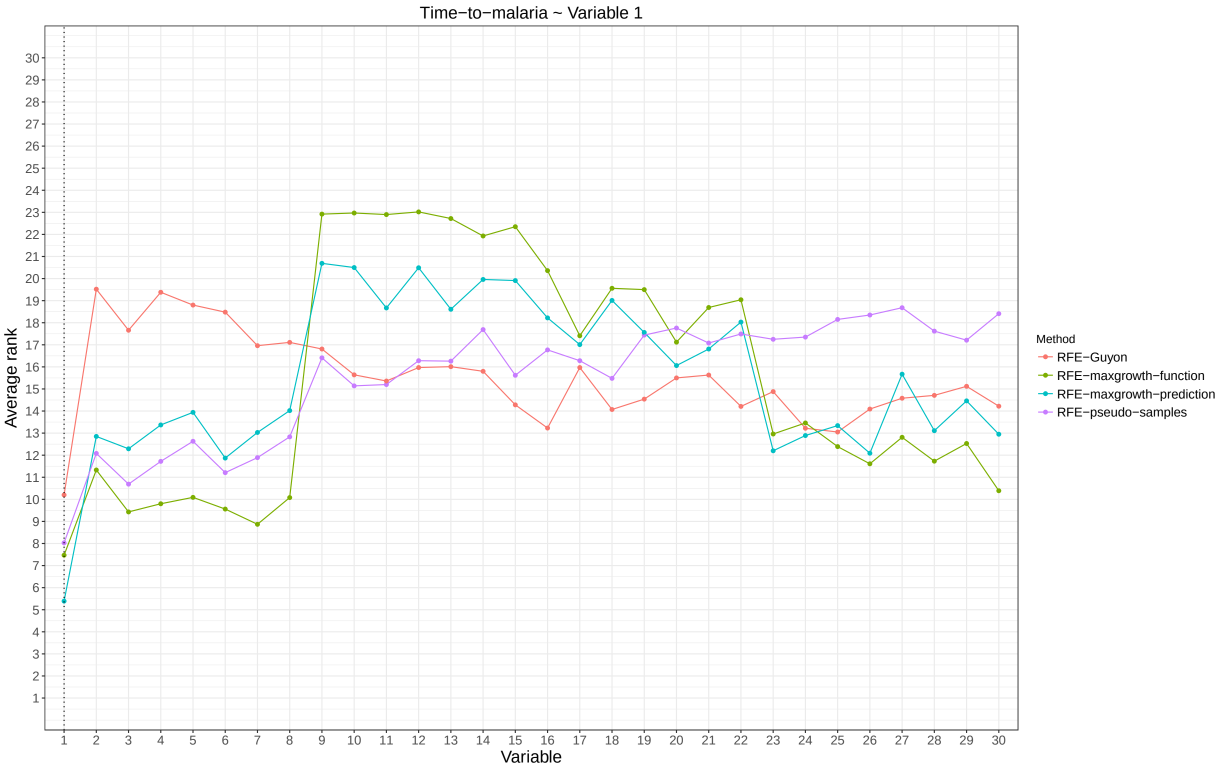


Figure 10.1: Average rank by variable and method for the 100 simulated datasets for Scenario 1. Dotted line represents the variable used to generate the time-to-malaria variable. The lower the rank the more relevant the variable is for the specific algorithm.

For scenario 2 (Figure 10.2), the true relevant variables are identified for all 4 algorithms, being the average rank pretty similar, except the RFE-maxgrowth-function. The specific order is RFE-Guyon, RFE-maxgrowth-prediction, RFE-pseudo-samples and RFE-maxgrowth-function. The average rank for the other non-relevant variables is similar for all methods.

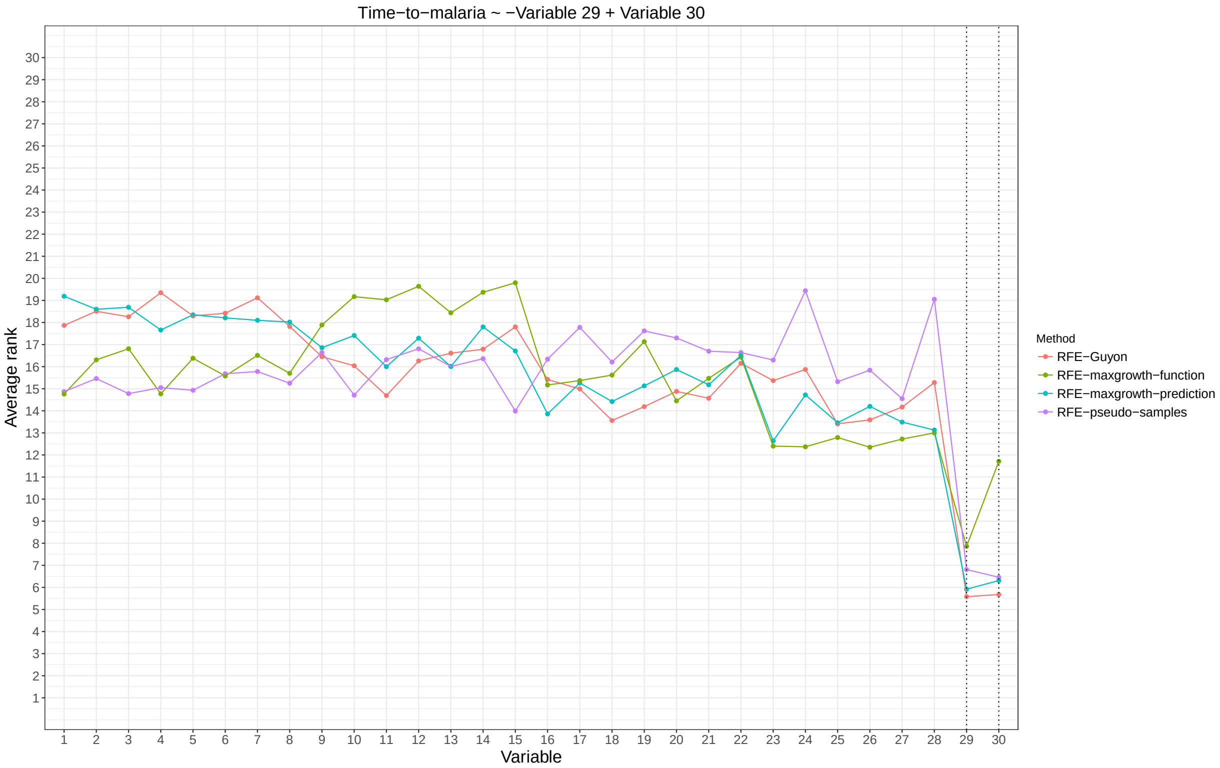


Figure 10.2: Average rank by variable and method for the 100 simulated datasets for Scenario 2. Dotted line represents the variable used to generate the time-to-malaria variable. The lower the rank the more relevant the variable is for the specific algorithm.

In scenario 3 (Figure 10.3), 5 variables are relevant in the true model. The algorithms are able to detect the relevant independent variables, except the RFE-maxgrowth-function, that for this set of variables (variables 20, 29 and 30) is the worst method. For the other 3 algorithms, and this set of variables, the RFE-pseudo-samples is slightly better than the others and RFE-Guyon slightly worst. For the other 2 highly correlated variables (Variable 1 and Variable 8) the two best methods are clearly RFE-pseudo-samples and RFE-maxgrowth-function.

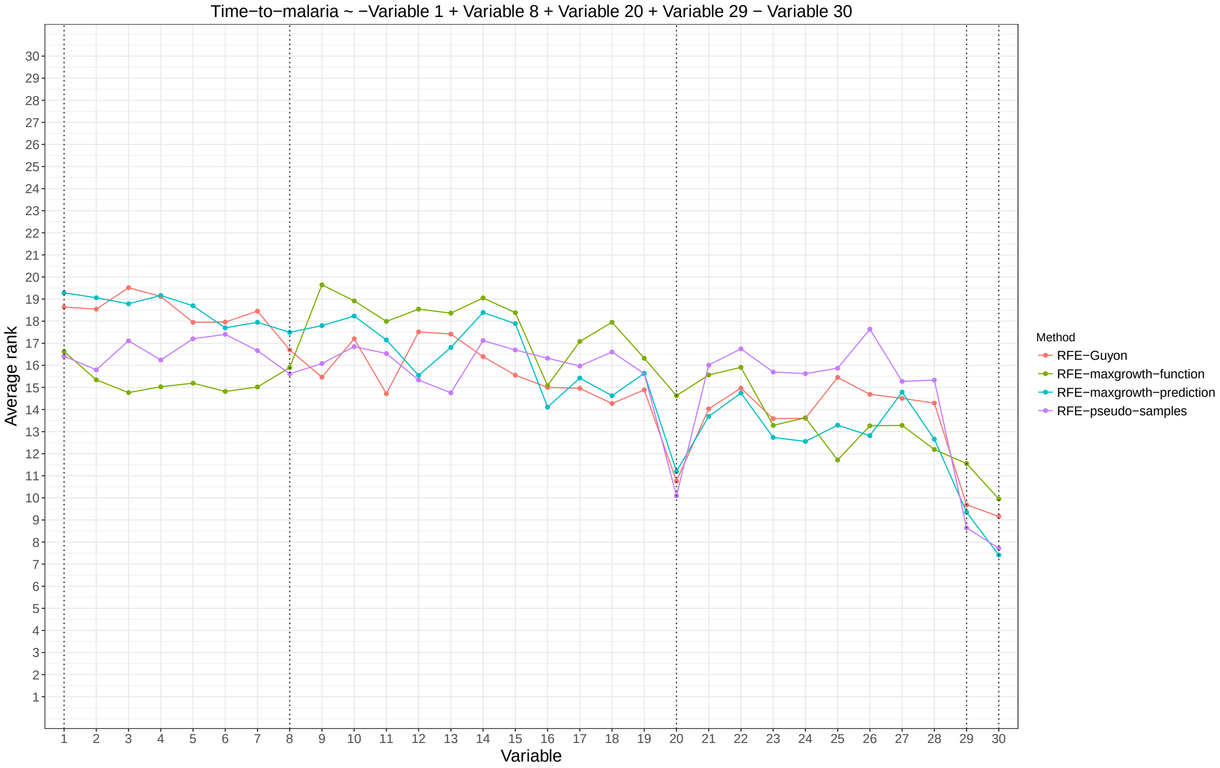


Figure 10.3: Average rank by variable and method for the 100 simulated datasets for Scenario 3. Dotted line represents the variable used to generate the time-to-malaria variable. The lower the rank the more relevant the variable is for the specific algorithm.

Scenario 4 (Figure 10.4) methods, except RFE-Guyon, detect the two relevant variables. However, RFE-maxgrowth-function identifies as relevant, with a pretty similar rank, variables 3 to 8 (highly correlated with the true relevant ones). In RFE-pseudo-samples algorithm it is observed that the lower the correlation with the true relevant variables, the larger the rank.

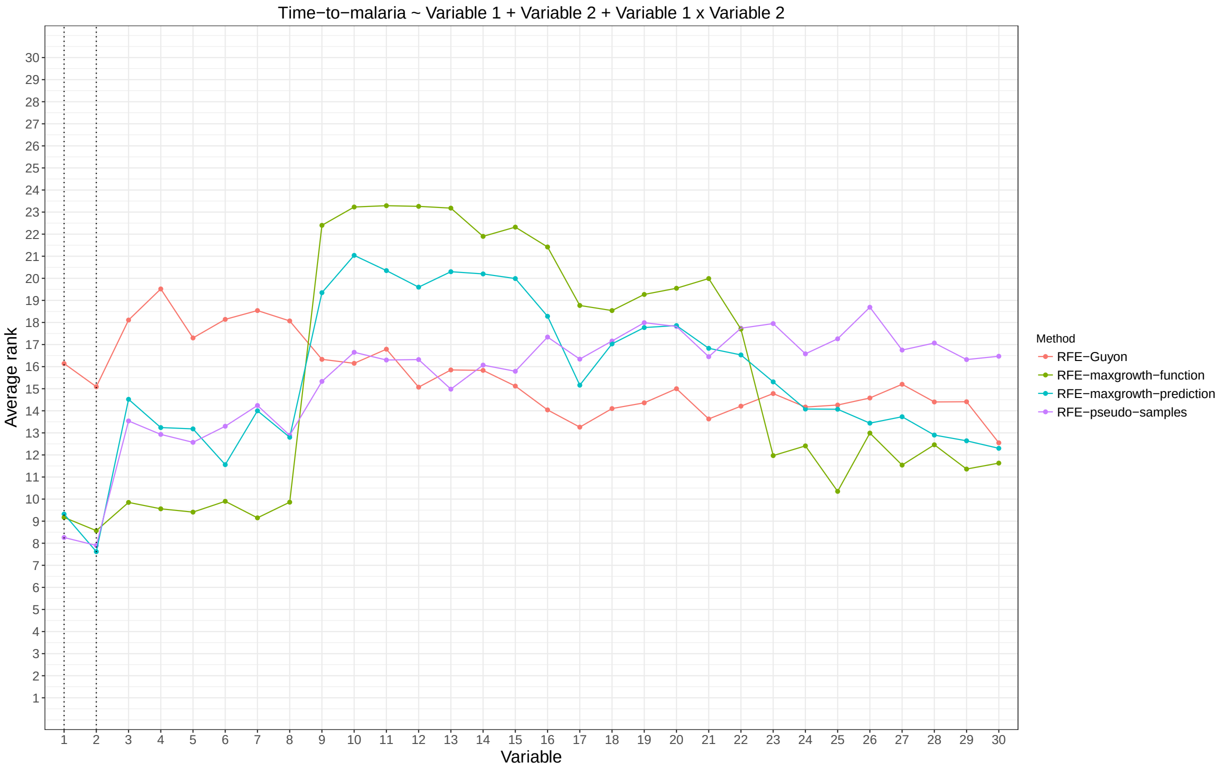


Figure 10.4: Average rank by variable and method for the 100 simulated datasets for Scenario 4. Dotted line represents the variable used to generate the time-to-malaria variable. The lower the rank the more relevant the variable is for the specific algorithm.

For Scenario 5 (Figure 10.5) three variables are relevant (1, 20 and 30). An interaction and a quadratic term are included. RFE-pseudo-samples is clearly the method that identifies best the relevant variables. The other three algorithms are not able to detect clearly the three variables. Although RFE-maxgrowth-function is able to identify as relevant, with a similar rank, variables 1 to 8.

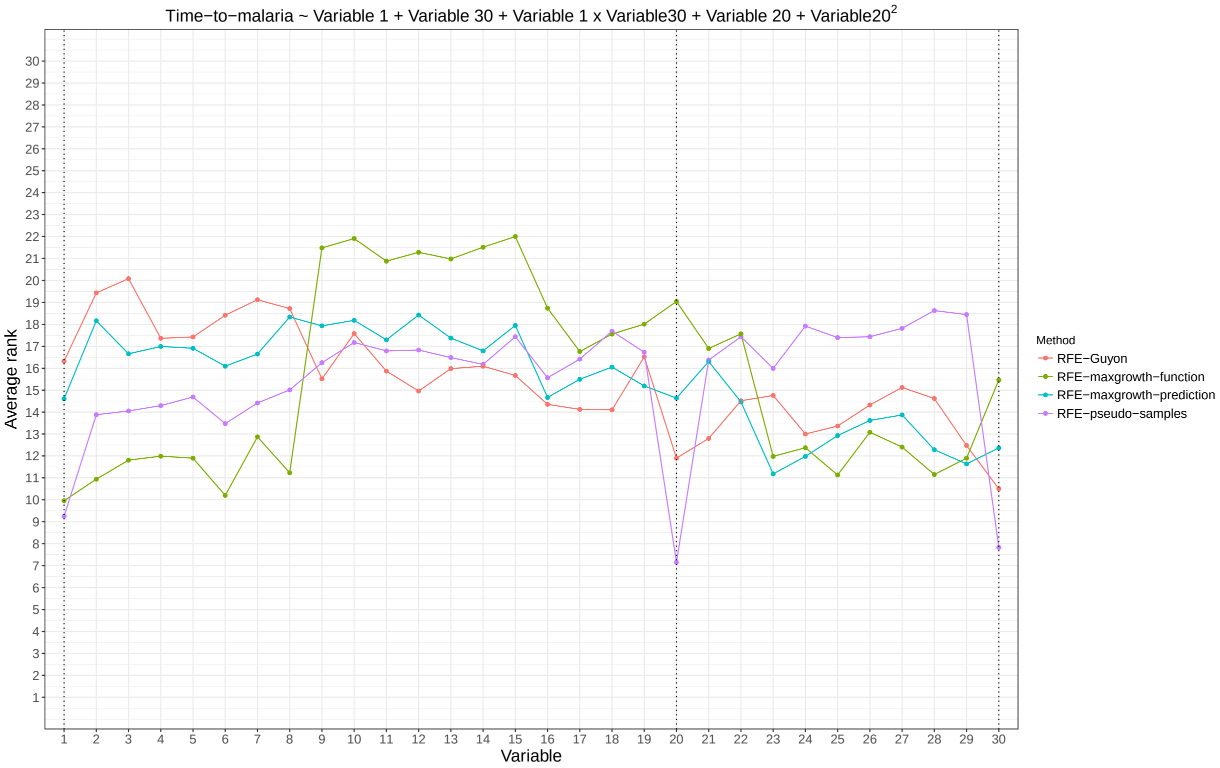


Figure 10.5: Average rank by variable and method for the 100 simulated datasets for Scenario 5. Dotted line represents the variable used to generate the time-to-malaria variable. The lower the rank the more relevant the variable is for the specific algorithm.

In Scenario 6 (Figure 10.6) Variable 1 and Variable 30 are relevant, being the former included as main effect with a quadratic term and the latter exponentiated. All methods, except RFE-maxgrowth-function, are able to detect the importance of Variable 30. With respect Variable 1, RFE-pseudo-samples, RFE-Maxgrowth-function present a similar rank around 10.5. The other two algorithms, RFE-Guyon and RFE-maxgrowth-prediction are not able to identify as relevant Variable 1 since the ranks obtained are similar to other non-relevant variables.

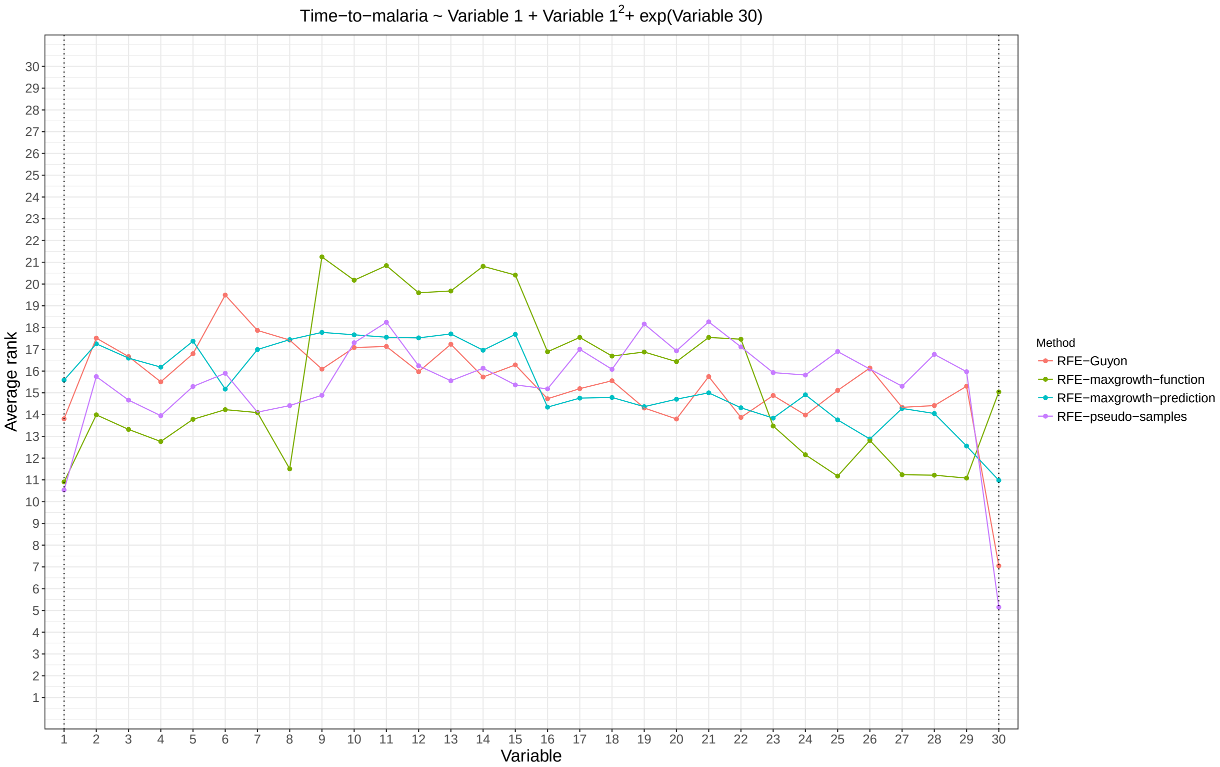


Figure 10.6: Average rank by variable and method for the 100 simulated datasets for Scenario 6. Dotted line represents the variable used to generate the time-to-malaria variable. The lower the rank the more relevant the variable is for the specific algorithm.

10.3.2 Visualization and interpretation of proposed methods

RFE-pseudo-samples

An example of the results for Scenario 2, the 100 simulated datasets and first iteration of the RFE algorithm is shown in Figure 10.7. There are two variables that present a completely different pattern compared to the others, which are Variable 29 and Variable 30. Besides that, the association with the response is opposite, since for Variable 30 the larger the pseudo-sample value the larger the decision value and for Variable 29, the larger the pseudo-sample the lower the decision value. The other variables are pretty constant along the pseudo-samples range. The other summary results for the pseudo-samples are shown in Appendix D.

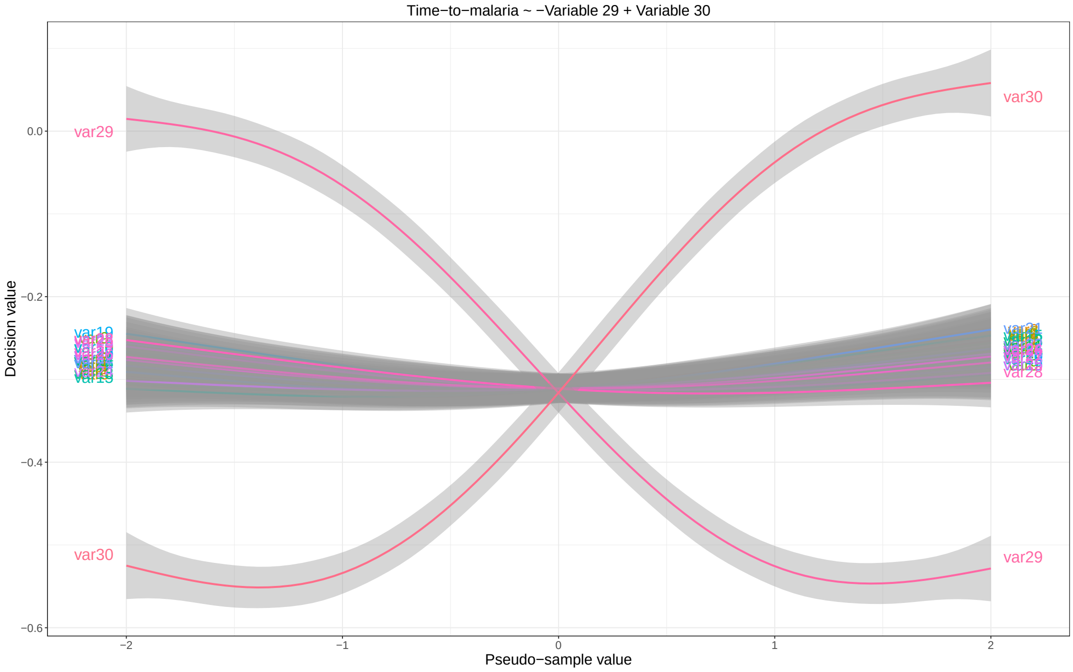


Figure 10.7: Scenario 2 results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval.

RFE-KPCA-maxgrowth prediction and function

In Figure 10.8 an example for RFE-maxgrowth-prediction approach, Scenario 1 and iteration 25 is shown. The 5 variables left, and more relevant are 1, 2, 25, 26 and 28. The first two are highly correlated in average and the others are independent by study design. The reference, in this case, is *prediction* approach, but is equivalent to *function* one, is the top-left figure. The first component (PC1) is the one that classifies the malaria group, most of malaria observations are negative and non-malaria positive. For the reference, the directions of points are going from non-malaria to malaria along the PC1 and PC2. With respect the other variables, only Variable 1 and Variable 2, present a pattern in terms of directions similar to the reference, while the others (25, 26 and 28) look pretty random. The interpretation of this is: variables 1, 2 and reference perform similarly, thus Variable 1 and Variable 2 are relevant, and the others are not. Besides that, since 25, 26 and 28 are random among them, they are not associated with the response and they are not correlated.

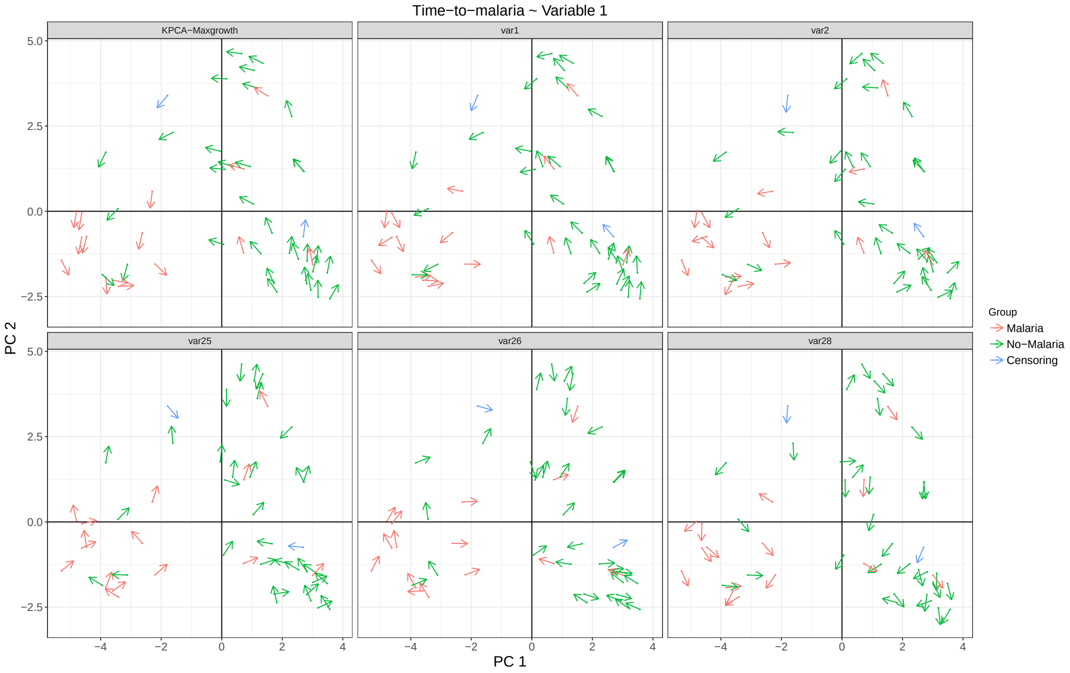


Figure 10.8: Scenario 1 results for a random simulated dataset and iteration 25 of the RFE-KPCA-maxgrowth-prediction approach. First component of the KPCA (PC1) and second component (PC2) are represented in the X-axis and Y-axis, respectively. Malaria events, non-malaria events (at the end of follow-up time) and censored observations (during follow-up) are represented.

Chapter 11

Discussion and Future Research

11.1 Discussion

In biomedical research is important to know which variables are associated to the studied outcome and the degree of this association. When dealing with non-linear kernels the feature space is different than the input one, thus extracting information from the variables in this space can not provide us information related to the original space. Although non-linear kernels, specially the Gaussian one, are widely used, there is no much work done comparing methods into the relevance variable issue in the non-linear SVM. Besides that, an important aspect is the interpretation and visualization of the association predictor-response, however, almost no work has been done related to this issue in the SVM context. Understanding, not only whether a variable is relevant or not, but the type of the association, is a key component in any medical research.

The three algorithms we propose perform generally better than the gold standard RFE-Guyon for non-linear kernel. As expected, results for all methods are better when the true relevant variables are independent, i.e., they are no correlated with the other variables in the SVM model. However, this scenario is not usual in biomedical data and it is specially unusual when dealing with a mid-to-large antigen panel.

Among the proposed methods, the RFE-pseudo-samples one performs overall much better than the others in all tested scenarios. Besides that, the algorithm allows interpreting and visualizing each model in a easy way being more informative than RFE-Guyon. One issue we have observed is that in the algorithm there are two main parts: one is the representation of the variables and the other is the metric used to define the relevance of the variable. This distinction is important because one visual conclusion can be different from the one obtained with the metric, therefore the method seems highly dependent on the metric used to measure the variability.

With regards the RFE-maxgroth approaches both prediction and function perform similarly, being the former slightly better. Actually, the *prediction* approach can be interpreted as an instance of the *function*, however, the former is much less time consuming. Contrary to what we expected, the RFE-maxgrowth-function does not perform as accurately as the other three compared methods; this approach is based on the explicit decision function, for this reason we expected much better results. One explanation could be that by approaching the decision function with a non-linear kernel as a combination of variables we are loosing more information than by using the RFE-maxgrowth-prediction.

In the other alternative we have tested, the RFE-maxgrowth-prediction algorithm, the prediction is included as an extra variable into the KPCA space. One thing to consider is that, when including this extra variable, we are constructing a space different to the one without including the prediction variable. This seems an advantage because we are taking into account, in some sense, the patterns that defines event and non-event into the KPCA. On the contrary, in the RFE-maxgrowth-function the KPCA space does not take any into account any specific variable directly related to the classes.

The RFE-maxgrowth interpretation is more complex but can be more informative than RFE-pseudo-samples specially for a specific subset of variables and observations. The reason being, we have to interpret the components of the KPCA, the directions of maximum growth of each input variable and compare its direction between the event and non-event observations; this can only be achieved with a reduced number of variables due to the amount of information in each generated figure.

The main disadvantage of the three proposed methods is that we are not including in a direct way the event and non-event classes, we are using, basically, the decision values (or the decision function).

One important aspect to consider is the computational cost of the proposed RFE-algorithms. The RFE-pseudo-samples method depends basically on the decision function and the pseudo-samples range, and as expected, the computational cost is not significant, being the fastest method. On the contrary the RFE-maxgrowth algorithms are more complex, specially the *function* version whose execution time depends on pairwise variable gradient.

11.2 Future research

A very important aspect for future research related to the RFE-pseudo-samples algorithm is the study of the values used for the pseudo-samples. In the present thesis we have used zero values for the variables different than the one tested. It could be interesting to evaluate how the method works using quartiles, centroids or variable-specific values, i.e., using a different value for each variable defining a specific observation such that the median event-observation, similarly to what is done in risk score equations (Wilson et al., 1998, Marrugat et al., 2003, Conroy et al., 2003). This could be useful for identifying variable interactions and other variables not identified using zero value.

Another point of interest is the one related to the metric used in the algorithm. In the present thesis we have used the MAD as a measure of the variability, but other metrics, project-specific could be used: maximum-minimum decision value difference, overall absolute value of the slope of the curve or more sophisticated like measuring the non-linearity of a variable trajectory as the maximal change of the angles of each pseudo-sample section with respect the horizontal axis as it is done in Postma et al. (2011).

Natural extensions of the RFE-KPCA-maxgrowth research might be divided in two main lines of work. On the one hand, we have not considered the selection of the principal components and we have chosen the first two. In KPCA-maxgrowth-prediction, as we are including the prediction as a variable, would be interesting to investigate how the method is improved when selecting components that represents the prediction variable by selecting the most discriminative for the event/non-event observations. On the other hand, the KPCA does not take into account the class of the observation, is an unsupervised method. As the main objective of our proposed algorithm is to select the most relevant variables associated with the response, other similar approaches to KPCA, but taking into account the class of the observations, could be implemented like kernel Fisher discriminant analysis (KFDA) (Mika et al., 1999), for instance. Thus, a similar approach to KPCA-maxgrowth but in KFDA context could be tested.

Part IV

Mal067 Correlates of Protection and Naturally Acquired Immunity

Chapter 12

Analysis of the Mal067 Study

The Mal067 data has been introduced in Section 1.3. This dataset consists in 459 participants from two African sites: Bagamoyo (Tanzania) and Manhiça (Mozambique). The main objective of the study is to identify which cytokines and chemokines are correlated with protection in each vaccination status group (RTS,S or comparator) within the participants that have received the third dose of the assigned vaccine.

The design of the study is presented in Section 12.1. Section 12.2 provides a description of the multiplex bead array assay and data pre-processing. The methods section is described in Section 12.3 and Section 12.3.1, the former describes the classical statistical methods applied to the Mal067 data and the latter the SVM for survival analysis methodology. The main results are shown in Section 12.4.

12.1 Study design and population

The African research centers participating in the study are: Ifakara Health Institute and Bagamoyo Research and Training Centre (IHI-BRTC in Tanzania) and Manhiça Health Research Center, Fundação Manhiça (FM-CISM, Mozambique). The two sites had low-medium malaria transmission intensity at the time of the study (RTS,S Clinical Trials Partnership, 2015, SCTP et al., 2012).

This study aimed to include all subjects from the pediatric Phase III clinical trial (Mal055, trial registered with ClinicalTrials.gov, number NCT00866619) who were eligible for the multicenter immunology ancillary study (Mal067), who met criteria for the modified according-to-protocol (ATP) cohort of the Phase III trial, from whom peripheral blood mononuclear cells (PBMC) was collected for cellular determinations, and from whom there was available supernatant from stimulated cells. Inclusion in the cellular component of the Mal067 immunology study in Manhiça targeted all children (5 to 17 months) and infants (6 to 12 weeks) from one of the recruiting peripheral health posts (Palmeira neighborhood) after ethical approvals for the Mal067 immunology study were obtained. In Bagamoyo, the first 400 children recruited after obtaining ethical approval had venous blood extracted for cellular determinations; infants were not included. After cryopreserving 5 x 10^6 PBMC in liquid nitrogen, the remaining cells were used for fresh antigen stimulations onsite and their culture supernatants were analyzed. Therefore, only vaccinees with enough PBMC for cryopreservation and additional ex vivo stimulations (at least 6.6 x 10^6 PBMC) were included in this study.

In the trial subjects were vaccinated with either RTS,S or a comparator vaccine, i.e., rabies vaccine (children) or meningococcal C conjugate vaccine (infants), administered at study month zero (M0), one and two. Approval for the study protocol was obtained from all relevant ethics review boards and national regulatory authorities.

12.2 Multiplex bead array assay and data pre-processing

Fifty μL of all supernatants were tested in single replicates and supernatants belonging to a same individual (baseline and M3 timepoints) were tested in the same plate. All assays were performed in a central lab (ISGlobal[1]) by three operators after completing training and passing proficiency testing. Plates were balanced across sites, age cohort, vaccine status and malaria-cases and malaria-controls. Plate design, standard curve dilutions and use of controls were optimized in a pre-pilot study to diminish variability and increase sensitivity. If performance of a plate did not meet the quality criteria established, samples were retested in another plate. Each plate included serial dilutions of a standard sample provided by the vendor with known concentration of each analyte, two blank controls and three additional controls (called positive controls) in duplicate with high, medium and low concentrations prepared from a reference sample for quality assurance/quality control (QA/QC) purposes. Samples were acquired on a Luminex® 100/200 instrument. To quantitate analyte concentrations and perform weekly QA/QC assessment, the `drLumi` R package was used (Sanz et al., 2015). The package fitted the standard curves based on five-parameter log-logistic models; if the algorithm did not converge, a four-parameter log-logistic model was fitted. Lower and higher limits of detection (LLOD and HLOD) were based on the upper (for LLOD) or lower (for HLOD) limit of the 95% confidence interval (CI) of the lower or upper asymptote of standard curves, respectively. Limits of quantification (LOQ) were estimated based on a cutoff value of the coefficient of variation (CV) of the standard curve for each analyte. LOD and LOQ were estimated for each plate.

Non-quantitated samples are below lower LOQ or above higher LOQ and, thus, cannot have MFI reliably mapped into concentration through a 5-parameter log-logistic standard curve because their MFI is located near or on the asymptotes of the curve. Those samples have generally very low concentration or very high concentration of the marker. Markers with less than 30% of non-quantitated had non-quantitated samples imputed and those with more than 30% of non-quantitated samples (e.g., the cytokines IL-6 and IL-8) were excluded from analysis. Non-quantitation was not associated with any comparison group of interest. Therefore, imputation aimed to allow inclusion of theses observations in continuous analysis by *jittering* the very low and high values through sampling for each subject a value from a uniform distribution with the following boundaries: i) for samples with MFI between LLOD and LLOQ, the concentration at

[1] http://www.isglobal.org/en/

LLOD and LLOQ were used as boundaries; ii) for samples with MFI below LLOD, a value of 0.01 or 0.1 and the LLOD were used as boundaries, depending on the value of the LLOD (if < 1 or not); for samples with MFI between HLOQ and HLOD, the HLOQ and the minimum of HLOD and the concentration corresponding to 25,000 MFI (maximum detectable MFI of the Luminex® reader) were used as boundaries.

12.3 Statistical analysis

Since vaccination with RTS,S could confound naturally acquired immunity, we have evaluated the impact of vaccination status on AMA1 specific marker levels by contrasting RTS,S and comparator vaccinees at M3 thought t-tests, Kaplan-Meier survival curves for RTS,S and comparator group with the corresponding 95% confidence bands have been estimated. For comparing the survival curves the log-rank test is calculated. Spearman correlations between all cytokines stratified by vaccination group have been computed. We also comparatively describe the distribution of each marker AMA1 specific response in subjects who did and did not have the event one year post start of follow-up (M3) thought t-tests.

To assess whether the AMA1 specific marker level is correlated with an effective immune response, i.e., protection from malaria, we have estimated hazard of clinical malaria as a function of the marker level in Cox regression models. Hazards ratios with the 95% confidence interval are estimated for each of the 28 individual markers separately in Cox models with the baseline hazards stratified by vaccination status. Tests for the assumption of proportional hazards are carried out based on Schoenfeld residuals for each individual marker. Figures of the estimated coefficients (Y-axis) as a function of time-to-event (X-axis) are also drawn to asses visually the proportionality of hazards.

12.3.1 Support vector machines for survival data

All methods described in Chapter 7 are compared into the Mal067 dataset applying a 5-fold nested cross-validation. Specifically, for each kernel-based method the grid search for each tuning parameter is the same as it is in the simulation Section 7.1.5:

- Kernel Cox regression: the values used for tuning the Gaussian kernel are 0.25, 0.5, 1, 2 and 4.
- Weighted support vector machines: two approaches are used, one defining the weights of the censored observations as proportional to the follow-up time, and another one using conditional survival. The values used for tune the parameter C are 0.1, 1, 10 and 100, and for the kernel 0.25, 0.5, 1, 2 and 4.
- Support vector machines with uncertain classes: related to the kernel two approaches are tested, linear and Gaussian kernels; related to the probabilities of the censored observations both proportional and conditional survival based methods are considered. For the Gaussian

kernel the potential values tested are 0.25, 0.5, 1, 2 and 4. The C and $\widetilde{C}$ possible values are 0.1, 1, 10 and 100. The η value has been fixed to be 0.0001.

- Support vector machines learning using privileged information: Gaussian kernel and linear kernel are considered for the decision and correcting space, respectively. For the latter proportionality of follow-up time and conditional survival are considered. For the Gaussian kernel the values used in the grid search are 0.25, 0.5, 1, 2 and 4. The cost values of misclassification parameter C are 0.1, 1, 10 and 100. The weight used for the correcting space has been considered to be 0.1, 1, 10 and 100.
- Semi-supervised support vector machines using invariances: the parameters tested have been 0.25, 0.5, 1, 2 and 4 for both Gaussian kernel and Gaussian density, depending on the local invariance. The parameter values considered for ρ_2 are 0.1, 1, 10 and 100 for both gradient and averaging alternative approaches.

12.3.2 Tuning parameters and performance of the methods

The tuning parameters for each SVM method and kernel Cox regression are calculated following a 5-fold nested cross-validation approach. As we want to tune the parameters of the model and evaluate its performance a conventional k-fold cross-validation would overestimate the performance. The reason being, the test set for tuning parameters is the same than the one used to test parameters (Stone, 1974, Varma and Simon, 2006). The specific, nested cross-validation algorithm used, is shown in Algorithm 4.

Friedman et al. (2001) stated that, overall five or ten-fold cross-validation are recommended as a good compromise between variance and bias. The function that we have considered to select the tuning parameters and that defines the cross-validation performance, is the accuracy defined in the same way as in equation (7.7). Besides that, to complement, the same metrics established in Section 7.3 are considered: normalized mutual information, area under the ROC curve and Matthews correlation.

12.3.3 Relevance of variables approach

As the RFE-pseudo-samples approach is the one that performs best from the 4 compared (Section 11.1) is the one applied to the Mal067 data. The RFE-algorithm is carried out with the best SVM model obtained in the nested cross-validation approach. Fifty cutoff points of the range of the data are used and the range of the pseudo-samples is described to evaluate which is the interval of pseudo-samples in which most of the antigens lay in. One hundred bootstrap datasets are generated and the RFE-pseudo-samples algorithm is applied to all of them. Therefore, the final ranks are based on the average of the 100 relevant ranks.

```
Data   : Dataset with $p$ variables, time-to-event and status.
Input  : $K$-folds.
Output: Summary statistics in terms of accuracy.

Randomly assign each observation in one of $K$ groups;
$R \leftarrow$ combinations of parameters vector to be tested in the grid search;
for $i = 1$ to $K$ do
    $Test.Set \leftarrow$ Observations in Data $\in$ partition $i$ ;
    $Training.Set \leftarrow$ Observations in Data $\notin$ partition $i$;
    $K^* \leftarrow$ Partitions vector different than $i$ (1 to $[K-1]$);
    for $j = 1$ to $(K-1)$ do
        $Test.Set^*_j \leftarrow$ Observations in $Training.Set \in$ partition $K^*_j$ ;
        $Training.Set^*_j \leftarrow$ Observations in $Training.Set \notin$ partition $K^*_j$;
        for $r = 1$ to $R$ do
            $SVM.inner_r \leftarrow$ SVM using $Training.Set^*_j$ with combination of parameters $r$;
            $accuracy_{jr} \leftarrow$ accuracy of $SVM.inner_r$ into $Test.Set^*_j$;
        end
    end
    $accuracy.r \leftarrow$ average $accuracy_{jr}$ for each combination of parameters $r$;
    $best.parameters \leftarrow$ combination of parameters $r$ with highest accuracy;
    $SVM.outer_i \leftarrow$ SVM using $Training.Set$ and parameters vector $best.parameters$;
    $Test.outer_i \leftarrow$ Accuracy of $SVM.outer_i$ into $Test.Set$;
end
$Method.performance \leftarrow$ mean of $(Test.outer_1, \ldots, Test.outer_K)$;
return $Method.performance$
```

Algorithm 4: Pseudo-code for k-fold nested cross-validation methodology given one specific SVM method.

12.4 Results

In this section main results related to the Mal067 data are presented for both RTS,S and comparator vaccination status.

12.4.1 Data descriptives

The distribution of the main characteristics of the Mal067 participants is shown in Table 12.1. A total of 459 subjects were analyzed, 449 of them completed the one-year follow-up, and 59 of them had a clinical malaria event. The proportion of participants lost-to-follow-up was around 2% (5 participants) in the RTS,S cohort and around 3% (5 participants) in comparator cohort.

Table 12.1: Characteristics of participants at post-vaccination (M3) by vaccination status.

		Comparator (n=154)	**RTS,S (n=305)**	**p value**
Site, n(%)	Bagamoyo	78 (50.6%)	169 (55.4%)	0.386
	Manhiça	76 (49.4%)	136 (44.6%)	
Age cohort, n(%)	5-17 months	101 (65.6%)	190 (62.3%)	0.556
	6-12 weeks	53 (34.4%)	115 (37.7%)	
Sex, n(%)	Female	89 (57.8%)	152 (49.8%)	0.130
	Male	65 (42.2%)	153 (50.2%)	
Malaria event, n(%)	No	130 (84.4%)	270 (88.5%)	0.274
	Yes	24 (15.6%)	35 (11.5%)	
Time to follow-up, Med [Q1;Q3]		365 [365;365]	365 [365;365]	0.043

The distribution of each cytokine and chemokine $\log_{10}$(AMA1/DMSO) ratio by malaria case and vaccination group is shown in Table 12.2. For the RTS,S cohort the statistically significant cytokines, at a 5% α level, are G-CSF, IFN-α, IL-10 and MCP-1. For the comparator cohort there are no significant cytokines. Related to the Cox models, for the RTS,S cohort the significant cytokines are the same as in the t-test analysis being all of them a protective effect factor.

The pairwise correlation coefficients (Spearman) between all analytes is shown in Figure 12.1 and Figure 12.2 for RTS,S and comparator vaccinees respectively. Both groups have similar correlations pattern and there are no remarkable negative association, i.e., all Spearman correlations are greater or equal than 0. For both RTS,S and comparator vaccines, G-CSF cytokine is highly correlated (more than 0.7) with IL-10 and IL-1ra.

Table 12.2: Comparison of cytokine AMA1/DMSO ratio (log10 transformed) by malaria and non malaria events stratified by vaccination status. Hazard ratio of the bivariate Cox model and the corresponding 95% confidence interval is shown.

	RTS,S				Comparator			
Analyte	**No malaria**	**Malaria**	**p value**	**HR (95% CI)**	**No malaria**	**Malaria**	**p value**	**HR (95% CI)**
EGF	0.05 (0.15)	0.06 (0.10)	0.915	1.13 (0.11;11.5)	0.02 (0.20)	0.07 (0.15)	0.205	4.32 (0.45;41.6)
Eotaxin	0.03 (0.09)	0.03 (0.16)	0.800	1.60 (0.04;61.2)	0.02 (0.17)	0.02 (0.07)	0.987	1.02 (0.08;12.3)
FGF	0.03 (0.23)	0.03 (0.16)	0.985	1.01 (0.23;4.50)	0.03 (0.26)	0.09 (0.20)	0.249	2.18 (0.58;8.22)
G-CSF	0.02 (0.10)	-0.03 (0.17)	0.017	0.07 (0.01;0.62)	0.01 (0.19)	0.02 (0.06)	0.915	1.13 (0.11;11.2)
GM-CSF	0.04 (0.15)	0.01 (0.24)	0.295	0.43 (0.09;2.10)	0.04 (0.19)	0.06 (0.06)	0.561	1.81 (0.25;13.2)
HGF	0.02 (0.13)	0.00 (0.11)	0.482	0.40 (0.03;5.22)	-0.01 (0.26)	0.06 (0.19)	0.160	2.76 (0.67;11.4)
IFN-α	0.02 (0.05)	-0.02 (0.10)	0.001	0.00 (0.00;0.06)	0.01 (0.15)	-0.02 (0.15)	0.501	0.48 (0.06;4.10)
IFN-γ	0.00 (0.20)	0.01 (0.27)	0.978	1.02 (0.20;5.11)	0.02 (0.25)	0.01 (0.15)	0.952	0.95 (0.17;5.30)
IL-10	0.05 (0.13)	0.00 (0.24)	0.041	0.19 (0.04;0.93)	0.05 (0.21)	0.06 (0.08)	0.872	1.19 (0.15;9.41)
IL-12	0.02 (0.12)	-0.02 (0.24)	0.080	0.25 (0.05;1.18)	0.04 (0.20)	0.02 (0.08)	0.688	0.64 (0.07;5.64)
IL-13	0.01 (0.06)	0.00 (0.09)	0.267	0.08 (0.00;6.89)	0.01 (0.14)	0.02 (0.05)	0.694	1.73 (0.11;26.8)
IL-15	0.04 (0.16)	0.00 (0.17)	0.083	0.27 (0.06;1.19)	0.06 (0.21)	0.05 (0.12)	0.825	0.79 (0.09;6.65)
IL-17	0.03 (0.08)	0.04 (0.09)	0.646	2.46 (0.05;115)	0.01 (0.17)	0.02 (0.08)	0.596	2.23 (0.12;42.7)
IL-1β	0.01 (0.08)	0.00 (0.15)	0.250	0.17 (0.01;3.47)	0.01 (0.18)	0.03 (0.06)	0.589	1.78 (0.22;14.6)
IL-1ra	0.03 (0.11)	0.01 (0.16)	0.256	0.21 (0.01;3.14)	0.04 (0.20)	0.07 (0.23)	0.453	1.92 (0.35;10.5)
IL-2	0.06 (0.18)	0.00 (0.24)	0.068	0.22 (0.04;1.12)	0.05 (0.21)	0.07 (0.12)	0.729	1.40 (0.21;9.35)
IL-2r	0.02 (0.05)	0.01 (0.07)	0.368	0.07 (0.00;22.4)	0.01 (0.18)	0.03 (0.06)	0.547	1.88 (0.24;14.7)
IL-4	0.01 (0.05)	0.00 (0.07)	0.387	0.06 (0.00;32.2)	0.01 (0.17)	0.01 (0.03)	0.969	1.05 (0.07;15.2)
IL-5	0.00 (0.09)	0.01 (0.11)	0.618	2.59 (0.06;109)	0.01 (0.14)	0.03 (0.07)	0.327	5.77 (0.17;193)
IL-7	0.00 (0.07)	0.02 (0.08)	0.241	16.3 (0.15;1741)	0.00 (0.17)	0.03 (0.14)	0.446	2.64 (0.22;31.8)
IP-10	0.01 (0.05)	0.01 (0.05)	0.804	0.45 (0.00;239)	0.01 (0.16)	0.02 (0.05)	0.657	1.75 (0.15;20.6)
MCP-1	0.01 (0.13)	-0.07 (0.27)	0.001	0.10 (0.02;0.40)	0.00 (0.21)	0.00 (0.13)	0.949	1.07 (0.13;8.48)
MIG	0.02 (0.14)	0.03 (0.11)	0.696	1.61 (0.15;17.4)	0.01 (0.19)	0.02 (0.08)	0.728	1.50 (0.16;14.4)
MIP-1α	-0.01 (0.14)	-0.01 (0.22)	0.936	0.91 (0.10;8.60)	0.02 (0.23)	-0.01 (0.12)	0.521	0.53 (0.07;3.73)
MIP-1β	0.03 (0.34)	0.00 (0.49)	0.618	0.79 (0.30;2.03)	0.05 (0.45)	-0.06 (0.35)	0.243	0.56 (0.21;1.48)
RANTES	0.04 (0.14)	0.07 (0.16)	0.171	2.98 (0.62;14.3)	0.04 (0.20)	0.03 (0.11)	0.930	0.91 (0.10;7.94)
TNF	0.70 (0.41)	0.82 (0.41)	0.121	1.84 (0.85;3.97)	0.80 (0.49)	0.69 (0.48)	0.258	0.63 (0.29;1.40)
VEGF	0.02 (0.06)	0.01 (0.09)	0.303	0.10 (0.00;8.08)	0.02 (0.17)	0.03 (0.07)	0.797	1.37 (0.13;15.0)

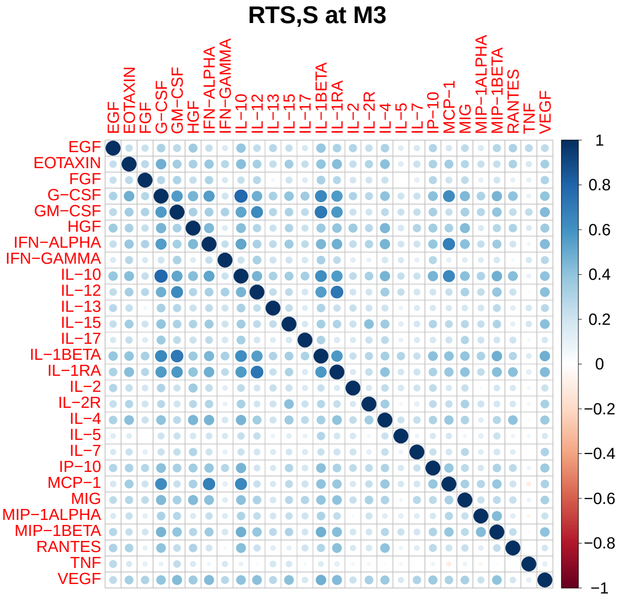


Figure 12.1: Spearman correlation between all cytokines for RTS,S vaccination status group at M3 timepoint.

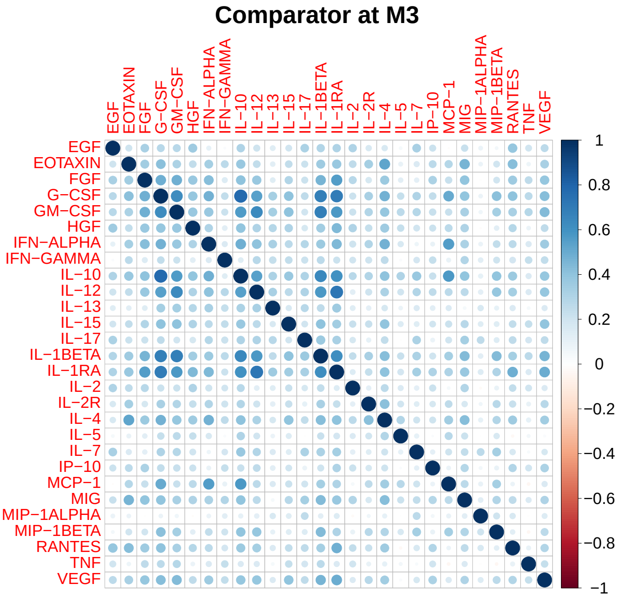


Figure 12.2: Spearman correlation between all cytokines for comparator vaccination status group at M3 timepoint.

The Kaplan-Meier survival curves for each vaccination group are shown in Figure 12.3. Survival probability at 12 months follow-up is higher for RTS,S cohort compared to the comparator one, although this difference is not statistically significant according to the log-rank test (p value = 0.167).

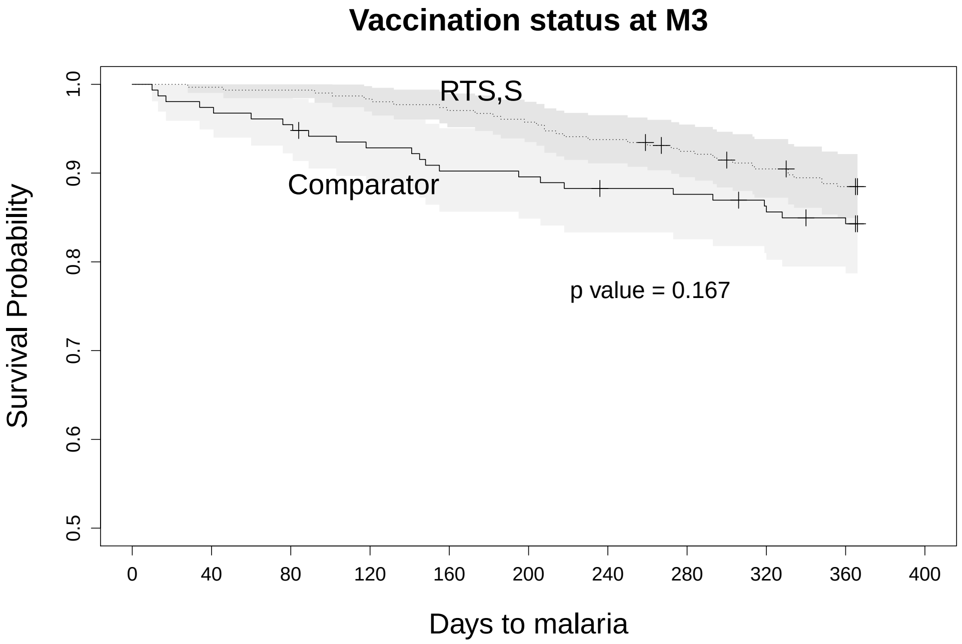


Figure 12.3: Kaplan-Meier survival curves by vaccination status with the 95% confidence intervals. The p value of the log-rank test comparing both survival curves is shown. Censored participants are represented as +.

We assessed the assumption of proportionality of hazards over time for each analyte through Schoenfeld residuals. Two cytokines are statistically significant in the RTS,S cohort: TNF and IFN-γ with p value lower than 0.001 and 0.023 respectively (Appendix Table E.1), suggesting a non-proportionality for these two analytes. Visual inspection of Schoenfeld residuals (Appendix Figure E.1), suggests that the non-proportionality is confirmed for TNF, although is not a remarkable non-proportionality. For IFN-γ, the plot does not indicate a clear non-proportionality. In the comparator cohort no markers are statistically significant (Appendix Table E.1) and after visual inspection (Appendix Figure E.2) results are confirmed.

12.4.2 Model performance

Table 12.3 shows the results after applying the 5-fold nested cross-validation by each one of the two vaccination groups. Metrics based on normalized mutual information and Matthews correlation are not reproducible in repetitions of cross-validation possibly because the number of cases in each cross-validation fold was limited.

Table 12.3: Summary results of the 5-fold nested cross-validation by method and vaccination group. Average accuracy and area under the ROC curve (C-statistic) is shown.

	RTS,S		**Comparator**	
Method	**Accuracy**	**AUC**	**Accuracy**	**AUC**
Cox model	0.85	0.63	0.69	0.58
Kernel Cox	0.73	0.60	0.48	0.59
wSVM-KM	0.88	0.63	0.84	0.69
wSVM-Prop	0.86	0.58	0.84	0.65
pSVM-linear-KM	0.88	0.68	0.83	0.70
pSVM-linear-prop	0.88	0.63	0.83	0.70
pSVM-radial-KM	0.88	0.64	0.84	0.68
pSVM-radial-prop	0.86	0.60	0.82	0.63
LUPI-linear-KM	0.88	0.66	0.84	0.71
LUPI-linear-prop	0.85	0.64	0.82	0.70
inSVM-gradient	0.88	0.64	0.82	0.74
inSVM-averaging	0.88	0.64	0.83	0.72

The performance of the Kaplan-Meier approaches, compared to the proportional time within the same method are better, or at least equal. This is not the case specially for the AUC metric for both cohorts. Cox model performs better than kernel Cox regression for both RTS,S and comparator cohort. The best performance for RTS,S cohort is pSVM-linear-KM with an accuracy of 0.88 and AUC of 0.68. The lowest accuracy is observed in kernel Cox with 0.73 but the lowest AUC is found in wSVM-proportional approach (0.58). With respect comparator vaccinees, the best performance is observed in LUPI-linear-KM with an accuracy of 0.84 and AUC of 0.71. The worst accuracy is found in kernel Cox (0.48) and worst AUC in Cox model (0.58). The accuracy results, overall for all methods, are larger in RTS,S cohort compared to the comparator one, but the AUC results are larger in comparator vaccinees, except for Cox and kernel Cox regression.

12.4.3 Relevant cytokines, chemokines and growth factors

To perform the relevant antigens analysis the pSVM-radial-KM method have been used since it is one of the best methods and is based on a non-linear kernel. The RFE-pseudo-samples summary of all 100 bootstrap samples for the first RFE iteration (the model with all markers) is shown in Figure 12.4 for the RTS,S cohort and Figure 12.5 for comparator vaccination group. For RTS,S cohort the most relevant analytes are: FGF, RANTES, IL-5 and TNF, being all of them risk factor for malaria. In comparator cohort, the most relevant markers are: IL-15 and FGF being the former protector and the latter risk factor. The variability of results, shown by the width of the confidence bands, is higher in comparator group.

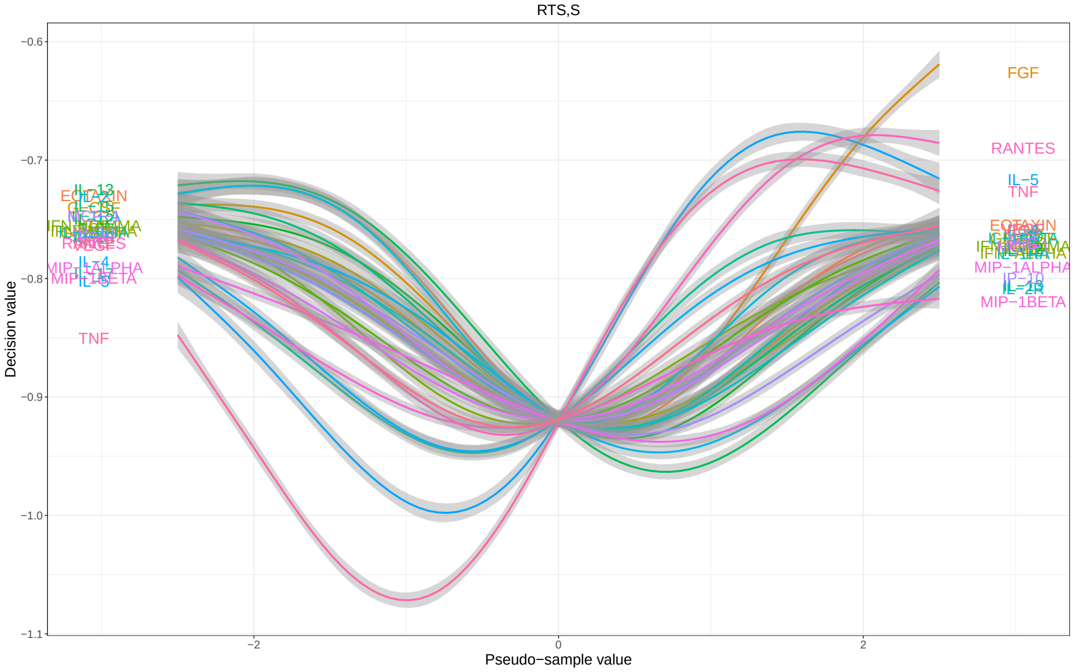


Figure 12.4: Summary of the first iteration for 100 bootstrap samples for the RTS,S cohort. RFE-pseudo-samples approach is applied. Smoothness is created using non-parametric local regression (LOESS) with the corresponding 95% confidence interval.

In Figure 12.6 relevant ranks for both RTS,S and comparator are shown summarizing the 100 bootstrap samples after applying the RFE algorithm. For RTS,S cohort the 5 most relevant analytes are: RANTES, IL-12, G-CSF, Eotaxin and EGF with an average rank of 6.96, 8.13, 9.49, 10.33 and 11.06, respectively. For comparator cohort the 5 most relevant antigens are: IL-15, IP-10, IL-2, MIP-1α and HGF with an average relevance rank of 9.72, 9.76, 9.93, 10.43 and 10.73. In appendix Table E.2 the specific mean and standard deviation summarizing the rank of the 100 bootstrap samples is shown.

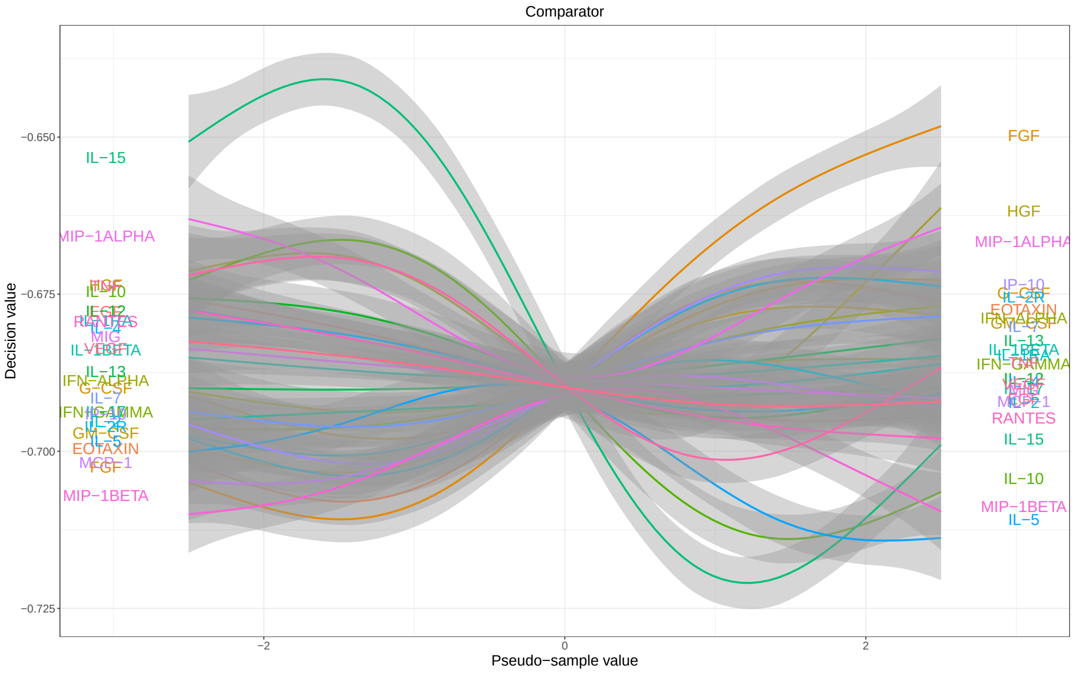


Figure 12.5: Summary of the first iteration for 100 bootstrap samples for the comparator cohort. RFE-pseudo-samples approach is applied. Smoothness is created using non-parametric local regression (LOESS) with the corresponding 95% confidence interval.

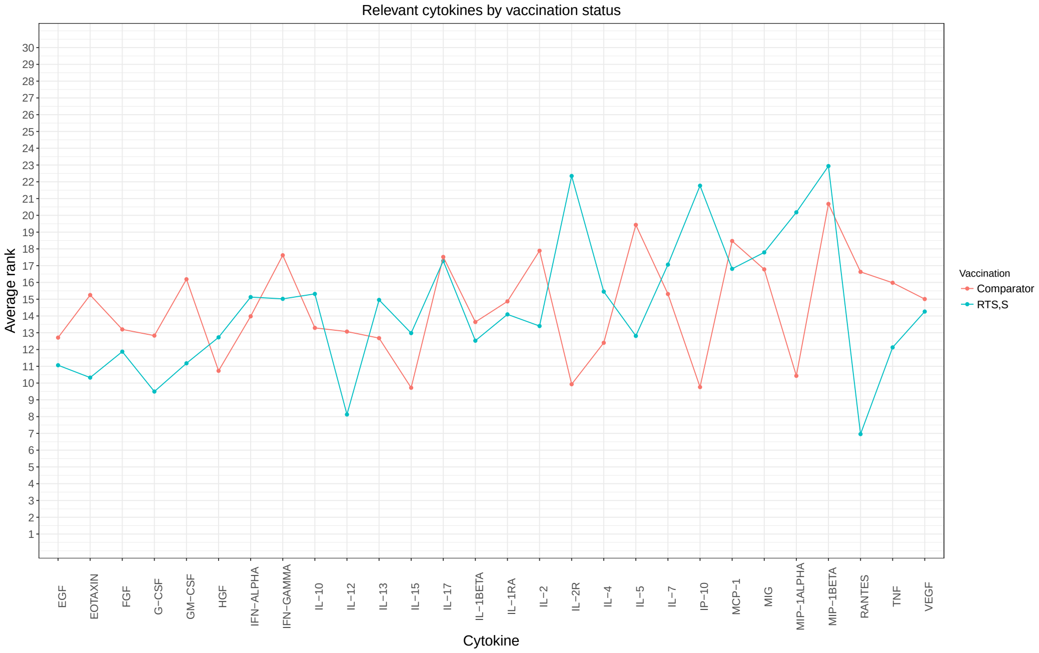


Figure 12.6: Comparison of the relevant antigens for both RTS,S and comparator cohort applying the RFE-pseudo-samples algorithm. Average relevant rank is measured as the average of 100 bootstrap samples. The lower the rank the most relevant the cytokine is for the specific vaccination group.

Chapter 13

Discussion and Future Research

13.1 Discussion

This part of the thesis has explored the relevant cytokines, chemokines and growth factors associated with the RTS,S and comparator vaccines for the Mal067 subjects. It is accepted that the antigens associated with the naturally acquired immunity are IFN-γ, TNF and IL-10.

The main characteristic of the Mal067 data, both RTS,S and comparator groups, is the proportion of events and censoring, i.e., there is approximately a 2% of censored data during the follow-up. In terms of SVM methods it has been discussed how the classes unbalancing can affect results. However, is not clear which cutoff is the one that defines a balanced data from an unbalanced one and in survival is more complicated because we have censored data. Another issue is the one related with proportion of censoring along the follow-up time. All proposed survival-SVM methods are based on SVM for binary classification, thus tend to be equivalent to the classical SVM in no censoring situations. For that reason, the proposed methods perform pretty similar, although the Kaplan-Meier approach performs better both in accuracy and discrimination metrics. The Cox model in RTS,S cohort performs similarly to SVM methods, probably due to the fact that there is more than one event per variable (must be noted that a nested cross-validation have been applied). In comparator cohort the performance is much lower due to the fact that there are fewer events per variable. Kernel Cox performance is similar to the Cox one in terms of AUC but not in terms of accuracy. One explanation to this is that the kernel Cox score discriminates better the relative risk between two participants but the estimation of absolute risk is worst.

Regarding the RFE-pseudo-samples approach an interesting issue is related to the variability of the bootstrap samples. Although, comparator sample size is much lower than RTS,S the standard deviation of the bootstrap samples and the bands in the first iteration figure it's an information to complement how accurate an reliable results are.

With regards, the specific relevant analytes from three accepted antigens associated to naturally acquired immunity, IFN-γ, TNF and IL-10, only IL-10 is significant for the RTS,S cohort applying a univariate test. When analyzing the first iteration of the RFE-pseudo-samples algorithm, i.e., the model with all antigens, TNF appears as relevant in the RTS,S cohort and IL-10 could be also identified in the comparator cohort. This relevance is maintained when performing the RFE algorithm specially for the RTS,S cohort: TNF ranks position 7 out 28; for the comparator cohort IL-10 ranks 12. As mentioned in Chapter 8.2 one aspect to consider is the non-proportionality due to a time-varying coefficient. With respect TNF and as explained in Section 12.4, we have observed non-proportionality in the RTS,S cohort, we haven't addressed this specific non-proportionality but perhaps this aspect could influence observed results.

13.2 Future research

One point of interest for future research is to compare the presented results in this thesis with other methods such as partial Cox regression (Li and Gui, 2004) to check whether results are consistent or not. One important issue to analyze is a specific subgroup analysis by specific age cohort and site to discard confounding effects in our observed results. Specially important, is the analysis by age cohort since the naturally acquired immunity it is highly dependent on age. Following the same rationale, analysing data at baseline, prior receiving the first dose of the vaccine would be of interest to measure baseline levels of AMA1 by cytokine and compare them to the M3 timepoint and due to the fact that differences between age cohort are larger.

Part V

Conclusions

Chapter 14

Conclusions

This thesis has focused on SVM methods for analysing survival data with censored observations. A part of the document has considered algorithms for visualizing and ranking variables in the SVM framework according to their relevance. The third considered aspect has been modelling and exploring the relevance of a 30-panel cytokines and chemokines associated with a correlate of protection induced by the RTS,S vaccine. The research has explored three main aspects:

1. The study of methods for dealing with time-to-event data into the SVM for binary classification framework.
2. The study of algorithms for visualizing and ranking the relevance of variables into the SVM for survival analysis framework.
3. The classification of relevant cytokines and chemokines associated with a correlate of protection induced by the RTS,S vaccine.

With regards to the first issue:

- It has been shown that:
  - The proposed conditional survival approach improves the prediction performance compared to the proportionality of time method.
  - The LUPI model for survival analysis performs similar to the wSVM.
  - The proposed inSVM model performs similar to the best survival-SVM method and Cox regression in proportionality of hazards scenarios and around 4 events per variable.
  - The performance of the proposed inSVM method is, from all studied methods, the most robust to modifications of censoring (proportion and distribution) and violation of proportionality of hazards assumption.
  - The proposed wSVM performance, independently of the approach used for estimating the weights, is influenced by the censoring proportion and proportionality of hazards assumption violation.

- It has been developed R code that:
  - Implements LUPI approach, allowing any kernel for decision and correcting space.
  - Implements pSVM method.
  - Implements wSVM approach.
  - Implements inSVM method for both gradient and averaging local invariances. It implements both alternatives in terms of observations used in the estimation (only unknown classes or all observations).

With regards to the second issue:

- It has been proposed a RFE-pseudo-samples algorithm that:
  - It allows to visualize and interpret the association and relevance of each variable with the time-to-event response variable.
  - It ranks the variables according to their relevance.
- It has been proposed a RFE-KPCA-maxgrowth algorithm, with two alternative approaches, that:
  - It allows to compare the direction of maximum growth of an SVM decision function and compare it with all predictor variables.
  - It ranks the variables according to their relevance.
- It has been shown that:
  - The RFE-pseudo-samples approach improve the ranking of the Guyon-RFE for non-linear kernels and RFE-KPCA-maxgrowth algorithm.
  - The KPCA-maxgrowth-prediction performs better than KPCA-maxgrowth function and similar to Guyon-RFE for non-linear kernels.

With regards to the third issue:

- The study of the methods have revealed that the proposed SVM for survival analysis approaches perform better than the proportional hazards model and kernel Cox regression for both RTS,S and comparator cohorts.
- The study of the relevant cytokines and chemokines has shown that:
  - For RTS,S cohort the five most relevant cytokines are: RANTES, IL-12, G-CSF, Eotaxin and EGF.
  - For comparator cohort the five most relevant antigens are: IL-15, IP-10, IL-2r, MIP-1α and HGF.

Bibliography

Agnandji, S. T., Lell, B., Soulanoudjingar, S. S., Fernandes, J. F., Abossolo, B. P., Conzelmann, C., Methogo, B. G., Doucka, Y., Flamen, A., Mordmuller, B., Issifou, S., Kremsner, P. G., Sacarlal, J., Aide, P., Lanaspa, M., Aponte, J. J., Nhamuave, A., Quelhas, D., Bassat, Q., Mandjate, S., Macete, E., Alonso, P., Abdulla, S., Salim, N., Juma, O., Shomari, M., Shubis, K., Machera, F., Hamad, A. S., Minja, R., Mtoro, A., Sykes, A., Ahmed, S., Urassa, A. M., Ali, A. M., Mwangoka, G., Tanner, M., Tinto, H., D'Alessandro, U., Sorgho, H., Valea, I., Tahita, M. C., Kabore, W., Ouedraogo, S., Sandrine, Y., Guiguemde, R. T., Ouedraogo, J. B., Hamel, M. J., Kariuki, S., Odero, C., Oneko, M., Otieno, K., Awino, N., Omoto, J., Williamson, J., Muturi-Kioi, V., Laserson, K. F., Slutsker, L., Otieno, W., Otieno, L., Nekoye, O., Gondi, S., Otieno, A., Ogutu, B., Wasuna, R., Owira, V., Jones, D., Onyango, A. A., Njuguna, P., Chilengi, R., Akoo, P., Kerubo, C., Gitaka, J., Maingi, C., Lang, T., Olotu, A., Tsofa, B., Bejon, P., Peshu, N., Marsh, K., Owusu-Agyei, S., Asante, K. P., Osei-Kwakye, K., Boahen, O., Ayamba, S., Kayan, K., Owusu-Ofori, R., Dosoo, D., Asante, I., Adjei, G., Adjei, G., Chandramohan, D., Greenwood, B., Lusingu, J., Gesase, S., Malabeja, A., Abdul, O., Kilavo, H., Mahende, C., Liheluka, E., Lemnge, M., Theander, T., Drakeley, C., Ansong, D., Agbenyega, T., Adjei, S., Boateng, H. O., Rettig, T., Bawa, J., Sylverken, J., Sambian, D., Agyekum, A., Owusu, L., Martinson, F., Hoffman, I., Mvalo, T., Kamthunzi, P., Nkomo, R., Msika, A., Jumbe, A., Chome, N., Nyakuipa, D., Chintedza, J., Ballou, W. R., Bruls, M., Cohen, J., Guerra, Y., Jongert, E., Lapierre, D., Leach, A., Lievens, M., Ofori-Anyinam, O., Vekemans, J., Carter, T., Leboulleux, D., Loucq, C., Radford, A., Savarese, B., Schellenberg, D., Sillman, M., and Vansadia, P. (2011). First results of phase 3 trial of RTS,S/AS01 malaria vaccine in African children. *N. Engl. J. Med.*, 365(20):1863–1875.

Akbani, R., Kwek, S., and Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. In *Proceedings of the 15th European Conference on Machine Learning (ECML*, pages 39–50.

Alonso-Atienza, F., Rojo-Álvarez, J. L., Rosado-Muñoz, A., Vinagre, J. J., García-Alberola, A., and Camps-Valls, G. (2012). Feature selection using support vector machines and bootstrap methods for ventricular fibrillation detection. *Expert Systems with Applications*, 39(2):1956–1967.

Aytug, H. (2015). Feature selection for support vector machines using generalized Benders decomposition. *European Journal of Operational Research*, 244(1):210–218.

Becker, N., Toedt, G., Lichter, P., and Benner, A. (2011). Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. *BMC bioinformatics*, 12(1):1.

Becker, N., Werft, W., Toedt, G., Lichter, P., and Benner, A. (2009). penalizedSVM: a R-package for feature selection SVM classification. *Bioinformatics*, 25(13):1711–1712.

Bejon, P., White, M. T., Olotu, A., Bojang, K., Lusingu, J. P., Salim, N., Otsyula, N. N., Agnandji, S. T., Asante, K. P., Owusu-Agyei, S., et al. (2013). Efficacy of RTS,S malaria vaccines: individual-participant pooled analysis of phase 2 data. *The Lancet infectious diseases*, 13(4):319–327.

Bender, R., Augustin, T., and Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. *Statistics in medicine*, 24(11):1713–1723.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252.

Bi, J., Bennett, K., Embrechts, M., Breneman, C., and Song, M. (2003). Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, 3(Mar):1229–1243.

Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In Hausser, D., editor, *Proc. annual Conf. Computational Learning Theory*, pages 144–152. ACM Press, Pittsburg, PA.

Bradley, P. S. and Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In *ICML*, volume 98, pages 82–90.

Brank, J., Grobelnik, M., Milic-Frayling, N., and Mladenic, D. (2002). Feature selection using linear support vector machines. In *Proceedings of the 3rd International Conference on Data Mining Methods and Databases for Engineering.*

Breslow, N. (1972). Comment on 'Regression and life tables' by dr Cox. *Journal of the Royal Statistical Society, Series B*, 34:216–217.

Butler, N. S., Harris, T. H., and Blader, I. J. (2013). Regulation of immunopathogenesis during Plasmodium and Toxoplasma infections: more parallels than distinctions? *Trends in parasitology*, 29(12):593–602.

Caputo, B., Sim, K., Furesjo, F., and Smola, A. (2002). Appearance-based object recognition using SVMs: Which kernel should i use? In *Proceedings of NIPS Workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer Vision*, Whistler.

Chapelle, O., Sindhwani, V., and Keerthi, S. S. (2008). Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9(Feb):203–233.

Chapelle, O. and Zien, A. (2005). Semi-supervised classification by low density separation. In Cowell, R. G. and Ghahramani, Z., editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Jan 6-8, 2005, Savannah Hotel, Barbados*, pages 57–64. Society for Artificial Intelligence and Statistics.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chen, Y.-W. and Lin, C.-J. (2006). Combining SVMs with various feature selection strategies. In *Feature extraction*, pages 315–324. Springer.

Concato, J., Peduzzi, P., Holford, T. R., and Feinstein, A. R. (1995). Importance of events per independent variable in proportional hazards analysis. I. Background, goals, and general strategy. *J Clin Epidemiol*, 48(12):1495–1501.

Conroy, R., Pyörälä, K., Fitzgerald, A. e., Sans, S., Menotti, A., De Backer, G., De Bacquer, D., Ducimetiere, P., Jousilahti, P., Keil, U., et al. (2003). Estimation of ten-year risk of fatal cardiovascular disease in europe: the SCORE project. *European heart journal*, 24(11):987–1003.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 3(20):273–297.

Courtin, D., Oesterholt, M., Huismans, H., Kusi, K., Milet, J., Badaut, C., Gaye, O., Roeffen, W., Remarque, E. J., Sauerwein, R., et al. (2009). The quantity and quality of African children's IgG responses to merozoite surface antigens reflect protection against Plasmodium falciparum malaria. *PloS one*, 4(10):e7590.

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334.

Cox, D. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220.

Crawley, J., Chu, C., Mtove, G., and Nosten, F. (2010). Malaria in children. *Lancet*, 375(9724):1468–1481.

Derksen, S. and Keselman, H. (1992). Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, 45(2):265–282.

Doolan, D. L., Dobaño, C., and Baird, J. K. (2009). Acquired immunity to malaria. *Clinical microbiology reviews*, 22(1):13–36.

Duchateau, L. and Janssen, P. (2007). *The frailty model*. Springer Science & Business Media.

Evers, L. (2009). *survpack: methods for fitting high-dimensional survival models*. R package version 0.1-4.

Evers, L. and Messow, C. M. (2008). Sparse kernel methods for high-dimensional survival data. *Bioinformatics*, 24(14):1632–1638.

Franc, V., Zien, A., and Schölkopf, B. (2011). Support vector machines as probabilistic models. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 665–672.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin.

Goldberg, Y. and Kosorok, M. R. (2012). Support vector regression for right censored data. *arXiv preprint arXiv:1202.5130*.

Grandvalet, Y., Mariéthoz, J., and Bengio, S. (2005). Interpretation of SVMs with an application to unbalanced classification. In *Advances in Neural Information Processing Systems, NIPS 18*. Citeseer.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422.

Harrell, F. (2015). *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer.

Harrell, F. E., Lee, K. L., Califf, R. M., Pryor, D. B., and Rosati, R. A. (1984). Regression modelling strategies for improved prognostic prediction. *Statistics in Medicine*, 3(2):143–152.

Hastie, T. (2016). *svmpath: The SVM Path Algorithm*. R package version 0.955.

Hosmer, D. W., Lemeshow, S., and May, S. (2011). Applied survival analysis.

Inoue, S.-I., Niikura, M., Mineo, S., and Kobayashi, F. (2013). Roles of IFN-γ and $\gamma\delta$ T cells in protective immunity against blood-stage malaria. *Frontiers in immunology*, 4:258.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209.

Kalbfleisch, J. D. and Prentice, R. L. (1973). Marginal likelihoods based on Cox's regression and life model. *Biometrika*, 60(2):267–278.

Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20.

Karush, W. (1939). Minima of function of several variables with inequalities as side constraints. Master's thesis. Master's thesis, Dept. Mathematics, Chicago University.

Keerthi, S. and Lin, C.-J. (2003). Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural computation*, 15(7):1667–1689.

Khan, F. and Zubek, V. (2008). Support vector regression for censored data (SVRc): A novel tool for survival analysis. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, IEEE International Conference, pages 863–868.

Kim, D.-H. and Jeong, H.-C. (2006). Weighted LS-SVM regression for right censored data. *Communications for Statistical Applications and Methods*, 13(3):765–776.

Kimeldorf, G. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *J. Math. Anal. Appl.*, (33):82–95.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324.

Krooshof, P. W., Üstün, B., Postma, G. J., and Buydens, L. M. (2010). Visualization and recovery of the (bio) chemical interesting variables in data analysis with support vector machine classification. *Analytical chemistry*, 82(16):7000–7007.

Kuhn, H. and Tucker, A. (1951). Nonlinear programming. In *Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics*, pages 481–492. Univ. California Press, Berkeley.

Kuhn, M. (2016). *caret: classification and regression training.* R package version 6.0-70.

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

Lee, W., Zhang, X., and Teh, Y. (2006). Semi-supervised learning in reproducing kernel Hilbert spaces using local invariances. *NUS Technical Report TRB3/06.*

Li, H. and Gui, J. (2004). Partial Cox regression analysis for high-dimensional microarray gene expression data. *Bioinformatics*, 20(suppl 1):i208–i215.

Li, H. and Luan, Y. (2003). Kernel Cox regression models for linking gene expression profiles to censored survival data. *Pacific Symposium on Biocomputing*, 8(2):65–76.

Liang, Y., Chai, H., Liu, X. Y., Xu, Z. B., Zhang, H., and Leung, K. S. (2016). Cancer survival analysis using semi-supervised learning method based on Cox and AFT models with L1/2 regularization. *BMC Med Genomics*, 9:11.

Little, R. and Rubin, D. (1987). Multiple imputation for nonresponse in surveys.

Liu, Q., Chen, C., Zhang, Y., and Hu, Z. (2011). Feature selection for support vector machines with RBF kernel. *Artificial Intelligence Review*, 36(2):99–115.

Maldonado, S. and Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences*, 179(13):2208–2217.

Marrugat, J., Solanas, P., D'Agostino, R., Sullivan, L., Ordovas, J., Cordón, F., Ramos, R., Sala, J., Masià, R., Rohlfs, I., et al. (2003). Estimación del riesgo coronario en España mediante la ecuación de Framingham calibrada. *Revista española de cardiología*, 56(3):253–261.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2015). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.6-7.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K.-R. (1999). Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 41–48. IEEE.

Moncunill, G., Aponte, J. J., Nhabomba, A. J., and Dobaño, C. (2013). Performance of multiplex commercial kits to quantify cytokine and chemokine responses in culture supernatants from Plasmodium falciparum stimulations. *PLoS One*, 8(1):e52587.

Nash, J. C. et al. (2014). On best practice optimization methods in R. *Journal of Statistical Software*, 60(2):1–14.

Nash, J. C., Varadhan, R., et al. (2011). Unifying optimization algorithms to aid software system users: optimx for R. *Journal of Statistical Software*, 43(9):1–14.

Niaf, E., Flamary, R., Lartizien, C., and Canu, S. (2011). Handling uncertainties in SVM classification. *Statistical Signal Processing Workshop (SSP)*, pages 757–760.

Peduzzi, P., Concato, J., Feinstein, A. R., and Holford, T. R. (1995). Importance of events per independent variable in proportional hazards regression analysis. II. Accuracy and precision of regression estimates. *J Clin Epidemiol*, 48(12):1503–1510.

Perez-Mazliah, D. and Langhorne, J. (2015). CD4T-cell subsets in malaria: TH1/TH2 revisited. *CD4+ T cell differentiation in infection: amendments to the Th1/Th2 axiom*, page 50.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Postma, G., Krooshof, P., and Buydens, L. (2011). Opening the kernel of kernel partial least squares and support vector machines. *Analytica chimica acta*, 705(1):123–134.

Reverter, F., Vegas, E., and Oller, J. M. (2014). Kernel-PCA data integration with enhanced interpretability. *BMC systems biology*, 8(2):1.

RTS,S Clinical Trials Partnership (2015). Efficacy and safety of RTS,S/AS01 malaria vaccine with or without a booster dose in infants and children in Africa: final results of a phase 3, individually randomised, controlled trial. *Lancet*, 386(9988):31–45.

Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517.

Sama, W., Dietz, K., and Smith, T. (2006). Distribution of survival times of deliberate Plasmodium falciparum infections in tertiary syphilis patients. *Transactions of the Royal Society of Tropical Medicine and Hygiene*, 100(9):811–816.

Sanz, H., Aponte, J., Harezlak, J., Dong, Y., Murawska, M., and Valim, C. (2015). *drLumi: multiplex immunoassays data analysis.* R package version 0.1.2.

Scholkopf, B., Mika, S., Burges, C. J., Knirsch, P., Muller, K.-R., Ratsch, G., and Smola, A. J. (1999). Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks*, 10(5):1000–1017.

Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press.

SCTP, R., Agnandji, S. T., Lell, B., Fernandes, J. F., Abossolo, B. P., Methogo, B., Kabwende, A. L., Adegnika, A. A., Mordmueller, B., Issifou, S., et al. (2012). A phase 3 trial of RTS,S/AS01 malaria vaccine in African infants. *The New England journal of medicine*, 367(24):2284–95.

Serra-Toro, C., Traver, V. J., and Pla, F. (2014). Exploring some practical issues of SVM+: Is really privileged information that helps? *Pattern Recognition Letters*, 42:40–46.

Shiao, H.-T. and Cherkassky, V. (2013). SVM-based approaches for predictive modeling of survival data. In *Proceedings of the International Conference on Data Mining (DMIN)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Shim, J. and Hwang, C. (2009). Support vector censored quantile regression under random censoring. *Computational Statistics and Data Analysis*, 53(4):912–919.

Shivaswamy, P. K., Chu, W., and Jansche, M. (2007). A support vector approach to censored targets. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 655–660. IEEE.

Stanisic, D. I. and Good, M. F. (2016). Examining cellular immune responses to inform development of a blood-stage malaria vaccine. *Parasitology*, 143(02):208–223.

Stevenson, M. M., Ing, R., Berretta, F., and Miu, J. (2011). Regulating the adaptive immune response to blood-stage malaria: role of dendritic cells and CD4 (+) Foxp3 (+) regulatory T cells. *Int J Biol Sci*, 7(9):1311–1322.

Stevenson, M. M. and Riley, E. M. (2004). Innate immunity to malaria. *Nature Reviews Immunology*, 4(3):169–180.

Steyerberg, E. (2008). *Clinical prediction models: a practical approach to development, validation, and updating*. Springer Science & Business Media.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society. Series B (Methodological)*, pages 111–147.

Sutherland, C. J., Drakeley, C. J., and Schellenberg, D. (2007). How is childhood development of immunity to plasmodium falciparum enhanced by certain antimalarial interventions? *Malaria Journal*, 6(1):161.

Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

Tang, Y., Zhang, Y.-Q., Chawla, N. V., and Krasser, S. (2009). SVMs modeling for highly imbalanced classification. *Trans. Sys. Man Cyber. Part B*, 39(1):281–288.

Therneau, TM. and Grambsch, PM. (2000). *Modeling survival data: extending the Cox model.* Springer, New York.

Udhayakumar, V., Kariuki, S., Kolczack, M., Girma, M., Roberts, J. M., Oloo, A. J., Nahlen, B. L., and Lal, A. A. (2001). Longitudinal study of natural immune responses to the Plasmodium falciparum apical membrane antigen (AMA-1) in a holoendemic region of malaria in western Kenya: Asembo Bay Cohort Project VIII. *The American journal of tropical medicine and hygiene*, 65(2):100–107.

Van Belle, V., Pelckmans, K., Suykens, J., and Van Huffel, S. (2007). Support vector machines for survival analysis. In *Proceedings of the Third International Conference on Computational Intelligence in Medicine and Healthcare (CIMED2007)*, pages 1–8.

Van Belle, V., Pelckmans, K., Suykens, J. A., and Van Huffel, S. (2008). Survival SVM: a practical scalable algorithm. In *ESANN*, pages 89–94.

Van Belle, V., Pelckmans, K., Suykens, J. A., and Van Huffel, S. (2010). Additive survival least-squares support vector machines. *Statistics in Medicine*, 29(2):296–308.

Van Belle, V., Pelckmans, K., Van Huffel, S., and Suykens, J. A. (2011). Support vector methods for survival analysis: a comparison between ranking and regression approaches. *Artificial intelligence in medicine*, 53(2):107–118.

Vapnik, V. and Vashist, V. (2009). 2009 special issue: A new learning paradigm: Learning using privileged information. *Neural Network*, 22(5–6):544–557.

Varma, S. and Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91.

Veropoulos, K., Campbell, C., and Cristianini, N. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on AI*, pages 55–60.

Vittinghoff, E. and McCulloch, C. E. (2007). Relaxing the rule of ten events per variable in logistic and Cox regression. *Am. J. Epidemiol.*, 165(6):710–718.

Weihs, C., Ligges, U., Luebke, K., and Raabe, N. (2005). klaR analyzing german business cycles. In Baier, D., Decker, R., and Schmidt-Thieme, L., editors, *Data Analysis and Decision Support*, pages 335–343, Berlin. Springer-Verlag.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2000). Feature selection for SVMs. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, pages 647–653. MIT Press, Neural Information Processing Systems Foundation.

WHO (2015). World malaria report 2015.

Wilson, P. W., D'Agostino, R. B., Levy, D., Belanger, A. M., Silbershatz, H., and Kannel, W. B. (1998). Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847.

Wipasa, J. and Riley, E. M. (2007). The immunological challenges of malaria vaccine development. *Expert opinion on biological therapy*, 7(12):1841–1852.

Wu, G. and Chang, E. Y. (2003). Class-boundary alignment for imbalanced dataset learning. In *ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC*, pages 49–56.

Yang, X., Song, Q., and Wang, Y. (2007). A weighted support vector machine for data classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(05):961–976.

Zhu, J. and Hastie, T. (2005). Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14(1):185–205.

Resum en català

1.1 Introducció

El procés de creació d'una vacuna eficaç contra la malària és complex degut a les característiques del paràsit responsable. En la interacció vacuna-malaltia s'han de tenir en compte molts aspectes per millorar la vacuna, i sobretot entendre-la. En aquest context, trobar noves maneres de predir i modelar la malaltia, així com de trobar els marcadors més importants associats amb la malaltia, millorarà el coneixement i com a conseqüència l'eficàcia de la vacuna. Aquest és precisament l'objectiu de l'estudi analitzat en aquesta tesi, Mal067, identificar correlats de protecció de la vacuna contra la malària RTS,S.

La tecnologia Luminex® permet analitzar un elevat nombre de proteïnes amb una mostra de sang molt petita. La combinació d'aquest tipus de tècniques amb un elevat nombre de variables comparat amb el nombre d'observacions, fa que els mètodes estadístics clàssics tinguin problemes a l'hora d'analitzar aquestes dades. Les màquines de suport vectorial (SVM) no presenten problemes a l'hora d'analitzar dades amb moltes observacions i poques variables. Aquesta tècnica va ser ideada inicialment per classificar individus que pertanyien a dos grups diferents i a partir d'aquesta idea inicial diferents extensions s'han anat desenvolupant al llarg del temps. Per altra banda, en moltes ocasions, la recerca científica recull dades de l'estil temps fins a esdeveniment (dades de supervivència). Els mètodes desenvolupats per estendre SVM a dades temps fins a esdeveniment s'han focalitzat en una branca molt concreta, la de regressió, deixant de banda la recerca basant-se en l'aproximació de resposta binaria. A més, un aspecte no resolt al voltant dels models SVM és el referent a visualització i mesura de la rellevància de variables en el context de kernels no lineals. Aquest punt es torna molt important en la recerca de la millora d'una vacuna ja que, identificar correlats de protecció, antigens per exemple, induïts per la vacuna és crucial.

1.1.1 Infecció de malària

La malària és una infecció transmesa per la picada del mosquit femella *Anopheles*. Quan un mosquit pica a una persona infectada, agafa una petita quantitat de sang que conté el paràsit. El paràsit va passant per diferents fases i transformacions, fins que una setmana després, aproximadament, pot tornar a ser transmès. Així doncs, en la propera picada, el mosquit injecta la saliva i el paràsit, i transmet la malària a una altra persona. *Plasmodium falciparum*, un dels cinc tipus de paràsits transmissors de la malària, és el causant de la malària més severa. Durant l'any 2015 van haver-hi uns 214 milions de casos i 438.000 morts degudes a la malària (WHO, 2015), majoritàriament en nens de l'Àfrica sub-Sahariana, on *P.falciparum* és l'especie predominant. Les principals co-morbiditats associades a malària són: anèmia en nens i dones embarassades, poc pes al naixement, naixements prematurs i danys neurològics.

Existeixen diverses vacunes, cadascuna enfocada en una etapa de la infecció. La vacuna en estat més avançat és la vacuna RTS,S. Recentment, l'estudi Mal055, un assaig clínic en Fase III, multi-cèntric ha avaluat l'eficàcia de la vacuna a un any. L'eficàcia ha estat al voltant d'un 56% en nens d'entre 5 i 17 mesos d'edat, i al voltant d'un 31% en nens d'entre 6 i 12 setmanes d'edat (RTS,S Clinical Trials Partnership, 2015). La vacuna, però, confereix protecció limitada en el temps. Entendre la immunitat conferida per la vacuna ajudarà a desenvolupar-la i millorar la seva eficàcia i duració de la protecció.

Pel que fa a la definició de malària es va definir com participants que van anar al centre de salut i van presentar algun valor de *P.falciparum* en sang.

1.1.2 Dades de citocines, quimiocines i factors de creixement

La immunitat cel·lular és un procés complex que involucra diversos tipus de cèl·lules. Aquestes cèl·lules es poden quantificar mitjançant tecnologia multiplex, que habitualment es realitza utilitzant la plataforma Luminex®. Aquest assaig permet mesurar citocines, quimiocines i factors de creixement simultàniament utilitzant un petit volum de sang.

Generalment, en aquests assajos es mesura la fluorescència de les mostres utilitzant diferents grups de perles dipositats en plaques de 96 pouets. Després de realitzar tot el procés s'obté com a mesura final la mediana de la intensitat de la fluorescència (MFI) de totes les perles per mostra. El MFI es converteix en concentració mitjançant corbes estàndard, creades a partir de les anomenades mostres estàndard (amb concentració coneguda). Aquesta concentració per mostra és la mesura que s'utilitza com a mesura per comparar els diferents grups de individus, és a dir, és la mesura de la concentració d'aquell antigen en la mostra.

El panell de 30 antígens que es va utilitzar per analitzar el component cel·lular de l'estudi Mal067 va ser: factor de creixement epidèrmic (EGF), Eotaxin, factor de creixements dels fibroblasts (FGF), factor estimulant de colònies de granulòcits (G-CSF), factor estimulant de colònies de granulòcits i macròfags (GM-CSF), factor de creixement d'hepatòcits (HGF), interferó (IFN)-α, IFN-γ, interleucina (IL)-1RA, IL-1β, IL-2, IL-2R, IL-4, IL-5, IL-6, IL-7, IL-8, IL-10, IL-12(p40/p70), IL-13, IL-15, IL-17, proteïna induïda IFN-γ (IP-10), proteïna quimio-tàctica de monòcits (MCP-1), monòcits induïts per IFN-γ (MIG), proteïna inflamatòria macròfaga (MIP)-1α, MIP-1β, cèl·lula T normal expressada i secretada regulada en activació (RANTES), factor de tumor necròtic (TNF), i factor de creixement de l'endoteli vascular (VEGF).

Dintre d'aquest panell, i normalment, qualsevol panell hi han blocs d'antígens amb diferents nivells de correlació degut a la seva naturalesa.

1.1.3 Immunitat natural adquirida

Cada mostra de cada participant de l'estudi Mal067 va ser estimulada amb 5 antigens diferents per avaluar diferents aspectes de la vacuna. L'estimulació en la que es focalitza l'anàlisi d'aquesta tesi és l'antigen apical de membrana 1 (AMA1), suspesa en dimetilsulfòxid (DMSO). Per cada mostra es va mesurar la concentració de l'antigen per l'estimulació AMA1 i DMSO, el control negatiu. Per tant, cada participant té els seus propis valors d'AMA1 i DMSO. Per tal de comparar els valors entre participants es necessari comparar la raó AMA1/DMSO. Normalment, aquesta nova variable és asimètrica i s'acostuma a aplicar la transformació logarítmica.

La importància de l'estimulació AMA1 esta relacionada amb el que es coneix com a immunitat natural adquirida (NAI). A l'Àfrica sub-Sahariana la majoria d'habitants estan infectats contínuament per *P.falciparum*, i la majoria d'adults infectats en rares ocasions experimenten la malaltia. Els nivells de paràsits serien letals en una persona no exposada a la malària, però a ells els permet realitzar les seves tasques diàries. S'ha demostrat que nivells d'anticossos d'anti-AMA1 estan presents en aquells que han adquirit immunitat natural. Així doncs, entendre els mecanismes de protecció de immunitat adquirida és important en el desenvolupament de noves vacunes, i en el cas de l'estudi Mal067, de la vacuna RTS,S.

1.1.4 Anàlisis de la supervivència

Un dels models que s'utilitza habitualment en estadística per analitzar dades de tipus temps fins a esdeveniment és el model de riscos proporcionals de Cox. Aquest model queda expressat per la funció de risc

$$\lambda(t|\boldsymbol{x}_i) = \lambda_0(t)\exp(\langle \boldsymbol{x}_i, \boldsymbol{\beta}\rangle) \tag{1.1}$$

on $\lambda(t|\boldsymbol{x}_i)$ és el risc en el moment t per un individu i amb vector de covariables $\boldsymbol{x}_i$, $\lambda_0(t)$ és la funció de risc basal i $\boldsymbol{\beta}$ és el vector de coeficients del model. El model assumeix que la funció de risc basal és comú a tots els individus de la població. En aquest model, el risc d'un subjecte s'incrementa de manera multiplicativa amb les covariables. En el model de Cox la funció de risc basal es pot modelitzar de manera paramètrica o semi-paramètrica. El cas més utilitzat és la versió semi-paramètrica. La funció de log-versemblança parcial queda expressada com:

$$\log(\mathcal{L}(\boldsymbol{\beta})) = \sum_{i=1}^{n} \delta_i \left(\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle - \log\left(\sum_{j\in R_i} \exp(\langle \boldsymbol{x}_j, \boldsymbol{\beta}\rangle) \right)\right) \tag{1.2}$$

Maximitzant la funció de versemblança parcial, es poden obtenir els estimadors màxim versemblants que són assimptoticament no esbiaixats, eficients i amb distribució normal.

Tot i ser un model molt flexible. El model de riscos proporcionals de Cox fa una sèrie d'assumpcions (Therneau, TM. and Grambsch, PM., 2000):

- La variable temps fins esdeveniment és una variable continua. A la realitat, aquesta assumpció no es compleix, ja que en dades reals acostumen a haver-hi temps iguals (empats). Tot i això hi han diferents aproximacions per tractar aquest aspecte.

- Linealitat i additivitat dels predictors amb respecte el log-risc.
- Proporcionalitat de riscos al llarg del temps (o raó de riscos constants).
- Les observacions han de ser independents i idènticament distribuïdes. En cas de que aquesta situació no es doni existeixen extensions com els models de fragilitat compartida, que permeten tractar amb dades no independents.

Un altre aspecte important és el referent al numero mínim d'esdeveniments per variable (EPV). La funció de versemblança parcial depèn del nombre d'esdeveniments i no de la grandària mostral total. Simulacions portades a terme per Concato et al. (1995) i per Peduzzi et al. (1995) conclouen que el mínim EPV pel model de riscos proporcionals de Cox és 10. Vittinghoff and McCulloch (2007) va trobar que els resultats incloent de 5 a 9 EPV eren comparables a aquells incloent de 10 a 16 EPV. En tot cas, els tres estudis van estar d'acord en que els resultats on l'anàlisi tenint en compte variables confusores no es pot dur a terme sense violar el mínim numero d'EPV, han de ser interpretats amb cura.

1.2 Màquines de suport vectorial

La metodologia SVM va ser desenvolupada per Cortes and Vapnik (1995). Intuïtivament aquests models estan basats en classificar dos grups d'observacions mitjançant un hiperplà (superfície de decisió lineal). Aquest hiperplà maximitza la distància, precisament, entre el pla i les observacions. El gran avantatge dels SVM és que permeten implementar funcions no lineals, nuclis o kernels, per tal de representar les dades en un espai de dimensions diferents a l'original mitjançant l'anomenat *truc kernel.*

L'expressió SVM més coneguda expressada en termes de hiperplans, s'expressa com:

$$\begin{aligned}
&\underset{\boldsymbol{w},\boldsymbol{\xi}}{\text{minimitzar}} && \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i && \\
&\text{subjecte a} && \xi_i \geq 0, && i = 1,\dots,n, \\
& && y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i, && i = 1,\dots,n
\end{aligned} \tag{1.3}$$

on $\boldsymbol{w}$ és el vector de pesos del hiperplà, $\boldsymbol{x}_i$ és el vector de covariables de l'observació i, $y_i \in \{\pm 1\}$ és la classe a la que pertany l'observació[1] i, C és la constant de regularització que permet violacions de les restriccions i maximitzen el marge global i b és el llindar o biaix. El problema d'optimització a (1.3) és quadràtic, amb un mínim sempre observable que és el problema d'optimització primal. Aquest problema s'ha de resoldre en el dual.

Una vegada calculada la Lagrangiana i aplicant les simplificacions corresponents, obtenim el

[1] En els models SVM per classificació binària, pertànyer a una classe o grup de individus s'expressa com a ±1.

problema dual

$$
\begin{aligned}
\underset{\boldsymbol{\alpha}}{\text{minimitzar}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle - \sum_{i=1}^{n}\alpha_i \\
\text{subjecte a} \quad & 0 \leq \alpha_i \leq C, \qquad\qquad i = 1,\dots,n, \\
& \sum_{i=1}^{n}\alpha_i y_i = 0, \qquad\qquad i = 1,\dots,n
\end{aligned}
\tag{1.4}
$$

Després de trobar el vector $\boldsymbol{\alpha}$ optim es pot demostrar que la funció de decisió queda expressada com:

$$
f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{n}\alpha_i\langle \boldsymbol{x}_i, \boldsymbol{x}\rangle y_i + b\right) \tag{1.5}
$$

Aplicant el truc kernel, podem substituir els productes escalars de l'expressió de SVM i de la funció de decisió per un kernel definit positiu. Els kernels més utilitzats són el Gaussià i el lineal:

- Lineal:

  $$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{x}, \boldsymbol{x}'\rangle$$

  aquest és el producte escalar habitual.

- Gaussià:

  $$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\sigma\|\boldsymbol{x} - \boldsymbol{x}'\|^2)$$

  o en una formulació diferent,

  $$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2})$$

  on $\sigma > 0$. Una de les principals característiques és que $k(\boldsymbol{x}, \boldsymbol{x}) = 1$.

1.3 SVM per supervivència i rellevància de variables

1.3.1 Extensions SVM per supervivència

Les principals aproximacions a dades de supervivència utilitzant SVM identificades a la literatura es poden classificar en:

Regressió kernel de Cox. Aquest mètode no és una extensió específica del SVM però és el pas natural a mètodes kernel de la regressió de Cox. Li and Luan (2003) van aplicar mètodes kernel a la versemblança parcial del model de Cox, expressant-la com a una funció de pèrdua.

Aproximació utilitzant regressió de suport vectorial. Prenent com a referència la regressió de suport vectorial basada en la funció de pèrdua ϵ, Shivaswamy et al. (2007) van presentar un mètode enfocat bàsicament a la predicció del temps de supervivència, incloent el indicador de censura. Altres aproximacions són la de Van Belle et al. (2010) que utilitzen una versió de SVM basades en mínims quadrats per ordenar noves observacions amb respecte el seu temps de supervivència; o l'aproximació de Kim and Jeong (2006) on proposen utilitzar mínims quadrats ponderats mitjançant pesos basats en l'estimador Kaplan-Meier de supervivència.

Classificació ordinal suport vectorial. A Van Belle et al. (2007), Van Belle et al. (2008) i Evers and Messow (2008) l'aproximació esta basada en ordenar les observacions a partir de la maximització de l'estadístic C de Harrell (Harrell et al., 1984). L'objectiu principal del model és predir un index pronòstic.

Máquines de suport vectorial per classificació binària. Shiao and Cherkassky (2013) van proposar utilitzar el mètode SVM basat en aprenentatge utilitzant informació privilegiada (Vapnik and Vashist, 2009), que utilitza la censura com a informació privilegiada només disponible a les dades d'entrenament. L'altra aproximació que els autors proposen és utilitzar SVM amb incerteses (Niaf et al., 2011), aquest mètode permet afegir un determinat grau de confiança a la classe de cada observació permeten un grau de incertesa en la definició de classe. En ambdós mètodes, Shiao and Cherkassky (2013) proposen introduir la informació de la censura com la proporció de temps de seguiment de l'observació.

1.3.2 Implementació en R

Pel que fa a mètodes SVM per supervivència no hi han funcions específiques o paquets implementats en R, excepte el paquet `survpack` (Evers, 2009) que implementa la regressió kernel Cox i diferents extensions d'aquest mètode.

1.3.3 Comentaris generals i problemes adreçats en aquesta tesi

La major part d'extensions realitzades per supervivència estan enfocades a regressió. De la revisió realitzada a la literatura no s'han trobat diferències significatives a l'hora de comparar els mètodes de SVM per supervivència amb la regressió de riscos proporcionals de Cox o la regressió kernel Cox, indicant que s'ha de fer més investigació amb respecte a aquesta àrea. A més, no hem trobat investigació rellevant pel que fa a extensions en supervivència basant-se en SVM per classificació binària. Per tant, hi ha una oportunitat de investigar noves aproximacions i comparar-les amb el model de Cox o kernel Cox. L'aproximació de SVM per classificació binària per part de Shiao and Cherkassky (2013) enfoca les dades censurades en base a la proporcionalitat de temps de seguiment, és a dir, utilitzant la proporció de temps que s'ha seguit a l'observació. Això implica, que una observació censurada al principi del seguiment tindrà el mateix pes, sigui quina sigui la supervivència final de la cohort. Un altre aspecte pendent de fer recerca és investigar noves aproximacions basades en classificació binària, es necessiten més mètodes per tenir una perspectiva global del problema de dades de supervivència en el context de SVM.

Específicament, la present tesi s'aproxima als buits trobats en la recerca:

- Proposant un mètode alternatiu a la proporcionalitat de seguiment suggerida per Shiao and Cherkassky (2013) quan s'utilitza el mètode SVM amb incerteses i SVM amb informació privilegiada, mitjançant la supervivència condicionada al temps de seguiment.

- Proposant una aproximació des de la perspectiva SVM per classificació binària semi-supervisada al tractament de les censures.
- Avaluant el mètode SVM per classificació binària poderada aplicada a dades de supervivència.
- Implementant en R els mètodes proposats.

1.3.4 Rellevància de variables

Existeixen tres gran mètodes per identificar la rellevància de les variables en el context SVM (i *machine learning* en general):

Mètodes filtre. Avaluen la rellevància de les variables mirant les propietats intrínseques de les dades sense tenir en compte la informació proporcionada per l'algoritme de classificació. Com exemple, Chen and Lin (2006) proposen alguns mètodes filtre combinats amb SVM. Un d'ells és la puntuació de Fisher, que també és el mètode proposat per Maldonado and Weber (2009). Basades en aquesta puntuació les variables amb un valor més baix es descarten i només les restants s'inclouen en el model SVM.

Mètodes embolcall. L'avaluació d'un grup específic de variables s'obté mitjançant un model de classificació. És a dir, un algoritme de recerca de importància de variables embolcalla l'algoritme de classificació. Guyon et al. (2002) proposen un dels mètodes més utilitzats per selecció de variables en el context SVM. El mètode es conegut com Eliminació Recursiva de Variables (RFE-SVM) i l'algoritme consisteix, utilitzant el kernel lineal, en donat un model SVM amb totes les variables, anar excloent a cada iteració la variable amb un valor del vector pes $\boldsymbol{w}$ menor. D'aquesta manera el resultat final obtingut d'aplicar aquest algoritme és un rànquing de les variables segons la seva rellevància.

En el mateix article els autors proposen una aproximació per kernels no lineals. La idea és aplicar l'algoritme RFE-SVM però eliminant aquelles variables amb menor canvi en el valor de la funció de cost assumint els mateixos valors pels paràmetres a cada iteració.

Mètodes d'encastat. La cerca pel conjunt de variables més rellevants es realitza en la construcció de l'algoritme de classificació. Totes les aproximacions trobades a la literatura estan limitades al kernel lineal i a la vegada estan basades en modificacions en el terme de penalització. Aytug (2015) va formular dues versions del que anomenen *problema de selecció de variables*, una on explícitament restringeix el nombre de variables i una altra que penalitza el nombre de variables. Becker et al. (2009) i Becker et al. (2011) proposen una versió penalitzada dels mètodes SVM amb diferents termes de penalització. Depenent de la forma d'aquesta penalització els autors proposen les següents versions de SVM: Ridge SVM, LASSO SVM, Elastic net SVM, SCAD SVM i Elastic-SCAD SVM. Aquests mètodes es poden aplicar únicament amb el kernel lineal.

1.3.5 Implementació en R

El paquet `penalizedSVM` implementa els mètodes descrits en Becker et al. (2009) i Becker et al. (2011). L'algoritme RFE està implementat i disponible al paquet `caret` (Kuhn, 2016), encara que la mesura per la importància de variables no està basada en el vector de pesos si no en l'error quadràtic mitjà, R^2, precisió o Kappa depenent del mètode de classificació o regressió.

1.3.6 Comentaris generals i problemes adreçats en aquesta tesi

El principal avantatge dels mètodes SVM és la possibilitat de incorporar kernels no lineals. Un algoritme de selecció de variables hauria de ser capaç de tractar amb aquest tipus de kernels. Els mètodes embolcall són els que millor s'aproximen a aquesta idea: i) són més eficients que els mètodes filtre i ii) els mètodes encastats estan focalitzats a kernels lineals. El mètode embolcall de referència és l'algoritme RFE proposat per Guyon et al. (2002). Un altre aspecte important, on els mètodes embolcall no són prou eficients és en la visualització i interpretació dels resultats obtinguts durant el procés RFE. Per altra banda, aquest algoritme, per kernels no lineals, permet fer un rànquing de variables però no comparar el comportament de les variables en una iteració específica, i tampoc interpretar els resultats tenint en compte: associació amb la variable resposta, associació amb les altres variables i magnitud d'aquesta associació.

Específicament, la present tesi s'aproxima als buits trobats en la recerca:

- Proposant un mètode basat en l'algoritme RFE que permet visualitzar la rellevància de variables graficant les prediccions d'un model SVM.
- Proposant dos algoritmes basats en el mètode RFE i en la representació de variables en l'espai de les components principals kernelitzades (KPCA).

1.4 Objectius

1.4.1 Objectius principals

Els objectius principals d'aquesta tesi són:

1. Desenvolupar mètodes SVM per tractar dades temps fins esdeveniment en el context de SVM per classificació binària.
2. Desenvolupar mètodes per visualització i rànquing de variables en el context de mètodes SVM per dades censurades.
3. Identificar les citocines, quimiocines i factors de creixement rellevants associats amb correlats de protecció induïts per la vacuna RTS,S en l'estudi Mal067.

1.4.2 Objectius específics pels mètodes SVM per dades de supervivència

Els objectius específics pels mètodes proposats per SVM en supervivència són:

1. Avaluar l'aproximació de supervivència condicionada comparada amb la de proporcionalitat de seguiment en els mètodes proposats.

2. Avaluar el comportament global dels mètodes proposats comparats amb el model de riscos proporcional de Cox i la regressió kernel Cox en escenaris específics segons: distribució de la censura, grandària mostral i compliment de l'assumpció de proporcionalitat de riscos.

3. Implementar el codi en R dels models SVM comparats.

1.4.3 Objectius específics per visualització i rellevància de variables

Els objectius específics per visualització i rànquing de variables són:

1. Avaluar el comportament dels tres mètodes proposats en termes de rànquing comparats amb l'aproximació RFE per kernels no lineals, sota diferents escenaris segons les variables directament associades amb la variable resposta.

2. Interpretar i visualitzar les figures obtingudes pels mètodes proposats.

1.4.4 Objectius específics per l'anàlisi de les dades de l'estudi Mal067

Els objectius específics per l'anàlisi de les dades de l'estudi Mal067 són:

1. Trobar el mètode que millor s'adequa a les dades Mal067, incloent: els mètodes SVM proposats, el model de riscos proporcionals de Cox i la regressió kernel Cox per ambdues cohorts, RTS,S i comparadora.

2. Visualitzar els resultats sobre rellevància d'anàlits i fer un rànquing de tots els antígens respecte la seva rellevància, pel millor model trobat per cada vacuna, i pel millor algoritme basat en RFE trobat en l'estudi de simulació.

1.5 Màquines de suport vectorial aprenent utilitzant informació privilegiada

Aquesta aproximació, també coneguda com a LUPI, va ser descrita per Vapnik (Vapnik and Vashist, 2009) i es basa en utilitzar la informació únicament disponible en el procés d'entrenament de les dades. El problema d'optimització s'expressa com:

$$\begin{aligned}
\underset{\boldsymbol{w},\boldsymbol{w}^*,b,b^*}{\text{minimizar}} \quad & \frac{1}{2}(\|\boldsymbol{w}\|^2 + \gamma\|\boldsymbol{w}^*\|^2) + C\sum_{i=1}^{n}\xi_i \\
\text{subjecte a} \quad & \xi_i = (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \\
& y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*), \quad i = 1, \ldots, n, \\
& (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \geq 0, \qquad\qquad\qquad\qquad\quad i = 1, \ldots, n
\end{aligned}$$

on $\boldsymbol{w}$, $\boldsymbol{x}_i$ i b són els vector de pesos, el vector de covariables i el biaix habitual expressat en la notació clàssica SVM i les equivalents versions * són de l'espai de correcció. La funció de decisió, després d'aplicar el truc kernel, es defineix com:

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{n} y_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b\right)$$

1.5.1 Aplicació a les dades de supervivència

En les dades d'entrenament la variable temps fins esdeveniment es coneguda per totes les observacions però no ho és en les dades a predir. Així doncs, la informació de les dades censurades pot ser utilitzada com informació privilegiada. Segons Shiao and Cherkassky (2013) la censura en forma de informació privilegiada es pot expressar com:

$$\boldsymbol{x}_i^* = (T_i, p_i)$$

on T_i és el temps observat (esdeveniment o censura) i p_i és la certesa de ser no esdeveniment, proporcional al temps observat. Així doncs per un esdeveniment serà 0 i en observacions censurades serà T_i/τ, sent τ el temps de seguiment establert. La nostra proposta és definir la informació privilegiada basada en la probabilitat de presentar l'esdeveniment, condicionada al moment en que l'observació està censurada, utilitzant l'estimador Kaplan-Meier.

1.5.2 Paràmetres i kernels

Shiao and Cherkassky (2013) van comparar el kernel Gaussià amb el kernel lineal pels dos espais (correcció i decisió) obtenint resultats semblants. Basant-nos en aquesta conclusió utilitzarem el kernel lineal per l'espai de correcció i el Gaussià per l'espai de decisió. A més, es compararem tant el mètode de censura de proporcionalitat en el temps com el de supervivència condicionada.

1.6 Màquines de suport vectorial per classes incertes

Aquesta extensió de SVM desenvolupada per Niaf et al. (2011) permet definir de manera no perfecta algunes observacions pel que respecta a la seva classe. Per a aquestes observacions es dona un nivell de confiança o de probabilitat. Aquest mètode també se'l coneix com a *pSVM*. El problem d'optimització primal s'expressa com

$$\begin{array}{lll}
\underset{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\xi}^-,\boldsymbol{\xi}^+,b}{\text{minimitzar}} & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \widetilde{C}\sum_{i=n+1}^{m}(\xi_i^- + \xi_i^+) & \\
\text{subjecte a} & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i, & i = 1, \ldots, n, \\
& z_i^- - \xi_i^- \leq \langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b \leq z_i^+ + \xi_i^+, & i = n+1, \ldots, m, \\
& \xi_i \geq 0, & i = 1, \ldots, n, \\
& \xi_i^- \geq 0, & i = n+1, \ldots, m, \\
& \xi_i^+ \geq 0, & i = n+1, \ldots, m
\end{array}$$

on els paràmetres C i $\widetilde{C}$ controlen el pes de les classes certes i de les incertes respectivament, i z_i^- i z_i^+ són els límits que depenen del grau de certesa o probabilitat p_i que haguem especificat a cada observació i. La funció de decisió després d'aplicar el truc kernel és

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) - \sum_{i=n+1}^{m} (\mu_i^+ - \mu_i^-) k(\boldsymbol{x}_i, \boldsymbol{x}) + b\right)$$

1.6.1 Aplicació a les dades de supervivència

Shiao and Cherkassky (2013), seguint la mateixa idea que per l'aproximació LUPI, van suggerir utilitzar com a probabilitat per les observacions censurades la proporció de temps fins al final de seguiment. La nostra proposta és utilitzar la supervivència condicional estimada mitjançant l'estimador Kaplan-Meier.

1.6.2 Paràmetres i kernels

La nostra proposta, per avaluar el mètode, és utilitzar el kernel lineal i el kernel Gaussià. A més, de l'aproximació proporcional al temps, la censura també s'implementarà amb l'aproximació de supervivència condicionada.

1.7 Màquines de suport vectorial semi-supervisades amb invariàncies locals

Una perspectiva diferent que proposem a les dades censurades en SVM és el de dades semi-supervisades. En el context de dades semi-supervisades hi han observacions on les classes són conegudes i unes altres no ho són. De les classes conegudes i desconegudes s'aprèn per tal de crear una funció de decisió que discrimini les observacions amb classes conegudes.

En un context diferent al de SVM, Lee et al. (2006) proposen un mètode d'aprenentatge semi-supervisat amb kernels reproductors a espais de Hilbert utilitzant invariàncies locals que caracteritzen explícitament el comportament de la funció objectiu al voltant de les classes conegudes i desconegudes. Lee et al. (2006) expressen la funció a minimitzar com

$$\rho_1 \|g\|^2 + \rho_2 \sum_{i=l+1}^{n} l_1(\mathrm{L}_i(g)) + \sum_{i=1}^{l} l_2(y_i, g(\boldsymbol{x}_i))$$

on $\rho_1 > 0$ i $\rho_2 > 0$ són els pesos relatius a les funcions de pèrdua l_1 per les classes desconegudes i l_2 per les classes conegudes. Per $\mathrm{L}_i(g)$ s'expressa el funcional lineal per les classes desconegudes i observació i. Com a funcions l_1 i l_2 es pot utilitzar qualsevol funció de pèrdua. Les funcions que utilitzarem seran la *hinge* per l_2 i ϵ-insensible per l_1, d'aquesta manera equiparem aquest problema d'optimització al clàssic SVM i tenim un problema convex. Així doncs, el problema d'optimització queda de la següent manera

$$\begin{array}{lll}
\underset{g,b}{\text{minimitzar}} & \rho_1\|g\|^2 + \rho_2 \sum_{i=l+1}^{n} (\xi_i + \xi_i^*) + \sum_{i=1}^{l} \gamma_i & \\
\text{subjecte a} & -\langle g, z_i\rangle - b \leq \epsilon + \xi_i, & i = l+1, \ldots, n, \\
 & \langle g, z_i\rangle + b \leq \epsilon + \xi_i^*, & i = l+1, \ldots, n, \\
 & \xi_i \geq 0, & i = l+1, \ldots, n, \\
 & \xi_i^* \geq 0, & i = l+1, \ldots, n, \\
 & y_i(\langle g, \phi(\boldsymbol{x}_i)\rangle + b) \geq 1 - \gamma_i, & i = 1, \ldots, l, \\
 & \gamma_i \geq 0, & i = 1, \ldots, l
\end{array}$$

Sent la funció de decisió després d'aplicar el truc kernel

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{l} \beta_i^* y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + \sum_{i=l+1}^{n} z_i(\boldsymbol{x})(\alpha_i - \alpha_i^*)\right)$$

on el terme z_i serà diferent depenent de la invariància local que apliquem.

1.7.1 Invariàncies locals

Els autors, Lee et al. (2006), proposen, entre altres, dos tipus de invariàncies basades en l'assumpció que el hiperplà no varia als voltants de cada observació. Específicament les dues invariàncies són:

- Invariància gradient: restringeix el valor del gradient de la funció en les observacions amb classe coneguda.
- Invariància mitjana: restringeix el valor de la funció a cada punt, de tal manera que el valor de l'observació és semblant al valor mitjà de les observacions que té al seu voltant.

1.7.2 Aplicació a les dades de supervivència

Les dades censurades durant el seguiment es tractaran com una classe desconeguda. Els esdeveniments i els no esdeveniments al final del seguiment es tractaran com classes conegudes.

1.7.3 Paràmetres i kernels

El nombre de paràmetres i kernels per afinar depèn del tipus de invariància local utilitzada. Per la invariància gradient, que es basa en kernel Gaussià, aquest serà l'utilitzat. Per la invariància mitjana el kernel Gaussià és el que s'utilitzarà, i també s'utilitzarà la funció Gaussiana que defineix la finestra a partir de la qual s'han de mesurar el valor mitjà.

1.8 Màquines de suport vectorial ponderades

Un altra aproximació que no s'ha avaluat en la literatura és la ponderació d'observacions en SVM. La idea consisteix en assignar un pes a cadascuna de les observacions segons la seva importància relativa (Yang et al., 2007). Aquesta metodologia està focalitzada principalment en el tractament de valors extrems. El problema d'optimització queda expressat com

$$\begin{aligned} \underset{\boldsymbol{w},\boldsymbol{\xi}}{\text{minimitzar}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} W_i\xi_i \\ \text{subjecte a} \quad & \xi_i \geq 0, && i = 1,\ldots,n \\ & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i, && i = 1,\ldots,n \end{aligned}$$

sent $\boldsymbol{W}$ el vector ponderació que dona un determinat pes a cadascuna de les observacions. Després d'aplicar la Lagrangiana podem trobar la funció de decisió, havent aplicat el truc kernel, com:

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) y_i + b\right)$$

1.8.1 Aplicació a les dades de supervivència

Les dades censurades es poden entendre com una observació ponderada o parcial. Una observació censurada just abans del final del seguiment hauria de ser tractada *pràcticament* com una observació completa, així doncs, seguint el mateix criteri que pels mètodes anteriors els pesos es calcularan utilitzant la proporcionalitat de temps seguit i la supervivència condicionada mitjançant l'estimador Kaplan-Meier.

1.8.2 Paràmetres i kernels

Només s'utilitzarà el kernel Gaussià.

1.9 Simulacions per SVM amb dades censurades

Per avaluar el comportament dels quatre mètodes SVM per dades de supervivència proposats, s'ha dut a terme un estudi mitjançant simulacions. S'han generat diferents combinacions d'escenaris atenent a diversos aspectes.

1.9.1 Generació de covariables

S'han simulat 30 variables seguint un patró similar al de l'estudi Mal067. A cada base de dades simulada s'han generat 30 variables a partir d'una distribució normal multivariant. La mitjana de cada variable s'ha extret d'una distribució Uniforme(0,03, 0,06). La matriu de covariàncies s'ha calculat en cada base de dades de tal manera que de les 30 variables, s'han creat 4 grups de variables segons el grau de correlació de Pearson entre elles: no correlació, baixa, mitjana i alta.

1.9.2 Generació de variable temps fins a malària

S'han comparat dos escenaris referents a la variable temps fins a malària basats en el model de riscos proporcionals de Cox. En un, es compleix la proporcionalitat de riscos i en l'altre no. La variable temps fins a malària, seguint Sama et al. (2006) i Bejon et al. (2013), s'ha simulat basant-se en una distribució Gompertz. Així doncs, seguint Bender et al. (2005) es pot simular una variable temps fins esdeveniment que segueix una distribució Cox-Gompertz:

$$T = \frac{1}{\alpha} \log\left(1 - \frac{\alpha \log(U)}{\gamma \exp(\langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle)}\right)$$

on U és una variable que segueix una Uniforme(0,1), $\alpha \in (-\infty, \infty)$ i $\gamma > 0$ són els paràmetres d'escala i forma d'una distribució Gompertz.

En el context de no proporcionalitat de riscos, l'expressió anterior es pot modificar de tal manera que podem violar la proporcionalitat de riscos afegint una variabilitat comú a 5 grups de individus (com si es tractes d'una variable no observada), simulant d'aquesta manera un model de fragilitat compartida (Duchateau and Janssen, 2007).

Tots els escenaris es van simular de tal manera que la supervivència als 18 mesos estigues al voltant de 0.6 (Agnandji et al., 2011).

1.9.3 Generació de la variable censura

Es van simular dos escenaris pel que fa a la proporció de censura de les dades: un amb un 10% i un altre amb un 30%. Pel que fa a la distribució, es van simular 3 tipus de distribució:

- Distribució uniforme: la censura es reparteix amb la mateixa probabilitat al llarg del seguiment.
- Distribució asimètrica positiva: la distribució es dona més al final del seguiment que al principi. Per simular aquesta distribució s'ha utilitzat una distribució Exponencial.
- Distribució asimètrica negativa: la distribució es dona més al començament del seguiment que al final. S'ha utilitzat una distribució Exponencial invertida.

1.9.4 Nombre d'observacions

S'han simulat dos escenaris pel que fa al nombre d'observacions: un amb 300 i un altre amb 50.

1.9.5 Comparació de mètodes

Els mètodes que s'han comparat han estat:

- Model de riscos proporcionals de Cox
- Regressió kernel Cox: els valors avaluats pel kernel Gaussià han estat 0.25, 0.5, 1, 2 i 4.

- Màquines de suport vectorial ponderades: s'han comparat dues aproximacions, una definint els pesos segons la proporcionalitat de temps en el seguiment i l'altra segons la supervivència condicionada. Els valors avaluats pel paràmetre C han estat 0.1, 1, 10 i 100, i pel kernel 0.25, 0.5, 1, 2 i 4.

- Màquines de suport vectorial per classes incertes: s'ha utilitzat el kernel lineal i Gaussià. Els dos mètodes per calcular el grau de incertesa són proporcionalitat en el temps i supervivència condicionada. Pel kernel Gaussià els valors que s'han avaluat són 0.25, 0.5, 1, 2 i 4. Els valors avaluats per C i $\widetilde{C}$ són 0.1, 1, 10 i 100. El valor η s'ha fixat a 0.0001.

- Màquines de suport vectorial amb informació privilegiada: el kernel Gaussià i el kernel lineal s'han utilitzat per l'espai de decisió i correcció respectivament. Per l'espai de correcció s'ha utilitzat tant l'aproximació de temps proporcional a la censura com la de supervivència condicionada. Pel kernel Gaussià els valors avaluats han estat 0.25, 0.5, 1, 2 i 4. El valor de C s'ha avaluat d'entre els valors 0.1, 1, 10 i 100. El pes per a l'espai de correcció, γ, s'ha avaluat d'entre 0.1, 1, 10 i 100.

- Màquines de suport vectorial semi-supervisades amb invariàncies locals: s'han comparat les dues aproximacions, tant per gradient com per mitjana. Els paràmetres pel kernel Gaussià i per la densitat Gaussiana (només per la invariància mitjana) avaluats han estat 0.25, 0.5, 1, 2 i 4. El paràmetre ρ_2 s'ha avaluat entre 0.1, 1, 10 and 100 tant per gradient com per mitjana.

1.9.6 Criteris d'avaluació i afinació de paràmetres

Per afinar els paràmetres i avaluar el comportament de cada mètode i escenari, s'ha utilitzat un procés d'entrenament-validació-prova dels models. Aquest procés s'ha realitzat en dues etapes:

1. Afinat de paràmetres: per a cada combinació de paràmetres, s'han modelat 10 bases de dades d'entrenament i, per cadascuna, 10 de validació. A cada base de dades de validació es va mesurar la precisió, és a dir, per a cada combinació de paràmetres, es va mesurar la precisió 10 vegades i es va fer el mitjana. La combinació de paràmetres amb una precisió més elevada va ser la seleccionada.

2. Avaluació dels mètodes: es van utilitzar 10 bases de dades d'entrenament, diferents de les utilitzades en la fase d'afinat de paràmetres, per modelar els mètodes amb la millor combinació de paràmetres. Per cadascun d'aquests 10 models, es van provar 10 bases de dades diferents i mesurar diverses mètriques. Per tant, s'han utilitzat 100 dades simulades per mesurar el comportament dels mètodes proposats.

1.9.7 Mesures per avaluar el comportament dels mètodes

Es van utilitzar quatre mesures per avaluar el comportament dels mètodes proposats i de referència:

- Precisió: mesura la proporció d'observacions classificades correctament en relació al total.
- Coeficient de correlació de Matthews: és un coeficient que presenta valors en l'interval $[-1, +1]$. Un valor de -1 indica desacord total entre predicció i vertadera malària, un valor de 0 indica una predicció completament aleatòria i un valor de -1 total desacord entre predicció i realitat.
- Informació mútua normalitzada: representa la reducció de incertesa d'una variable quan s'observa una altra. Un valor de 0 indica una predicció totalment aleatòria i de 1 que la predicció és perfecte.
- Àrea sota la corba ROC: mesura el grau de discriminació d'un model entre individus que han patit l'esdeveniment i individus que no, sent un valor de 0,5 una discriminació dels individus totalment aleatòria i de 1 perfecte.

1.9.8 Resultats de la simulació

Per tots els escenaris de 300 observacions i proporcionalitat de riscos, el model de Cox i pSVM (amb kernel lineal) van presentar resultats molt semblants, juntament amb el mètode de invariàncies locals (tant gradient com mitjana), aquest últim amb mesures properes als dos millors mètodes descrits. Específicament, la precisió va estar al voltant de 0,89 pel model de Cox, 0,87 pel kernel lineal pSVM i 0,84 per SVM amb invàriancies locals. Pel que fa a l'àrea sota la corba, el model de Cox va obtenir un valor de 0,96, pSVM 0,95 i invariàncies locals 0,92. La distribució i proporció de la censura no va afectar els resultats generals, encara que dels tres mètodes descrits, invariàncies locals-gradient va ser el més afectat. Pel que fa als escenaris on es violava la proporcionalitat de riscos, els resultats a l'escenari de 300 observacions es van veure més afectats que en l'escenari de 50 observacions.

1.9.9 Discussió

L'aproximació de supervivència condicionada es comporta lleugerament millor que el mètode de proporcionalitat en el seguiment en tots els models comparats. En condicions favorables al model de Cox i regressió kernel Cox, els mètodes proposats pSVM i de invariàncies locals es comporten de manera semblant, sent més robustos i tenint uns millors resultats a modificacions de la grandària mostral i violació de la proporcionalitat de riscos.

1.9.10 Recerca futura

De cara a recerca futura, seria interessant introduir més informació referent a la supervivència de la cohort en els model SVM per supervivència com LUPI i pSVM, de tal manera que les dades censurades tinguessin en compte la supervivència final i la variabilitat d'aquesta. Un exemple d'això podria ser, a més d'introduir l'estimació Kaplan-Meier de supervivència condicionada, introduir la variabilitat de l'estimació puntual, és a dir, l'interval de confiança. Un altre aspecte important a tenir en compte en recerca futura, és el tipus de no proporcionalitat de riscos. Existeixen diferents tipus de violació de la condició de proporcionalitat de riscos, un d'ells, no avaluat en aquesta tesi, és la deguda a coeficients variants en el temps, és a dir, que el risc associat a una variable no és constant al llarg del temps. Avaluar el comportament dels mètodes sota aquesta situació i estudiar maneres de tenir en compte aquesta informació resultaria de molta utilitat ja que és una situació relativament comú en recerca biomèdica.

1.10 RFE-pseudo-mostres

El primer mètode proposat sobre rellevància de variables és una extensió de les idees proposades per Krooshof et al. (2010) i Postma et al. (2011). En aquesta tesi proposem un algoritme mitjançant RFE utilitzant les anomenades *pseudo-mostres*. El mètode proposat queda definit de la següent manera:

1. Optimitzar els valors i paràmetres del SVM.
2. Crear una matriu de pseudo-mostres on diferents valors, equidistants, engloben tot l'interval observat d'una variable p, mantenint la resta de variables com a 0. Tal com es representa a continuació:

$$\begin{array}{cc} \begin{array}{ccccc} \text{Var.1} & \text{Var.2} & \text{Var.3} & & \text{Var.p} \end{array} & \\ \left(\begin{array}{ccccc} z_1 & 0 & 0 & \dots & 0 \\ z_2 & 0 & 0 & \dots & 0 \\ z_3 & 0 & 0 & \dots & 0 \\ & & \vdots & & \\ z_p & 0 & 0 & \dots & 0 \end{array}\right) & \begin{array}{l} \text{pseudo-mostra}_1 \\ \text{pseudo-mostra}_2 \\ \text{pseudo-mostra}_3 \\ \\ \text{pseudo-mostra}_p \end{array} \end{array}$$

3. Predir els valors de decisió per cadascuna de les pseudo-mostres (no la classe).
4. Fer un gràfic posant en l'eix de les X els valors de la variable en qüestió i en l'eix Y el valor de decisió.

Una vegada s'han representat gràficament els valors de decisió per totes les variables, s'ha de tenir en compte:

- Valors associats de cada variables amb la resposta.
- La variabilitat en els valors de decisió que poden ser indicatiu de la rellevància de la variable.

1.10.1 Rànquing de variables

El mètode de rànquing de variables que proposem esta basat en la variabilitat en els valors obtinguts en la funció de decisió. Aplicant l'algoritme RFE, aquella variable que presenti una menor variabilitat en els valors de decisió és la variable que s'exclou com a menys rellevant. Per mesurar la variabilitat proposem la desviació absoluta de la mediana (MAD).

1.11 RFE-kernel components principals per representació de variables

Reverter et al. (2014) proposen un mètode, en un context diferent al de SVM, utilitzant components principals kernelitzades (KPCA), per representar, per cadascuna de les variables, la direcció de màxim creixement local. Donades dues component principals, representades en un gràfic, es pot representar el màxim creixement per cada variable per cadascuna de les observacions mitjançant vectors. De la mateixa manera, els autors també presenten un mètode per representar funcions de variables. Si una variable és rellevant, mostrarà una direcció clara en totes les observacions i si no les direccions seran aleatòries. En aquest context, l'algoritme que proposem és el següent:

1. Modelar SVM amb les dades d'entrenament (incloent totes les variables).

2. Crear l'espai de KPCA utilitzant els paràmetres de SVM (quan sigui possible), amb totes les variables.

3. Representar les dues primeres components de manera gràfica.

4. Calcular i representar el màxim creixement de cadascuna de les variables i observacions. Així mateix també per la funció de decisió del SVM.

5. Calcular l'angle mitjà de cadascuna de les variables-observacions amb la funció de decisió, que és utilitzada com a referència.

6. Basant-se en alguna mètrica calcular quina variable és menys rellevant.

7. Excloure la variable menys rellevant.

8. Repetir el procés fins que només quedi una variable, aplicant d'aquesta manera l'algoritme RFE.

El que es pot representar, segons Reverter et al. (2014), és per cada observació les variables com vectors, que indiquen la direcció de màxim creixement per cada variable o funció de variables. Si dues variables estan positivament correlacionades, les direccions de màxim creixement per totes les observacions han d'anar en la mateixa direcció i sentit. Si estan correlacionades negativament la direcció hauria de ser globalment en sentit oposat i si les variables no estan correlacionades les direccions haurien de ser aleatòries.

1.11.1 Rànquing de variables

La nostra proposta es pot dividir en dues aproximacions alternatives:

- Incloure la predicció dels valors de decisió de SVM per cadascuna de les observacions com una nova variable, que anomenem *variable de referència* i després comparar cadascuna de les variables amb la de referència.

- Incloure la direcció de la funció de decisió del model SVM i prendre-la com la *direcció de referència.*

En ambdues aproximacions es pot calcular una mesura de la similitud entre una variable i la referència (predicció o decisió) mitjançant la mitjana de l'angle de totes les observacions amb la referència. Així, si per una observació, l'angle de la direcció de màxim creixement d'una variable p amb la referència és de 0 graus, indicarà que el vector de direccions es superposa i estan perfectament correlacionades. Si pel contrari, l'angle és de 180 graus, aniran en la mateixa direcció però sentit oposat, indicant que estan inversament correlacionades. Fent la mitjana de l'angle de totes les observacions obtenim un resum de la similitud de cada variable amb la referència, i conseqüentment si la variable és rellevant o no. Assumint, que hi ha soroll en les dades reals, una variable es pot classificar com a rellevant al comparar-la amb les altres: la variable amb un angle mitjà més proper al angle mig, tenint en compte totes les variables, serà classificada com menys rellevant.

1.12 Simulació d'escenaris i generació de dades

La simulació d'escenaris s'ha realitzat simulant per a cada simulació 30 variables seguint una distribució normal, estructurada en 4 blocs diferents segons la correlació: no correlació, baixa, moderada i alta.

La variable temps fins a malària s'ha simulat seguint una distribució Gompertz en un context de riscos proporcionals. El numero d'observacions de cada base de dades va ser de 50. La distribució de la censura va seguir una Uniforme amb un 10% de censura.

1.12.1 Escenaris segons rellevància de variables

S'han simulat 6 escenaris atenent a les variables implicades en la generació de temps fins a malària, per tal d'avaluar cada mètode proposat:

1. Variable 1

2. -Variable 29 + Variable 30

3. -Variable 1 + Variable 8 + Variable 20 + Variable 29 - Variable 30

4. Variable 1 + Variable 2 + Variable 1 × Variable 2

5. Variable 1 + Variable 30 + Variable 1 × Variable 30 + Variable 20 + Variable 20^2
6. Variable 1 + Variable 1^2 + exp (Variable 30)

1.12.2 Comparació de mètodes

El model SVM que es va utilitzar ha estat el de pSVM amb l'estimació del pes de la censura basat en la supervivència condicionada. Una vegada el model es va optimitzar els mètodes que es van comparar van ser:

- RFE-Guyon pel kernel Gaussià: es va prendre com el mètode de referència.
- RFE-KPCA-predicció: el KPCA es va basar en kernel Gaussià i el paràmetre utilitzat va ser el de pSVM.
- RFE-KPCA-decisió: el KPCA es va basar en kernel Gaussià i el paràmetre utilitzat va ser el de pSVM.
- RFE-pseudo-mostres: per crear-les es van utilitzar 50 punts equidistants de la variable en qüestió. Els valors de les pseudo-mostres van anar de -2 a 2, donat que les variables estan distribuïdes normalment al voltant de 0 aproximadament.

1.12.3 Mesures per avaluar els mètodes

Es van simular 100 bases de dades diferents per a cada escenari. De cadascun es va mesurar el rànquing obtingut d'aplicar l'algoritme RFE de les 30 variables, i es va resumir amb la posició mitjana i la desviació estàndard. Per als 3 mètodes proposats s'han creat, com exemple, resultats d'algunes iteracions per interpretar els resultats obtinguts.

1.12.4 Resultats

El mètode de les pseudo-mostres ha presentat resultats iguals o superiors en tots els escenaris al mètode de referència RFE-Guyon i als RFE-KPCA. Pel que fa als mètodes RFE-KPCA els resultats han estat millor en general que els de RFE-Guyon. Tots els mètodes han posicionat correctament les variables rellevants quan aquestes eren independents de les altres que hi havien en les dades. Pel contrari, en situacions de correlació o de interaccions el millor mètode ha estat el de les pseudo-mostres i després, depenent de l'escenari, RFE-KPCA-predicció.

1.12.5 Discussió

Els mètodes proposats es comporten tan bé o millor que el RFE-Guyon. Destaca, RFE-pseudo-mostres, comportant-se millor fins i tot en escenaris de no-linealitat o de interacció de variables. Els mètodes proposats, a més, permeten visualitzar i interpretar els resultats en termes de tipus d'associació entre variables i variable resposta. A nivell computacional, no són mètodes exigents, a excepció de RFE-KPCA-decisió que és molt costos. Els mètodes proposats són una bona eina com a complement o per elles mateixes per trobar les variables rellevants en un conjunt de dades.

1.12.6 Recerca futura

La recerca referent a pseudo-mostres hauria d'anar enfocada a provar diferents pseudo-mostres basades en la distribució de les dades, com per exemple quartils. Un altre punt que cal investigar és el referent a la mètrica a utilitzar, en aquesta tesi s'ha utilitzat la desviació amb respecte la mediana, però s'haurien de provar altres mesures. Amb respecte el mètode RFE-KPCA el problema principal és que en el KPCA no es té en compte la variable resposta, així doncs, les components es creen únicament amb la informació de les variables al ser un mètode no-supervisat. Una aproximació a investigar seria seguint la mateixa aproximació però utilitzant l'anàlisi discriminant de Fisher (Mika et al., 1999) amb kernels, per exemple, ja que la classe es tindria en compte.

1.13 Anàlisi de les dades Mal067

1.13.1 Disseny de l'estudi

La base de dades de l'estudi Mal067 consta de 459 nens de dos centres africans: Bagamoyo (Tanzània) i Manhiça (Moçambic), d'entre 6 i 12 setmanes d'edat i d'entre 5 a 17 mesos. L'objectiu principal de l'estudi és identificar quines citocines, quimiocines i factors de creixement estan correlacionats amb protecció en cada tipus de vacuna (RTS,S o comparadora) d'entre els participants que han rebut la tercera dosi de la vacuna assignada.

Del panell original de 30 antígens, dos IL-6 i IL-8, van haver de ser exclosos de l'estudi degut als problemes en la quantificació de les mostres. Finalment, doncs, es van analitzar 28 antígens.

1.13.2 Anàlisis estadístic

Les concentracions, mesurades com $\log_{10}$(AMA1/DMSO), entre RTS,S i vacuna comparadora i malària i no malària per antigen, es van comparar mitjançant la t-Student. La correlació entre anàlits dintre de cada grup de vacunació es va mesurar mitjançant la correlació d'Spearman. Es van estimar les corbes de supervivència Kaplan-Meier amb el corresponent interval de confiança al 95% per cadascun dels grups de vacunació i es van comparar mitjançat la prova log-rank. Es van realitzar models de Cox per tal d'avaluar el risc de cadascun dels antigens pel que fa a presentar malària a un any de seguiment. La proporcionalitat de riscos en cada model es va avaluar mitjançant els residus escalats de Schoenfeld (amb el p valor i de manera visual).

1.13.3 Mètodes SVM per dades censurades i rellevància d'antígens

Es van comparar i optimitzar tots els models proposats en aquesta tesi així com el model de Cox i la regressió kernel Cox. Els models són:

- Model de riscos proporcionals de Cox
- Regressió kernel Cox.

- Màquines de suport vectorial ponderades.
- Màquines de suport vectorial amb classes incertes.
- Màquines de suport vectorial amb informació privilegiada.
- Màquines de suport vectorial semi-supervisades amb invariàncies locals.

Exceptuant el model de Cox, els paràmetres d'afinament juntament amb l'avaluació de la precisió del model, es va realitzar mitjançant validació-creuada niuada amb 5 iteracions, seguint el descrit a Stone (1974), Varma and Simon (2006).

Després de trobar el model amb una precisió més alta, aquest es va utilitzar com a base per aplicar l'algoritme RFE-pseudo-mostres que va ser el que va mostrar millors resultats en l'estudi de simulacions de mètodes de rellevància de variables. A més, per a cada cohort es va aplicar l'algoritme per a 100 mostres amb reemplaçament, per tal de tenir una mesura de la variabilitat de la rellevància.

1.13.4 Resultats

Per la cohort RTS,S els antigens amb una risc associat a malària van ser: G-CSF, IFN-α, IL-10 i MCP-1. Per la vacuna comparadora, en canvi, no van haver-hi antígens estadísticament significatius. No van haver-hi diferències estadísticament significatives entre els dos grup vacunats pel que fa a la supervivència a 12 mesos (p valor = 0,167), tot i que la supervivència per RTS,S va ser superior al final del seguiment. Amb respecte la proporcionalitat de riscos dels antigens, per la cohort RTS,S, després de realitzar una inspecció visual juntament amb els p valors de la prova dels residus de Schoenfeld es va descartar violació de la proporcionalitat de riscos. Pel que fa a la cohort de la vacuna comparadora, cap anàlit va violar l'assumpció de proporcionalitat de riscos.

El model SVM que va presentar millors resultats va ser, per ambdues cohorts, pSVM amb kernel Gaussià. Aquest model va ser l'utilitzat per aplicar l'algoritme RFE-pseudo-mostres. Per la cohort RTS,S els antígens més rellevants van ser: RANTES, IL-12, G-CSF, Eotaxin i EGF. Per la cohort comparadora, els antígens més rellevants van ser: IL-15, IP-10, IL-2, MIP-1α i HGF.

1.13.5 Discussió

Tots el mètodes comparats, basats en SVM, tendeixen a ser equivalents a SVM per classificació binària en situacions de cap censura. Per aquesta raó, tots els mètodes SVM presenten resultats molt similars en les dades Mal067. Pel que fa al mètode RFE-pseudo-mostres, la variabilitat en els resultats en la cohort comparadora és major que en la cohort RTS,S. Això pot ser degut a la menor grandària mostral, i nombre d'esdeveniments, d'aquesta cohort.

1.13.6 Recerca futura

Un punt important sobre el que investigar és comparar els resultats obtinguts en aquesta tesi amb altres mètodes com poden ser regressió parcial de Cox (Li and Gui, 2004) per comprovar si els resultat obtinguts són consistents o no. Un altre aspecte important és l'anàlisi de subgrups: per edat i centre per descartar possibles efectes confusors. Un altre punt important, especialment relacionat amb NAI, és el de mesurar els valors de cada anàlit en el moment basal, és a dir, abans de rebre la primera dosi de la vacuna i mesurar l'evolució dels valors dels antigens al moment de rebre la tercera dosi.

1.14 Conclusions

Aquesta tesi ha explorat tres aspectes:

1. L'estudi de mètodes per tractar variables temps fins esdeveniment en el context de SVM per classificació binària.
2. L'estudi d'algoritmes per visualitzar i fer un rànquing de les variables rellevants en el context de SVM per dades de supervivència.
3. La classificació d'antígens associats a correlats de protecció induits per la vacuna RTS,S.

Amb respecte el primer punt:

- S'ha mostrat que:
  - El mètode proposat de supervivència condicionada millora la predicció de les dades comparat amb el mètode de proporcionalitat en el temps.
  - El mètode LUPI es comporta de manera similar al SVM ponderat.
  - El mètode semi-supervisat SVM amb invariàncies locals es comporta de manera similar al millor mètode observat en les simulacions per a cada escenari de proporcionalitat de riscos.
  - El comportament del mètode SVM amb invariàncies locals, de tots els mètodes estudiats, és el més robust a modificacions de censura (tant en proporció com en distribució) i violació de l'assumpció de proportionalitat de riscos.
  - El mètode SVM ponderat, independentment de l'aproximació per estimar els pesos, queda influenciat per la proporció de censura i la proporcionalitat de riscos.
- S'ha desenvolupat codi d'R que:
  - Implementa l'aproximació LUPI.
  - Implementa l'aproximació pSVM.
  - Implementa l'aproximació SVM ponderat.
  - Implementa l'aproximació SVM amb invariàncies locals pel mètode gradient i mitjana.

Amb respecte el segon punt:

- S'ha proposat un algoritme RFE-pseudo-mostres que:
  - Permet visualitzar i interpretar l'associació i rellevància de cada variable amb la variable resposta temps fins esdeveniment.
  - Fa un rànquing de les variables segons la seva rellevància.
- S'ha proposat un algoritme RFE-KPCA amb dues alternatives que:
  - Permet comparar la direcció de màxim creixement d'una funció de decisió i comparar-la amb les variables predictores.
  - Fa un rànquing de les variables segons la seva rellevància.
- S'ha demostrat que:
  - L'aproximació RFE-pseudo-mostres millora el rànquing de RFE-Guyon per kernels no lineals i els dos mètodes alternatius basats en RFE-KPCA.
  - L'alternativa RFE-KPCA per predicció es comporta millor que l'alternativa per funció de decisió i de manera similar a Guyon-RFE per kernels no lineals.

Amb respecte el tercer punt:

- L'estudi dels mètodes ha mostrat que els mètodes SVM proposats per dades de supervivència es comporten millor que el model de riscos proporcionals de Cox i regressió kernel Cox, tant per la cohort RTS,S com la cohort comparadora.
- L'estudi de les citocines, quimiocines i factors de creixement ha mostrat que:
  - Els cinc antígens més rellevants a la cohort RTS,S són: RANTES, IL-12, G-CSF, Eotaxin i EGF.
  - Els cinc antígens més rellevants a la cohort comparadora són: IL-15, IP-10, IL-2r, MIP-1α i HGF.

Appendix

Appendix A

Theoretical Aspects

A.1 Rationale support vector machines

The rationale behind SVM is to find a hyperplane as decision surface that splits the space into two parts. Therefore, a hyperplane can be seen as a binary classifier.

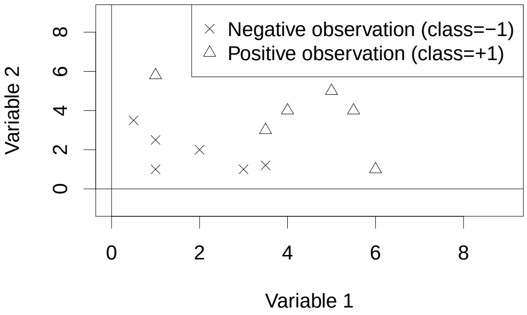


Figure A.1: Example of 12 observations, 2 classes (defined as $\{\pm 1\}$) and 2 variables (Variable 1 and Variable 2).

A.1.1 Hyperplane

An equation of a hyperplane is defined by a point (P_0) and a perpendicular vector to the plane ($\boldsymbol{w}$) at that point. We can define the vectors, $\boldsymbol{x}_0 = \overrightarrow{0P_0}$ and $\boldsymbol{x} = \overrightarrow{0P}$ where P is an arbitrary point of a hyperplane. A condition for P to be on the plane is that the vector $(\boldsymbol{x} - \boldsymbol{x}_0)$ is perpendicular to $\boldsymbol{w}$ (Figure A.2):

$$\langle \boldsymbol{w}, (\boldsymbol{x} - \boldsymbol{x}_0) \rangle = 0 \Rightarrow \langle \boldsymbol{w}, \boldsymbol{x} \rangle - \langle \boldsymbol{w}, \boldsymbol{x}_0 \rangle = 0 \tag{A.1}$$

If we define $b = -\langle \boldsymbol{w}, \boldsymbol{x}_0 \rangle$ we obtain that $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$. If the parameter b is modified then we obtain parallel hyperplanes along the direction of $\boldsymbol{w}$.

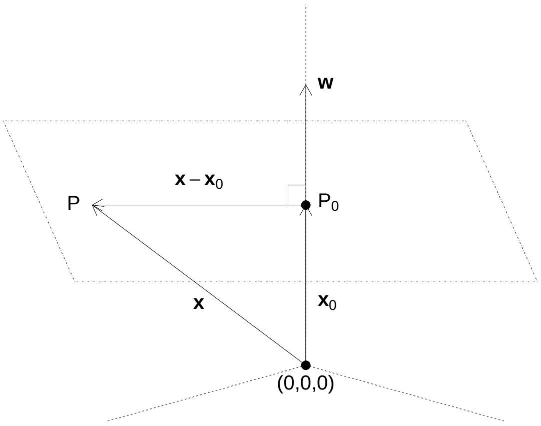


Figure A.2: Representation of a hyperplane in a 3-dimensional space.

A.1.2 Distance between two hyperplanes

Given two parallel hyperplanes as the ones showed in Figure A.3 defining D as the distance between both we can express that the vector $\boldsymbol{x_2}$ is the sum of vectors $\boldsymbol{x_1}$ and $t\boldsymbol{w}$. We can see then that the distance can be defined as $D = \|t\boldsymbol{w}\| = |t|\,\|\boldsymbol{w}\|$. As per rationale (A.1) we can define

$$\begin{aligned}\langle \boldsymbol{w}, \boldsymbol{x}_2 \rangle + b_2 = 0 \Rightarrow \langle \boldsymbol{w}, (\boldsymbol{x}_1 + t\boldsymbol{w}) \rangle + b_2 &= \langle \boldsymbol{w}, \boldsymbol{x}_1 \rangle + t\|\boldsymbol{w}\|^2 + b_2 \\ &= (\langle \boldsymbol{w}, \boldsymbol{x}_1 \rangle + b_1) - b_1 + t\|\boldsymbol{w}\|^2 + b_2 \\ &\Rightarrow -b_1 + t\|\boldsymbol{w}\|^2 + b_2 = 0 \Rightarrow t = \frac{b_1 - b_2}{\|\boldsymbol{w}\|^2} \\ &\Rightarrow D = |t|\,\|\boldsymbol{w}\| = \frac{|b_1 - b_2|}{\|\boldsymbol{w}\|}\end{aligned}$$

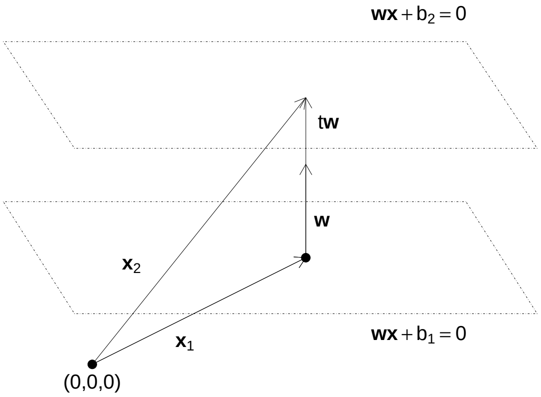


Figure A.3: Representation of two parallel hyperplanes in a 3-dimensional space.

A.1.3 Hyperplane as classifier and hard support vector machines

If we want to find a hyperplane to separate negative observations from the positive ones we will find that an infinite number of such hyperplanes exists. The main goal of SVM is to find the one that maximizes the margin between data points on the boundaries, what are called *support vectors* (Figure A.4). The distance between the shown hyperplanes is $D = \frac{2}{\|\boldsymbol{w}\|}$. Since we want to maximize the margins we need to minimize $\|\boldsymbol{w}\|$, and for mathematical and optimization purposes we want to minimize the equivalent $\frac{1}{2}\|\boldsymbol{w}\|$. In addition, we need to impose constraints that all observations are correctly classified:

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \leq -1 \text{ if } y_i = -1$$
$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \geq +1 \text{ if } y_i = +1$$

or equivalently $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1$ what give us the main expression of the hard margin SVM:

$$\begin{aligned} &\underset{\boldsymbol{w},b}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{subject to} \quad y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 \quad i = 1, \dots, n \end{aligned} \tag{A.2}$$

Then given a new observation $\boldsymbol{x}_j$, the classifier is $f(\boldsymbol{x}_j) = \text{sign}(\langle \boldsymbol{w}, \boldsymbol{x}_j \rangle + b)$

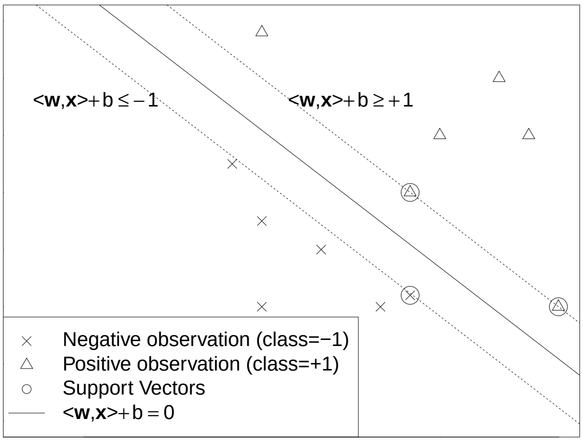

<w,x>+ b ≤ − 1
<w,x>+ b ≥ + 1
× Negative observation (class=−1)
△ Positive observation (class=+1)
○ Support Vectors
<w,x>+ b = 0

Figure A.4: Representation of the hyperplane that classifies binary data and the corresponding support vectors.

A.1.4 Soft margin support vector machines

In practice, a separating hyperplane may not exist due to the fact that in reality there are overlapping classes. To allow for the possibility of overlapping classes slack variables, $\xi_i \geq 0$ for all $i = 1, \dots, n$, are introduced in order to relax the constraints in (A.2):

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i \quad i = 1, \dots, n$$

A classifier that generalizes well is then found by controlling both the classifier capacity (via $\|\boldsymbol{w}\|$) and the sum of the slacks $\sum_{i=1}^{n} \xi_i$. The latter can be shown to provide an upper bound on the number of training errors. Then the minimizing objective function is given by

$$\begin{aligned} \underset{\boldsymbol{w},b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i \quad i = 1,\dots,n \\ & \xi_i \geq 0 \quad\quad\quad\quad\quad\quad\quad\; i = 1,\dots,n \end{aligned} \tag{A.3}$$

The expression (A.3) is the usual support vector machines optimization function, and originally known as *soft margin support vector machines.*

A.2 Taylor series of Gaussian kernel

Using the Gaussian kernel defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{x}'\|^2)$$

the Taylor series expansion is applied, and assuming that $x_i, x_j \in \mathbb{R}$ and $\gamma > 0$ we obtain for two given scalars:

$$\begin{aligned} \exp(-\gamma\|x_i - x_j\|^2) &= \exp(-\gamma(x_i - x_j)^2) = \exp(-\gamma x_i^2 + 2\gamma x_i x_j - \gamma x_j^2) \\ &= \exp(-\gamma x_i^2 - \gamma x_j^2)\sum_{k=0}^{\infty}\frac{(2\gamma x_i x_j)^k}{k!} \\ &= \exp(-\gamma x_i^2 - \gamma x_j^2)\left(1 + \sqrt{\frac{2\gamma}{1!}}x_i\sqrt{\frac{2\gamma}{1!}}x_j + \sqrt{\frac{(2\gamma)^2}{2!}}x_i^2\sqrt{\frac{(2\gamma)^2}{2!}}x_j^2 + \dots\right) \\ &= \langle \phi(x_i), \phi(x_j)\rangle \end{aligned}$$

where

$$\phi(x) = \exp(-\gamma x^2)\left[1, \sqrt{\frac{2\gamma}{1!}}x, \sqrt{\frac{(2\gamma)^2}{2!}}x^2, \dots\right]$$

A.3 Loss functions

A.3.1 ϵ-insensitive loss

The ϵ-insensitive loss function is the following:

$$l(y, f(\boldsymbol{x}))_{\epsilon} = \max(0, |y - f(\boldsymbol{x})| - \epsilon)$$

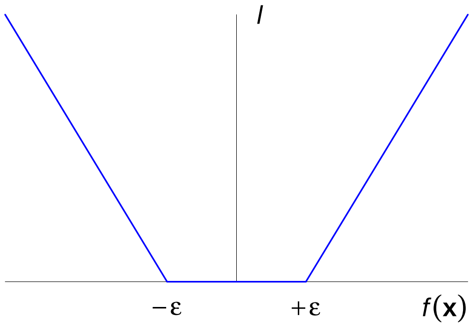


Figure A.5: Representation of the ϵ-insensitive loss function.

A.3.2 Hinge loss

The hinge loss function is the following:

$$l(y, f(\boldsymbol{x})) = \max(0, 1 - yf(\boldsymbol{x}))$$

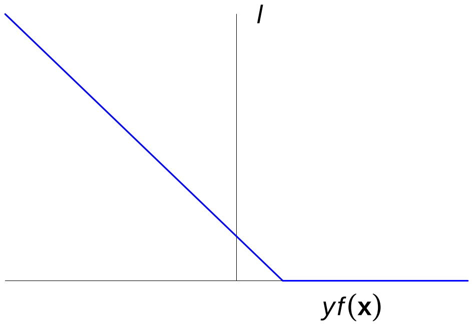


Figure A.6: Representation of the hinge loss function.

Appendix B

Primal-Dual Optimization Problems

B.1 Support vector machines learning using privileged information

In the LUPI paradigm the function to be minimized is:

$$\begin{aligned}
\underset{\boldsymbol{w},\boldsymbol{w}^*,b,b^*}{\text{minimize}} \quad & \frac{1}{2}(\|\boldsymbol{w}\|^2 + \gamma\|\boldsymbol{w}^*\|^2) + C\sum_{i=1}^{n}\xi_i \\
\text{subject to} \quad & \xi_i = (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*), && i = 1,\dots,n, \\
& y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*), && i = 1,\dots,n, \\
& (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \geq 0, && i = 1,\dots,n
\end{aligned}$$

To solve this problem the Lagrangian should be constructed:

$$\begin{aligned}
L(\boldsymbol{w}, b, \boldsymbol{w}^*, b^*, \boldsymbol{\alpha}, \boldsymbol{\beta}) = & \frac{1}{2}(\|\boldsymbol{w}\|^2 + \gamma\|\boldsymbol{w}^*\|^2) \\
& + C\sum_{i=1}^{n}(\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) - \sum_{i=1}^{n}\alpha_i(y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \\
& - 1 + (\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*)) - \sum_{i=1}^{n}\beta_i(\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*)
\end{aligned} \tag{B.1}$$

At the saddle point the derivatives of L with respect primal variables must vanish, the necessary conditions are:

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} - \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i = 0$$

$$\boldsymbol{w} = \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i \tag{B.2}$$

$$\frac{\partial L}{\partial \boldsymbol{w}^*} = 0 \implies \gamma\boldsymbol{w}^* + \sum_{i=1}^{n}C\boldsymbol{x}_i^* - \sum_{i=1}^{n}\alpha_i\boldsymbol{x}_i^* - \sum_{i=1}^{n}\beta_i\boldsymbol{x}_i^* = 0$$

$$\boldsymbol{w}^* = \frac{1}{\gamma}\sum_{i=1}^{n}\boldsymbol{x}_i^*(\alpha_i + \beta_i - C) \tag{B.3}$$

$$\frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{B.4}$$

$$\frac{\partial L}{\partial b^*} = 0 \implies \sum_{i=1}^{n} C - \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \beta_i = 0$$

$$\sum_{i=1}^{n} (\alpha_i + \beta_i - C) = 0 \tag{B.5}$$

Substituting (B.2) and (B.3) into the Lagrange function (B.1) and taking into account (B.4) and (B.5) we obtain the following expressions:

$$\begin{aligned}
R(\boldsymbol{\alpha}, \boldsymbol{\beta}) \overset{(B.2)}{=}\; & \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \frac{\gamma}{2} \| \boldsymbol{w}^* \|^2 + C \sum_{1=n}^{n} ((\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \\
& - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\sum_{j=1}^{n} \alpha_j y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + b \right) - 1 + ((\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \right] - \sum_{i=1}^{n} \beta_i ((\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \\
=\; & \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \frac{\gamma}{2} \| \boldsymbol{w}^* \|^2 + C \sum_{i=1}^{n} ((\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \\
& - \left[\sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{n} \alpha_j y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + b \right] + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i ((\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) - \sum_{i=1}^{n} \beta_i ((\langle \boldsymbol{w}^*, \boldsymbol{x}_i^* \rangle + b^*) \\
\overset{(B.3)}{=}\; & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=n}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \frac{\gamma}{2} \left[\frac{1}{\gamma^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C) \langle \boldsymbol{x}_i^*, \boldsymbol{x}_j^* \rangle \right] \\
& + \sum_{i=1}^{n} (\alpha_i + \beta_i - C) \left[\frac{1}{\gamma} \sum_{j=1}^{n} \langle \boldsymbol{x}_j^*, \boldsymbol{x}_i^* \rangle (\alpha_j + \beta_j - C) \right] \\
=\; & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=n}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\
& + \frac{1}{2\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C) \langle \boldsymbol{x}_i^*, \boldsymbol{x}_j^* \rangle \\
& - \frac{1}{\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C) \langle \boldsymbol{x}_i^*, \boldsymbol{x}_j^* \rangle \\
=\; & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \frac{-1}{2\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C) \langle \boldsymbol{x}_i^*, \boldsymbol{x}_j^* \rangle \quad \text{(B.6)}
\end{aligned}$$

Applying the kernel trick, the function (B.6) to be maximized can be formulated as:

$$R(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{2\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C) k(\boldsymbol{x}_i^*, \boldsymbol{x}_j^*)$$

B.2 Support vector machines with uncertain classes

In the SVM with uncertain classes the function to be minimized is

$$\begin{aligned}
\underset{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\xi}^-,\boldsymbol{\xi}^+b}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \widetilde{C}\sum_{i=n+1}^{m}(\xi_i^- + \xi_i^+) \\
\text{subject to} \quad & y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) \geq 1 - \xi_i, & i = 1,\dots,n, \\
& z_i^- - \xi_i^- \leq \langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b \leq z_i^+ + \xi_i^+, & i = n+1,\dots,m, \\
& \xi_i \geq 0 & i = 1,\dots,n, \\
& \xi_i^- \geq 0 & i = n+1,\dots,m, \\
& \xi_i^+ \geq 0 & i = n+1,\dots,m
\end{aligned} \tag{B.7}$$

The problem can be re-written it introducing the Lagrange multipliers:

$$\begin{aligned}
L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^-, \boldsymbol{\xi}^+, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}^+, \boldsymbol{\mu}^-, \boldsymbol{\gamma}^+, \boldsymbol{\gamma}^-) = {} & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \widetilde{C}\sum_{i=n+1}^{m}(\xi_i^- + \xi_i^+) \\
& - \sum_{i=1}^{n}\alpha_i(y_i\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) - (1 - \xi_i)) - \sum_{i=1}^{n}\beta_i\xi_i \\
& - \sum_{i=n+1}^{m}\mu_i^-((\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) - (z_i^- - \xi_i^-)) - \sum_{i=n+1}^{m}\gamma_i^-\xi_i^- \\
& - \sum_{i=n+1}^{m}\mu_i^+((z_i^+ + \xi_i^+) - (\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b)) - \sum_{i=n+1}^{m}\gamma_i^+\xi_i^+
\end{aligned} \tag{B.8}$$

At the saddle point the derivatives of L with respect primal variables must vanish, the necessary conditions are:

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w} - \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i - \sum_{i=n+1}^{m}\mu_i^-\boldsymbol{x}_i + \sum_{i=n+1}^{m}\mu_i^+\boldsymbol{x}_i = 0 \\
\boldsymbol{w} = \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i + \sum_{i=n+1}^{m}\mu_i^-\boldsymbol{x}_i - \sum_{i=n+1}^{m}\mu_i^+\boldsymbol{x}_i \\
= \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i - \sum_{i=n+1}^{m}(\mu_i^+ - \mu_i^-)\boldsymbol{x}_i
\end{aligned} \tag{B.9}$$

$$\begin{aligned}
\frac{\partial L}{\partial b} = 0 \implies -\sum_{i=1}^{n}\alpha_i y_i - \sum_{i=n+1}^{m}\mu_i^- + \sum_{i=n+1}^{m}\mu_i^+ = 0 \\
\sum_{i=1}^{n}\alpha_i y_i = \sum_{i=n+1}^{m}(\mu_i^+ - \mu_i^-)
\end{aligned} \tag{B.10}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \implies C - \alpha_i - \beta_i = 0 \tag{B.11}$$

$$\frac{\partial L}{\partial \xi_i^-} = 0 \implies \widetilde{C} - \mu_i^- - \gamma_i^- = 0 \tag{B.12}$$

$$\frac{\partial L}{\partial \xi_i^+} = 0 \implies \widetilde{C} - \mu_i^+ - \gamma_i^+ = 0 \tag{B.13}$$

Substituting the previous expression into the Lagrange function we obtain:

$$
\begin{aligned}
R(\boldsymbol{\alpha}, \boldsymbol{\mu}^+, \boldsymbol{\mu}^-) &\overset{(B.9)}{=} \frac{1}{2}\Bigg[\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle - 2\sum_{i=1}^{n}\sum_{j=n+1}^{m}\alpha_i y_i\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle(\mu_j^+ - \mu_j^-) + \\
&\sum_{i=n+1}^{m}\sum_{j=n+1}^{m}(\mu_i^+ - \mu_i^-)(\mu_j^+ - \mu_j^-)\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle\Bigg] + C\sum_{i=1}^{n}\xi_i + \widetilde{C}\sum_{i=n+1}^{m}(\xi_i^- + \xi_i^+) \\
&- \sum_{i=1}^{n}\alpha_i\left[y_i\left(\sum_{j=1}^{n}\alpha_j y_j\langle\boldsymbol{x}_j, \boldsymbol{x}_i\rangle - \sum_{j=n+1}^{m}(\mu_j^+ - \mu_j^-)\rangle\boldsymbol{x}_i, \boldsymbol{x}_j\langle + b\right) - (1 - \xi_i)\right] \\
&- \sum_{i=1}^{n}\beta_i\xi_i - \sum_{i=n+1}^{m}\gamma_i^-\xi_i^- - \sum_{i=n+1}^{m}\gamma_i^+\xi_i^+ \\
&- \sum_{i=n+1}^{m}\mu_i^-\left[\left(\sum_{j=1}^{n}\alpha_j y_j\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle - \sum_{j=n+1}^{m}(\mu_j^+ - \mu_j^-)\rangle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle + b\right) - (z_i^- - \xi_i^-)\right] \\
&- \sum_{j=n+1}^{m}\mu_i^+\left[(z_i^+ + \xi_i^+) - \left(\sum_{j=1}^{n}\alpha_j y_j\langle\boldsymbol{x}_j, \boldsymbol{x}_i\rangle - \sum_{j=n+1}^{m}(\mu_j^+ - \mu_j^-)\langle\boldsymbol{x}_j, \boldsymbol{x}_i\rangle + b\right)\right] \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle - \sum_{i=1}^{n}\sum_{j=n+1}^{m}\alpha_i y_i\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle(\mu_j^+ - \mu_j^-) \\
&+ \frac{1}{2}\sum_{i=n+1}^{m}\sum_{j=n+1}^{m}(\mu_i^+ - \mu_i^-)(\mu_j^+ - \mu_j^-)\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle + C\sum_{i=1}^{n}\xi_i + \widetilde{C}\sum_{i=n+1}^{m}(\xi_i^- + \xi_i^+) \\
&- \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i y_i\alpha_j y_j\langle\boldsymbol{x}_j, \boldsymbol{x}_i\rangle + \sum_{i=1}^{n}\sum_{j=n+1}^{m}\alpha_i y_i(\mu_j^+ - \mu_j^-)\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle - \sum_{i=1}^{n}\alpha_i y_i b \\
&+ \sum_{i=1}^{n}\alpha_i(1 - \xi_i) - \sum_{i=n+1}^{m}\mu_i^- b - \sum_{i=1}^{n}\beta_i\xi_i \\
&- \sum_{i=n+1}^{m}\sum_{j=1}^{n}\mu_i^-\alpha_j y_j\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i=n+1}^{m}\sum_{j=n+1}^{m}\mu_i^-(\mu_j^+ - \mu_j^-)\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle \\
&+ \sum_{i=n+1}^{m}\mu_i^-(z_i^- - \xi_i^-) - \sum_{i=n+1}^{m}\gamma_i^-\xi_i^- - \sum_{i=n+1}^{m}\mu_i^+(z_i^+ + \xi_i^+) + \sum_{i=n+1}^{m}\sum_{j=1}^{n}\mu_i^+\alpha_j y_j\langle\boldsymbol{x}_j, \boldsymbol{x}_i\rangle \\
&- \sum_{i=n+1}^{m}\sum_{j=n+1}^{m}\mu_i^+(\mu_j^+ - \mu_j^-)\langle\boldsymbol{x}_j, \boldsymbol{x}_i\rangle + \sum_{i=n+1}^{m}\mu_i^+ b - \sum_{i=n+1}^{m}\gamma_i^+\xi_i^+ \\
&\overset{(B.10)}{=} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\langle\boldsymbol{x}_i\boldsymbol{x}_j\rangle + \sum_{i=n+1}^{m}\sum_{j=1}^{n}\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle\alpha_j y_j(\mu_i^+ - \mu_i^-) \\
&+ \frac{1}{2}\sum_{i=n+1}^{m}\sum_{j=n+1}^{m}(\mu_i^+ - \mu_i^-)(\mu_j^+ - \mu_j^-)\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle + C\sum_{i=1}^{n}\xi_i + \widetilde{C}\sum_{i=n+1}^{m}(\xi_i^- + \xi_i^+) \\
&+ \sum_{i=1}^{n}\alpha_i(1 - \xi_i) - \sum_{i=n+1}^{m}\sum_{i=n+1}^{m}\gamma_i^-\xi_i^- - \sum_{i=1}^{n}\beta_i\xi_i \\
&+ \sum_{i=n+1}^{m}\sum_{j=n+1}^{m}\mu_i^-(\mu_j^+ - \mu_j^-)\langle\boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i=n+1}^{m}\mu_i^-(z_i^- - \xi_i^-)
\end{aligned}
$$

$$\begin{aligned}
&- \sum_{i=n+1}^{m} \mu_i^+(z_i^+ + \xi_i^+) - \sum_{i=n+1}^{m}\sum_{j=n+1}^{m} \mu_i^+(\mu_j^+ - \mu_j^-)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle - \sum_{i=n+1}^{m} \gamma_i^+\xi_i^+ \\
&\overset{(B.11)}{=} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i=n+1}^{m}\sum_{j=1}^{n}\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \alpha_j y_j(\mu_i^+ - \mu_i^-) \\
&+ \frac{1}{2}\sum_{i=n+1}^{m}\sum_{j=n+1}^{m} (\mu_i^+ - \mu_i^-)(\mu_j^+ - \mu_j^-)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \widetilde{C}\sum_{i=n+1}^{m} (\xi_i^- + \xi_i^+) \\
&+ \sum_{i=n+1}^{m}\sum_{j=n+1}^{m} \mu_i^-(\mu_j^+ - \mu_j^-)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i=n+1}^{m} \mu_i^-(z_i^- - \xi_i^-) - \sum_{i=n+1}^{m} \gamma_i^-\xi_i^- \\
&- \sum_{i=n+1}^{m} \mu_i^+(z_i^+ + \xi_i^+) - \sum_{i=n+1}^{m}\sum_{j=n+1}^{m} \mu_i^+(\mu_j^+ - \mu_j^-)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle - \sum_{i=n+1}^{m} \gamma_i^+\xi_i^+ + \sum_{i=1}^{n}\alpha_i \\
&\overset{(B.12,B.13)}{=} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i=n+1}^{m}\sum_{j=1}^{n}\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \alpha_j y_j(\mu_i^+ - \mu_i^-) \\
&+ \frac{1}{2}\sum_{i=n+1}^{m}\sum_{j=n+1}^{m} (\mu_i^+ - \mu_i^-)(\mu_j^+ - \mu_j^-)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i=n+1}^{m}\sum_{j=n+1}^{m} \mu_i^-(\mu_j^+ - \mu_j^-)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \\
&- \sum_{i=n+1}^{m}\sum_{j=n+1}^{m} \mu_i^+(\mu_j^+ - \mu_j^-)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i=1}^{n}\alpha_i - \sum_{i=n+1}^{m} \mu_i^+ z_i^+ + \sum_{i=n+1}^{m} \mu_i^- z_i^-
\end{aligned}$$

Simplifying we obtain the function to be maximized:

$$\begin{aligned}
R(\boldsymbol{\alpha}, \boldsymbol{\mu}^+, \boldsymbol{\mu}^-) = &-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i=n+1}^{m}\sum_{j=1}^{n}\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \alpha_j y_j(\mu_i^+ - \mu_i^-) \\
&- \frac{1}{2}\sum_{i=n+1}^{m}\sum_{j=n+1}^{m} (\mu_i^+ - \mu_i^-)(\mu_j^+ - \mu_j^-)\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle + \sum_{i=1}^{n}\alpha_i - \sum_{i=n+1}^{m} \mu_i^+ z_i^+ + \sum_{i=n+1}^{m} \mu_i^- z_i^-
\end{aligned} \tag{B.14}$$

Applying the kernel trick, the function (B.14) to be maximized can be formulated as:

$$\begin{aligned}
R(\boldsymbol{\alpha}, \boldsymbol{\mu}^+, \boldsymbol{\mu}^-) = &-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=n+1}^{m}\sum_{j=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_j) \alpha_j y_j(\mu_i^+ - \mu_i^-) \\
&- \frac{1}{2}\sum_{i=n+1}^{m}\sum_{j=n+1}^{m} (\mu_i^+ - \mu_i^-)(\mu_j^+ - \mu_j^-)k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{n}\alpha_i - \sum_{i=n+1}^{m} \mu_i^+ z_i^+ + \sum_{i=n+1}^{m} \mu_i^- z_i^-
\end{aligned}$$

B.3 Support vector machines using local invariances

The initial expression as expressed by Lee et al. (2006), is

$$\rho_1 \|g\|^2 + \rho_2 \sum_{i=l+1}^{n} l_1(\mathrm{L}_i(g)) + \sum_{i=1}^{l} l_2(y_i, g(\boldsymbol{x}_i)))$$

Given that l_1 is the ϵ-insensitive loss and that l_2 is hinge-loss and g is the target function, we can express the function to be minimized as:

$$\begin{array}{lll} \underset{g,b}{\text{minimize}} & \rho_1 \|g\|^2 + \rho_2 \sum_{i=l+1}^{n} (\xi_i + \xi_i^*) + \sum_{i=1}^{l} \gamma_i & \\ \text{subject to} & -\langle g, z_i \rangle - b \leq \epsilon + \xi_i, & i = l+1, \dots, n, \\ & \langle g, z_i \rangle + b \leq \epsilon + \xi_i^*, & i = l+1, \dots, n, \\ & \xi_i \geq 0, & i = l+1, \dots, n, \\ & \xi_i^* \geq 0, & i = l+1, \dots, n, \\ & y_i(\langle g, \phi(\boldsymbol{x}_i) \rangle + b) \geq 1 - \gamma_i, & i = 1, \dots, l, \\ & \gamma_i \geq 0, & i = 1, \dots, l \end{array}$$

Then, we can construct the Lagrangian:

$$\begin{aligned} L(g, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\eta}, \boldsymbol{\eta}^*) = {} & \rho_1 \|g\|^2 + \rho_2 \sum_{i=l+1}^{n} (\xi_i + \xi_i^*) + \sum_{i=1}^{l} \gamma_i - \sum_{i=l+1}^{n} (\eta_i \xi_i + \eta_i^* \xi_i^*) \\ & - \sum_{i=l+1}^{n} \alpha_i (\epsilon + \xi_i + \langle g, z_i \rangle + b) \\ & - \sum_{i=l+1}^{n} \alpha_i^* (\epsilon + \xi_i^* - \langle g, z_i \rangle - b) \\ & + \sum_{i=1}^{l} \beta_i^* \left[1 - y_i(\langle g, \phi(\boldsymbol{x}_i) \rangle + b) - \gamma_i\right] - \sum_{i=1}^{l} \beta_i \gamma_i \end{aligned} \tag{B.15}$$

To minimize (B.15) we take the functional with respect g. Suppose g^* is the minimizer of (B.15). For any, $g \in \mathcal{H}$, let $g = g^* + \delta h$ where $\delta \in \mathbb{R}$ and $h \in \mathcal{H}$. Then,

$$
\begin{aligned}
L(g, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\eta}, \boldsymbol{\eta}^*) &= \rho_1 \|g^* + \delta h\|^2 + \rho_2 \sum_{i=l+1}^{n} (\xi_i + \xi_i^*) + \sum_{i=1}^{l} \gamma_i - \sum_{i=l+1}^{n} (\eta_i \xi_i + \eta_i^* \xi_i^*) \\
&- \sum_{i=l+1}^{n} \alpha_i (\epsilon + \xi_i + \langle g^* + \delta h, z_i \rangle + b) \\
&- \sum_{i=l+1}^{n} \alpha_i^* (\epsilon + \xi_i^* - \langle g^* + \delta h, z_i \rangle - b) \\
&+ \sum_{i=1}^{l} \beta_i^* \left[1 - y_i (\langle g^* + \delta h, \phi(\boldsymbol{x}_i) \rangle + b) - \gamma_i\right] - \sum_{i=1}^{l} \beta_i \gamma_i \\
&= \rho_1 \langle g^* + \delta h, g^* + \delta h \rangle + \rho_2 \sum_{i=l+1}^{n} (\xi_i + \xi_i^*) + \sum_{i=1}^{l} \gamma_i \\
&- \sum_{i=l+1}^{n} (\eta_i \xi_i + \eta_i^* \xi_i^*) \\
&- \sum_{i=l+1}^{n} \alpha_i (\epsilon + \xi_i + \langle g^*, z_i \rangle + \delta \langle h, z_i \rangle + b) \\
&- \sum_{i=l+1}^{n} \alpha_i^* (\epsilon + \xi_i^* - \langle g^*, z_i \rangle - \delta \langle h, z_i \rangle - b) \\
&+ \sum_{i=1}^{l} \beta_i^* \left(1 - y_i \langle g^*, \phi(\boldsymbol{x}_i) \rangle - y_i \delta \langle h, \phi(\boldsymbol{x}_i)) \rangle - b - \gamma_i\right) - \sum_{i=1}^{l} \beta_i \gamma_i \\
&= \rho_1 (\langle g^*, g^* \rangle + 2\delta \langle g^*, h \rangle + \delta^2 \langle h, h \rangle) + \rho_2 \sum_{i=l+1}^{n} (\xi_i + \xi_i^*) \\
&+ \sum_{i=1}^{l} \gamma_i - \sum_{i=l+1}^{n} (\eta_i \xi_i + \eta_i^* \xi_i^*) \\
&- \sum_{i=l+1}^{n} \alpha_i (\epsilon + \xi_i + \langle g^*, z_i \rangle + \delta \langle h, z_i \rangle + b) \\
&- \sum_{i=l+1}^{n} \alpha_i^* (\epsilon + \xi_i^* - \langle g^*, z_i \rangle - \delta \langle h, z_i \rangle - b) \\
&+ \sum_{i=1}^{l} \beta_i^* \left(1 - y_i \langle g^*, \phi(\boldsymbol{x}_i) \rangle - y_i \delta \langle h, \phi(\boldsymbol{x}_i)) \rangle - b - \gamma_i\right) - \sum_{i=1}^{l} \beta_i \gamma_i
\end{aligned}
\tag{B.16}
$$

Take derivative with respect to δ,

$$
\frac{\partial L}{\partial \delta} = 2\rho_1 \langle g^*, h \rangle + 2\delta \rho_1 \langle h, h \rangle - \sum_{i=l+1}^{n} \alpha_i \langle h, z_i \rangle - \sum_{i=l+1}^{n} \alpha_i^* (-\langle h, z_i \rangle) + \sum_{i=1}^{l} \beta_i^* (-y_i \langle h, \phi(\boldsymbol{x}_i) \rangle)
$$

Note that $\frac{\partial L}{\partial \delta} = 0$. Then,

$$
2\rho_1 \langle g^*, h \rangle - \sum_{i=l+1}^{n} \alpha_i \langle h, z_i \rangle + \sum_{i=l+1}^{n} \alpha_i^* \langle h, z_i \rangle - \sum_{i=1}^{l} \beta_i^* y_i \langle h, \phi(\boldsymbol{x}_i) \rangle = 0
$$

This equation holds for any $h \in \mathcal{H}$. Specifically, setting $h = k(., \boldsymbol{x})$ gives,

$$2\rho_1 \langle g^*, k(\cdot, \boldsymbol{x}) \rangle - \sum_{i=l+1}^{n} \alpha_i \langle k(\cdot, \boldsymbol{x}), z_i \rangle + \sum_{i=l+1}^{n} \alpha_i^* \langle k(\cdot, \boldsymbol{x}), z_i \rangle - \sum_{i=1}^{l} \beta_i^* y_i \langle k(\cdot, \boldsymbol{x}), k(\cdot, \boldsymbol{x}_i) \rangle = 0$$

$$\Rightarrow 2\rho_1 g^*(\boldsymbol{x}) - \sum_{i=l+1}^{n} \alpha_i z_i(\boldsymbol{x}) + \sum_{i=l+1}^{n} \alpha_i^* z_i(\boldsymbol{x}) - \sum_{i=1}^{l} \beta_i^* y_i k(\boldsymbol{x}, \boldsymbol{x}_i) = 0$$

Thus

$$g^*(\boldsymbol{x}) = \frac{1}{2\rho_1} \left(\sum_{i=l+1}^{n} \alpha_i z_i(\boldsymbol{x}) - \sum_{i=l+1}^{n} \alpha_i^* z_i(\boldsymbol{x}) + \sum_{i=1}^{l} \beta_i^* y_i k(\boldsymbol{x}, \boldsymbol{x}_i) \right)$$

And

$$g^*(\cdot) = \frac{1}{2\rho_1} \left(\sum_{i=l+1}^{n} \alpha_i z_i(\cdot)(\alpha_i - \alpha_i^*) + \sum_{i=1}^{l} \beta_i^* y_i k(\cdot, \boldsymbol{x}_i) \right)$$

Together with the other derivatives:

$$g^*(\cdot) = \frac{1}{2\rho_1} \left(\sum_{i=l+1}^{n} \alpha_i z_i(\cdot)(\alpha_i - \alpha_i^*) + \sum_{i=1}^{l} \beta_i^* y_i k(\cdot, \boldsymbol{x}_i) \right) \tag{B.17}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \implies \rho_2 - \eta_i - \alpha_i = 0 \tag{B.18}$$

$$\frac{\partial L}{\partial \xi_i^*} = 0 \implies \rho_2 - \eta_i^* - \alpha_i^* = 0 \tag{B.19}$$

$$\frac{\partial L}{\partial \gamma_i} = 0 \implies -\beta_i^* - \beta_i = 0 \tag{B.20}$$

$$\frac{\partial L}{\partial b} = 0 \implies -\sum_{i=1}^{l} \beta_i^* y_i = 0 \tag{B.21}$$

Developing the expression in (B.15) we obtain:

$$\begin{aligned} L(g, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\eta}, \boldsymbol{\eta}^*) = & \rho_1 \|g^*\|^2 + \rho_2 \sum_{i=l+1}^{n} (\xi_i + \xi_i^*) + \sum_{i=1}^{l} \gamma_i - \sum_{i=l+1}^{n} (\eta_i \xi_i + \eta_i^* \xi_i^*) \\ & - \sum_{i=l+1}^{n} \alpha_i \epsilon - \sum_{i=l+1}^{n} \alpha_i \xi_i - \sum_{i=l+1}^{n} \alpha_i \langle g^*, z_i \rangle \\ & - \sum_{i=l+1}^{n} \alpha_i^* \epsilon - \sum_{i=l+1}^{n} \alpha_i^* \xi_i^* + \sum_{i=l+1}^{n} \alpha_i^* \langle g^*, z_i \rangle \\ & + \sum_{i=1}^{l} \beta_i^* - \sum_{i=1}^{l} \beta_i^* y_i \langle g^*, \boldsymbol{x}_i \rangle - \sum_{i=1}^{l} \beta_i^* y_i b - \sum_{i=1}^{l} \beta_i^* \gamma_i - \sum_{i=1}^{l} \beta_i \gamma_i \end{aligned} \tag{B.22}$$

Substituting (B.18), (B.19), (B.20) and (B.21) in (B.22) we derive the following expression:

$$\begin{aligned} L(g, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\eta}, \boldsymbol{\eta}^*) = & \rho_1 \|g^*\|^2 - \sum_{i=l+1}^{n} \alpha_i \epsilon - \sum_{i=l+1}^{n} \alpha_i \langle g^*, z_i \rangle - \sum_{i=l+1}^{n} \alpha_i^* \epsilon \\ & + \sum_{i=l+1}^{n} \alpha_i^* \langle g^*, z_i \rangle + \sum_{i=1}^{l} \beta_i^* - \sum_{i=1}^{l} \beta_i^* y_i \langle g^*, \boldsymbol{x}_i \rangle \end{aligned} \tag{B.23}$$

Substituting (B.17) in (B.23):

$$\begin{aligned}
L(g,b,\boldsymbol{\xi},\boldsymbol{\xi}^*,\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{\beta}^*,\boldsymbol{\alpha},\boldsymbol{\alpha}^*,\boldsymbol{\eta},\boldsymbol{\eta}^*) = {} & \frac{1}{2}\left[\sum_{i=1}^{l}\sum_{j=1}^{l}\beta_i^*\beta_j^* y_i y_j \langle \boldsymbol{x}_i,\boldsymbol{x}_j\rangle + \sum_{i=l+1}^{n}\sum_{j=l+1}^{n}\langle z_i,z_j\rangle(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\right. \\
& \left. +2\sum_{i=1}^{l}\sum_{j=l+1}^{n}\beta_i^* y_i z_j(\boldsymbol{x}_i)(\alpha_j-\alpha_j^*)\right] - \sum_{i=l+1}^{n}\alpha_i\epsilon \\
& -\frac{1}{2\rho_1}\left[\sum_{i=l+1}^{l+n}\sum_{j=1}^{l}\beta_j^* y_j z_i(\boldsymbol{x}_j)\alpha_i + \sum_{i=l+1}^{n}\sum_{j=l+1}^{n}\langle z_i,z_j\rangle(\alpha_j-\alpha_j^*)\alpha_i\right] \\
& +\frac{1}{2\rho_1}\left[\sum_{i=l+1}^{n}\sum_{j=1}^{l}\beta_j^* y_j z_i(\boldsymbol{x}_j)\alpha_i^* + \sum_{i=l+1}^{n}\sum_{j=l+1}^{n}\langle z_i,z_j\rangle(\alpha_j-\alpha_j^*)\alpha_i^*\right] \\
& -\frac{1}{2\rho_1}\left[\sum_{i=1}^{l}\sum_{j=1}^{l}\beta_i^*\beta_j^* y_i y_j \langle \boldsymbol{x}_i,\boldsymbol{x}_j\rangle + \sum_{i=1}^{l}\sum_{j=l+1}^{n}\beta_i^* y_i z_j(\boldsymbol{x}_i)(\alpha_j-\alpha_j^*)\right] \\
& -\sum_{i=1}^{l}\beta_i^* - \sum_{i=l+1}^{n}\alpha_i^*\epsilon
\end{aligned} \tag{B.24}$$

Assuming that $\rho_1 = \frac{1}{2}$ to be more similar to classical SVM the final expression to be maximized is:

$$\begin{aligned}
R(\boldsymbol{\alpha},\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*) = {} & -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\beta_i^*\beta_j^* y_i y_j \langle \boldsymbol{x}_i,\boldsymbol{x}_j\rangle - \frac{1}{2}\sum_{i=l+1}^{n}\sum_{j=l+1}^{n}\langle z_i,z_j\rangle(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*) \\
& -\sum_{i=1}^{l}\sum_{j=l+1}^{n}\beta_i^* y_i z_j(\boldsymbol{x}_i)(\alpha_j-\alpha_j^*) - \sum_{i=l+1}^{n}\epsilon(\alpha_i+\alpha_i^*) + \sum_{i=1}^{l}\beta_i^*
\end{aligned} \tag{B.25}$$

Applying the kernel trick and linear functional definition we obtain the final function to be maximized:

$$\begin{aligned}
R(\boldsymbol{\alpha},\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*) = {} & -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\beta_i^*\beta_j^* y_i y_j k(\boldsymbol{x}_i,\boldsymbol{x}_j) - \frac{1}{2}\sum_{i=l+1}^{n}\sum_{j=l+1}^{n}\langle z_i,z_j\rangle(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*) \\
& -\sum_{i=1}^{l}\sum_{j=l+1}^{n}\beta_i^* y_i z_j(\boldsymbol{x}_i)(\alpha_j-\alpha_j^*) - \sum_{i=l+1}^{n}\epsilon(\alpha_i+\alpha_i^*) + \sum_{i=1}^{l}\beta_i^*
\end{aligned} \tag{B.26}$$

Appendix C

Support Vector Machines for Survival Analysis Supplementary Simulation Results

C.1 Scenarios descriptives

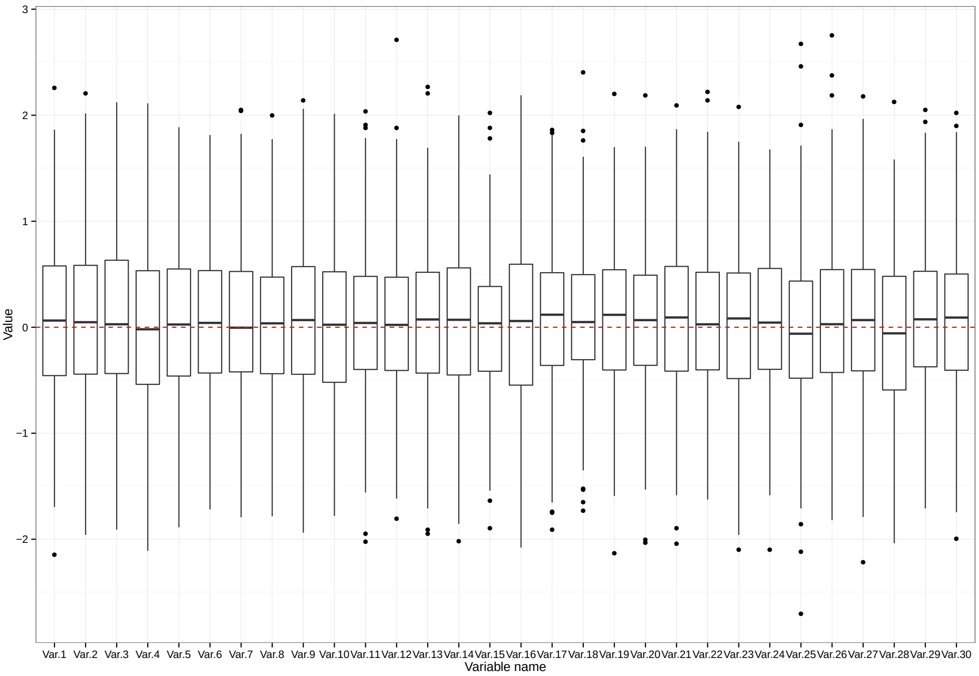


Figure C.1: Distribution of a simulated random dataset with 30 variables and 300 observations. Each variable distribution is represented by a boxplot. The dotted red line represents 0 mean.

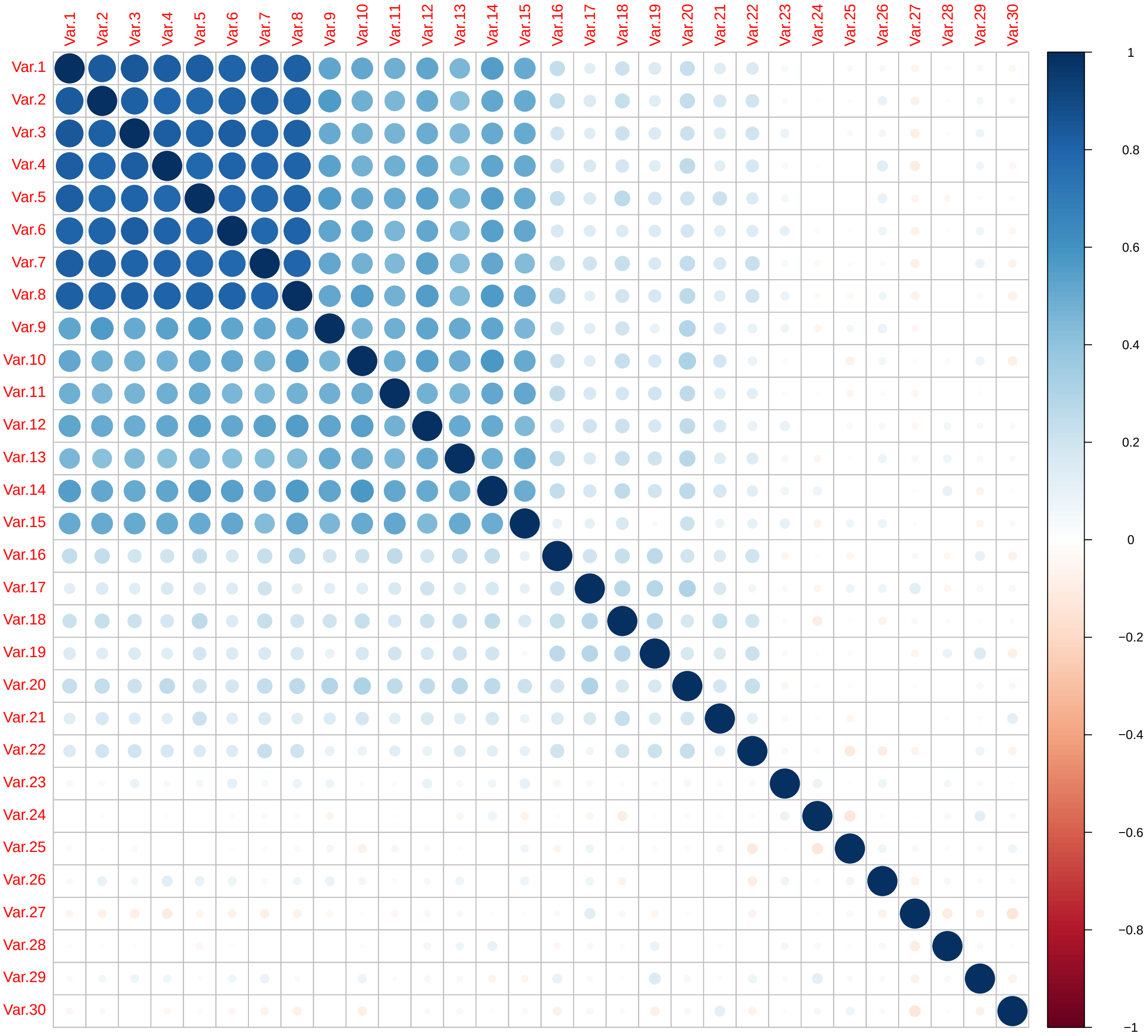


Figure C.2: Pearson correlation matrix of the dataset presented in Figure C.1 illustrated as a heatmap.

Table C.1: Summary of main descriptives for the proportional hazards scenarios, 30 variables and 300 observations in 100 simulated datasets. Schoenfeld residuals p value testing for proportionality of hazards is calculated and number of datasets in which proportionality of hazards does not hold is shown (Schoenfeld < 0.05). For censoring and events, median proportion and [Quartile 1; Quartile 3], is shown.

Theoretical censoring scenario		**Simulated scenario**		
Distribution	**Proportion (%)**	**Schoenfeld < 0.05**	**Censoring proportion (%)**	**Events proportion (%)**
Positive	10	6	12 [11 , 13]	40 [35,51]
	30	7	32 [28 , 32]	34 [27,40]
Negative	10	8	8 [7 , 12]	42 [37,51]
	30	5	30 [27 , 33]	38 [34,46]
Zero	10	5	9 [7 , 12]	41 [36,51]
	30	7	28 [27 , 32]	36 [30,42]

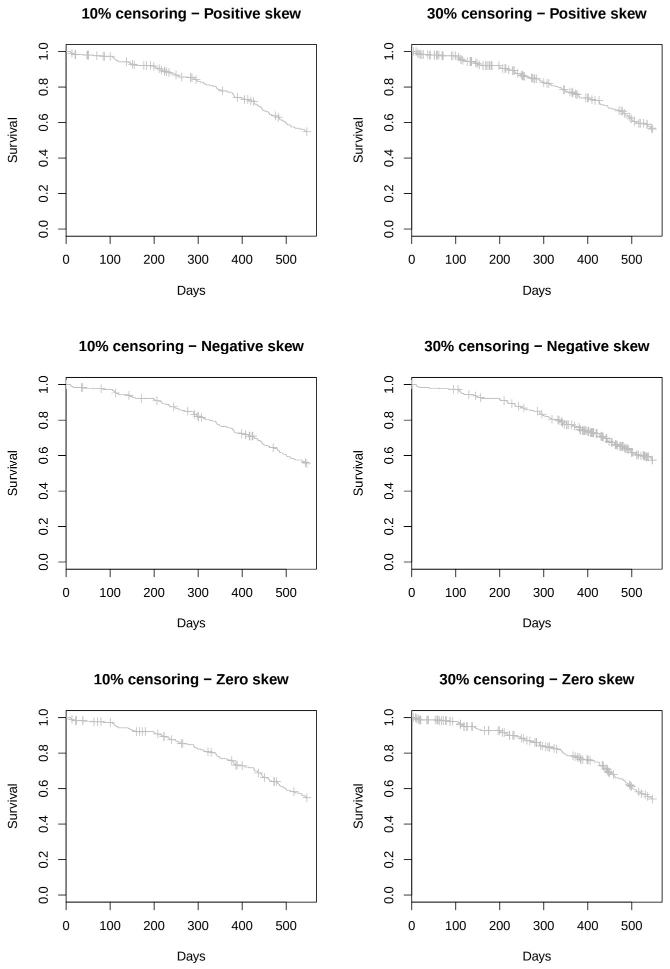


Figure C.3: Example of a simulated dataset for the proportional hazards scenario (composed by 30 variables and 300 observations). Censored observations are represented by + symbol.

Table C.2: Summary of main descriptives for the proportional hazards scenarios, 30 variables and 50 observations in 100 generated datasets. Schoenfeld residuals p value testing for proportionality of hazards is not calculated due to the fact that there are problems with Cox models convergence (low sample size). For censoring and events, median proportion and [Quartile 1; Quartile 3], is shown.

Theoretical censoring scenario		Simulated scenario		
Distribution	**Proportion (%)**	**Schoenfeld $<$ 0.05**	**Censoring proportion (%)**	**Events proportion (%)**
Positive	10	–	12 [10 , 12]	44 [38,46]
	30	–	32 [30 , 32]	36 [32,40]
Negative	10	–	8 [8 , 8]	42 [38,48]
	30	–	28 [28 , 30]	36 [32,38]
Zero	10	–	10 [8 , 10]	42 [38,47]
	30	–	28 [28 , 29]	36 [32,41]

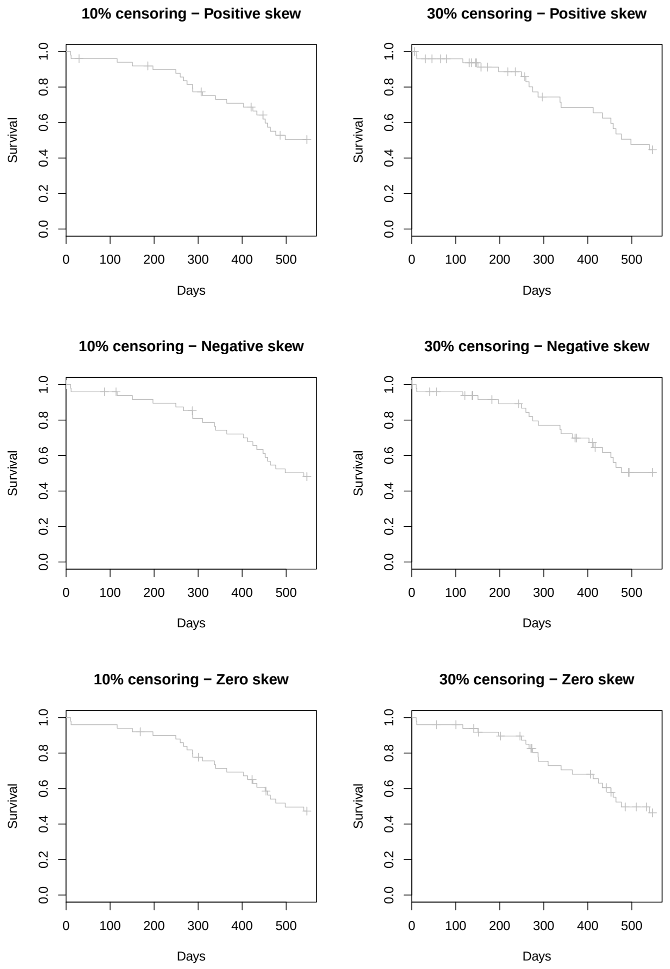


Figure C.4: Example of a simulated dataset for the proportional hazards scenario (composed 30 variables and 50 observations). Censored observations are represented by + symbol.

Table C.3: Summary of main descriptives for the non-proportional hazards scenarios, 30 variables and 300 observations in 100 generated datasets. Schoenfeld residuals p value testing for proportionality of hazards is also calculated and number of datasets in which proportionality of hazards does not hold is shown (Schoenfeld < 0.05). For censoring and events, median proportion and [Quartile 1; Quartile 3], is shown.

Theoretical censoring scenario		**Simulated scenario**		
Distribution	**Proportion (%)**	**Schoenfeld $<$ 0.05**	**Censoring proportion (%)**	**Events proportion (%)**
Positive	10	90	13 [12 , 13]	38 [37,39]
	30	90	32 [31 , 32]	33 [32,35]
Negative	10	89	8 [7 , 8]	40 [39,41]
	30	88	29 [28 , 31]	37 [36,38]
Zero	10	92	8 [8 , 9]	40 [38,41]
	30	90	28 [27 , 28]	34 [33,36]

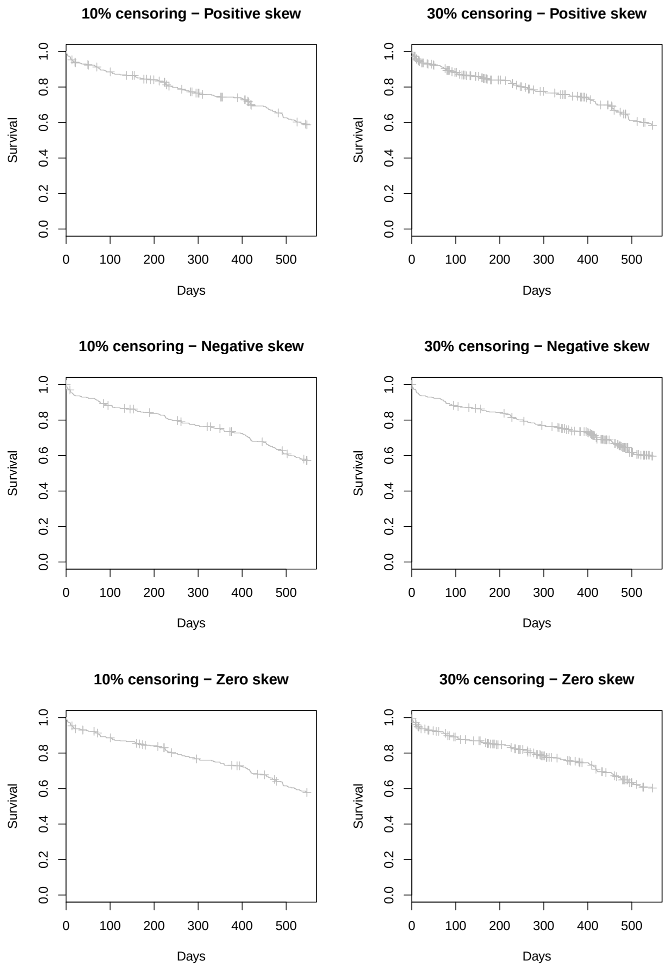


Figure C.5: Example of a simulated dataset for the non-proportional hazards scenario (composed by 30 variables and 300 observations). Censored observations are represented by + symbol.

Table C.4: Summary of main descriptives for the non-proportional hazards scenarios, 30 variables and 50 observations in 100 generated datasets. Schoenfeld residuals p value testing for proportionality of hazards are not calculated due to the fact that there are problems with Cox models convergence (low sample size). For censoring and events, median proportion and [Quartile 1; Quartile 3], is shown.

Theoretical censoring scenario		**Simulated scenario**		
Distribution	**Proportion (%)**	**Schoenfeld < 0.05**	**Censoring proportion (%)**	**Events proportion (%)**
Positive	10	–	12 [12 , 13]	42 [42,44]
	30	–	32 [32 , 32]	35 [33,38]
Negative	10	–	8 [8 , 10]	41 [39,44]
	30	–	28 [28 , 30]	38 [36,40]
Zero	10	–	10 [8 , 12]	42 [42,42]
	30	–	30 [28 , 30]	34 [32,36]

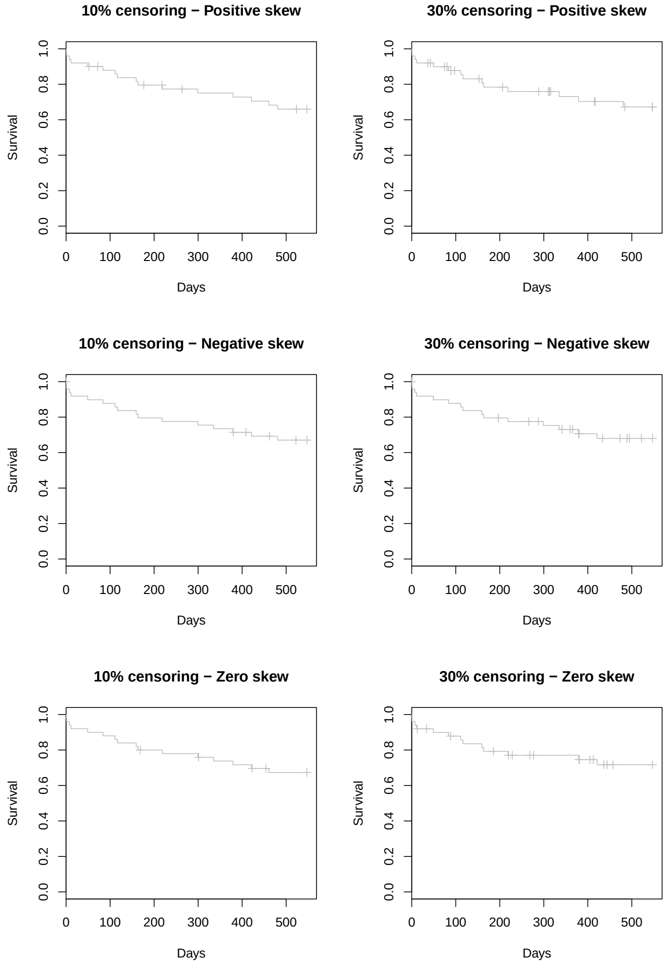


Figure C.6: Example of a simulated dataset for the non-proportional hazards scenario (composed by 30 variables and 50 observations). Censored observations are represented by + symbol.

C.2 Tables of simulations results

Table C.5: Proportional hazards, positive skew, 10% and 30% censoring and 300 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.89 (0.02)	0.77 (0.03)	0.49 (0.05)	0.96 (0.01)	0.89 (0.02)	0.78 (0.04)	0.50 (0.06)	0.96 (0.01)
Kernel Cox	0.80 (0.03)	0.59 (0.06)	0.27 (0.06)	0.86 (0.03)	0.79 (0.02)	0.58 (0.05)	0.25 (0.05)	0.86 (0.02)
wSVM-KM	0.75 (0.02)	0.45 (0.06)	0.15 (0.04)	0.87 (0.02)	0.67 (0.02)	0.37 (0.05)	0.11 (0.03)	0.85 (0.03)
wSVM-Prop	0.73 (0.02)	0.46 (0.06)	0.15 (0.04)	0.87 (0.02)	0.66 (0.02)	0.36 (0.04)	0.10 (0.03)	0.85 (0.03)
pSVM-linear-KM	0.87 (0.02)	0.72 (0.05)	0.44 (0.06)	0.95 (0.02)	0.86 (0.02)	0.71 (0.04)	0.42 (0.06)	0.94 (0.01)
pSVM-linear-prop	0.86 (0.02)	0.72 (0.05)	0.44 (0.06)	0.95 (0.02)	0.86 (0.02)	0.71 (0.04)	0.42 (0.06)	0.94 (0.01)
pSVM-radial-KM	0.79 (0.02)	0.57 (0.05)	0.25 (0.05)	0.87 (0.02)	0.79 (0.03)	0.58 (0.05)	0.27 (0.05)	0.86 (0.02)
pSVM-radial-prop	0.79 (0.02)	0.56 (0.05)	0.24 (0.05)	0.87 (0.02)	0.79 (0.03)	0.57 (0.05)	0.27 (0.05)	0.86 (0.02)
LUPI-linear-KM	0.78 (0.03)	0.56 (0.05)	0.27 (0.05)	0.84 (0.03)	0.78 (0.03)	0.54 (0.05)	0.26 (0.05)	0.84 (0.03)
LUPI-linear-prop	0.77 (0.03)	0.55 (0.05)	0.27 (0.05)	0.84 (0.03)	0.77 (0.03)	0.54 (0.05)	0.26 (0.05)	0.83 (0.03)
inSVM-gradient	0.83 (0.02)	0.66 (0.04)	0.35 (0.05)	0.92 (0.02)	0.80 (0.02)	0.59 (0.05)	0.27 (0.05)	0.88 (0.02)
inSVM-averaging	0.84 (0.02)	0.67 (0.05)	0.37 (0.05)	0.92 (0.02)	0.83 (0.02)	0.66 (0.05)	0.35 (0.06)	0.92 (0.02)

Table C.6: Proportional hazards, negative skew, 10% and 30% censoring and 300 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.89 (0.02)	0.78 (0.03)	0.50 (0.05)	0.96 (0.01)	0.89 (0.02)	0.79 (0.03)	0.51 (0.05)	0.96 (0.01)
Kernel Cox	0.81 (0.02)	0.61 (0.05)	0.29 (0.05)	0.87 (0.02)	0.81 (0.02)	0.61 (0.04)	0.28 (0.05)	0.87 (0.02)
wSVM-KM	0.74 (0.03)	0.48 (0.08)	0.17 (0.06)	0.86 (0.03)	0.71 (0.03)	0.43 (0.07)	0.14 (0.05)	0.87 (0.02)
wSVM-Prop	0.74 (0.03)	0.48 (0.08)	0.17 (0.06)	0.87 (0.03)	0.71 (0.03)	0.43 (0.07)	0.14 (0.05)	0.86 (0.02)
pSVM-linear-KM	0.86 (0.02)	0.72 (0.05)	0.44 (0.06)	0.94 (0.02)	0.87 (0.02)	0.73 (0.05)	0.44 (0.06)	0.94 (0.01)
pSVM-linear-prop	0.85 (0.03)	0.70 (0.05)	0.41 (0.06)	0.93 (0.02)	0.86 (0.03)	0.72 (0.05)	0.43 (0.07)	0.94 (0.02)
pSVM-radial-KM	0.78 (0.03)	0.55 (0.08)	0.22 (0.07)	0.87 (0.03)	0.79 (0.03)	0.59 (0.05)	0.30 (0.05)	0.86 (0.03)
pSVM-radial-prop	0.77 (0.03)	0.54 (0.07)	0.21 (0.07)	0.87 (0.02)	0.79 (0.03)	0.58 (0.05)	0.29 (0.05)	0.86 (0.03)
LUPI-linear-KM	0.76 (0.03)	0.54 (0.05)	0.29 (0.04)	0.83 (0.03)	0.77 (0.03)	0.55 (0.06)	0.27 (0.05)	0.84 (0.03)
LUPI-linear-prop	0.76 (0.03)	0.54 (0.05)	0.29 (0.04)	0.83 (0.03)	0.77 (0.03)	0.55 (0.06)	0.27 (0.05)	0.84 (0.03)
inSVM-gradient	0.83 (0.03)	0.66 (0.06)	0.35 (0.06)	0.92 (0.02)	0.80 (0.03)	0.61 (0.06)	0.30 (0.06)	0.89 (0.03)
inSVM-averaging	0.84 (0.02)	0.67 (0.05)	0.37 (0.06)	0.92 (0.02)	0.84 (0.03)	0.67 (0.05)	0.37 (0.06)	0.92 (0.02)

Table C.7: Non-proportional hazards, negative skew, 10% and 30% censoring and 300 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.71 (0.03)	0.39 (0.06)	0.10 (0.04)	0.77 (0.03)	0.70 (0.03)	0.38 (0.05)	0.09 (0.03)	0.76 (0.03)
Kernel Cox	0.67 (0.03)	0.33 (0.05)	0.1 (0.04)	0.71 (0.03)	0.68 (0.02)	0.33 (0.05)	0.08 (0.04)	0.7 (0.03)
wSVM-KM	0.65 (0.02)	0.24 (0.06)	0.01 (0.02)	0.71 (0.03)	0.63 (0.02)	0.19 (0.06)	0.01 (0.02)	0.70 (0.03)
wSVM-Prop	0.64 (0.02)	0.24 (0.06)	0.01 (0.02)	0.71 (0.03)	0.62 (0.02)	0.18 (0.06)	0.01 (0.02)	0.69 (0.02)
pSVM-linear-KM	0.69 (0.03)	0.38 (0.06)	0.13 (0.04)	0.76 (0.03)	0.68 (0.03)	0.36 (0.05)	0.13 (0.04)	0.75 (0.03)
pSVM-linear-prop	0.69 (0.03)	0.37 (0.06)	0.13 (0.04)	0.76 (0.03)	0.68 (0.03)	0.35 (0.06)	0.12 (0.04)	0.75 (0.03)
pSVM-radial-KM	0.66 (0.02)	0.29 (0.05)	0.04 (0.03)	0.71 (0.03)	0.65 (0.04)	0.31 (0.06)	0.20 (0.11)	0.71 (0.02)
pSVM-radial-prop	0.66 (0.02)	0.29 (0.05)	0.03 (0.03)	0.70 (0.03)	0.64 (0.04)	0.30 (0.06)	0.17 (0.11)	0.70 (0.02)
LUPI-linear-KM	0.66 (0.03)	0.33 (0.05)	0.15 (0.06)	0.70 (0.03)	0.66 (0.03)	0.33 (0.05)	0.13 (0.06)	0.70 (0.02)
LUPI-linear-prop	0.66 (0.03)	0.33 (0.05)	0.15 (0.06)	0.70 (0.03)	0.66 (0.03)	0.33 (0.05)	0.13 (0.06)	0.70 (0.02)
inSVM-gradient	0.69 (0.03)	0.37 (0.05)	0.11 (0.03)	0.76 (0.03)	0.67 (0.02)	0.34 (0.05)	0.13 (0.05)	0.72 (0.02)
inSVM-averaging	0.69 (0.03)	0.37 (0.05)	0.12 (0.03)	0.76 (0.03)	0.68 (0.03)	0.37 (0.05)	0.14 (0.04)	0.76 (0.03)

Table C.8: Non-proportional hazards, positive skew, 10% and 30% censoring and 300 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.71 (0.02)	0.39 (0.05)	0.10 (0.03)	0.77 (0.03)	0.71 (0.03)	0.39 (0.06)	0.10 (0.04)	0.76 (0.03)
Kernel Cox	0.68 (0.03)	0.34 (0.06)	0.07 (0.04)	0.70 (0.03)	0.67 (0.03)	0.33 (0.05)	0.09 (0.04)	0.70 (0.03)
wSVM-KM	0.65 (0.02)	0.20 (0.05)	0.01 (0.02)	0.69 (0.03)	0.61 (0.02)	0.16 (0.06)	0.01 (0.02)	0.69 (0.03)
wSVM-Prop	0.63 (0.02)	0.20 (0.06)	0.01 (0.02)	0.69 (0.03)	0.61 (0.02)	0.15 (0.06)	0.01 (0.02)	0.70 (0.03)
pSVM-linear-KM	0.70 (0.03)	0.38 (0.05)	0.11 (0.03)	0.76 (0.03)	0.69 (0.03)	0.36 (0.06)	0.13 (0.03)	0.75 (0.03)
pSVM-linear-prop	0.69 (0.02)	0.37 (0.05)	0.11 (0.03)	0.75 (0.03)	0.68 (0.03)	0.36 (0.06)	0.12 (0.04)	0.75 (0.03)
pSVM-radial-KM	0.66 (0.03)	0.28 (0.06)	0.03 (0.03)	0.70 (0.03)	0.67 (0.03)	0.32 (0.06)	0.14 (0.04)	0.71 (0.03)
pSVM-radial-prop	0.66 (0.03)	0.27 (0.06)	0.03 (0.03)	0.69 (0.03)	0.65 (0.03)	0.31 (0.05)	0.08 (0.05)	0.70 (0.03)
LUPI-linear-KM	0.67 (0.03)	0.33 (0.06)	0.09 (0.04)	0.70 (0.03)	0.66 (0.03)	0.32 (0.05)	0.13 (0.04)	0.70 (0.03)
LUPI-linear-prop	0.67 (0.03)	0.33 (0.06)	0.09 (0.04)	0.70 (0.03)	0.66 (0.03)	0.32 (0.05)	0.13 (0.04)	0.70 (0.03)
inSVM-gradient	0.71 (0.02)	0.37 (0.05)	0.09 (0.03)	0.75 (0.03)	0.69 (0.02)	0.36 (0.04)	0.11 (0.03)	0.75 (0.03)
inSVM-averaging	0.71 (0.02)	0.37 (0.05)	0.1 (0.03)	0.75 (0.03)	0.69 (0.03)	0.38 (0.05)	0.13 (0.03)	0.76 (0.03)

Table C.9: Proportional hazards, negative skew, 10% and 30% censoring and 50 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.57 (0.11)	0.34 (0.14)	0.13 (0.07)	0.56 (0.10)	0.50 (0.07)	0.18 (0.13)	0.21 (0.12)	0.53 (0.05)
Kernel Cox	0.72 (0.08)	0.44 (0.15)	0.17 (0.11)	0.77 (0.08)	0.73 (0.06)	0.45 (0.13)	0.16 (0.11)	0.77 (0.07)
wSVM-KM	0.62 (0.03)	0.20 (0.13)	0.03 (0.05)	0.75 (0.08)	0.56 (0.04)	0.12 (0.11)	0.02 (0.01)	0.75 (0.07)
wSVM-Prop	0.61 (0.03)	0.19 (0.13)	0.03 (0.05)	0.75 (0.08)	0.55 (0.04)	0.12 (0.1)	0.02 (0.01)	0.74 (0.07)
pSVM-linear-KM	0.76 (0.07)	0.53 (0.13)	0.27 (0.11)	0.85 (0.06)	0.73 (0.07)	0.47 (0.13)	0.20 (0.1)	0.82 (0.07)
pSVM-linear-prop	0.75 (0.07)	0.49 (0.14)	0.24 (0.11)	0.84 (0.07)	0.73 (0.07)	0.46 (0.14)	0.20 (0.11)	0.81 (0.08)
pSVM-radial-KM	0.65 (0.05)	0.28 (0.17)	0.11 (0.13)	0.76 (0.08)	0.67 (0.07)	0.35 (0.15)	0.39 (0.27)	0.75 (0.07)
pSVM-radial-prop	0.64 (0.06)	0.24 (0.18)	0.09 (0.15)	0.75 (0.08)	0.63 (0.07)	0.31 (0.14)	0.20 (0.33)	0.75 (0.07)
LUPI-linear-KM	0.69 (0.09)	0.4 (0.16)	0.22 (0.14)	0.74 (0.09)	0.70 (0.08)	0.41 (0.15)	0.19 (0.12)	0.73 (0.08)
LUPI-linear-prop	0.69 (0.09)	0.4 (0.16)	0.22 (0.14)	0.74 (0.09)	0.70 (0.08)	0.40 (0.15)	0.19 (0.12)	0.73 (0.08)
inSVM-gradient	0.74 (0.06)	0.46 (0.14)	0.17 (0.11)	0.79 (0.07)	0.74 (0.07)	0.48 (0.14)	0.23 (0.12)	0.82 (0.06)
inSVM-averaging	0.75 (0.07)	0.49 (0.14)	0.19 (0.11)	0.83 (0.07)	0.74 (0.06)	0.49 (0.13)	0.23 (0.12)	0.83 (0.06)

Table C.10: Proportional hazards, positive skew, 10% and 30% censoring and 50 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.67 (0.14)	0.47 (0.18)	0.24 (0.14)	0.68 (0.14)	0.52 (0.10)	0.28 (0.14)	0.07 (0.08)	0.55 (0.09)
Kernel Cox	0.74 (0.06)	0.48 (0.13)	0.18 (0.1)	0.79 (0.07)	0.73 (0.07)	0.45 (0.16)	0.17 (0.11)	0.77 (0.08)
wSVM-KM	0.62 (0.06)	0.28 (0.16)	0.09 (0.08)	0.79 (0.07)	0.54 (0.05)	0.15 (0.09)	0.02 (0)	0.78 (0.07)
wSVM-Prop	0.60 (0.06)	0.27 (0.16)	0.08 (0.08)	0.79 (0.07)	0.54 (0.05)	0.14 (0.08)	0.02 (0.01)	0.77 (0.07)
pSVM-linear-KM	0.77 (0.06)	0.54 (0.12)	0.26 (0.11)	0.86 (0.06)	0.74 (0.07)	0.47 (0.14)	0.21 (0.11)	0.82 (0.07)
pSVM-linear-prop	0.77 (0.07)	0.54 (0.13)	0.26 (0.12)	0.85 (0.06)	0.74 (0.07)	0.47 (0.14)	0.21 (0.11)	0.82 (0.07)
pSVM-radial-KM	0.67 (0.06)	0.33 (0.17)	0.23 (0.25)	0.79 (0.07)	0.64 (0.07)	0.30 (0.15)	0.40 (0.41)	0.76 (0.08)
pSVM-radial-prop	0.66 (0.07)	0.31 (0.17)	0.22 (0.27)	0.79 (0.07)	0.64 (0.08)	0.29 (0.17)	0.35 (0.41)	0.76 (0.08)
LUPI-linear-KM	0.73 (0.06)	0.47 (0.12)	0.22 (0.1)	0.79 (0.07)	0.71 (0.06)	0.42 (0.13)	0.21 (0.11)	0.76 (0.07)
LUPI-linear-prop	0.73 (0.06)	0.47 (0.12)	0.22 (0.1)	0.79 (0.07)	0.71 (0.06)	0.42 (0.13)	0.21 (0.11)	0.76 (0.07)
inSVM-gradient	0.76 (0.06)	0.52 (0.12)	0.22 (0.10)	0.85 (0.06)	0.72 (0.07)	0.45 (0.14)	0.22 (0.12)	0.79 (0.07)
inSVM-averaging	0.76 (0.07)	0.51 (0.14)	0.22 (0.11)	0.84 (0.06)	0.75 (0.06)	0.50 (0.11)	0.24 (0.10)	0.83 (0.06)

Table C.11: Non-Proportional hazards, negative skew, 10% and 30% censoring and 50 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.58 (0.05)	0.14 (0.17)	0.01 (0.08)	0.53 (0.07)	0.58 (0.07)	0.08 (0.17)	0.07 (0.08)	0.54 (0.06)
Kernel Cox	0.63 (0.06)	0.23 (0.13)	0.06 (0.08)	0.63 (0.07)	0.63 (0.07)	0.23 (0.14)	0.05 (0.07)	0.64 (0.08)
wSVM-KM	0.62 (0.04)	0.14 (0.16)	0.01 (0.06)	0.63 (0.08)	0.59 (0.04)	0.12 (0.11)	0.01 (0.06)	0.63 (0.08)
wSVM-Prop	0.60 (0.03)	0.14 (0.15)	0.01 (0.06)	0.63 (0.08)	0.59 (0.04)	0.12 (0.11)	0.01 (0.06)	0.63 (0.08)
pSVM-linear-KM	0.64 (0.07)	0.26 (0.14)	0.08 (0.06)	0.68 (0.08)	0.62 (0.08)	0.22 (0.15)	0.11 (0.08)	0.64 (0.09)
pSVM-linear-prop	0.62 (0.06)	0.23 (0.14)	0.08 (0.08)	0.66 (0.08)	0.61 (0.07)	0.22 (0.14)	0.11 (0.08)	0.64 (0.09)
pSVM-radial-KM	0.63 (0.04)	0.17 (0.15)	0.01 (0.05)	0.63 (0.08)	0.61 (0.07)	0.16 (0.16)	0.22 (0.31)	0.63 (0.08)
pSVM-radial-prop	0.61 (0.04)	0.16 (0.15)	0.01 (0.07)	0.63 (0.08)	0.59 (0.09)	0.13 (0.16)	0.06 (0.35)	0.63 (0.08)
LUPI-linear-KM	0.63 (0.07)	0.23 (0.15)	0.07 (0.11)	0.64 (0.09)	0.62 (0.07)	0.22 (0.14)	0.09 (0.10)	0.64 (0.08)
LUPI-linear-prop	0.63 (0.07)	0.23 (0.15)	0.07 (0.11)	0.64 (0.09)	0.62 (0.07)	0.22 (0.14)	0.09 (0.10)	0.64 (0.08)
inSVM-gradient	0.65 (0.06)	0.28 (0.14)	0.07 (0.08)	0.68 (0.09)	0.63 (0.07)	0.23 (0.14)	0.07 (0.08)	0.64 (0.08)
inSVM-averaging	0.66 (0.06)	0.28 (0.13)	0.07 (0.08)	0.67 (0.08)	0.64 (0.07)	0.26 (0.14)	0.11 (0.09)	0.68 (0.07)

Table C.12: Non-Proportional hazards, positive skew, 10% and 30% censoring and 50 observations scenarios results. Mean (standard deviation) is shown.

	10% Censoring				30% Censoring			
Method	**Accuracy**	**Matthews**	**NMI**	**AUC**	**Accuracy**	**Matthews**	**NMI**	**AUC**
Cox model	0.56 (0.04)	0.08 (0.11)	0.02 (0.07)	0.52 (0.05)	0.57 (0.05)	0.12 (0.13)	0.07 (0.08)	0.53 (0.06)
Kernel Cox	0.60 (0.07)	0.19 (0.14)	0.07 (0.07)	0.62 (0.08)	0.61 (0.07)	0.20 (0.15)	0.08 (0.09)	0.64 (0.08)
wSVM-KM	0.59 (0.04)	0.14 (0.16)	0.01 (0.06)	0.61 (0.09)	0.58 (0.04)	0.08 (0.14)	0.01 (0.06)	0.63 (0.08)
wSVM-Prop	0.57 (0.04)	0.14 (0.15)	0.01 (0.06)	0.61 (0.09)	0.59 (0.04)	0.08 (0.14)	0.01 (0.06)	0.63 (0.08)
pSVM-linear-KM	0.62 (0.08)	0.21 (0.17)	0.07 (0.08)	0.65 (0.10)	0.61 (0.07)	0.20 (0.14)	0.08 (0.08)	0.64 (0.09)
pSVM-linear-prop	0.61 (0.08)	0.21 (0.17)	0.07 (0.08)	0.65 (0.10)	0.59 (0.07)	0.18 (0.14)	0.09 (0.08)	0.64 (0.09)
pSVM-radial-KM	0.63 (0.04)	0.11 (0.14)	0.01 (0.05)	0.62 (0.09)	0.59 (0.07)	0.12 (0.16)	0.11 (0.16)	0.63 (0.08)
pSVM-radial-prop	0.60 (0.05)	0.11 (0.15)	0.01 (0.07)	0.61 (0.09)	0.54 (0.09)	0.08 (0.15)	0.61 (0.54)	0.62 (0.08)
LUPI-linear-KM	0.63 (0.08)	0.22 (0.15)	0.13 (0.09)	0.62 (0.09)	0.63 (0.07)	0.23 (0.15)	0.11 (0.12)	0.63 (0.07)
LUPI-linear-prop	0.60 (0.08)	0.21 (0.15)	0.12 (0.09)	0.62 (0.09)	0.60 (0.07)	0.22 (0.15)	0.11 (0.12)	0.63 (0.07)
inSVM-gradient	0.64 (0.07)	0.25 (0.15)	0.06 (0.07)	0.67 (0.09)	0.63 (0.07)	0.25 (0.13)	0.11 (0.09)	0.67 (0.08)
inSVM-averaging	0.64 (0.07)	0.24 (0.15)	0.06 (0.07)	0.67 (0.09)	0.64 (0.07)	0.26 (0.14)	0.11 (0.08)	0.67 (0.08)

Appendix D

Relevance of Variables Supplementary Simulation Results

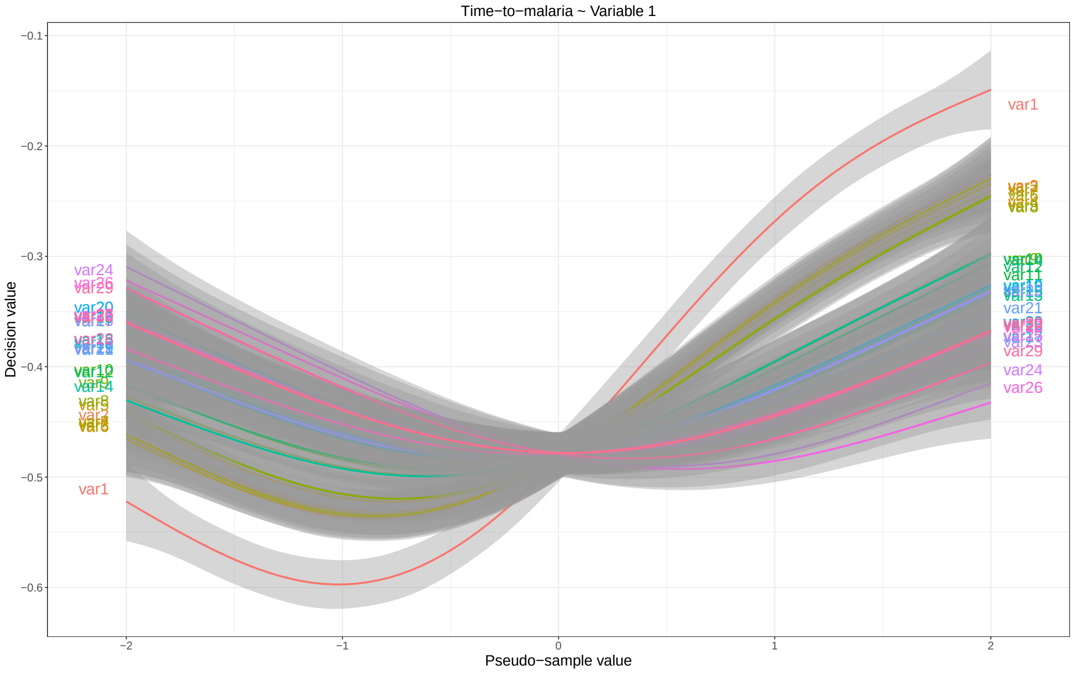


Figure D.1: Scenario 1 results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval.

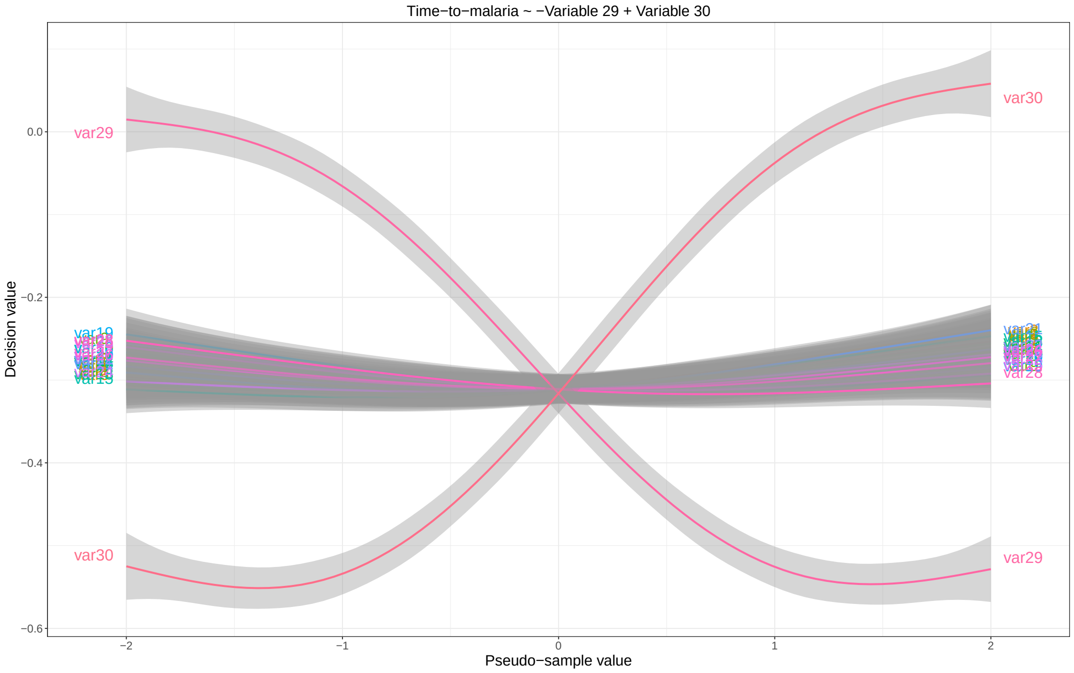


Figure D.2: Scenario 2 results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval.

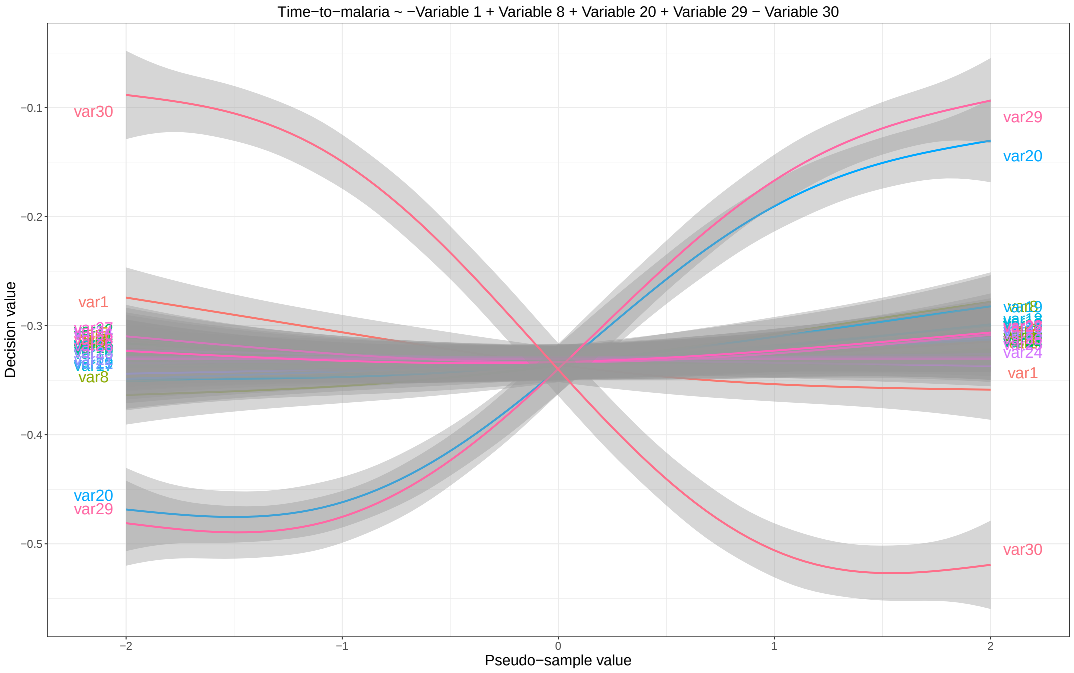


Figure D.3: Scenario 3 results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval.

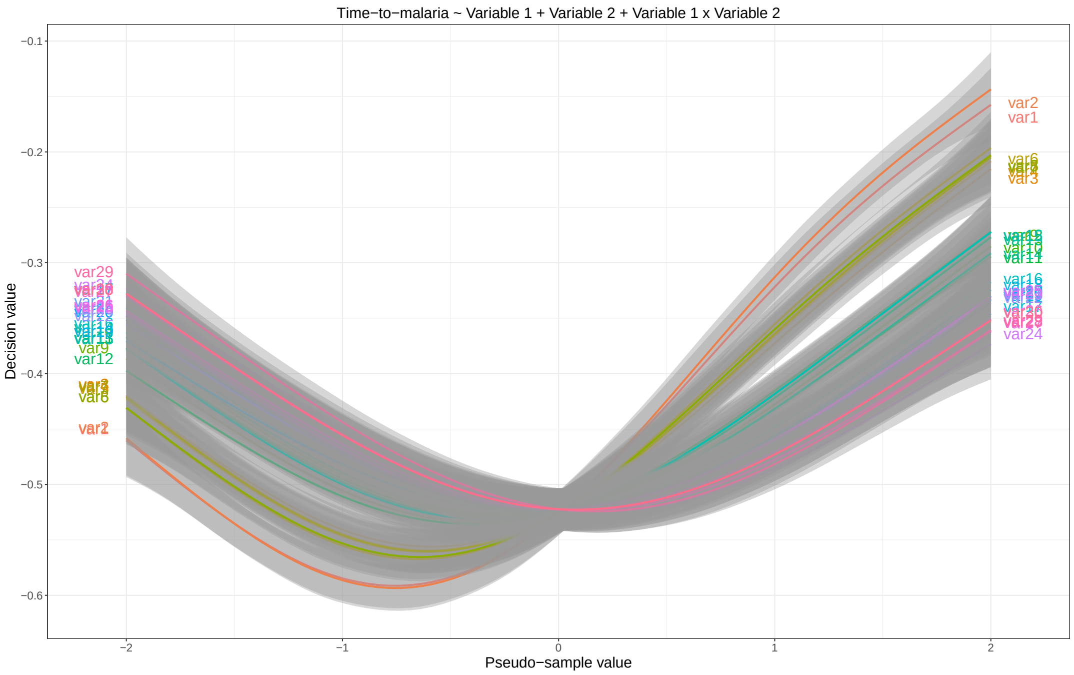


Figure D.4: Scenario 4 results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval.

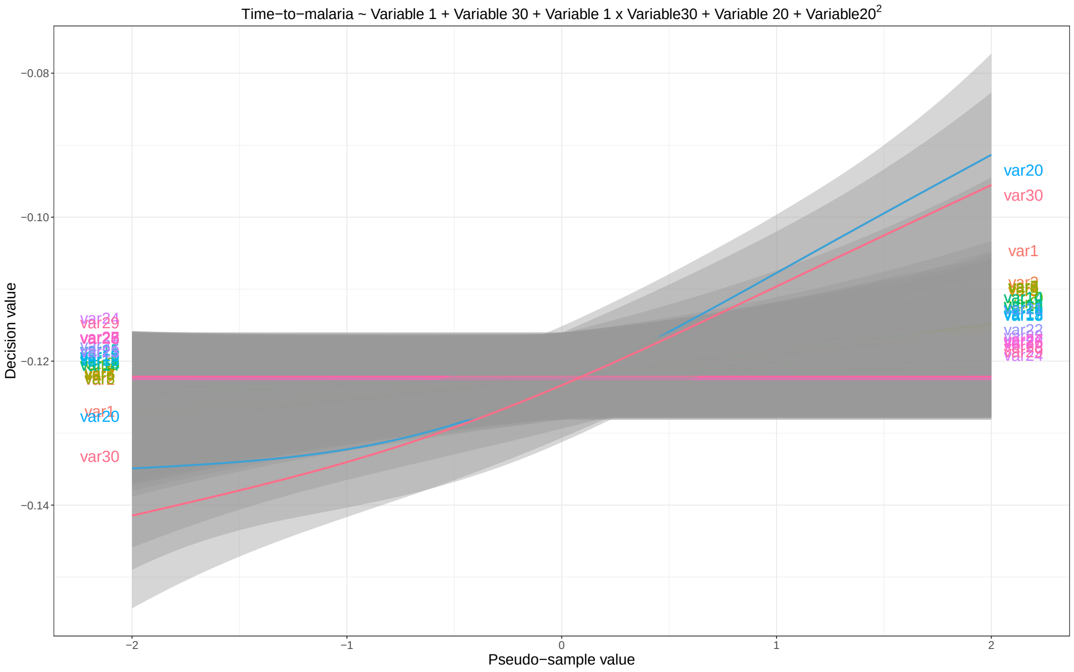


Figure D.5: Scenario 5 results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval.

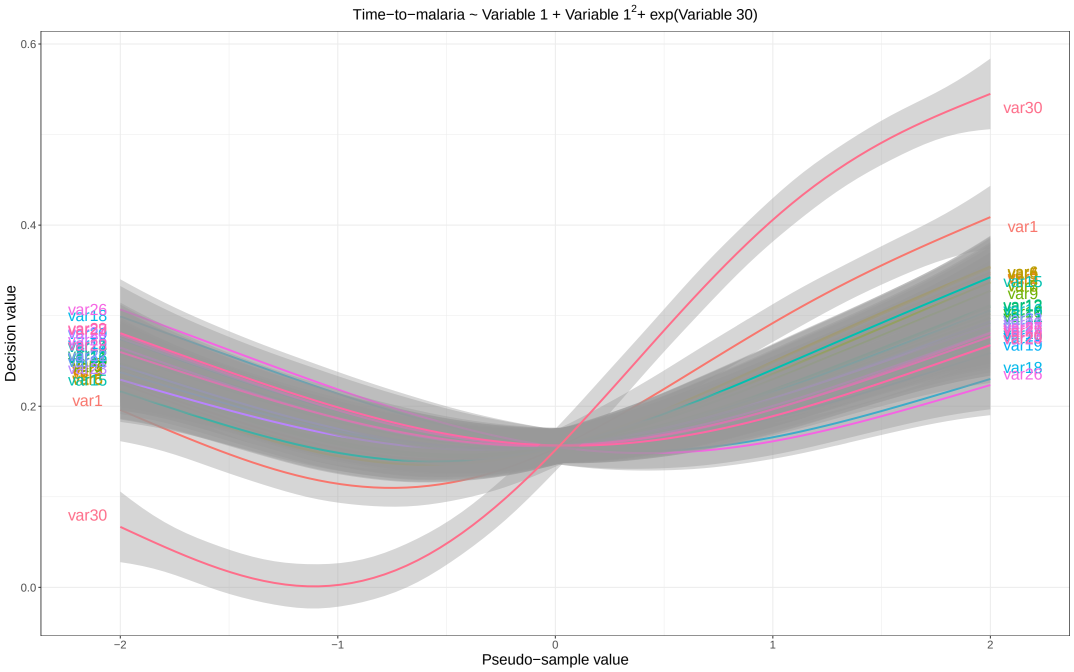


Figure D.6: Scenario 6 results for all 100 simulated datasets, all 30 variables and first iteration of the RFE-pseudo-samples algorithm. The pseudo-samples distribution for each variable is shown with a non-parametric local regression estimation (LOESS) with the corresponding 95% confidence interval.

Appendix E

Mal067 Supplementary Results

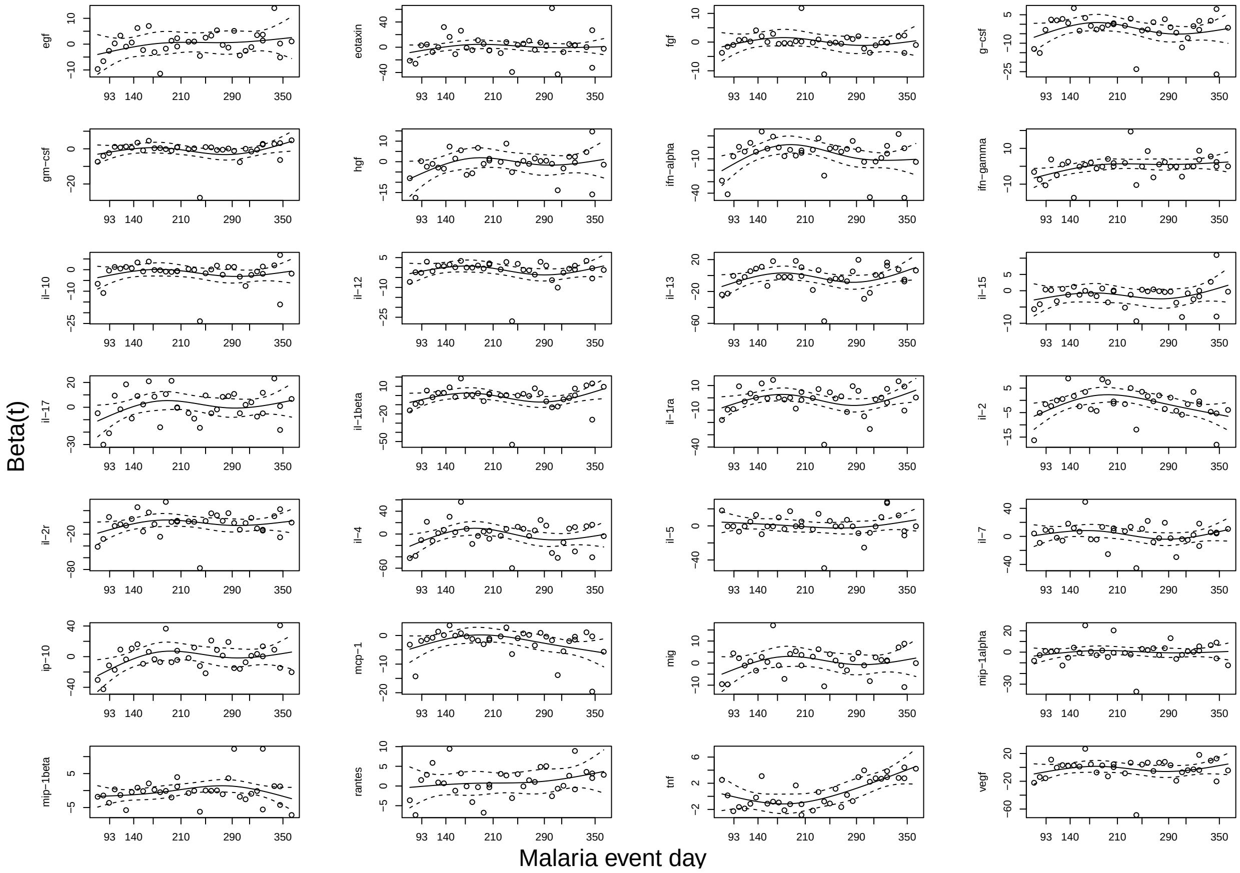


Figure E.1: Scaled Schoenfeld residuals for the 28 cytokines and RTS,S cohort. Y-axis shows the estimation of the Cox proportional hazards coefficient as a function of time and the X-axis the day in which an event occurred.

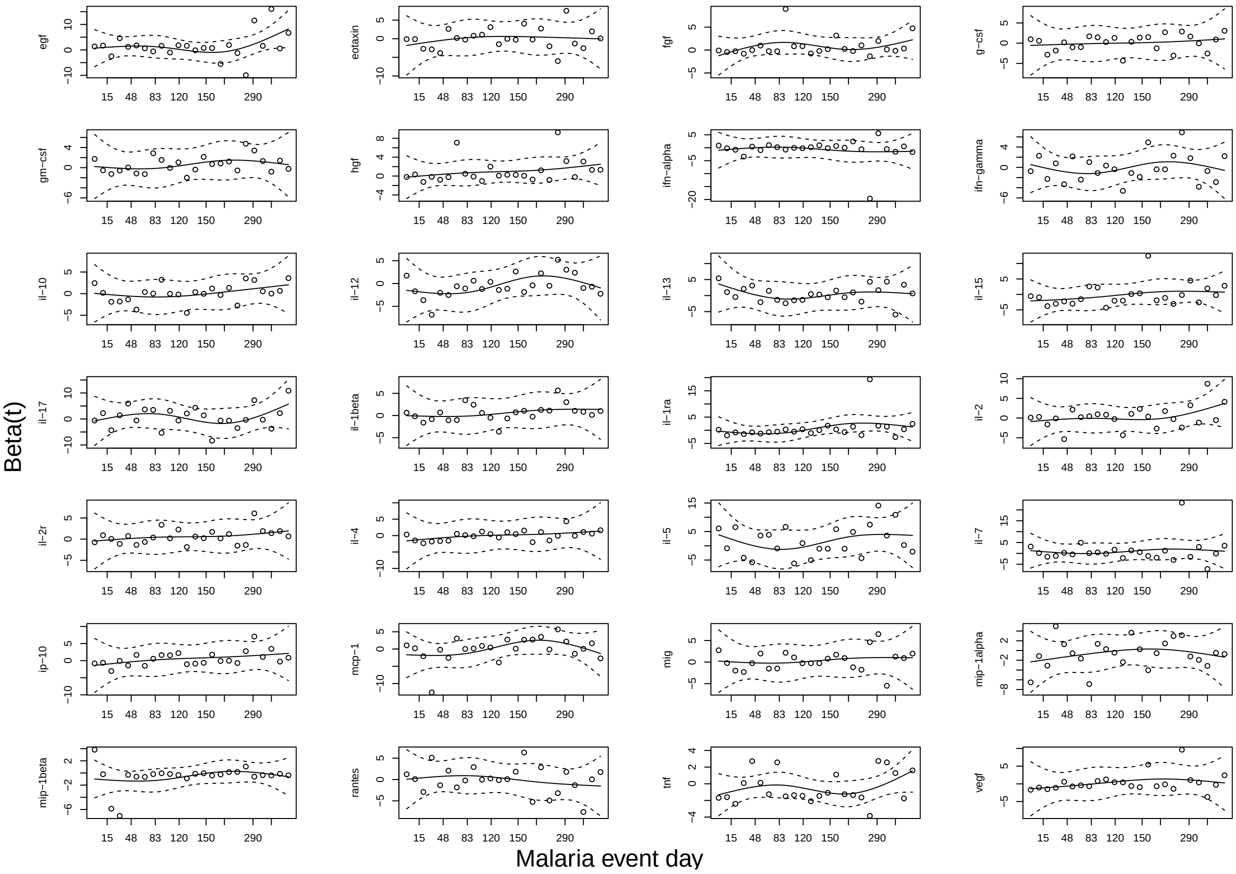


Figure E.2: Scaled Schoenfeld residuals for the 28 cytokines for the comparator cohort. Y-axis shows the estimation of the Cox proportional hazards coefficient as a function of time and the X-axis the day in which an event occurred.

Table E.1: P values for the scaled Schoenfeld residuals by analyte and vaccination status after fitting a bivariate Cox proportional hazards model. The p values correspond to the ones on Figure E.1 (RTS,S) and Figure E.2 (comparator).

Analyte	RTS,S	Comparator
EGF	0.244	0.344
Eotaxin	0.707	0.748
FGF	0.698	0.630
G-CSF	0.568	0.717
GM-CSF	0.651	0.633
HGF	0.446	0.327
IFN-α	0.634	0.654
IFN-γ	0.023	0.693
IL-10	0.746	0.498
IL-12	0.579	0.356
IL-13	0.506	0.915
IL-15	0.607	0.321
IL-17	0.365	0.786
IL-1β	0.882	0.556
IL-1ra	0.784	0.139
IL-2	0.288	0.328
IL-2r	0.553	0.586
IL-4	0.735	0.574
IL-5	0.880	0.538
IL-7	0.506	0.722
IP-10	0.199	0.453
MCP-1	0.195	0.325
MIG	0.692	0.702
MIP-1α	0.788	0.678
MIP-1β	0.398	0.360
RANTES	0.253	0.520
TNF	<0.001	0.411
VEGF	0.726	0.562

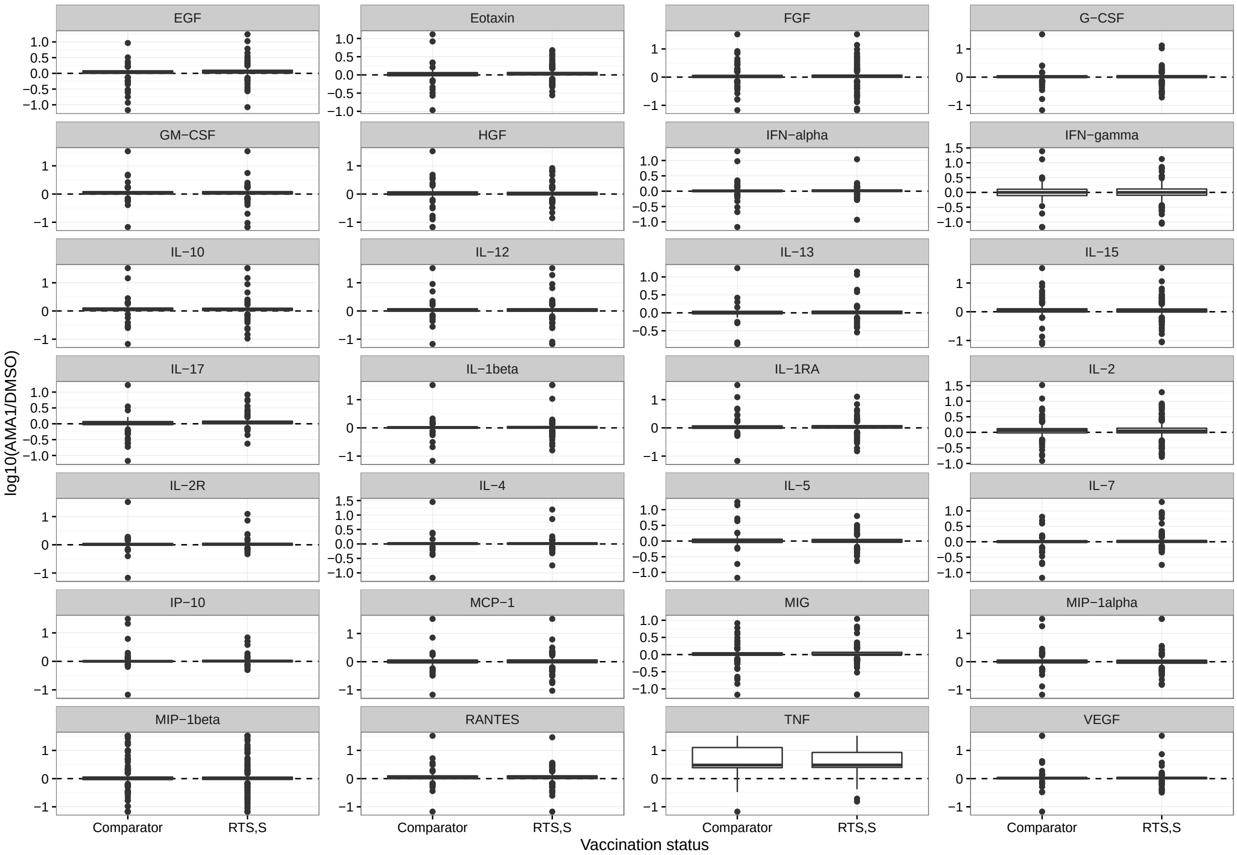


Figure E.3: Distribution of $\log_{10}$(AMA1/DMSO) ratio by cytokine and vaccination status.

Table E.2: Summary of relevant cytokines rank for RTS,S and comparator cohort (based on RFE-pseudo-samples algorithm). Mean (standard deviation) of the 100 bootstrap datasets is shown. The lower the mean the more relevant the cytokine/chemokine is.

RTS,S		**Comparator**	
Analyte	Mean (SD)	Analyte	Mean (SD)
RANTES	6.96 (5.96)	IL-15	9.72 (5.69)
IL-12	8.13 (4.75)	IP-10	9.76 (7.22)
G-CSF	9.49 (5.62)	IL-2r	9.93 (9.05)
Eotaxin	10.33 (6.84)	MIP-1α	10.43 (7.63)
EGF	11.06 (7.19)	HGF	10.73 (8.25)
GM-CSF	11.18 (6.62)	IL-4	12.40 (8.91)
FGF	11.87 (6.91)	IL-13	12.68 (8.80)
TNF	12.13 (6.88)	EGF	12.71 (9.52)
IL-1β	12.53 (7.30)	G-CSF	12.83 (6.77)
HGF	12.73 (6.12)	IL-12	13.07 (7.91)
IL-5	12.81 (8.52)	FGF	13.20 (7.06)
IL-15	12.98 (6.38)	IL-10	13.29 (7.20)
IL-2	13.40 (8.53)	IL-1β	13.64 (9.24)
IL-1ra	14.09 (6.15)	IFN-α	13.98 (8.75)
VEGF	14.26 (5.83)	IL-1ra	14.87 (7.26)
IL-13	14.96 (8.25)	VEGF	15.01 (9.55)
IFN-γ	15.02 (8.98)	Eotaxin	15.25 (7.24)
IFN-ALPHA	15.13 (7.37)	IL-7	15.31 (7.39)
IL-10	15.32 (6.61)	TNF	15.98 (6.56)
IL-4	15.45 (6.58)	GM-CSF	16.19 (6.79)
MCP-1	16.81 (7.07)	RANTES	16.63 (6.20)
IL-7	17.06 (7.01)	MIG	16.78 (7.05)
IL-17	17.28 (9.08)	IL-17	17.52 (6.89)
MIG	17.79 (6.21)	IFN-γ	17.62 (6.50)
MIP-1α	20.18 (8.78)	IL-2	17.89 (6.83)
IP-10	21.77 (6.36)	MCP-1	18.47 (6.66)
IL-2r	22.35 (5.70)	IL-5	19.43 (6.78)
MIP-1β	22.94 (8.16)	MIP-1β	20.68 (5.84)

Appendix F

R code

This section shows the implemented code created specifically for the present thesis: to estimate the parameters of the proposed SVM methods presented in Chapter 6. The code lacks of quality control parts because it was developed specifically for internal use. Therefore, all arguments were specified in the correct format.

F.1 Support vector machines learning using privileged information

The code shown is the R code to estimate the parameters of the expression shown in Section 6.1, equation (6.3) and the function to predict the class given a new observation in equation (6.2).

Estimation of parameters function

Arguments

`train.data` Training data.

`vars` Character vector defining the variables used in the decision space.

`vars.correc` Character vector defining the variables used in the correcting space.

`class.var` Character vector specifying the name of the variable that defines the class of the observations.

`sigma` The parameter value of the radial Gaussian kernel of the decision space (when Gaussian kernel is specified).

`gamma` The parameter value of the radial Gaussian kernel of the correcting space (when Gaussian kernel is specified).

`kernel.dec` Character vector defining whether to use linear kernel or Gaussian kernel in the decision space.

`kernel.correct` Character vector defining whether to use linear kernel or Gaussian kernel in the correcting space.

`scale.arg` Logical indicating whether to scale variables.

`rho` Cost (or weight) of the correcting space.

C Cost of the constraint violations.

Function

```
svm.lupi <- function(train.data, vars, vars.correct, class.var, sigma = NULL,
                gamma=NULL, kernel.dec="linear", kernel.correct = "radial",
                scale.arg=TRUE, rho, C){

      dat <- train.data
      n <- nrow(dat)

      #### Kernel functions
      if(kernel.dec=="linear") kernelfun.dec <- vanilladot()
      if(kernel.dec=="radial") kernelfun.dec <- rbfdot(sigma)
      if(kernel.correct=="linear") kernelfun.cor <- vanilladot()
      if(kernel.correct=="radial") kernelfun.cor <- rbfdot(gamma)

      ### Define some matrices and scale
      yvector <- dat[, class.var]
      xmat.dec <- dat[, vars]
      xmat.corr <- dat[, vars.correct]
      if(scale.arg){
        xmat.dec <- apply(xmat.dec, 2, function(x) scale(x, TRUE, TRUE))
        xmat.corr <- apply(xmat.corr, 2, function(x) scale(x, TRUE, TRUE))
      }

      ### Defining indexes of vectors
      alpha <- 1:nrow(dat)
      beta <- (nrow(dat)+1):(nrow(dat)*2)

      kern.matrix.dec <- kernelfun.dec(t(xmat.dec), t(xmat.dec))
      kern.matrix.cor <- kernelfun.cor(t(xmat.dec), t(xmat.dec))

      minfun <- function(x){
        a1 <- sum(x[alpha])

        a2.1 <- x[alpha]*yvector
        a2.1 <- sapply(a2.1, function(xx) xx*a2.1)
        a2 <- -1/2 * sum(a2.1 * kern.matrix.dec)
```

```
     a3.1 <- x[alpha] + x[beta] - C
     a3.1 <- sapply(a3.1, function(xx) xx*a3.1)
     a3 <- -1/(2*rho) * sum(a3.1 * kern.matrix.cor)

     ans <- a1 + a2 + a3
     return(-ans)
   }

   ll <- rep(0,n*2)
   uu <- rep(Inf,n*2)
   input <- rep(0.1, n*2)

   eqn=function(x){
     z1 <- sum(yvector * x[alpha])
     z2 <- sum(x[alpha] + x[beta] - C )
     return(c(z1,z2))
   }

  out <- solnp(input, minfun, eqfun = eqn, eqB = c(0,0),
                     LB = ll, UB= uu,  control = list(trace=0))

  y.data <- dat[,class.var]
  mat.data <- dat[,vars]

   ### Estimate bias
   fx <- function(x){
     value <- sum(as.matrix(kern.matrix.dec[,x])*yvector*out$pars[alpha])
     bias.ans <- yvector[x]  - value
     bias.ans
   }

   bias <- sapply(1:nrow(dat),function(yy) fx(yy))
   bias <- mean(bias)

   ans <- list(par= out$pars, sol=out, sigma = sigma, gamma = gamma, rho = rho,
               C=C, original.data = train.data, vars = vars,
               vars.correct = vars.correct, y.data = yvector,
               mat.data = dat[, c(vars)], alphas = out$pars[alpha],
               original.dat = dat, betas = out$pars[beta] ,bias = bias,
               kernelfun.dec = kernelfun.dec, kernelfun.cor=kernelfun.cor)
```

```
      return(ans)
}
```

Decision function

Arguments

`x`	Numeric vector with the observation to predict.
`out.object`	Object with the information provided by the function `svm.lupi`
`bias.arg`	Logical indicating whether to add the bias in the decision value.

Function

```
ff.svm.lupi <- function(x, out.object, bias.arg=TRUE){

  kernelfun.dec <- out.object$kernelfun.dec
  bias <- out.object$bias
  vars <- out.object$vars
  y.data <- out.object$y.data
  mat.data <- out.object$mat.data
  alphas <- out.object$alphas

  new.mat.data <- as.matrix(rbind(mat.data,x))
  nselec <- nrow(new.mat.data)
  value.k <- kernelMatrix(kernelfun.dec, new.mat.data, new.mat.data)[nselec,-nselec]
  ans <- alphas*y.data*value.k
  if(bias.arg) ans <- sum(ans) + bias
  if(!bias.arg) ans <- sum(ans)
  return(ans)
}
```

F.2 Support vector machines with uncertain classes

The code shown is the R code to estimate the parameters of the expression shown in equation (6.6) and the function to predict the class given a new observation in equation (6.7).

Estimation of parameters function

Arguments

`dat`	Training data.

`vars` Character vector specifying the name of the variables.

`class.var` Character vector specifying the name of the variable that defines the class.

`kernel.arg` Character vector specifying the type of the kernel, “linear” or “radial”.

`sigma` Numeric value defining the parameter of the radial Gaussian (when Gaussian kernel is specified).

`C` Cost value related to the certain classes.

`C1` Cost value related to the uncertain classes.

`rho` Distance of the probability prediction for a given point.

`verbose` Indicator value defining whether to print the output of the `solnp` function. This argument is directly passed to the function. Default 0 (does not show the output).

Function

```
svm.uncertain <- function(dat, vars, class.var, kernel.arg = "linear",
                          sigma, C, C1, rho=0.001, verbose=0){

    class.vector <- dat[,class.var]

    #### Kernel functions
    if(kernel.arg=="linear") kmeth <- vanilladot()
    if(kernel.arg=="radial") kmeth <- rbfdot(sigma)

    y.cert <- dat[dat[,class.var]%in%c(-1,1),class.var]
    y.uncert <- dat[dat[,class.var]%nin%c(-1,1),class.var]
    n.certain <-  length(y.cert)
    n.uncertain <-  length(y.uncert)
    datx.cert <- dat[dat[,class.var]%in%c(-1,1),vars]
    datx.uncert <- dat[dat[,class.var]%nin%c(-1,1),vars]

    # Probabilities
    pi <- dat[dat[,class.var]%nin%c(-1,1),class.var]
    a <- log( (1/rho) -1 )
    zminus <- (-1/a)*log((1/(pi-rho))-1)
    zplus <- (-1/a)*log((1/(pi+rho))-1)

    # K1 estimation. Only certain.
    kern.mat <- kernelMatrix(kmeth, as.matrix(datx.cert))
    kern.mat.certain <- kern.mat
    y.mat <- sapply(y.cert, function(x) y.cert*x)
```

```
K1 <- kern.mat*y.mat

# K2 estimation. Certain and uncertain
kern.mat <- apply(datx.cert, 1, function(x)
            kernelMatrix(kmeth, as.matrix(rbind(x,datx.uncert)))[1,])
kern.mat <- t(kern.mat)[,-1]
K2 <- kern.mat*y.cert

# K3 estimation. Only certain.
kern.mat <- kernelMatrix(kmeth, as.matrix(datx.uncert))
K3 <- kern.mat

### Create matrices to estimate the parameters
G1 <- cbind(K1, -K2, K2)
G2 <- cbind(t(-K2), K3, -K3)
G3 <- cbind(t(K2), -K3, K3)
G <- rbind(G1, G2, G3)

e <- c(rep(1,n.certain),  -zplus ,  zminus )
f <- c(y.cert, rep(-1, n.uncertain), rep(1, n.uncertain))

### Function to be minimized
minfun <- function(x){
  ans <- as.numeric((1/2 * (t(x)%*%G%*%x)) - (t(e)%*%x))
  return(ans)
}
eqn <- function(x){
  h <- as.numeric(f%*%x)
  return(h)
}

ll <- rep(0,n.certain+(n.uncertain*2))
uu <- c(rep(C, n.certain), rep(C1, n.uncertain*2))
input <- rep(min(c(C,C1))/2, n.certain+(n.uncertain*2))
out <- solnp(input, minfun, eqfun = eqn, eqB =0,
                    LB = ll, UB= uu, control = list(trace=verbose))

# Bias
fx <- function(x){
  labelled.sol <- out$pars[1:n.certain]
```

```
      tmp <- as.matrix(kernelMatrix(kmeth, as.matrix(datx.cert)))
      value1 <- sum(as.matrix(tmp[,x])*y.cert*labelled.sol)
      tmp <- as.matrix(kernelMatrix(kmeth, rbind(as.matrix(datx.cert)[x,],
                 as.matrix(datx.uncert))))
      muplus <- out$pars[(n.certain+1):(n.certain+n.uncertain)]
      muminus <- out$pars[(n.certain+n.uncertain+1):(n.certain + 2*n.uncertain)]
      value2 <- sum((muplus-muminus)*tmp[1,-1])
      bias.ans <-  y.cert[x]  - (value1 - value2)
      bias.ans
    }

    bias <- unlist(lapply(1:nrow(datx.cert),function(yy) fx(yy)))
    bias <- mean(bias)

    alpha <- out$pars[1:n.certain]
    muplus <- out$pars[(n.certain+1):(n.certain+n.uncertain)]
    muminus <- out$pars[(n.certain+n.uncertain+1):(n.certain + 2*n.uncertain)]
    ans <- list(sol=out, par=out$pars ,bias = bias, datx.cert=datx.cert,
                kmeth = kmeth, zplus = zplus, zminus = zminus,
                datx.uncert=datx.uncert, y.cert=y.cert, n.certain = n.certain,
                f=f, alpha=alpha, muplus =  muplus, muminus = muminus)
    return(ans)
}
```

Decision function

Arguments

`xvector`	Numeric vector with the observation to predict.
`out`	Object with the information provided by the function `svm.uncertain`
`bias.arg`	Logical indicating whether to add the bias in the decision value.

Function

```
ff.svm.uncertain <- function(xvector, out, bias.arg=TRUE){
  parameters <- out$par
  datx.cert <- out$datx.cert
  datx.uncert <- out$datx.uncert
  kmeth <- out$kmeth
  n.certain <- out$n.certain
  y.cert <- out$y.cert
  zplus <- out$zplus
  bias <- out$bias
  alpha <- out$alpha
  muplus <- out$muplus
  muminus <- out$muminus
  bias <- out$bias

  kern.mat <- kernelMatrix(kmeth, as.matrix(rbind(xvector, as.matrix(datx.cert))))
  ans1 <- sum(alpha*y.cert*kern.mat[1,-1])

  kern.mat <- kernelMatrix(kmeth, as.matrix(rbind(xvector, as.matrix(datx.uncert))))
  ans2 <- sum((muplus-muminus)*kern.mat[1,-1])

  ans <- ans1 - ans2
  if(bias.arg) ans <- ans + bias

  return(ans)
}
```

F.3 Semi-supervised support vector machines with local invariances

The code shown is the R code to estimate the parameters of the expression shown in equation (6.12) and the function to predict the class given a new observation in equation (6.13).

Specific functions to calculate local invariances

Arguments

`x,xi,xp`	A vector of an specific observation.
`sigmap`	Sigma value of the Gaussian distribution for the averaging approach.
`sigmak`	Sigma value of the Gaussian kernel for the averaging approach.
`sigma`	Sigma value of the Gaussian kernel for the gradient approach.
`kmeth`	Kernel method (function of the kernel method used) for both averaging and gradient approach.
`d`	d value from the averaging approach used.
`jcomp, qcomp`	j and q component of the x vector to calculate the gradient (gradient specific).

Functions

```
### Function to calculate z_ij(x) for the averaging approach
linv.average.z <- function(xi, x, sigmap, sigmak, d, kmeth){
  ans1 <- (sigmak^d / (sigmak + sigmap)^d ) *
                  exp( (-1 / (2*(sigmak + sigmap)^2)) * sum((xi-x)^2))
  ans2 <-  exp( (-1 / (2*(sigmak^2))) * sum((xi-x)^2))
  ans <- ans1 - ans2
  return(ans)
}

### Function to calculate dot product of z_ij(x) for the averaging approach
linv.average.zz <- function(xi, xj, sigmap, sigmak, d, kmeth){
  ans1 <- (sigmak^d / (sigmak + 2*sigmap)^d ) *
                  exp(- (1/ (2*(sigmak + 2*sigmap)^2)) * sum((xi-xj)^2))
  ans2 <- (sigmak^d / (sigmak + sigmap)^d ) *
                  exp(- (1/ (2*(sigmak + sigmap)^2)) * sum((xi-xj)^2 ))
  ans3 <- linv.average.z(xj, xi, sigmap, sigmak, d)
  ans <- ans1 - ans2 - ans3
  return(ans)
}


### Function to calculate z_ij(x) for the gradient approach
linv.diff.z <- function(x, xi, jcomp, sigma, kmeth){
  mat <- as.matrix(rbind(x,xi))
  kmatrix.value <- kernelMatrix(kmeth, mat)[1,2]
  ans <- (1/(sigma^2)) * (as.vector(x)[jcomp] - as.vector(xi)[jcomp])
                * kmatrix.value
  ans <- as.numeric(ans)
  return(ans)
}

### Function to calculate dot product of z_ij(x) for the gradient approach
linv.diff.zz <- function(xi, jcomp, xp, qcomp, sigma, kmeth){
  mat <- as.matrix(rbind(xi,xp))
  kmatrix.value <- kernelMatrix(kmeth, mat)[1,2]
  if(jcomp!=qcomp){
    ans <- (-1/sigma^4) * (xi[jcomp] - xp[jcomp]) * (xi[qcomp] - xp[qcomp]) *
                        kmatrix.value
  } else {
    ans <- (1/sigma^4) * (sigma^2 - (xi[jcomp] - xp[jcomp])^2) *
                        kmatrix.value
  }
  return(as.numeric(ans))
}
```

Estimation of parameters function

Arguments

`dat` Training data.

`epsilon` ϵ value of the dual function.

`rho2` ρ_2 value of the dual function.

`class.var` Character vector specifying the name of the variable that defines the class of the observations.

`sigma.k` σ value of the Gaussian kernel (averaging or gradient approach).

`sigma.p` σ value of the Gaussian distribution (only for averaging-specific approach).

`all.instances` Logical indicating whether to include all observations in the local invariances parts of the algorithm or only the unknown observations.

`cost.rho2.o` Cost value of the function when no local invariances are used (i.e., cost function of the classical SVM).

`verbose` Logical indicating whether to print intermediate steps of the function.

`scale.arg` Logical indicating whether to scale the variables.

`inv.method` "gradient" or "averaging" method to be applied.

`kernel.method` Character specifying "linear" or "radial" for the respective kernels to be used (only optimized for Gaussian).

Function

```
svm.linvariances <- function(dat,  epsilon=0.001, rho2=0.01, class.var="class",
         sigma.k=0.12, sigma.p=0.1, all.instances=TRUE, cost.rho2.o=1,
         verbose=FALSE, scale.arg=FALSE, inv.method="gradient",
         kernel.method="linear"){

  # Dataset should only have class and predictor variables
  nfeatures <- ncol(dat) - 1

  # Define some parameters to be used
  n.sup <- sum(!is.na(dat[,class.var]))
  n.unsup <- sum(is.na(dat[,class.var]))
  x.sup <- dat[!is.na(dat[,class.var]),]
  x.sup[,class.var] <- NULL
  x.unsup <- dat[is.na(dat[,class.var]),]
  x.unsup[,class.var] <- NULL
```

```
if(all.instances==TRUE | n.unsup==0){
  n.unsup <- n.sup + n.unsup
  x.unsup <- rbind(x.sup, x.unsup)
  labelled.index <- 1:n.sup
}

# Scale arguments
if(scale.arg){
  xvars <- names(x.sup)
  x.sup <- as.data.frame(apply(x.sup, 2, scale))
  names(x.sup) <- xvars
  x.unsup <- as.data.frame(apply(x.unsup, 2, scale))
  names(x.unsup) <- xvars
}

ydata <- dat[!is.na(dat[,class.var]),class.var]
if(kernel.method=="radial") kmeth <- rbfdot(1/(2*sigma.k^2))
if(kernel.method=="linear") kmeth <- vanilladot()

kernmat <- kernelMatrix(kmeth, as.matrix(x.sup))
kernmat.unsup <- kernelMatrix(kmeth, as.matrix(x.unsup))


# Y matrix (labelled data)
Ymat <- sapply(ydata, function(x) x*ydata)

# Length of parameters
n.alpha1 <- 1:n.unsup
n.alpha <- (n.unsup + 1):(n.unsup *2)
n.beta <- ( (n.unsup*2)+1 ):( (n.unsup*2) + n.sup )

# B term
Bterm <- 1/2 *  (kernmat*Ymat)
E <- -rep(1,n.sup)

if(rho2==0){
  cc <- matrix(E)
  HH <- Bterm
  ll <- matrix(rep(0,n.sup))
  uu <- matrix(rep(cost.rho2.o,n.sup))
```

```
    minfun <- function(x){
      minvalue <- (x%*%HH%*%x) + x%*%cc
      minvalue
    }

    sol <- optim(rep(0.1,n.sup), minfun, lower=ll, upper=uu, method="L-BFGS-B",
                 control = list(maxit=2000))
    # sol <- malschains(minfun, lower=ll, upper=uu)

  } else {

    if(inv.method=="average"){

      # A term
      pij <- matrix(NA, n.unsup, n.unsup)
      for(i in 1:n.unsup){
        for(j in i:n.unsup){
          if(verbose)  cat(paste0(i,"---",j), sep="\n")
          tmp <- linv.average.zz(x.unsup[i,], x.unsup[j,], sigma.p, sigma.k, nfeatures)
          pij[i,j] <- tmp
          pij[j,i] <- tmp
        }
      }

      A1 <- 1/2 * pij
      A2 <- -1/2 * pij
      A3 <- t(A2)
      A4 <- 1/2 * pij

      # C term
      pij <- matrix(NA, n.unsup, n.sup)
      for(i in 1:n.unsup){
        for(j in 1:n.sup){
          if(verbose) cat(paste0(i,"---",j), sep="\n")
          pij[i,j] <-  linv.average.z(x.unsup[i,], x.sup[j,],
                                 sigma.p, sigma.k, nfeatures) * ydata[j]
        }
      }
```

```
   C1 <- pij
   C2 <- -pij

   # Order 1 terms (D + E)
   D1 <- rep(epsilon,n.unsup)
   D2 <- rep(epsilon,n.unsup)

   # Re-structured if needed
   newA1 <- A1
   newA2 <- A2
   newA3 <- A3
   newA4 <- A4
   newB <- Bterm
   newC1 <- C1
   newC2 <- C2

   ll <- rep(0, length(n.alpha1)+length(n.alpha)+length(n.beta))
   uu <- c(rep(rho2,  length(n.alpha1)+length(n.alpha)), rep(1, length(n.beta)))
   input <- rep(0.1, length(n.alpha1)+length(n.alpha)+length(n.beta))
}
 if(inv.method=="gradient"){

   n.alpha1 <- 1:(n.unsup*nfeatures)
   n.alpha <- (length(n.alpha1)+1):((n.unsup*nfeatures)+length(n.alpha1))
   n.beta <- (((n.unsup*nfeatures)+length(n.alpha1))+1):
                     (((n.unsup*nfeatures)+length(n.alpha1))+n.sup)
   labelled.index <- (1:nrow(dat))[!is.na(dat[,class.var])]

   # A term
   zi <- cbind(sort(rep(1:nfeatures,n.unsup)),1:(n.unsup*nfeatures))
   pij <- matrix(NA, n.unsup*nfeatures, n.unsup*nfeatures)
   pijC <- matrix(NA, n.unsup*nfeatures, n.sup)
   for(i in 1:nfeatures){

     for(k in 1:n.unsup){
       for(kk in 1:n.sup){
         pijC[zi[zi[,1]==i,2][k],kk] <- linv.diff.z(x.sup[kk,],x.unsup[k,],
                                     i, sigma.k, kmeth)*ydata[kk]

       }
```

```
    }

    for(j in i:nfeatures){
      pij.tmp <- matrix(NA, ncol = nrow(x.unsup), nrow=nrow(x.unsup))
      if(i==j){
        diag(pij.tmp) <- (1/sigma.k^2)
        tmp.dat <- x.unsup

        for(k in 1:(nrow(tmp.dat)-1)){
          for(kk in (k+1):nrow(tmp.dat)){
            pij.tmp[k, kk] <- (1/sigma.k^4) *
                    (sigma.k^2 - (x.unsup[k,i] - x.unsup[kk,i])^2)
             * kernmat.unsup[k,kk]
            pij.tmp[kk, k] <- pij.tmp[k, kk]
          }
        }
        pij[ zi[zi[,1]==i,2] , zi[zi[,1]==j,2]] <- pij.tmp

      } else {
        diag(pij.tmp) <- 0
        tmp.dat <- x.unsup
        for(k in 1:(nrow(tmp.dat)-1)){
          for(kk in (k+1):nrow(tmp.dat)){
            pij.tmp[k, kk] <- (-1/sigma.k^4) *
                     (x.unsup[k,i] - x.unsup[kk,i])
              * (x.unsup[k,j] - x.unsup[kk,j]) *
                      kernmat.unsup[k,kk]
            pij.tmp[kk, k] <- pij.tmp[k, kk]
          }
        }
        pij[ zi[zi[,1]==i,2] , zi[zi[,1]==j,2]] <- pij.tmp
        pij[ zi[zi[,1]==j,2] , zi[zi[,1]==i,2]] <- pij.tmp
      }
      if(verbose) cat(paste0("A code, i: ",i," j: ",j),sep="\n")
    }
  }


  A1 <- 1/2 * pij
  A2 <- -1/2 * pij
```

```
      A3 <- t(A2)
      A4 <- 1/2 * pij

      # C term
      C1 <- pijC
      C2 <- -pijC

      # Order 1 terms (D + E)
      D1 <- rep(epsilon,n.unsup*nfeatures)
      D2 <- rep(epsilon,n.unsup*nfeatures)


      # Re-structured if needed
      newA1 <- A1
      newA2 <- A2
      newA3 <- A3
      newA4 <- A4
      newB <- Bterm
      newC1 <- C1
      newC2 <- C2


      ll <- rep(0, length(n.alpha1)+length(n.alpha)+length(n.beta))
      uu <- c(rep(rho2,  length(n.alpha1)+length(n.alpha)), rep(1, length(n.beta)))
      input <- rep(min(c(rho2/2,0.5)), length(n.alpha1)+length(n.alpha)+length(n.beta))
    }

    ## Define minimization function
    minfun <- function(x){
      a1 <- x[n.alpha1]%*%newA1%*%x[n.alpha1]
      a2 <- x[n.alpha1]%*%newA2%*%x[n.alpha]
      a3 <- x[n.alpha]%*%newA3%*%x[n.alpha1]
      a4 <- x[n.alpha]%*%newA4%*%x[n.alpha]
      b <- x[n.beta]%*%newB%*%x[n.beta]

      c1 <- x[n.alpha1]%*%newC1%*%x[n.beta]
      c2 <- x[n.alpha]%*%newC2%*%x[n.beta]

      d1 <- sum(x[n.alpha1]*epsilon)
      d2 <- sum(x[n.alpha]*epsilon)
      d3 <-  -sum(x[n.beta])
      ans <-  a1 + a2 + a3 + a4 + b + c1 + c2 + d1 + d2 + d3
      ans <- as.numeric(ans)
      return(ans)
    }

    ## Gradient function of optimization problem (to make optimization faster)
    grr <- function(x){
      a1.deriv <- newA1%*%x[n.alpha1]  + t(newA1)%*%x[n.alpha1]  + newA2%*%x[n.alpha]
                         + t(x[n.alpha]%*%newA3) + newC1%*%x[n.beta] + epsilon
      a.deriv <- t(x[n.alpha1]%*%newA2) + newA3%*%x[n.alpha1] + newA4%*%x[n.alpha]
                        + t(newA4)%*%x[n.alpha] +  newC2%*%x[n.beta] + epsilon
      beta.deriv <- newB%*%x[n.beta]  + t(newB)%*%x[n.beta] + t(x[n.alpha1]%*%newC1)
                                + t(x[n.alpha]%*%newC2) -1
      ans <-  c(a1.deriv, a.deriv, beta.deriv)
      return(ans)
    }


    if(verbose) cat("Starting optim function",sep="\n")
    sol <- optim(input , minfun, grr, lower=ll, upper=uu,
                 method="L-BFGS-B", control = list(maxit=2000))
    if(verbose) cat("End optim function",sep="\n")

  }

  # b parameter estimation
  ### Estimate bias
  if(rho2==0) n.beta <- 1:length(sol$par)

  fx <- function(x){

    kernmat <- as.matrix(kernelMatrix(kmeth, as.matrix(x.sup)))[,x]
    f1 <- sum((kernmat*ydata)*sol$par[n.beta])


    newdata <- as.data.frame(x.sup[x,])
    n.unsup <- nrow(x.unsup)
    f2 <- NULL
    z <- sort(rep(1:nfeatures,n.unsup))
    zobs <- rep(1:n.unsup,nfeatures)

    if(inv.method=="gradient"){
      for(iii in 1:(n.unsup*nfeatures)){
       comp <- z[iii]
       ni <- zobs[iii]
       f2[iii] <- linv.diff.z(x.sup[x,], as.vector(x.unsup[ni,]),
                      comp,sigma.k, kmeth)
      }
    }
    if(inv.method=="average"){
     for(iii in 1:n.unsup){
      f2[iii] <- linv.average.z(as.vector(x.unsup[iii,]),
                      x.sup[x,],sigma.p,sigma.k,nfeatures,kmeth)
     }
    }

    f2p1 <- f2%*%sol$par[n.alpha1]
    f2p2 <- f2%*%sol$par[n.alpha]
    f2p3 <- f2%*%(sol$par[n.alpha1]-sol$par[n.alpha])

    ans <- ydata[x] - (as.numeric(f1) + as.numeric(f2p3))


    return(ans)
   }

   bias <- sapply(1:nrow(x.sup),function(yy) fx(yy))
   bias <- mean(bias)

   ans <- list(sol=sol, x.unsup = x.unsup, x.sup = x.sup, dat = dat,
               ll = ll, uu = uu, sigma.k = sigma.k, sigma.p=sigma.p,
               Ymat = Ymat, nfeatures = nfeatures, ydata = ydata,
               bias = bias, n.alpha1= n.alpha1, n.alpha = n.alpha,
         kmeth=kmeth, n.beta=n.beta, par.alpha1 = sol$par[n.alpha1],
         par.alpha = sol$par[n.alpha],par.beta = sol$par[n.beta],
         inv.method = inv.method, rho2=rho2)

   return(ans)
  }
```

Decision function

Arguments

`x` Numeric vector with the observation to predict.

`out.object` Object with the information provided by the function `svm.linvariances`

`bias.arg` Logical indicating whether to add the bias in the decision value.

```
ff.svm.linvariances <- function(x, out.object, bias.arg = TRUE){

  inv.method <- out.object$inv.method
  rho2 <- out.object$rho2
  if(inv.method=="average"){
    parameters <- out.object$sol$par
    sigma.k <- out.object$sigma.k
    sigma.p <- out.object$sigma.p
    kmeth <- out.object$kmeth
    x.sup <- out.object$x.sup
    mat <- as.matrix(rbind(x.sup, x))
    dat <- out.object$dat
    Ymat <- out.object$Ymat
    ndata <- nrow(dat)
    ydata <- out.object$ydata
    nfeatures <- out.object$nfeatures
    bias <- out.object$bias
    n.alpha1 <- out.object$n.alpha1
    n.alpha <- out.object$n.alpha
    n.beta <- out.object$n.beta
    n.param <- length(parameters)
    n.sup <- nrow(x.sup)
    kernmat <- kernelMatrix(kmeth, mat)
    nkernmat <- nrow(kernmat)

    if(rho2>0){
      f1 <- sum((kernmat[nkernmat,-nkernmat]*ydata)*parameters[n.beta])
      x.unsup <- out.object$x.unsup
      newdata <- as.data.frame(x)
      n.unsup <- nrow(x.unsup)
      f2 <- NA
      for(j in 1:n.unsup){
        f2[j] <- linv.average.z(x.unsup[j,],x, sigma.p, sigma.k, nfeatures)
      }

      f2p1 <- f2%*%parameters[n.alpha1]
      f2p2 <- f2%*%parameters[n.alpha]
      f2p3 <- f2%*%(parameters[n.alpha1]-parameters[n.alpha])

      ans <- as.numeric(f1) + as.numeric(f2p3) + bias

    } else {
      f1 <- sum((kernmat[nkernmat,-nkernmat]*ydata)*parameters)
      ans <- as.numeric(f1) +  bias
    }
  }

  if(inv.method=="gradient"){
    parameters <- out.object$sol$par
    sigma.k <- out.object$sigma.k
    sigma.p <- out.object$sigma.p
    kmeth <- out.object$kmeth
    x.sup <- out.object$x.sup
    mat <- as.matrix(rbind(x.sup, x))
    dat <- out.object$dat
    Ymat <- out.object$Ymat
    ndata <- nrow(dat)
    ydata <- out.object$ydata
    nfeatures <- out.object$nfeatures
    bias <- out.object$bias
    n.alpha1 <- out.object$n.alpha1
    n.alpha <- out.object$n.alpha
    n.beta <- out.object$n.beta
    n.param <- length(parameters)
    n.sup <- nrow(x.sup)
    kernmat <- kernelMatrix(kmeth, mat)
    nkernmat <- nrow(kernmat)

    if(rho2>0){
      f1 <- sum((kernmat[nkernmat,-nkernmat]*ydata)*parameters[n.beta])
      x.unsup <- out.object$x.unsup
      newdata <- as.data.frame(x)
      n.unsup <- nrow(x.unsup)
      f2 <- NULL

      z <- sort(rep(1:nfeatures,n.unsup))
      zobs <- rep(1:n.unsup,nfeatures)

      for(i in 1:(n.unsup*nfeatures)){
        comp <- z[i]
        ni <- zobs[i]
        f2[i] <- linv.diff.z(x, as.vector(x.unsup[ni,]),comp,sigma.k, kmeth)
      }

      f2p1 <- f2%*%parameters[n.alpha1]
      f2p2 <- f2%*%parameters[n.alpha]
      f2p3 <- f2%*%(parameters[n.alpha1]-parameters[n.alpha])

      ans <- as.numeric(f1) + as.numeric(f2p3) + bias
    } else {
      f1 <- sum((kernmat[nkernmat,-nkernmat]*ydata)*parameters)
      ans <- as.numeric(f1) +  bias
    }
  }
  return(ans)
}
```

F.4 Weighted support vector machines

The code shown is the R code to estimate the parameters of the expression shown in equation (6.19) and the function to predict the class given a new observation of equation (6.20)

Estimation of parameters function

Arguments

`dat` Training data.

`class.var` Character vector specifying the name of the variable that defines the class of the observations.

`C` Cost parameter.

`weights` Weight vector for the observations.

`sigma`	Sigma parameter of the Gaussian Kernel when specified in the `kernel.method` argument.
`verbose`	Depreciated.
`scale.arg`	Logical indicating whether to scale variables.
`kernel.method`	Character specifying “linear” or “radial” for the respective kernels to be used.

Function

```
svm.weighted <- function(dat,  class.var="class", C=0.01, weights=1,
                         sigma=0.12,  verbose=FALSE,
                         scale.arg=FALSE, kernel.method="linear"){

  nfeatures <- ncol(dat) - 1

  # Define some parameters to be used
  n.sup <- sum(!is.na(dat[,class.var]))
  x.sup <- dat[!is.na(dat[,class.var]),]
  x.sup[,class.var] <- NULL

  # Scale arguments
  if(scale.arg){
    xvars <- names(x.sup)
    x.sup <- as.data.frame(apply(x.sup, 2, scale))
    names(x.sup) <- xvars
  }

  ydata <- dat[!is.na(dat[,class.var]),class.var]
  ydata <- as.numeric(as.character(ydata))
  if(kernel.method=="radial") kmeth <- rbfdot(sigma)
  if(kernel.method=="linear") kmeth <- vanilladot()

  kernmat <- kernelMatrix(kmeth, as.matrix(x.sup))


  # Y matrix (labelled data)
  Ymat <- sapply(ydata, function(x) x*ydata)

  # Length of parameters
  n.alpha <- 1:nrow(dat)

  # B term
```

```
Bterm <- 1/2 *  (kernmat*Ymat)
E <- -rep(1,n.sup)

cc <- matrix(E)
HH <- Bterm
ll <- matrix(rep(0,n.sup))

if(length(weights)==1)  uu <- matrix(rep(C,n.sup))
if(length(weights)!=1)  uu <- matrix(C*weights)

minfun <- function(x){
  minvalue <- as.numeric(t(x)%*%HH%*%x + t(cc)%*%x)
  minvalue
}

eqn <- function(x){
  h <- as.numeric(ydata%*%x)
  return(h)
}

sol <- solnp(uu/2, minfun, eqfun = eqn, eqB =0, LB = ll, UB= uu,
control = list(trace=verbose))
sol$par <- sol$pars

fx <- function(x){
  labelled.sol <- sol$par
  tmp <- as.matrix(kernelMatrix(kmeth, as.matrix(x.sup)))
  value <- sum(as.matrix(tmp[,x])*ydata*labelled.sol)
  bias.ans <-  ydata[x]  - value
  return(bias.ans)
}

bias <- sapply(1:nrow(x.sup),function(yy) fx(yy))
bias <- mean(bias)

ans <- list(sol=sol,  x.sup = x.sup, dat = dat, ll = ll, uu = uu,
            sigma = sigma,  Ymat = Ymat, nfeatures = nfeatures, ydata=ydata,
            ydata = ydata, bias = bias,n.alpha = n.alpha, kmeth=kmeth,
            par.alpha = sol$par)
return(ans)
```

```
}
```

Decision function

Arguments

`x` Numeric vector with the observation to predict.

`out.object` Object with the information provided by the function `svm.weighted`

`bias.arg` Logical indicating whether to add the bias in the decision value.

Function

```
ff.svm.weighted <- function(x, out.object, bias.arg = TRUE){

  parameters <- out.object$sol$par
  kmeth <- out.object$kmeth
  x.sup <- out.object$x.sup
  mat <- as.matrix(rbind(x.sup, x))
  ydata <- out.object$ydata
  bias <- out.object$bias
  kernmat <- kernelMatrix(kmeth, as.matrix(mat))
  nkernmat <- nrow(kernmat)
  bias <- out.object$bias
  f1 <- sum((kernmat[nkernmat,-nkernmat]*ydata)*parameters)

  if(bias.arg)  ans <- as.numeric(f1) +  bias
  if(!bias.arg)  ans <- as.numeric(f1)

  return(ans)
}
```