

## University of Calabria

Department of Electronics, Informatics and Systems

PhD course: *Operations Research*

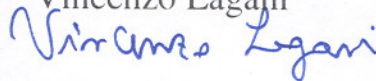
Cycle XXI

Scientific sector: Operations Research (MAT/09)

### **MKL – CT: Multiple Kernel Learning for Censored Targets**

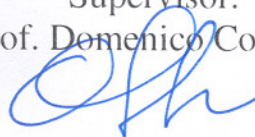
Candidate:

Vincenzo Lagani



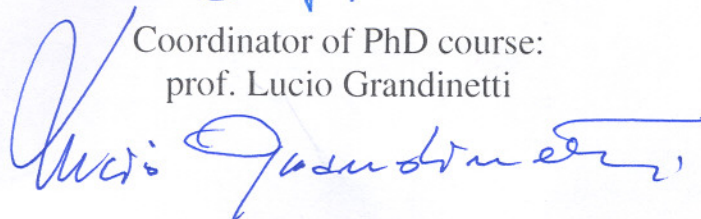
Supervisor:

prof. Domenico Conforti



Coordinator of PhD course:

prof. Lucio Grandinetti



*...dedicated to the fearless people who encouraged, trusted and loved me*

# Index

Introduction .....	1
1. Analysis of censored data.....	3
1.1. Introduction.....	3
1.2. Statistical approaches for the analysis of censored data.....	5
1.3. Machine learning approaches for the analysis of censored data.....	6
1.3.1. Data pre processing approaches .....	7
1.3.2. SVCR: Support Vector Regression for Censored Target .....	8
1.3.3. Survival Regression Tree and Survival Regression Forest.....	11
1.4. Features selection and censored data. ....	11
2. Kernel Learning .....	13
2.1. Learning the kernel matrix .....	13
2.2. Kernel Learning with Semi Definite Programming.....	14
2.3. Hyper Kernels.....	16
2.4. Multiple Kernel Learning. ....	17
3. Multiple Kernel Learning for Censored Target.....	21
3.1. Merging Support Vector Machine for Censored Target with Multiple Kernel Learning.....	21
3.2. CT – MKL: optimization model. ....	23
3.3. CT – MKL: solving algorithm. ....	30
3.3.1. The wrapper algorithm .....	30
3.3.2. The chunking algorithm.....	32
3.4. CT – MKL: implementation. ....	34
4. Experimentation.....	39
4.1. Experiments Organization. ....	39
4.1.1. Experimentation protocol .....	41
4.1.2. Performance measure .....	43
4.2. Heterogeneous survival data: effect of the interaction among genetic and physiological factors for the ageing process. ....	45
4.2.1. Carolei dataset Description.....	45
4.2.2. Experiments on Carolei dataset.....	48

4.3. CT – MKL on left, right, double and interval censored data.....	50
4.3.1. Fried dataset Description .....	51
4.3.2. Experiments on Fried datasets .....	52
4.4. CT – MKL applied on real survival datasets.....	57
4.4.1. Experiments on Bfeed dataset.....	57
4.4.2. Experiments on Nwtco dataset.....	59
4.4.3. Experiments on Pneumon dataset .....	60
4.4.4. Experiments on Std dataset.....	62
4.5. Comparison of MKL – CT and SVCR algorithms in terms of time spent during the training phase.....	63
4.6. Critical discussion of results.....	64
Conclusions .....	66

## Introduction

The analysis of censored data is a research stream of statistics science that attracted a great interest in the last decades. The term “censored data” indicates a uncertain measure known only through its upper and a lower bound. In particular, censored data are highly frequent in longitudinal studies, where the data to be collected consist in the times of occurrence of a particular event. Whether such event does not occur before the end of the study, one can only assert that the end of the study represents a lower bound for the actual time – to – event.

Recently, several Support Vector Machine (SVM) models were devised in order to deal with censored data. SVM consist in a class of Mathematical Programming models able to solve classification, regression and data description problems .In particular, the majority of SVM models can be stated and solved as Convex Quadratic Programming problems.

Beside SVM for censored data, in the last years other SVM models were introduced, able to automatically determine the best kernel function for the problem under study. Kernel functions can be thought as parameters of SVM models, that usually are chosen “a priori”. The performances of SVM models almost totally depend by the choice of the kernel function most suitable for the problem under analysis.

Summarizing, the current scientific literature offers both SVM models for dealing with censored data and SVM model able to automatically determine the best kernel to be used. An unique SVM model able to contemporary deal with censored data and to automatically select the best kernel is still missing.

Thus, the present thesis work consists in:

1. formulating the MKL – CT model, i.e. Multiple Kernel Learning for Censored Target. Such model unifies the characteristics of SVM model

for censored data and the features of MKL – SVM models for the automation of kernel selection procedure;

2. adapting MKL solving algorithms to the resolution of MKL – CT model;
3. modifying open source codes in order to solve the optimization problem underlying the MKL – CT approach;
4. evaluating the effective validity of MKL – CT model through a wide experimentation carried out on simulated and “real world” data

It is worthwhile to underline the heavy role played by Operations Research and Optimization methods in the present thesis. In fact the MKL –CT model consists in a Quadratically Constrained Quadratic Programming model, that can be formulated as a Semi Infinite Linear Programming problem. Meanwhile, the solution of the final MKL – CT model can be obtained by using an “ad hoc” exact solving algorithm, belonging to the class of “exchange methods”.

In synthesis, even if this thesis is mainly oriented to the Machine Learning research field, the methods and models used during this work come from the latest developments of Optimization science.

Chapters are organized as following:

1. the first chapter introduces the main concepts related to censored data, and describes some of the methods used in order to analyze this type of data, including the SVM models for censored target;
2. the second chapter briefly outlines the SVM models able to automatic select the kernel function, with particular regard to MKL methods;
3. MKL – CT model and solving algorithm are deeply described in the third chapter;
4. last chapter describes the experimentation performed and critically analyzes the obtained results.

# 1. Analysis of censored data

## 1.1. Introduction

Statistical modelling of *time to event* processes is a widely studied research field, with practical applications in many areas, including medicine, bioinformatics, actuarial sciences and reliability analysis.

For examples, numerous medical studies are carried out by registering the individual time – to – event of the subjects belonging to a selected population; the event under study can be the arising of specific symptoms, the development of an infection, or death. The aim of these studies generally is to analyze the effect of some factors (e.g. treatment, age) on the time to occurrence of the studied event.

If the event occurred in all subjects, many methods of regression analysis would be applicable. However, usually at the end of follow-up period some of the individuals have not experienced any event, and thus their actual time – to – event is *censored*, i.e. we only know that the event was not experienced until the limit of the follow up period. This type of censored data are defined *survival data*, and requires specific strategies in order to be analyzed; in particular, it is necessary to take in account the partial information provided by the censored times to event.

Survival data are also known as *right censored data*: the history of some subjects is not known after a certain time point, i.e., on the *right side* of an imaginary time line.

Other two types of censored data exist: when for some subjects it is only known that the event occurred before a certain time point, the data are defined *left censored*; when the censored times to event are known to lie between two time points, the data are said to be *interval censored*.

We can compactly model the three types of censored data using a mathematical formulation:

Let define a dataset  $D$  as a set of  $m$  tuples  $\langle x_i, l_i, u_i \rangle$ , where each  $x_i$  represents the  $i^{th}$  subject under study. The generic  $x_i$  can be indicated as subject, case or instance, and it is composed by  $n$  variables (called also attributes, features or

covariates), i.e.  $x_i = \langle x_i^1, x_i^2, \dots, x_i^j, \dots, x_i^n \rangle$ . The values  $l_i$  and  $u_i$  represent respectively the lower bound and the upper bound of the time to event  $y_i$  of the generic  $i^{\text{th}}$  subject.

Given this formulation, we can easily represent left, right and interval censored cases, as well as the cases with a known time to event:

- left censored cases are characterized by  $l_i = -\infty$ ;
- right censored cases by  $u_i = +\infty$ ;
- interval censored are represented allowing  $u_i > l_i$ ;
- cases with known time to event have  $u_i = l_i$ .

Other formalization could be adopted in order to represent censored data; we chose the above given formulation because it is highly intuitive and it will allow an easier explication of some concepts in the following chapters.

Moreover, it should be noted that the adopted formalization is not able to represent some more complex information, such as time varying variables, presence of multiple events, censored covariates. However, these kind of data (and the respective methods of analysis) will be not considered during the present thesis.

Instead, the main aim of the algorithms and methods presented in the rest of this work can be described with the following proposition:

**Proposition 1:** Given a dataset  $D$ , find a function  $f : R^n \rightarrow R$  able to represent the relationship between the set of variable  $x$  and the time to event  $y$ , i.e.  $y = f(x)$ . The time to event  $y$  is known only through its lower and upper bound, respectively  $l$  and  $u$ .



## **1.2. Statistical approaches for the analysis of censored data**

Over the last decades, censored data analysis received increasing interest by the statistical community. Several parametric and semi parametric algorithms were developed in order to modelling time to event on a set of covariate, even in presence of censored data: one of the first and most notable technique is the semi parametric Cox regression model [1]. More recently, other methodologies were developed, including parametric survival models [2], accelerated failure time models [3], spline based extensions [4], fractional polynomials [5] and Bayesian methods [6].

Among the afore cited techniques, Cox regression and parametric survival models deserve a more detailed explication, for their wide use in the analysis of censored data.

Cox regression models were originally developed for survival data, and in their original form they can deal only with right censored cases. Cox regression lies on the definition of the survival function  $S(t) = \Pr(T > t)$ , that is the probability of surviving after the time  $t$ . The hazard function, defined as  $h(t) = -S'(t)/S(t)$ , express the risk of experiencing the event at time  $t$ .

In his famous work [1], Sir Cox proposed to model the hazard function with the following formula:

$$h(t, x) = h_0(t) \cdot e^{b^T x}$$

where  $h_0(t)$  is a baseline hazard function, common to all the subjects, and the terms  $e^{b^T x}$  takes in account the influences of a set of covariates  $x$ , representing the particular status of each subject.

The points of strength of Cox models are:

1. the estimation procedure of parameters vector  $b$  is based on the maximization of the log likelihood and is able to exploit the information carried out by the censored cases;

2. the baseline hazard function must not be estimated in order to calculate the value of the parameters vector  $b$ .

On the other hand, Cox models assume that the hazard ratio between two subject will be constant over the time, i.e. the ratio between the hazards of subject A and subject b will remain the same independently by the time. Of course this assumption represent a limitation for the Cox models.

Parametric survival models assume that the survival function of a population can be modelled with a given parametric distribution. Some of the most used distributions are shown in Table 1. Survival functions are unconditional, in the sense that they do not take in account the vector of covariates characterizing the different subjects. Then, survival functions must be turned in conditional models, by replacing one of the free parameters with a (suitably transformed) linear predictor. The linear predictor is simply the inner product of a parameters vector  $b$  and the vector  $x$  of the covariates under study.

**Table 1: Survival distributions for parametric survival models. The cumulative normal distribution  $\phi(z)$  is defined as  $\phi(z) = \int_{-\infty}^z N(\gamma;0,1)d\gamma$ .**

Distribution	$S(t)$
Weibull	$\exp(-\lambda \cdot t^\gamma)$
Exponential	$\exp(-\lambda \cdot t)$
Log – Normal	$1 - \phi(\lambda \cdot \ln(t))$
Log – Logistic	$(1 + \lambda \cdot t^\gamma)^{-1}$

### **1.3. Machine learning approaches for the analysis of censored data**

Various algorithm aimed to deal directly with censored data were developed in the context of the machine learning community.

Generally speaking, Machine Learning algorithms for survival analysis mainly consist in previously existent algorithms specifically modified or re – formulated in order to take in account censored samples. Among the various works present in the literature, it is worthwhile to report the following examples: Regression tree

for censored data [7], Artificial Neural Networks [8], Survival Ensemble and Survival Random Forest [9], Supervised PCA [10] and SVM like algorithms [11, 12, 13]. Moreover, there exist also some examples of statistical techniques hybridized with machine learning elements, such as Kernel Cox regression model [12] and Kernel accelerated time survival analysis [14].

Even if not limited to the above list, machine learning techniques able to deal with censored data remains relatively few, and moreover free software implementations are very rare. Exceptions are Survival Random Forest and Hierarchical Mixture of Experts (HME) models, both available for the *R* software in their respective packages “randomSurvivalForest”<sup>1</sup> and “hme”<sup>2</sup>.

The following paragraphs will deeper explain some of the techniques aforementioned, with particular emphasis on SVCR and Random Survival Forest (RSF). SVCR has been successively used during the experimental tests.

### 1.3.1. Data pre processing approaches

Several strategies based on data manipulation have been proposed in order to let machine learning algorithms deal with survival/censored datasets. This approaches are mainly concerning on right censored data.

The simplest method considers survival for a fixed time period, and consequently gives a binary classification problem [15]. Censored observations are removed and biases are introduced. It is clear that this approach is rather basic and does not really deal with the problem of censoring.

A second approach is based on multiple replications of each subject [15]. According to this method, each instance  $x_i$  is replicated several times, with two more attributes added to each replica:

1. an increasing “time stamp”  $x_i^{n+1}$ ;
2. a class attribute  $y_{class}$  indicating whether the event occurred or not at the time  $x_i^{n+1}$

---

<sup>1</sup> <http://cran.r-project.org>

<sup>2</sup> <http://www.maths.bris.ac.uk/~maxle/software.html>

The converted dataset can be processed using the standard algorithm for classification problems. This approach has been widely used in the field of neural network, and several publications testify its effectiveness [16 – 19]. Nevertheless, the replication of the original instances can produce scalability problems, due to the dimensions reached by the converted dataset.

The third approach consists in the imputation of censored outcome using the information carried by the uncensored cases. This method require a hierarchical structure of regression models: one or more models are firstly trained on the basis of the uncensored instances in order to estimate the times to event for the censored cases. Then, the final model is trained utilizing all the cases, using the estimated times to event for the censored cases [20].

### 1.3.2. SVCR: Support Vector Regression for Censored Target

SVCR represents one of the last attempts of adapting the well known SVM models to the analysis of censored target. Numerous tutorial regarding the standard SVM models are available for the interested readers [21].

Recalling the concepts expressed by **Proposition 1**, the main idea at the basis of the SVCR models is that the function  $f(x_i)$  should respect the following condition:

$$l_i \leq f(x_i) \leq u_i \quad i = 1 \dots m \quad (1)$$

that is, the estimations of times to event  $y$  provided by the function  $f$  should respect the upper and lower bounds  $u$  and  $l$ .

Let suppose that the function  $f$  consist of a simple linear combination of variables  $x$ , i.e.  $f(x_i) = w^T \cdot x_i + b$ . Then, condition (1) could be expressed with the following constraints:

$$w^T \cdot x_i + b - u_i \leq 0 \quad \forall i \in U \quad (2)$$

$$l_i - w^T \cdot x_i - b \leq 0 \quad \forall i \in L \quad (3)$$

where  $U$  is defined as  $U = \{i \mid u_i < +\infty\}$  and  $L = \{i \mid l_i > -\infty\}$ .

Usually real world data do not follow a strict liner trend, as required by constraints (2) and (3); then some slack variables should be added in order to make such constrains generally feasible:

$$w^T \cdot x_i + b - u_i \leq \zeta_i^* \quad \forall i \in U \quad (4)$$

$$l_i - w^T \cdot x_i - b \leq \zeta_i \quad \forall i \in L \quad (5)$$

$$\zeta_i \geq 0, \zeta_i^* \geq 0 \quad \forall i \quad (6)$$

Following the regularization theory, that is one of the main pillar of SVM methods, the weights  $w$  can be uniquely determined by imposing the minimization of  $w$  norm [22]. Then, the final primal optimization model is the following:

$$\min_{w, b, \zeta, \zeta^*} \frac{1}{2} \|w\|^2 + C \left( \sum_i \zeta_i^* + \sum_i \zeta_i \right) \quad (7)$$

$$w^T \cdot x_i + b - u_i \leq \zeta_i^* \quad \forall i \in U \quad (8)$$

$$l_i - w^T \cdot x_i - b \leq \zeta_i \quad \forall i \in L \quad (9)$$

$$\zeta_i \geq 0, \zeta_i^* \geq 0 \quad \forall i \quad (10)$$

where  $C$  is a user defined parameter.

Model (7) – (8) is the primal form of the SVCR optimization problem. Assigning dual variables  $\alpha^*$  and  $\alpha$  to constraints (8) and (9), the following dual problem can be easily obtained:

$$\min_{\alpha, \alpha^*} \frac{1}{2} \sum_i \sum_j (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(x_i, x_j) - \Lambda^T \alpha + \Psi^T \alpha^* \quad (11)$$

$$e_L^T \alpha - e_U^T \alpha^* = 0 \quad (12)$$

$$0 \leq \alpha_i, \alpha_i^* \leq 0 \quad \forall i \quad (13)$$

In constrain (12), vectors  $e_L$  and  $e_U$  are defined as:

$$[e_L]_i = \begin{cases} 0 & \text{if } l_i = -\infty \\ 1 & \text{otherwise} \end{cases}$$

$$[e_U]_i = \begin{cases} 0 & \text{if } u_i = +\infty \\ 1 & \text{otherwise} \end{cases}$$

Similarly, vectors  $\Lambda$  and  $\Psi$  are defined ad:

$$\Lambda_i = \begin{cases} 0 & \text{if } l_i = -\infty \\ l_i & \text{otherwise} \end{cases}$$

$$\Psi_i = \begin{cases} 0 & \text{if } u_i = +\infty \\ u_i & \text{otherwise} \end{cases}$$

The kernel function  $K(x_i, x_j)$  replaced the scalar product in the objective function (11) in order to allow a non linear decision function.

Model (11) – (13) formally is equivalent to the standard dual Support Vector Regression model [22]. Numerous fast and scalable algorithm have been developed for the solution of this type of quadratic program, as for example the SMO algorithm [23]. The size of the problem is equal to  $2 \cdot n$ , while the complexity is in the order of  $O(n^2)$ .

A few lines should be spend in order to define the elegant properties of this model. Firstly, SVCR is totally non parametric, in the sense that no assumption are made regarding the distribution of the data or about the shape of the survival/hazard function. The absence of assumptions regarding data distribution is a great advantage of SVCR in respect of classical statistical models, as for example Cox regression.

Moreover, SVCR is able to provide estimates of the time to event  $y$  for left, right and interval censored data, while other machine learning or statistical methods are ale to deal only with right censored cases.

Finally, SVCR optimization problem can be faced with all the efficient algorithms specifically developed and implemented for solving the standard SVM optimization problem; pre – existent codes for SVM solution can be easily adapted for SVCR particularities.

### **1.3.3. Survival Regression Tree and Survival Regression Forest.**

Survival regression tree have been firstly introduced by M.R. Segal [7] for the analysis of right censored data. The main idea consists in recursively subdividing the dataset  $D$  until it is not possible to recognize two groups with significantly different survival functions within the final subsamples.

The algorithm examines singularly each variable, and test whether it is possible to subdivide the sample in two or more subgroups with different survival functions; once tested every variable, the sample is subdivided following the variable that allows the creation of the subsamples most dissimilar. The iteration is repeated on each subsample until any subdivision is possible. A statistical test is usually used in order to assess the diversity of two survival functions.

An evolution of such algorithm is represented by the Random Survival Forest (RSF) algorithm [24]. RSF algorithm merge the Breiman's Random Forest [25] with the survival regression trees, trying to exploiting the advantages of both methods. Random Forest algorithms construct several decision trees, randomly or pseudo randomly choosing the nodes of each tree. When a new instance is presented to the Random Forest to be evaluated, each tree give an evaluation, and then a voting procedure is used over all the predictions. RSF use this same schema, adding the subdivision rule based on a statistical test for comparing two or more survival functions.

A freeware implementation of RSF algorithm is freely available as a package of the R software

## **1.4. Features selection and censored data.**

The selection of relevant features selection is one of the most relevant research area of data analysis. Countless literature is available about features selection

methods specifically devised for selecting the most relevant features for classification and regression problems.

However, much less work has been done regarding features selection in presence of censored outcomes. In particular, only few algorithms have been specifically devised in order to directly deal with censored data.

Univariate and wrapper features selection methods can be used also with censored data, choosing the appropriate statistical test (e.g. log rank test) for the univariate selection and a suitable performance function for the wrapper algorithms (for example, the C index [26]).

Beyond univariate and wrapper methods, other features selection algorithms for censored dataset have been developed within the bioinformatics area.

In bioinformatics studies, datasets are usually composed by thousands of variables and relatively few (hundreds) cases. Then an effective selection of the most relevant variables is mandatory. An extensive review of such methods has been written by Bøvelstad et al [27].



## 2. Kernel Learning

### 2.1. Learning the kernel matrix

Kernel based methods gained increasing popularity over the last years. One of the main reasons of kernel methods success relies on the so – called “kernel trick”, i.e., the possibility of replacing simple dot products with a more complex, and usually more effective, kernel function.

Kernel functions are not a trick merely useful in order to catch non linear trends present inside data; using the kernel functions, it is possible to embed domain specific knowledge inside general purpose data analysis algorithm, specifying a “similarity measure” able to incorporate the specificity of the problem under analysis. Then, several kernel functions have been devised and successfully experimented in various application field, from genetic sequences analysis to speech recognition [28] [29].

However, kernel methods presents also some drawbacks. The principal issues consists in the need of choosing a suitable kernel function: the performance of any kernel based algorithm is strictly depending by the choice of the similarity function. Generally, machine learning algorithms present several parameters to be set by the user; a reasonable choice of such parameters allow the algorithm to best fit the data. On the other side, unwary parameters setting techniques can easily lead to data over fitting, so time expensive performance estimation techniques, as for example cross validation, must be used in order to provide unbiased estimates of the performance for each tested parameters configuration.

From this point of view, the selection of the most suitable kernel can be seen as part of the parameters setting procedure. Moreover, it should be noted that kernels usually have some own parameters to be set, adding a further level to the parameters setting problem. When the choice of the kernel is enumeratively performed, the kernel selection process can be thought as a discrete optimization problem: given a dataset  $D$ , a kernel based algorithm  $A$ , a set  $\bar{K}$  of kernel functions,  $\bar{K} = \{K_1, K_2, \dots, K_{|\bar{K}|}\}$ , and a performance estimation procedure  $P$ , find the kernel  $\hat{K}$  that optimize the performance of the algorithm  $A$  on the dataset  $D$ .

Recently, new methods have been proposed in order to partially overcome the kernel selection problem, when SVM models are used. Such methods can be comprehensively named as Kernel Learning algorithms, since they attempt to contemporarily learn SVM model variables and the most suitable kernel directly from data.

The advantages of such approach are evident: a direct estimation of the kernel metric allows to avoid time consuming kernel selection procedures. Moreover, from a theoretical point of view, kernel estimation should provide more suitable similarity function compared to the usual selection procedure based on the experimentation of several “a priori” chosen kernels.

On the other hand, Kernel Learning approaches present some disadvantages: the computational effort required in order to train a Kernel Learning based SVM is usually far away more consistent than the effort required by the training of a standard SVM model. Moreover, known Kernel Learning algorithms can not learn new similarity functions without the specification of the typology of kernel to be learnt or without an initial set of kernels to be together combined.

Kernel Learning approaches first appeared in 2004, with Lanckriet et al. paper [30], proposing a Kernel Learning approach based on Semi Definite Programming (SDP) [31]. Other approaches have been successively developed by Ong et al. [32] and by Sonnenburg et al. [33], respectively proposing Hyper Kernels and Multiple Kernel Learning (MKL) methods.

The following paragraphs will introduce the main ideas of the afore mentioned approaches, with their respective points of strength and disadvantages. MKL methodology will be widely explained and discussed in the chapter 3, since the main results of the present thesis work are based on MKL techniques.

## **2.2. Kernel Learning with Semi Definite Programming.**

Lanckriet et al. proposed to estimate the kernel matrix as the linear combination of a set  $\overline{K}$  of known kernel functions:

$$K = \sum_{k=1}^{|\overline{K}|} \beta_k \cdot K_k \quad (14)$$

Starting from this simple idea, the authors demonstrated that Kernel Learning can be formulated as an SDP problem. Let recall the standard dual SVM model for classification tasks:

$$\max_{\alpha} 2 \cdot \alpha^T e - \alpha^T G(K) \alpha : C \geq \alpha \geq 0, \alpha^T y = 0 \quad (15)$$

where  $y$  represents the labels vector ( $y_i = \pm 1$ ) and  $G(K)$  is defined as:

$$[G(K)]_{ij} = y_i \cdot y_j \cdot K(x_i, x_j)$$

Merging expression (14) and (15), it is possible to demonstrate that the final model can assume the form:

$$\max_{\alpha, t} 2 \cdot \alpha - \mu^T r \cdot t \quad (16)$$

$$t \geq \frac{1}{r_k} \cdot \alpha^T G(K_k) \alpha \quad k = 1, \dots, |\bar{K}| \quad (17)$$

$$\alpha^T y = 0, C \geq \alpha \geq 0 \quad (18)$$

that is a SDP model, where  $r_k = \text{trace}(K_k)$  and under the adjunctive constraint  $\mu \geq 0$ .

The great advantage of SDP model (16) – (18) is its convexity, ensuring that no local minima might be found. On the other hand, model (16) – (18) has a computational complexity in the order of  $O(|\bar{K}| \cdot m^3)$ , far away superior to the usual  $O(m^2)$  complexity of standard SVM models.

Moreover, the authors experimented their model only within the limits of a transduction setting, and they did not provide a version of the model for the induction tasks.

### 2.3. Hyper Kernels.

The approach developed by Ong et al. is conceptually different from the work of Lanckriet et al., but the final model is again formulated as an SDP problem.

The main idea of the Hyper Kernel approach consists in minimizing the regularized risk with respect to both function  $f$  and the kernel function  $K$ .

Let recall the standard primal SVM model for classification task:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \zeta_i \quad (19)$$

$$y_i \cdot (w^T x_i + b) \geq 1 - \zeta_i, \quad \zeta \geq 0, \quad i = 1 \dots m \quad (20)$$

where  $y_i = \pm 1$  represents the label assigned to each case. The term  $\frac{1}{2} \|w\|^2$  is the regularization term, ensuring the required properties of robustness and uniqueness of the function  $f$ . The term  $C \sum_{i=1}^m \zeta_i$  represents the loss function, that is, the admitted inaccuracy of the function  $f$  with respect to the training data.

Recognizing that the function  $f$  could assume multiple form, and that the loss function could be generally represented as a function  $l(x_i, y_i, f)$ , then the model (19) – (20) can be rewritten as:

$$\min_f \frac{1}{2} \|f\|^2 + C \cdot \sum_{i=1}^m l(x_i, y_i, f) \quad (21)$$

The idea of the Hyper Kernel approach simply consists in adding an adjunctive terms  $\|K\|$  to model (21), in order to minimize also the regularized risk related to the choice of the kernel function:

$$\min_{f,K} \frac{1}{2} \|f\|^2 + C \cdot \sum_{i=1}^m l(x_i, y_i, f) + C_1 \cdot \|K\|^2 \quad (22)$$

where  $C_1$  is an user defined parameters. The introduction of the term  $\|K\|$  require the definition of a Hyper Kernel function, able to provide the dot product of two kernel functions as calculated in some Reproducing Hilbert Space.

Starting from model (22), it is possible to demonstrate that the final model assumes the shape of a SDP model. The final model is convex, and ensure the existence of a unique global minima.

However, also Hyper Kernel approach presents multiple disadvantages. In particular, the SDP models require the definition of more than  $m^2$  variables, where  $m$  is the number of training instance, leading to scalability problems. Moreover, the Hyper Kernel approach does not eliminate the problem of choosing a suitable kernel problem, because such approach requires an “a priori” definition of a Hyper Kernel function, so the problem of choosing a suitable kernel is only moved to a higher level. Lastly, Ong et al. justified their work as a solution able to effectively deal with the presence of attributes with different variances/scale factors; however, such types of issues can be more easily resolved by using a pre processing step, as normalization or standardization of the variables. So, the practical usefulness of such approach could be questionable.

#### **2.4. Multiple Kernel Learning.**

The MKL approach start from the same idea of Kernel Learning with SDP: the kernel function is defined as the linear combination of a set of already known kernel functions, as modelled in the expression (14).

The difference between the two methods consists in the final optimization model used in order to resolve the SVM model once the linear combination of kernel is embedded inside the standard Support Vector Machine model.

In order to develop their model, Sonnenburg et al. started from the classical primal SVM model for classification problem, modifying the regularization term:

$$\min \frac{1}{2} \left( \sum_{k=1}^{|\bar{K}|} \|w_k\| \right)^2 + C \sum_{i=1}^m \xi_i \quad (23)$$

$$y_i \left( \sum_{k=1}^{|\bar{K}|} w_k^T x_i + b \right) \geq 1 - \xi \quad i = 1 \dots n \quad (24)$$

$$\xi \geq 0 \quad (25)$$

The dual version of model (23) – (26) can be written as:

$$\min \gamma \quad (26)$$

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K_k(x_i, x_j) - \sum_{i=1}^n \alpha_i \leq \gamma \quad k = 1 \dots |\bar{K}| \quad (27)$$

$$y^T \alpha = 0, \quad 0 \leq \alpha \leq C \quad (28)$$

It should be noted that a different kernel  $K_k$  has been defined for each  $w_k$ . Moreover, the left side of inequality (28) is the objective function of a standard SVM model; let assume

$$S_k(\alpha) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K_k(x_i, x_j) - \sum_{i=1}^n \alpha_i \quad (29)$$

Interestingly, the dual version of model (26) – (28) can be written as the following:

$$\max \theta \quad (30)$$

$$\sum_{k=1}^{|\bar{K}|} \beta_k \cdot S_k(\alpha) \geq \theta \quad \forall \alpha \in \mathfrak{R}^n \mid 0 \leq \alpha \leq C, y^T \alpha = 0 \quad (31)$$

$$\sum_{k=1}^{|\bar{K}|} \beta_k = 1, \quad \beta_k \geq 0 \quad \forall k \quad (32)$$

The model (30) – (32) deserves some adjunctive explications. Firstly, it should be noted that the model is linear with respect to  $\theta$  and  $\beta$ ; however, line (31) potentially defines infinite linear constraints, because it should be considered one

constraint for each admissible values of the vector  $\alpha \in \mathfrak{R}^n$ . So expressions (30) – (32) define a Semi Infinite Linear Program (SILP) [34] model.

Compared with SDP and Hyper Kernel methods, MKL presents several advantages. In particular, it should be noted that:

1. MKL allows for effective Kernel selection. MKL provides vector  $\beta$  at the end of the optimization process; the  $\beta$  coefficients can be interpreted as the relative importance of each kernel, i.e., kernel functions more influencing on the solution will have a greater weight. Then, MKL can be used as a tool for kernel selection, in the sense that irrelevant kernel will not enter in the solution because their respective  $\beta_k$  will be zero. This particularity is considerably useful during the practical usage of SVM methods. In fact, MKL allows to contemporary experiment several kernels, avoiding the need of singularly test each kernel function. In some sense, with MKL it is possible to pass from an enumerative, discrete search of the best kernel to a search in the continuous space of kernels combination.
2. MKL allows the merge of heterogeneous data. Let re – write the line (24) in the following way:

$$y_i \left( \sum_{k=1}^{|\bar{K}|} w_k^T \Phi_k(x_i) + b \right) \geq 1 - \xi \quad i = 1 \dots n \quad (33)$$

where  $\Phi_k$  is a projection among the attributes of  $x_i$ , i.e. each  $\Phi_k$  selected a different subset of variables. Model (30) – (32) is not modified by this change; however, with such modification each kernel is applied only to a subset of dataset attributes. Let suppose that data under study are composed both by clinical and genetic information; in such a case, projections  $\Phi_k$  will subdivide the two typologies of data, allowing the contemporary experimentation of different kernels on clinical and genetic data. Under this respect, the coefficient  $\beta$  indicate the relative importance of each attributes subset, allowing an interpretation of the SVM models. Bigger  $\beta$  values will indicate a greater importance of the

respective attributes subset, and conversely a very small  $\beta$  value will indicate a irrelevant set of attributes.

3. Effective and scalable algorithm have been developed for the solution of model (30) – (32), and efficient implementation of such methods are public available. This point will be largely discussed in the next chapter.

For sake of clarity, it should be underlined that properties at point 1) and 2) are common to MKL and SDP approach, while the Hyper Kernel approach seems not able to perform kernel selection or features relevance analysis.



### **3. Multiple Kernel Learning for Censored Target.**

#### ***3.1. Merging Support Vector Machine for Censored Target with Multiple Kernel Learning.***

In previous chapters two important research areas were introduced and discussed: the analysis of censored data and the Multiple Kernel Learning techniques.

Both research areas address relevant problems: the analysis of censored data allows a complete utilization of the information/knowledge hidden in censored/truncated dataset; on the other hand, MKL methods greatly improve the already effective and efficient kernel based methods.

In particular, it is worth while to remind that MKL can be used both for automatic kernel selection and for the effective analysis of heterogeneous data (see chapter 2).

*However, the current literature does not report any attempt to merge this two distinct fields: that is, there is not any machine learning technique able to analyze censored data exploiting the advantages of MKL techniques. Let call this hypothetical technique “CT – MKL”.*

It could be stated that maybe there is not the need of creating a technique able to merge MKL and censored data analysis. Instead, it is easy to demonstrate that several “real world” problems could be faced in a much more effective way by an hypothetical CT – MKL technique.

Let have a practical example in order to better explain the need of creating a bridge between censored data analysis and MKL.

In the last decades, genetics studies focused on discovering the causes of complex phenotypes: questions like “which are the genetics characteristics responsible of mental skills?” or “which is the role of genetic predisposition in the process of ageing?” have been largely debated among biology and bioinformatics community.

However, the genetic component influencing the development of complex phenotypes rarely is preponderant with respect to other factors, like for example the characteristics of the ambient where the subjects under study grew and live, the alimentary diet, education, and so on.

Usually, in order to have a global picture of the causes of a particular complex phenotype, heterogeneous data must be collected, registering both genetics and not genetics information. Then all the information should be together examined, in order to take in account the possible interactions among the various factors.

Moreover, some complex phenotypes can be represented as an event occurring during the time; the most clear example is life duration, also known as life expectation. Usually, studies carried out on expectation of life must face the problem of censored data, due to subjects early dropping out from the study.

Synthesizing, the study of interactions among genetics characteristics, not genetics factors and life expectations is a typical problem that could gain great advantages from a technique able to merge MKL and censored data analysis; in fact, the presence of various factors suggest the use of a technique able to deal with heterogeneous data, like MKL; on the other side, the occurrence of censored cases need the application of a technique able to deal with censored data.

Moreover, it should be reminded that MKL can be used as an automatic kernel selection procedure; then, each kernel based methods able to deal with censored data could obtain advantages by MKL techniques, at least in terms of time savings during the parameters optimization phase.

Once explained that it is necessary to create a CT – MKL technique, it must be decided how to realize such ideas. A very straightforward solution would be merging MKL techniques with one of the SVM models able to deal with censored data. As pointed out in the brief literature review given in Chapter 1, SVCR presents several advantages with respect to other SVM implementation for censored data, i.e. the possibility of deal with right, left and interval censored data, and the possibility of using highly efficient optimization algorithms.

*Then, the most natural choice in order to create a CT – MKL algorithm seems to be the fusion between the SVCR model and MKL techniques.*

The objective is obtaining a model with the advantages of both SVCR and MKL techniques, in particular:

1. the ability of dealing with right, left and interval censored data;
2. the possibility of effectively treating heterogeneous data;
3. the ability of performing automatic kernels selection;
4. fast and effective algorithm for the solution of the underlying optimization problem.

The following paragraphs will introduce the CT – MKL model in detail, together with the optimization algorithm and the implementation strategies adopted in order to realize the software code.

### **3.2. CT – MKL: optimization model.**

In this paragraph we will derive the optimization model of CT – MKL technique. Let recall the primal SVCR model as given in paragraph 1.3.2:

$$\min_{w, b, \zeta, \zeta^*} \frac{1}{2} \|w\|^2 + C \left( \sum_{i \in U} \zeta_i^* + \sum_{i \in L} \zeta_i \right) \quad (34)$$

$$w^T \cdot x_i + b - u_i \leq \zeta_i^* \quad \forall i \in U \quad (35)$$

$$l_i - w^T \cdot x_i - b \leq \zeta_i \quad \forall i \in L \quad (36)$$

$$\zeta_i \geq 0 \quad \forall i \in L, \quad \zeta_i^* \geq 0 \quad \forall i \in U \quad (37)$$

With respect to model (34) – (37), we introduce a slightly modification, by setting up a further user defined parameter  $\varepsilon$ ; the modified model have the following form:

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2} \|w\|^2 + C \left( \sum_{i \in U} \zeta_i^* + \sum_{i \in L} \zeta_i \right) \quad (38)$$

$$w^T \cdot x_i + b - (u_i + \varepsilon) \leq \zeta_i^* \quad \forall i \in U \quad (39)$$

$$(l_i - \varepsilon) - w^T \cdot x_i - b \leq \zeta_i \quad \forall i \in L \quad (40)$$

$$\zeta_i \geq 0 \quad \forall i \in L, \quad \zeta_i^* \geq 0 \quad \forall i \in U \quad (41)$$

$\varepsilon$  introduction produces two different effects. Firstly, model (38) – (41) is more easily resolvable, because  $\varepsilon$  relaxes the constraints (39) and (40).

Secondly, the introduction of  $\varepsilon$  allows the use of the so – called “ $\varepsilon$  insensitive loss function”, i.e. the estimations  $f(x)$  of times to event  $y$  are required to be comprised in the interval  $[l - \varepsilon; u + \varepsilon]$ . In other world, high values of  $\varepsilon$  will provide a function  $f(x)$  able to better reproduce the general trends of the data, while low  $\varepsilon$  values will generate a regression function more faithful to the training data.

Once redefined the primal model of SVCR, let merge the MKL model (23) – (26) with model (38) – (41):

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2} \left( \sum_{k=1}^{|\bar{K}|} \|w_k\| \right)^2 + C \left( \sum_{i \in U} \zeta_i^* + \sum_{i \in L} \zeta_i \right) \quad (42)$$

$$\sum_k (w_k^T \cdot \Phi_k(x_i)) + b - (u_i + \varepsilon) \leq \zeta_i^* \quad \forall i \in U \quad (43)$$

$$(l_i - \varepsilon) - \sum_k (w_k^T \cdot \Phi_k(x_i)) - b \leq \zeta_i \quad \forall i \in L \quad (44)$$

$$\zeta_i \geq 0 \quad \forall i \in L, \quad \zeta_i^* \geq 0 \quad \forall i \in U \quad (45)$$

Note that  $w_k$  can be written as  $w_k = \beta_k \cdot w'_k$ , under the further constraints

$\sum_{k=1}^{|\bar{K}|} \beta_k = 1$  and  $\beta_k \geq 0 \quad \forall k$ . For sake of simplicity, thereafter  $x_{ik}$  will be used

instead of  $\Phi_k(x_i)$ ; moreover, index  $k$  will range from 1 to  $|\bar{K}|$ , unless otherwise

specified. In their paper [35], Bach et al. derived a dual version of problem (42) – (45) by treating the primal model as a Second Order Cone Programming (SOCP) optimization problem.

Let define the following cone constraint:  $Cone_k = \{(w_k, t_k) \in \mathfrak{R}^{|\bar{K}|+1} \mid \|w_k\| \leq t_k\}$ ; problem (42) – (45) can be now written as:

$$\min_{w, b, \zeta, \zeta^*} \frac{1}{2} \delta^2 + C \left( \sum_{i \in U} \zeta_i^* + \sum_{i \in L} \zeta_i \right) \quad (46)$$

$$\sum_k (w_k^T \cdot x_{ik}) + b - (u_i + \varepsilon) \leq \zeta_i^* \quad \forall i \in U \quad (47)$$

$$(l_i - \varepsilon) - \sum_k (w_k^T \cdot x_{ik}) - b \leq \zeta_i \quad \forall i \in L \quad (48)$$

$$\sum_k t_k \leq \delta \quad (49)$$

$$(w_k, t_k) \in Cone_k \quad \forall k \quad (50)$$

$$\zeta_i \geq 0, \zeta_i^* \geq 0 \quad \forall i \quad (51)$$

Note that  $\sum_k \|w_k\| \leq \sum_k t_k \leq \delta$ , then model (46) – (51) and model (42) – (45) are equivalent. Taking in account that constraint (50) is self dual, the Lagrangian function of problem (46) – (51) can be written as:

$$\begin{aligned} L = & \frac{1}{2} \delta^2 + C \left( \sum_{i \in U} \zeta_i^* + \sum_{i \in L} \zeta_i \right) + \sum_{i \in U} \alpha_i^* \cdot \left( \sum_k (w_k^T \cdot x_{ik}) + b - (u_i + \varepsilon) - \zeta_i^* \right) + \\ & + \sum_{i \in L} \alpha_i \cdot \left( (l_i - \varepsilon) - \sum_k (w_k^T \cdot x_{ik}) - b - \zeta_i \right) - \sum_{i \in L} B_i \zeta_i - \sum_{i \in U} B_i^* \zeta_i^* + \\ & + \gamma \left( \sum_k t_k - \delta \right) - \sum_k (\lambda_k^T \cdot w_k + \mu_k \cdot t_k) \end{aligned} \quad (52)$$

$$\alpha_i \geq 0 \quad \forall i \in L, \alpha_i^* \geq 0 \quad \forall i \in U \quad (53)$$

$$\gamma \geq 0 \quad (54)$$

$$(\lambda_k, \mu_k) \in Cone_k \quad (55)$$

Let derive the Lagrangian function (52) with respect to variables  $\delta, t, b, \zeta, \zeta^*, w$ :

$$\frac{\partial L}{\partial \delta} = \delta - \gamma = 0 \Rightarrow \delta = \gamma \quad (56)$$

$$\frac{\partial L}{\partial t_k} = \gamma - \mu_k = 0 \Rightarrow \gamma = \mu_k \quad \forall k \quad (57)$$

$$\frac{\partial L}{\partial b} = \sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0 \quad (58)$$

$$\frac{\partial L}{\partial \zeta_i} = C - \alpha_i - B_i = 0 \Rightarrow \alpha_i = C - B_i \quad \forall i \in L \quad (59)$$

$$\frac{\partial L}{\partial \zeta_i^*} = C - \alpha_i^* - B_i^* = 0 \Rightarrow \alpha_i^* = C - B_i^* \quad \forall i \in U \quad (60)$$

$$\frac{\partial L}{\partial w} = \sum_{i=1}^m (\alpha_i^* - \alpha_i) x_{ik} - \lambda_k = 0 \quad \forall k \quad (61)$$

Replacing expressions (56) – (61) in model (46) – (51), we obtain:

$$\min -\frac{1}{2} \gamma^2 - \sum_{i \in U} \alpha_i^* \cdot (u_i + \varepsilon) + \sum_{i \in L} \alpha_i \cdot (l_i - \varepsilon) \quad (62)$$

$$\|\lambda_k\| \leq \mu_k \quad \forall k \quad (63)$$

$$\mu_k = \gamma \quad \forall k \quad (64)$$

$$\lambda_k = \sum_{i=1}^m (\alpha_i^* - \alpha_i) \cdot x_{ik} \quad \forall k \quad (65)$$

$$\sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0 \quad (66)$$

$$0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1 \dots m \quad (67)$$

Model (62) – (67) can be further simplified by unifying constraints (63) – (65):

$$\min -\frac{1}{2}\gamma^2 - \sum_{i \in U} \alpha_i^* \cdot (u_i + \varepsilon) + \sum_{i \in L} \alpha_i \cdot (l_i - \varepsilon) \quad (68)$$

$$\left\| \sum_{i=1}^m (\alpha_i^* - \alpha_i) \cdot x_{ik} \right\| \leq \gamma \quad \forall k \quad (69)$$

$$\sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0 \quad (70)$$

$$0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1 \dots m \quad (71)$$

We can substitute  $\gamma^2$  with  $\gamma$  and transport part of the objective function in the constraints (65):

$$\min \gamma \quad (72)$$

$$\left\| \sum_{i=1}^m (\alpha_i^* - \alpha_i) \cdot x_{ik} \right\|^2 + \sum_{i \in U} \alpha_i^* \cdot (u_i + \varepsilon) - \sum_{i \in L} \alpha_i \cdot (l_i - \varepsilon) \leq \gamma \quad \forall k \quad (73)$$

$$\sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0 \quad (74)$$

$$0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1 \dots m \quad (75)$$

Now, let apply the “kernel trick” to model (72) – (75):

$$\min \gamma \quad (76)$$

$$\sum_{i=1}^m \sum_{j=1}^m (\alpha_i^* - \alpha_i) \cdot (\alpha_j^* - \alpha_j) \cdot K_k(x_{ik}, x_{jk}) + \sum_{i \in U} \alpha_i^* \cdot (u_i + \varepsilon) - \sum_{i \in L} \alpha_i \cdot (l_i - \varepsilon) \leq \gamma \quad \forall k \quad (77)$$

$$\sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0 \quad (78)$$

$$0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1 \dots m \quad (79)$$

In order to simplify the notation, thereafter constraints (77) will be rewritten as:

$$S_k(\alpha, \alpha^*) \leq \gamma \quad \forall k \quad (80)$$

where  $S_k(\alpha, \alpha^*)$  replaces the left side of constraints (77).

After the last transformation, it is easy to recognize that model (76) – (79) could be formulated as a min – max problem; in fact the model try to minimize  $\gamma$ , while  $\gamma$  is greater then the maximum of  $S_k(\alpha, \alpha^*)$ .

In order to exploit the underlying min – max nature of model (76) – (79), let's derive its dual version:

$$L = \gamma + \sum_k \beta_k \cdot (S_k(\alpha, \alpha^*) - \gamma) \quad (81)$$

where  $\beta_k \geq 0 \quad \forall k$ . Setting the derivative of (81) to zero, it is possible to obtain the following min – max problem:

$$\max_{\beta} \min_{\alpha, \alpha^*} \sum_k \beta_k \cdot S_k(\alpha, \alpha^*) \quad (82)$$

$$\sum_k \beta_k = 1 \quad (83)$$

$$\beta_k \geq 0 \quad \forall k \quad (84)$$

with the adjunctive constraints  $\sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0$  and  $0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1 \dots m$ .

Model (82) – (84) need to be further transformed in order to be solved; interesting, this model can be easily transformed in a Semi Infinite Linear Program (SILP):

$$\max \quad \theta \quad (85)$$

$$\sum_k \beta_k = 1 \quad (86)$$

$$\beta_k \geq 0 \quad \forall k \quad (87)$$

$$\sum_k \beta_k \cdot S_k(\alpha, \alpha^*) \geq \theta$$

$$\forall \alpha, \alpha^* \in \mathfrak{R}^n \mid \left( \sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0 \quad \text{and} \quad 0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1 \dots m \right) \quad (88)$$



It should be noted that problem (85) – (88) is linear with respect to variables  $\theta$  and  $\beta$ . Unfortunately, infinite linear constraints should be considered in order to find a solution; in fact there is a linear constraint for each possible configuration of variables  $\alpha, \alpha^*$  respecting the constraints

$$\Omega = \left\{ (\alpha_i^*, \alpha_i) \mid \sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0 \quad \text{and} \quad 0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1 \dots m \right\} \quad (89)$$

It must be noted that model (85) – (88) is formally equivalent to the model obtained by Sonnenburg et al. [33]. Using the results reported in [33], we can state that:

**Proposition 2:** model (85) – (88) has a solution because the corresponding primal (38) – (41) is feasible and bounded [36]. Note that problem (38) – (41) is a convex quadratic problem with a unique finite minimum, provided that  $u_i \geq l_i \forall i$  and moreover that elements of vectors  $u$  and  $l$  are finite.

**Proposition 3:** there is no duality gap because the cone  $M$  defined as

$$M = \text{cone} \left\{ \left[ S_1(\alpha, \alpha^*), \dots, S_{|\bar{K}|}(\alpha, \alpha^*) \right] \mid \sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0 \quad \text{and} \quad 0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1 \dots m \right\}$$

is a closed set [36] [34].

Regarding the Proposition 3, it must be noted that Sonnenburg et al. stated that the cone  $M$  is closed when the  $\varepsilon$  – insensitive loss function is used. The loss function used in model (38) – (41) and the  $\varepsilon$  – insensitive loss function produce formally equivalent dual SVM models, i.e. formally equivalent  $S_k(\alpha, \alpha^*)$ . Thus, the closedness of cone  $M$  for the  $\varepsilon$  – insensitive loss function implies the closedness of cone  $M$  for model (85) – (88).

### 3.3. CT – MKL: solving algorithm.

SILP problems can be effectively solved by using “exchange methods” algorithms. Such class of algorithms are known to converge [34] even if the convergence rates are not known. In the particular case of CT – MKL model, the adoption of exchange methods leads to a simple solution strategy, that can be synthesized as:

1. solve problem (85) – (88) considering a limited set of constraints (“restricted master problem” solution);
2. update the set of constraints adding a new, yet unsatisfied constraint.  
Stop if not unsatisfied constraints exists.

The principal issues of this strategy is the generation of the new constraint(s). Depending on adopted constraints generation procedure, two solving algorithms can be defined, the “wrapper algorithm” and the “chunking algorithm”, both proposed in [33].

#### 3.3.1. The wrapper algorithm

The wrapper algorithm is the simplest implementation of the solution strategy based on the exchange methods. Model (85) – (88) is subdivided into an inner and an outer sub problem. The two sub problems are alternatively resolved, using the output of the first sub problem as input for the second one, and vice versa.

In particular, the “restricted master problem” constitutes the outer loop. During the outer loop, model (85) – (88) is resolved as a simple linear program, by using the commercial solver CPLEX. Once determined the optimal  $\beta^*$  and  $\theta^*$ , they can be used in the inner loop in order to check the existence of a unsatisfied constraint.

The determination of an unsatisfied constraint can be easily performed; constraint (88) becomes, with  $\beta$  fixed to  $\beta^*$ :

$$\nu = \min \sum_{i=1}^m \sum_{j=1}^m (\alpha_i^* - \alpha_i) \cdot (\alpha_j^* - \alpha_j) \cdot K(x_i, x_j) + \sum_{i \in U} \alpha_i^* \cdot (u_i + \varepsilon) - \sum_{i \in L} \alpha_i \cdot (l_i - \varepsilon) \quad (90)$$

$$\sum_{i \in U} \alpha_i^* - \sum_{i \in L} \alpha_i = 0 \quad \text{and} \quad 0 \leq \alpha_i, \alpha_i^* \leq C \quad i = 1 \dots m \quad (91)$$

where  $K(x_i, x_j) = \sum_k \beta_k \cdot K_k(x_{ik}, x_{jk})$ . Model (90) – (91) is the dual version of SVCR model, and then it is easily resolvable with any decomposition technique specifically developed for SVM models.

Once obtained the optimal value  $\nu^*$ , a violated constraint can be found if  $\nu^* < \theta^*$ , and then the unsatisfied constraint can be added to the restricted master problem. The pseudocode of the algorithm is reported below;  $\varepsilon_{MKL}$  is a user defined convergence criterion, while  $\nu$  and  $\Omega$  were defined in lines (89) – (90).

**Algorithm 1: wrapper algorithm**

```

initialization
     $S^0 = 1, \theta^1 = -\infty, \beta_k^1 = \frac{1}{|K|} \forall k$ 

main loop:
    for  $t = 1, 2, \dots$ 
         $\alpha^t = \arg \min_{\alpha \in \Omega} \nu$ 
         $S^t = \sum_k \beta_k^t \cdot S_k^t$  where  $S_k^t = S_k(\alpha^t)$ 

        if  $\left| 1 - \frac{S^t}{\theta^t} \right| \leq \varepsilon_{MKL}$  then break

     $(\beta^{t+1}, \theta^{t+1}) = \arg \max_{\theta} \theta : \sum_k \beta_k = 1, \beta_k \geq 0 \forall k, \sum_k \beta_k \cdot S_k^r \geq \theta \quad r = 1, 2, \dots, t$ 

    end for
    
```

A further consideration about the wrapper algorithm. When the optimum is reached,  $S^t = \theta^t$ . For practical reason, it is not efficient to stop the algorithm only

when the equality hold; the parameter  $\varepsilon_{MKL}$  allows the main loop to break when  $S'$  and  $\theta'$  are “sufficiently” similar.

### 3.3.2. The chunking algorithm

Even if the wrapper algorithm is very easy to implement, there is the possibility of defining a more complex algorithm, i.e. the chunking algorithm, that is theoretically much faster.

The wrapper algorithm optimizes the  $\alpha$  variables at each iteration, even if the  $\beta$  variables are not yet optimal, and this procedure is unnecessary costly.

The idea of the chunking algorithm is to improve  $\beta$  variables immediately after the improvement of  $\alpha$  variables. In other word, the main loop of chunking algorithm cyclically repeats two steps:

1. slightly improve the values of  $\alpha$  variables. If the solution is optimal, stop.
2. slightly improve the values of  $\beta$  variables using the new  $\alpha$  found at step one. Go back to step one.

In some sense, the chunking algorithm optimize both sets of variables a step at the time. In order to implement the chunking algorithm, we need (a) an algorithm able to iteratively optimize the  $\alpha$  variables, and (b) an efficient procedure able to recomputed  $\beta$  once the new  $\alpha$  values are available.

Luckily, the state of the art algorithms for the solution of SVM models (SVM<sup>Light</sup>, LibSVM) are based on iterative decomposition techniques [37] [38], and so precondition (a) is easily satisfied. Regarding precondition (b), the new values of  $\beta$  variables can be easily computed using the linear program already used in the wrapper algorithm.

So, let suppose that a Decomposition Technique  $DT$  is used in order to resolve the SVM models;  $DT$  iteratively optimizes the  $\alpha$  variables through a finite sequence of iterations. Given  $DT$ , the pseudo code of chunking algorithm can be described as below. Symbols  $\nu$  and  $\Omega$  were defined in lines (89) – (90).

**Algorithm 2: chunking algorithm**

initialization

$$S^0 = 1, \theta^1 = -\infty, \beta_k^1 = \frac{1}{|K|} \forall k$$

main loop:

*for*  $t = 1, 2, \dots$ 1) Check optimality condition for the problem  $\min_{\alpha \in \Omega} \nu$ ; if optimal, then stop.2) Execute *one DT* iteration on problem  $\min_{\alpha \in \Omega} \nu$ ; let  $\alpha^t$  be the results of this single iteration.

3)  $S^t = \sum_k \beta_k^t \cdot S_k^t$  where  $S_k^t = S_k(\alpha^t)$

4) *if*  $\left| 1 - \frac{S^t}{\theta^t} \right| \geq \varepsilon_{MKL}$

$$(\beta^{t+1}, \theta^{t+1}) = \arg \max_{\theta} \theta$$

$$\sum_k \beta_k = 1, \beta_k \geq 0 \forall k, \sum_k \beta_k \cdot S_k^r \geq \theta \quad r = 1, 2, \dots, t$$

*else*

$$\theta^{t+1} = \theta^t$$

*end if**end for*

Evidently, the chunking algorithm is rather similar to the wrapper algorithm; the main difference consists in the use of intermediate  $\alpha$  values. Moreover, optimality is directly checked on problem  $\min_{\alpha \in \Omega} \nu$ , and then the termination of the algorithm depends by the optimality conditions of the used *DT*.

Finally, it is worthwhile to note that chunking algorithm is extremely modular, as well as the wrapper algorithm. In fact, both the chunking and the wrapper algorithm are composed by a first algorithm (*DT*), that iteratively optimize the SVM model, and by a second algorithm (e.g. the well known simplex method) that optimizes the LP problem reported at the fourth point.

The modularity of the wrapper/chunking algorithm allows a very easy and fast extension of MKL algorithm to different SVM models. For example, once implemented the MKL version of SVM classification model, it is sufficient to substitute the *DT* procedure with another decomposition technique in order to obtain the MKL version of SVM regression model, the MKL version of One class SVM models, and so on.

### **3.4. CT – MKL: implementation.**

MKL version of SVM model for classification and regression are fully implemented within the open source software suite SHOGUN<sup>3</sup>.

SHOGUN is a learning toolbox mainly focused on kernel methods and especially on large scale Support Vector Machine models. It is written in C++ programming language, meantime providing interfaces for Matlab, Python, Octave and R.

The main feature of the SHOGUN toolbox consists in the implementation of Multiple Kernel Learning algorithms, i.e. SHOGUN allows the user to train SVM classification or regression models using the MKL chunking algorithm (see Algorithm 2: chunking algorithm). In particular, the toolbox employs SVM<sup>Light</sup> as *DT*, and the commercial software CPLEX as linear solver.

As stated in paragraph 3.3.2, the chunking algorithm can be easily adapted in order to solve new SVM models, like SVCR. In fact, it is sufficient to utilize a *DT* able to solve the new SVM model. So, in order to implement the CT – MKL algorithm, it is sufficient to modify the SVM<sup>Light</sup> code contained in the SHOGUN toolbox, allowing SVM<sup>Light</sup> to optimize SVCR models.

So, let focus on the modifications performed on SVM<sup>Light</sup> code in order to solve the SVCR optimization problem.

SVM<sup>Light</sup> is able to deal with either classification and regression SVM models. Actually, SVM<sup>Light</sup> inner operation relies on the *optimize\_convergence* function; both classification and regression models are optimized by this function, after their specification in a standard primal form.

---

<sup>3</sup>website: <http://www.shogun-toolbox.org/>

The trick used in order to let SVM<sup>Ligth</sup> deal with SVCR models consists in providing treating SVCR models as regression models. This trick is applicable because:

1. SVCR model is formally reducible to a standard SVM regression model;
2. the function *optimize\_convergence* is able to optimize a wide range of SVM models, including the models that follow this template:

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2} \|w\|^2 + C \left( \sum_{i \in G} C_i \cdot \zeta_i^* + \sum_{i \in F} C_i \cdot \zeta_i \right) \quad (92)$$

$$w^T \cdot x_i + b - p_i \leq \zeta_i^* + \varepsilon \quad \forall i \in P \quad (93)$$

$$q_i - w^T \cdot x_i - b \leq \zeta_i + \varepsilon \quad \forall i \in Q \quad (94)$$

$$\zeta_i \geq 0 \quad \forall i \in Q, \quad \zeta_i^* \geq 0 \quad \forall i \in P \quad (95)$$

where  $P$  and  $Q$  are two generic subsets of the instances set,  $C_i$  are cost parameters, and  $p_i, q_i \in \mathfrak{R}$  are generic quantities assigned to each instance.  $P$  and  $Q$  can be overlapping.

It is easy to recognize that both regression SVM and SVCR model can be reduced to model (92) – (95). Moreover, it should be noted that the function *optimize\_convergence* actually optimizes the dual, kernelized version of model (92) – (95). Let examine the code used by the file `SVML_ligth.cpp` in order to standardize SVM regression model before submitting the optimization problem to the function *optimize\_convergence*:

SVR\_ligth.cpp code:

```
....
// set up regression problem in standard form
docs=new INT[2*totdoc];
label=new INT[2*totdoc];
c = new double[2*totdoc];
```

```

for(i=0;i<totdoc;i++) {
    docs[i]=i;
    j=2*totdoc-1-i;
    label[i]=+1;
    c[i]=labels->get_label(i);
    docs[j]=j;
    label[j]=-1;
    c[j] = labels->get_label(i);
}
totdoc*=2;
...

```

It is worthwhile to note that the standardization of the problem require the duplication of the instances, i.e., each instance of the original dataset is duplicated before calling *optimize\_convergence*. The SVR\_light.cpp code produce the following primal SVM regression model:

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^m \zeta_i^* + \sum_{i=1}^m \zeta_i \right) \quad (96)$$

$$w^T \cdot x_i + b - y_i \leq \zeta_i^* + \varepsilon \quad \forall i = 1 \dots m \quad (97)$$

$$y_i - w^T \cdot x_i - b \leq \zeta_i + \varepsilon \quad \forall i = 1 \dots m \quad (98)$$

$$\zeta_i, \zeta_i^* \geq 0 \quad \forall i = 1 \dots m \quad (99)$$

Now, let examine the code of the modified SVML\_ligth.cpp file:

Modified SVR\_light.cpp code:

```

...
// set up regression problem in standard form
docs=new INT[2*totdoc];
label=new INT[2*totdoc];
c = new double[2*totdoc];

```



```

for(i=0;i<totdoc;i++) {
    docs[i]=i;
    j=2*totdoc-1-i;
    label[i]=+1;
    c[i]=labels_low[i];
    docs[j]=j;
    label[j]=-1;
    c[j] = labels_up[i];
}
totdoc*=2;
...

```

The modified SVR\_light.cpp code produces the following primal SVCR model:

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^m \zeta_i^* + \sum_{i=1}^m \zeta_i \right) \quad (100)$$

$$w^T \cdot x_i + b - u_i \leq \zeta_i^* + \varepsilon \quad \forall i = 1 \dots m \quad (101)$$

$$l_i - w^T \cdot x_i - b \leq \zeta_i + \varepsilon \quad \forall i = 1 \dots m \quad (102)$$

$$\zeta_i, \zeta_i^* \geq 0 \quad \forall i = 1 \dots m \quad (103)$$

It is easy to recognize as models (96) – (99) and (100) – (103) are equivalent, and moreover that both models are reducible to problem (92) – (95).

Regarding model (100) – (103), each  $u_i$ ,  $l_i$  must be specified, even if they take infinite values (remind that right censored data are characterized by  $u_i = +\infty$ , while left censored data by  $l_i = -\infty$ ). The representation of infinite values within a C++ computer program could be achieved by using the function `std::numeric_limits<double>::max()` included in the library “limits”; however, the use of infinite values can cause numerical problems during the optimization procedures.

Therefore,  $u_i$  and  $l_i$  that should be set to  $\pm\infty$  are instead set to a very high/low value  $\pm T$ . Whether  $T$  is sufficiently large, then the solution of the model is not affected by the substitution.

## 4. Experimentation.

### 4.1. Experiments Organization.

Theoretical properties of new algorithms/methodologies must be practically demonstrated through extensive experimentations.

The first step of each experimental phase is the definition of investigation objectives. In other words, it is necessary to exactly define which are the properties/capabilities that we want to check, and the experimentation protocol used in order to perform the analysis.

In the previous chapters we stated that, from a theoretical point of view, the MKL – CT algorithm should provide numerous advantages. The two principal applications of MKL – CT can be described as:

1. Automatic kernel selection

one of the most notable problem related to the use of kernel methods consists in the selection of a suitable kernel. This problem holds also when kernel methods are applied to survival analysis. The MKL – CT approach can be used in order to choose the best kernel (or the best combination of kernels) among a pre defined set of kernel functions.

2. Resolving problems involving heterogeneous data.

In several context the data to be analyzed are heterogeneous, that is, variables under study can be grouped in clusters. Each cluster has a proper origin, nature and semantic. MKL –CT allows a separate treatment of variables, i.e. different kernels can be applied to each group of variables. The kernels are then linearly weighted, and the weights are automatically chosen by the algorithm. Interestingly, kernel weights can be also used in order to judge the relevance of each variables group.

The present chapter reports the experimentations conduced in order to demonstrate the effective validity of MKL – CT algorithm when applied to automatic kernel selection problems and to the analysis of heterogeneous data. Moreover, other MKL – CT properties have been experimented and discussed.

Computational results are organized in four subsection:

1. Heterogeneous survival data: effects of the interaction among genetic and physiological factors in the ageing process.

In this subsection a survival dataset including both genetic and physiological – clinical data is analyzed. The objective of the experimentation is to demonstrate the usefulness of MKL – CT for the analysis of heterogeneous data.

2. CT – MKL on left, right, double and interval censored data.

Four synthetic dataset were prepared and analyzed. Each dataset respectively presents left, right, double and interval censored outcomes. Objectives of these experimentations are the demonstration that MKL – CT algorithm can work on all types of censored data and the evaluation of MKL – CT performances for kernel selection problems.

3. CT – MKL applied on real survival datasets

In this subsection, four public survival datasets coming from “real world” studies are analyzed, in order to assess the validity of MKL – CT algorithm for kernel selection on real data.

4. Comparison of MKL – CT and SVCR algorithms in terms of time spent during the training phase

Finally, MKL – CT training times are reported and compared with the training times of SVCR.

Even if the four subsections have diverse objectives and use different datasets, the experiments present some common elements.

Firstly, for each dataset we used the same experimentation protocol (i.e. dataset subdivision in training and test set, number of cross validation etc.). The experimentation protocol is illustrated in the subparagraph 4.1.1.

Moreover, MKL – CT algorithm performances were always compared to the results of two other algorithms:

1. SVCR, i.e., a “single kernel” SVM model able to deal with censored data, see paragraph 1.3.2 and model (38) – (41) in paragraph 3.2
2. Parametric Survival Models (PSM, paragraph 1.2).

In particular, the utilization of SVCR models allowed the comparison of the automatic kernel selection procedure performed by MKL – CT with respect to the enumerative kernel selection method carried out for the SVCR models.

Parametric survival models were used for comparing MKL –CT algorithm with the “classical” statistics methods for survival analysis.

Finally, all the results were evaluated through two performance measures: the Average Absolute Error (AAE) and the Rank Score (RS). Both AAE and RS are largely discussed in the subparagraph 4.1.2.

#### **4.1.1. Experimentation protocol**

Each dataset was analyzed with the following protocol:

1. Pre – processing: missing values (whether present) were simply substituted by mean values. Each dataset was subdivided into a training and a test set, with a ratio between the number of training and test instances depending by the size of the entire dataset. Training set values were successively normalize in the interval  $[0;1]$ , while the test set was normalized according with the max/min values of training set attributes. Finally, training set was subdivided in five fold, each fold roughly containing the same ratio of censored and uncensored cases with respect to other folds.
2. Training phase: MKL – CT algorithm, SVCR and PSM were experimented on each dataset. For each algorithm, the learning parameters were optimized on the training set through a cross validation procedure, using the subdivision in five folds carried out in the pre – processing phase. Among all the possible parameters configurations, the setting with the minimum AAE was chosen.  
PSM presents only one parameter, i.e. its parametric function, chosen among Gaussian, Normal, Log – Normal, Weibull, Logistic, Log – Logistic and Exponential functions.

MKL – CT and SVCR needed the definition of  $C$  and  $\varepsilon$  parameters (see model (38) – (41) in paragraph 3.2 and Table 2), beyond the choice of the kernel function. Used kernels are summarized in Table 3.

*It is essential to note that SVCR models need to enumeratively evaluate each single kernel; so, given five possible values for the  $C$  parameter, three values for the  $\varepsilon$  parameter, and the seventeen kernel summarized in Table 2, the SVCR algorithm required 255 cross validation procedures for each dataset. On the other hand, MKL – CT is able to automatically detect the best combination of kernel to be used, then the only parameters to be chosen are the  $C$  and  $\varepsilon$  parameters, for a total of 15 different configuration to be tested trough cross validation. This difference between the two algorithms brings to a dramatic dissimilarity in terms of total time spent during the training phase.*

3. Test phase: once determined the best parameters set, for each dataset and for each algorithm a final model on the training set were produced; the final model was evaluated on the test set using AAE and RS measures

It is worth while to note that the afore described experimental protocol should provide unbiased estimation of the real performance of the tested algorithm, due to the combined use of the cross validation procedure and independent test sets.

**Table 2: values of parameters  $\varepsilon$  and  $C$**

Parameter	Values
$C$	[0.01, 0.1, 1, 10, 100]
$\varepsilon$	[0.01, 0.1, 1]

**Table 3: Kernel configurations**

Kernel Name	Formula	Parameters
Gaussian	$K_G(x_i, x_j) = \exp\left(-\frac{\ x_i - x_j\ ^2}{\gamma}\right)$	$\gamma = [0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100]$
Polynomial	$K_P(x_i, x_j) = \left(\sum_{k=1}^n x_i^k \cdot x_j^k + d\right)^g$	$d = [1, 0]$ $g = [2, 3]$ Normalized <sup>4</sup> = [true, false]

### 4.1.2. Performance measure

Performances evaluation is one of the main research streams in the Machine Learning field. The reason of the need of good performance measures is quite obvious; model selection is almost exclusively performed by comparing models performances, so it is necessary to define robust and reliable methods for measuring the validity of each model.

Censored data show several difficulties during the performance evaluation phase. Usually, for a generic censored instance  $x_i$ , we don't know its real outcome  $y_i$ , but only its upper and lower bounds  $u_i$  and  $l_i$ . However, a generic predictive algorithm will directly provide an estimation of  $y_i$ , i.e.  $\hat{y}_i$ . The problem consists in estimating the closeness between  $y_i$  and  $\hat{y}_i$  by using only  $\hat{y}_i$ ,  $u_i$  and  $l_i$ .

In order to overcome such problem, we used two different performance measures, namely the Absolute Average Error (AAE) and the Rank Score (RS) [11]

The AAE takes in account the cases when the estimated label  $\hat{y}_i$  lays out of the interval  $[l_i; u_i]$ . In other words, when  $\hat{y}_i$  belongs to the interval  $[l_i; u_i]$  the predictions is correct (or, equivalently, the error is equal to zero). Otherwise, the error is equal to the difference between  $\hat{y}_i$  and  $u_i$  (whether  $\hat{y}_i > u_i$ ) or between

---

<sup>4</sup> Normalized Kernel:  $K_n(x_i, x_j) = \frac{K(x_i, x_j)}{\sqrt{K(x_i, x_i) \cdot K(x_j, x_j)}}$

$\hat{y}_i$  and  $l_i$  (whether  $\hat{y}_i < l_i$ ). More formally, the absolute error  $AE_i$  for the generic  $i^{\text{th}}$  predictions can be calculated as:

$$AE_i = \max(\max(0, l_i - \hat{y}_i), \max(0, \hat{y}_i - u_i))$$

The final AAE is obtained by averaging the single  $AE_i$  on all the test instances. Interestingly, this measure can be used for any type of censored data, since AAE does not require any condition about the interval  $[l_i; u_i]$ . Unfortunately, censored data bring only limited information, and so the AAE can give only a limited help for evaluating the performance of a predictive algorithm. For example, let us considering the case of a right censored instance, i.e. the case  $u_i = +\infty$ . In this case, all the predictions laying in the interval  $[l_i; +\infty]$  will be equally correct, including those predictions that are unreasonable with respect to the problem under analysis.

Even if there are not known methods able to overcome such problem, we can contemporarily use another performance measure, in order to judge the predictions quality under multiple perspectives. Of course, this second performance measure should provide information not redundant with respect to the AAE.

Thus, we decided to use the Rank Score metric. RS is based on the same principles of the Area Under the Curve (AUC) measure. In order to introduce the RS, let us give two important definitions:

1. two instances  $x_i$  and  $x_j$  are *comparable* if their respective intervals  $[l_i; u_i]$  and  $[l_j; u_j]$  are not overlapping.
2. two predictions  $\hat{y}_i$  and  $\hat{y}_j$  are *swapped* if their intervals are not overlapping and if one of the following conditions holds:

$$\begin{cases} \hat{y}_i > \hat{y}_j & \text{and } u_i < l_j \\ \hat{y}_i < \hat{y}_j & \text{and } l_i > u_j \end{cases}$$



In practice, when comparability holds it is possible to give an order to the intervals. Meantime, two predictions are swapped when they are in the opposite order with respect to their respective interval.

Now, we can define the RS metric as:

$$RS = \frac{\#Comparable - \#Swapped}{\#Caomparable}$$

The RS metric ranges between 0 and 1; a value of 0.5 would mean random guess, while 1 means perfect ordering. It is worth while to note that RS shows the same characteristic of the already cited AUC metric.

## ***4.2. Heterogeneous survival data: effect of the interaction among genetic and physiological factors for the ageing process.***

In this section we analyze the effective advantages offered by the MKL – CT algorithm for the analysis of heterogeneous data. In particular, our objective is to determine whether the weighted combination of a set of kernel functions (as provided by the MKL – CT algorithm) is actually more effective than the use of a single kernel.

### **4.2.1. Carolei dataset Description**

The Department of Cell Biology of the University of Calabria carried out an extensive monitoring of the healthy status of the elderly population in Calabria (Southern Italy). In this frame, a consistent number of phenotypic and genetic data associated to the rate and the quality of aging have been collected. Here and thereafter, this dataset will be named “Carolei dataset”.

Carolei dataset includes 69–99 years old subjects (125 subjects, 45 males and 80 females; median ages 80 and 81 years respectively). All the subjects were born in Calabria (southern Italy) and their ancestry in the region had been ascertained up

to the grandparents generation. The sample had been recruited in the frame of a study carried out in the municipality of Carolei in order to evaluate the quality of aging of the elderly people living in this municipality. In this study, phenotypic information were collected by using the ECHA questionnaires (<http://biologia.unical.it/echa/results.htm>). Vital status at 36 months after the visit was traced for 104 subjects (83.2%) of the sample through the register of the population of this municipality. All the subjects had given informed consent for studies on aging.

#### *Anthropometric and geriatric measures*

The physical examination included the record of height, weight, knee-to-floor height and waist and hip circumferences. Cognitive function was assessed by Mini Mental State Examination (MMSE) test [39]. Since the test is affected by age and educational status, the scores were normalized for these variables. Hand Grip strength was measured by using a handheld dynamometer (SMEDLEY's dynamometer) while the subject was sitting with the arm close to his/her body. The test was repeated three times with the stronger hand. The maximum of these values was used in the analysis, after normalization for age, sex and height. Depression was assessed by the short form (15 items) of the Geriatric Depression Scale (GDS) [40]. Functional activity was assessed by using a modification of the Katz Index of ADL [41] and IADL index. The assessment was based on what the subject was able to do at the time of the visit. Health status was ascertained by medical visit carried out by a geriatrician, who also conducted a structured interview including questions on common diseases occurred in the past.

#### *DNA analysis*

DNA was prepared from blood buffy-coats according to standard procedures and stored at  $-20^{\circ}\text{C}$  until use. APOE genotyping (alleles e2, e3, e4) was carried out according to the protocol described in [42]. SSADH genotyping (alleles C and T) was carried out according to the protocol described in [43].

*Final dataset*

The final Carolei dataset includes 104 subjects, described by 32 features (see Table 4). This dataset is a typical example of right censored survival data, i.e., the event under study is “death”, and the registered outcome consists of both time – to – event and censored status. Over 104 subjects, only 21 subject died during the observational period; the remaining samples are censored.

Moreover, it should be noted that the dataset is evidently composed by two diverse groups of attributes: the first eighteen attributes describe the physiological status, while the remaining attributes are evidently related to the genetic profile of the subjects. Thus, the Carolei samples can be classified as “heterogeneous data”.

**Table 4: Carolei dataset description**

<b>Attribute Name</b>	<b>Attribute Description</b>	<b>Values</b>
Age	Age	Real
Sex	Sex	Binary
Height	height	Real
Weight	weight	Real
ADL	Activities of Daily Living	Real
IADL	Instrumental Activities of Daily Living	Real
MMSE	Minimal Mental State Examination	Real
HG	Hand Grip (corrected for age and education level)	Real
GDS	Geriatric Depression Scale	Real
SRHS	Self Reported Health Status	Real
diabetes	Presence of Diabetes	Binary
hypertension	Presence of Hypertension	Binary
AP	Presence of Angina Pectoris	Binary
HF	Presence of Heart Failure	Binary
IHR	Presence of Irregular Heart Rhythm	Binary
Heart Attack	Previous Heart Attack	Binary
stroke	Previous Stroke	Binary
cancer	Presence of Cancer	Binary

APOE_a22	Genotype a22 for SNP APOE	Binary
APOE_a33	Genotype a33 for SNP APOE	Binary
APOE_a23	Genotype a23 for SNP APOE	Binary
APOE_a34	Genotype a34 for SNP APOE	Binary
mtDNA_K	Genotype K for mtDNA	Binary
mtDNA_T	Genotype T for mtDNA	Binary
mtDNA_H	Genotype H for mtDNA	Binary
mtDNA_U	Genotype U for mtDNA	Binary
mtDNA_W	Genotype W for mtDNA	Binary
mtDNA_X	Genotype X for mtDNA	Binary
mtDNA_J	Genotype J for mtDNA	Binary
mtDNA_OTHERS	Other genotypes for mtDNA	Binary
SSADH_a12	Genotype a12 for SNP SSADH	Binary
SSADH_a11	Genotype a11 for SNP SSADH	Binary
Time to event	Days from the visit to death or to the end of follow up	Real
Censored	Censored status	Binary

#### 4.2.2. Experiments on Carolei dataset

The experiments performed on the Carolei dataset slightly differ from the experimentation protocol described in paragraph 4.1.1. The main differences regard the application of MKL – CT algorithm. As explained in paragraph 2.4, the MKL approach allows the application of each kernel to a diverse subset of attributes. Then, in order to separately treat the two attributes subgroups present in the Carolei dataset, we applied one sets of kernel functions to the physiological attributes group (variables 1 – 18) and another set of kernel functions to the genetic attributes group (variables 19 – 33).

Both kernel functions sets were composed by a series of Gaussian kernels, (see Table 3). Polynomial kernels were not used, neither for the MKL – CT algorithm neither for the SVCR models.

Moreover, we repeated the training phase twice, once using the cross validation procedure and then using the Leave – One – Out (LOO) method. We preferred to

also use the LOO performances estimation procedure because the size of the dataset is particular small, and the results provided by the cross validation could still be biased. Finally, it must be reported that the Carolei training test was composed by eighty – two samples (sixteen uncensored), while the test set counted twenty – two cases (five uncensored).

The results related to the Carolei dataset are reported in the following tables:

**Table 5: Carolei dataset, results obtained using the cross validation procedure**

	<b>MKL – CT</b>	<b>SVCR</b>	<b>PSM</b>
<b>AAE on training set</b>	208,3594	208,3591	1957,7797
<b>RS on training set</b>	0,5323	0,6454	0,3427
<b>AAE on test set</b>	299,0423	299,2000	777,2715
<b>RS on test set</b>	0,6842	NA <sup>5</sup>	0,6315

**Table 6: Carolei dataset, results obtained using the LOO procedure**

	<b>MKL – CT</b>	<b>SVCR</b>	<b>PSM</b>
<b>AAE on training set</b>	211,6524	211,3852	1570,3545
<b>RS on training set</b>	0,0621	0,0510	0,4625
<b>AAE on test set</b>	298,9073	298,0151	777,2715
<b>RS on test set</b>	0,6842	0,4632	0,6315

The tables of results indicates that both the MKL – CT algorithm and the SVCR models have comparable results, while PSM provided larger errors. However, it is interesting to analyze the kernels selected by the two kernel methods.

The SVCR algorithm selected a Gaussian kernel with  $\gamma = 0.01$  ( $C = 0.1$ ,  $\varepsilon = 0.1$ ) using the cross validation procedure, and a Gaussian kernel with  $\gamma = 50$  ( $C = 10$ ,  $\varepsilon = 0.01$ ) by using the LOO method.

The MKL – CT algorithm selected the same kernel combination by using the cross validation procedure and the LOO method. In particular, MKL – CT

---

<sup>5</sup> RS not calculable because all predictions were identical

selected only a Gaussian kernel with  $\gamma = 0.01$  ( $C = 0.01$ ,  $\varepsilon = 1$ ). That is, the weight of the selected kernel was one while other kernels weights were zeros. The selected kernel belonged to the set of kernels applied to the subgroups of attributes describing the genetic profiles of the subjects.

In other words, the MKL – CT approach provided results equivalent to SVCR results models by using only a part of the entire dataset, specifically by using the subgroup of genetic attributes.

From this point of view, the MKL – CT performed a feature selection procedure, in the sense that kernels assigned to the irrelevant features received null weights.

### ***4.3. CT – MKL on left, right, double and interval censored data.***

In the present section we demonstrate that the MKL – CT algorithm is able to deal with left, right, double and interval censored data. In order to achieve such objective, we used a public regression dataset, namely the Fried dataset<sup>6</sup>.

The uncensored outcome of the Fried dataset was modified in order to become censored; in particular, we censored the Fried dataset outcome four times, respectively creating a left censored, a right censored, an interval censored and a double censored dataset.

In other words, we analyzed Fried dataset four times, each time manipulating the outcome in order to have a different type of censored outcome.

We chose to modify a regression dataset because, as far as we know, there are not public datasets with interval or double censored outcomes. Moreover, the available left censored public datasets are usually composed by only a few covariates, and with a limited number of samples.

Finally, the use of the same set of covariates with different censored outcomes allowed us to analyze the effect of different censoring mechanisms on the performance of the used algorithms.

---

<sup>6</sup> <http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html>

### 4.3.1. Fried dataset Description

Fried dataset is composed by ten synthetic variables,  $z_1, z_2, \dots, z_{10}$ ; the values of each variable were randomly sampled from an uniform distribution. All variables range in the interval  $[0;1]$ . Beyond the ten covariates, Fried dataset includes also a deterministically generated outcome  $y$ :

$$y = 10 \cdot \sin(\pi \cdot z_1 \cdot z_2) + 20 \cdot (z_3 - 0.5)^2 + 10 \cdot z_4 + 5 \cdot z_5 + \sigma(0,1)$$

where  $\sigma(0,1)$  is a normal distributed noise, with mean zero and standard deviation one. The total number of samples was 40768, from which we randomly sampled 1004 instances, subdivided between the training set (512 cases) and the test set (492 cases). The outcome of the training set were successively censored in four different way, as described below; test set outcomes were not modified. For each censoring mechanism, approximately 25% of training cases were randomly selected in order to be censored.

*Left censoring mechanism:*

For the generic  $i^{\text{th}}$  selected case:

- a.  $l_i = -0.001$
- b.  $u_i = u_i + [\max_i(u_i) - u_i] \cdot rand_i$

where  $rand_i$  is a random value in the interval  $[0;1]$  sampled by a uniform distribution. Note that  $l_i = -0.001$  is equivalent to  $l_i = -\infty$  for this dataset, because  $y_i$  is expected to be strictly positive. Meanwhile,  $u_i$  is modified by adding a random value, without exceeding the limit of  $\max_i(u_i)$ .

*Right censoring mechanism:*

For the generic  $i^{\text{th}}$  selected case:

- a.  $l_i = rand_i \cdot l_i$
- b.  $u_i = 10 \cdot \max_i(u_i)$

Note that each  $u_i$  is set to a very big value, while each  $l_i$  is decremented by a random factor.

*Double censoring mechanism:*

Approximately half of the selected instances was submitted to the left censoring mechanism, while the other half was submitted to the right censoring mechanism.

*Interval censoring mechanism:*

For the generic  $i^{\text{th}}$  selected case:

- a.  $u_i = u_i + [\max_i(u_i) - u_i] \cdot rand_i$
- b.  $l_i = rand_i \cdot l_i$

After the application of the censoring mechanisms, we obtained four distinct datasets, namely Fried\_left, Fried\_rigth, Fried\_double and Fried\_censored datasets.

### 4.3.2. Experiments on Fried datasets

The experiments on Fried datasets followed the experimentation protocol described in paragraph 4.1.1.

Following tables reassume analysis results. Performance are reported in terms of AAE and RS metrics; for the SVCR and MKL – CT algorithms chosen kernels are reported. In particular, for the MKL – CT algorithm only the kernels with non zero weight are reported.



Table 7: results on Fried\_left dataset

	MKL – CT	SVCR	PSM
AAE on training set	1,07577	1,06218	1,5836
RS on training set	0,8904	0,8891	0,8382
AAE on test set	1,3001	1,2979	1,8696
RS on test set	0,88859	0,88734	0,8308

Table 8: kernel chosen by the MKL – CT and by the SCVCR algorithms (Fried\_left dataset).

MKL – CT		
Kernel name	Kernel parameters	Kernel weigth
Gaussian	$\gamma = 0.01$	0.31665
Gaussian	$\gamma = 0.1$	0.30733
Polynomial	$d = 1$ $g = 3$ Normalized= false	0.37602
SVCR		
Kernel name	Kernel parameters	Kernel weigth
Gaussian	$\gamma = 5$	1

Table 9: results on Fried\_right dataset

	MKL – CT	SVCR	PSM
AAE on training set	1,07527	1,05014	1,6134
RS on training set	0,8929	0,8929	0,8321
AAE on test set	1,2935	1,3439	1,8777
RS on test set	0,88964	0,88673	0,8303

Table 10: kernel chosen by the MKL – CT and by the SCVCR algorithms (Fried\_right dataset).

MKL – CT		
Kernel name	Kernel parameters	Kernel weigth
Gaussian	$\gamma = 0.01$	0.17742
Gaussian	$\gamma = 0.1$	0.43116
Polynomial	$d = 1$ $g = 3$ Normalized= false	0.39142
SVCR		
Kernel name	Kernel parameters	Kernel weigth
Polynomial	$d = 1$ $g = 3$ Normalized= false	1

Table 11: results on Fried\_double dataset

	MKL – CT	SVCR	PSM
AAE on training set	1,0678	1,02013	1,5990
RS on training set	0,891	0,8952	0,8361
AAE on test set	1,3084	1,3143	1,8751
RS on test set	0,8872	0,88635	0,8294

Table 12: kernel chosen by the MKL – CT and by the SCVCR algorithms (Fried\_double dataset).

MKL – CT		
Kernel name	Kernel parameters	Kernel weigth
Gaussian	$\gamma = 0.01$	0.079276
Gaussian	$\gamma = 0.1$	0.54662
Polynomial	$d = 1$ $g = 3$ Normalized= false	0.3741
SVCR		
Kernel name	Kernel parameters	Kernel weigth
Gaussian	$\gamma = 5$	1

Table 13: results on Fried\_interval dataset

	MKL – CT	SVCR	PSM
AAE on training set	1,07685	1,04481	1,6365
RS on training set	0,894	0,8931	0,8384
AAE on test set	1,299	1,2759	1,8753
RS on test set	0,88978	0,89074	0,8301

Table 14: kernel chosen by the MKL – CT and by the SCVCR algorithms (Fried\_interval dataset).

MKL – CT		
Kernel name	Kernel parameters	Kernel weigth
Gaussian	$\gamma = 0.01$	0.20297
Gaussian	$\gamma = 0.1$	0.41946
Polynomial	$d = 1$ $g = 3$ Normalized= false	0.37756
SVCR		
Kernel name	Kernel parameters	Kernel weigth
Gaussian	$\gamma = 5$	1

This results demonstrate that the MKL – CT algorithm and the SVCR models are quite similar in terms of results. In fact, looking at the results on the test set, we note that the MKL – CT algorithm shows better performances for the Freid\_right, Freid\_double, and Freid\_left datasets (for the last one, only in terms of RS score). However, the SVCR model is slightly better for the Freid\_interval dataset. Moreover, it should be noted that MKL – CT shows better results on the test set with respect to SVCR even if the results obtained through cross validation procedure seem to be favourable to the SVCR algorithm. So, we can state that MKL – CT algorithm sometimes shows a superior generalization capability with respect to SVCR models. Finally, it should be noted that PSM always present worse performances with respect to the other two methods.

#### **4.4. CT – MKL applied on real survival datasets.**

The experiments presented in the paragraph 4.3 already demonstrated the utility of CT – MKL algorithm for solving the problem of automatic kernel selection, even dealing with different types of censored outcomes.

In this section we will analyze the performance of MKL – CT algorithm when used for analyze “real world” survival dataset. In particular, we want to determine if the automatic kernel selection featured by the MKL – CT algorithm is superior with respect to SVCR enumerative kernel selection procedure.

Thus, we selected four survival datasets, publicly available from the statistical software R<sup>7</sup>: Bfeed, Pneumon, Std<sup>8</sup> and Nwtco<sup>9</sup> datasets. All these dataset were collected during real medical studies. We applied the experimentation protocol described in paragraph 4.1.1 to each selected dataset; the results of related analysis are reported in the following paragraphs.

##### **4.4.1. Experiments on Bfeed dataset**

The Bfeed dataset is aimed to study whether mothers are able to complete the breast feeding of their children. Covariates describe the socio/economic status of the mothers, while the outcome is measured as the length of breast feeding period (weeks). Associated to each sample there is a censoring indicator that can assume the values ‘breast feeding completed’ (not censored) or ‘completion of breast feeding period not observed’ (censored).

The total number of samples is nine hundreds twenty seven; the training set counts five hundreds samples, the test set the reaming ones. Relatively few instances are censored: the training set contains twenty two censored cases, the test set thirteen.

The following tables contains a descriptions of dataset variables and the results of the experimentation.

---

<sup>7</sup> [www.r-project.org](http://www.r-project.org)

<sup>8</sup> The first three datasets are available in the R package ‘KMSurv’

<sup>9</sup> The last dataset is available in the R package ‘Survival’

**Table 15: Bfeed dataset description**

Attribute Name	Attribute Description	Values
Race	Race of mother (1=white, 2=black, 3=other)	Real
Poverty	Mother in poverty (1=yes, 0=no)	Binary
Smoke	Mother smoked at birth of child (1=yes, 0=no)	Binary
Alcohol	Mother used alcohol at birth of child (1=yes, 0=no)	Binary
Age	Age of mother at birth of child	Real
YBirth	Year of birth	Real
YSchool	Education level of the mother (years of school)	Real
Pc3mth	Prenatal care after 3 <sup>rd</sup> month (1=yes, 0=no)	Binary
Duration	Duration of breast feeding, weeks	Real
Delta	Indicator of completed breast feeding	Binary

**Table 16: results on Bfeed dataset**

	MKL – CT	SVCR	PSM
<b>AAE on training set</b>	11,36431	11,25862	11,2548
<b>RS on training set</b>	0,5514	0,5487	0,5485
<b>AAE on test set</b>	11,9239	11,8811	11,6722
<b>RS on test set</b>	0,51653	0,51921	0,5301

**Table 17: kernel chosen by the MKL – CT and by the SCVCR algorithms (BFeed dataset).**

<b>MKL – CT</b>		
Kernel name	Kernel parameters	Kernel weigth
Polynomial	$d = 1$ $g = 3$ Normalized= false	1
<b>SVCR</b>		
Kernel name	Kernel parameters	Kernel weigth
Polynomial	$d = 1$ $g = 3$ Normalized= false	1

Interestingly, best results were obtained by using the PSM approach (log – logistic parametric function). We can argue that data distribution can be effectively represented by a “simple” parametric approach; in such cases, the application of a more complex, non parametric approach (like SVCR or MKL – CT) often deals to worse results, as demonstrated by the obtained performances.

#### 4.4.2. Experiments on Nwtco dataset

The Nwtco dataset collect data from an observational study about survival time of cancer patients. The training set counts one thousand samples, while the test set three thousands twenty eight cases, for a total of four thousands twenty eight samples. The number of censored cases is particularly elevated: 85% in the training set and approximately the same percentage (86%) in the test set.

**Table 18: Nwtco dataset description**

<b>Attribute Name</b>	<b>Attribute Description</b>	<b>Values</b>
Instit	Histology from local institution	Binary
Histol	Histology from central lab	Binary
Stage	Disease stage	Discrete
Study	Study	Discrete
Age	Age in months	Real
Edrel	Time for relapse	Real
Rel	Indicator for relapse	Binary

**Table 19: results on Nwtco dataset**

	<b>MKL – CT</b>	<b>SVCR</b>	<b>PSM</b>
<b>AAE on training set</b>	531,1934	530,6985	848,1591
<b>RS on training set</b>	0,6411	0,6418	0,6792
<b>AAE on test set</b>	458,681	459,2626	724,2196
<b>RS on test set</b>	0,66614	0,6654	0,6941

**Table 20: kernel chosen by the MKL – CT and by the SCVCR algorithms (Nwtco dataset).**

<b>MKL – CT</b>		
<b>Kernel name</b>	<b>Kernel parameters</b>	<b>Kernel weigth</b>
Gaussian	$\gamma = 0.01$	0.0376
Polynomial	$d = 1$ $g = 3$ Normalized= false	0.9624
<b>SVCR</b>		
<b>Kernel name</b>	<b>Kernel parameters</b>	<b>Kernel weigth</b>
Polynomial	$d = 1$ $g = 3$ Normalized= false	1

We observe that the best results were obtained with the MKL – CT algorithm. SVCR models are only slightly worse, while PSM performances were decisely poorer. Interestingly, kernels chosen by MKL – CT algorithm include the same kernel chosen with the SVCR; that is, the common kernel is essential for the formulation of precise prediction, but it is not enough.

#### 4.4.3. Experiments on Pneumon dataset

The Pneumon dataset collects data from a cohort of children hospitalized for the occurrence of pneumonia disease. The dataset presents thirteen predictors and three thousands four hundreds seventy cases, subdivided among training set (one thousand cases) and test set (the remaining samples). Also this dataset presents a very elevated number of censored cases: 97.8% in the training set and 98% in the test set.

**Table 21: Pneumon dataset description**

<b>Attribute Name</b>	<b>Attribute Description</b>	<b>Values</b>
Chldage	Age child had pneumonia, months	Real
Mthage	Age of mother, years	Binary
Urban	Urban environment for mother	Binary
Alcohol	Alcohol use during pregnancy	Binary



Smoke	Smoke during pregnancy	Binary
Region	Region of the country	Discrete
Poverty	Mother at poverty level	Binary
Bweight	Normal birthweight	Binary
Race	Race of the mother	Discrete
Education	Education of the mother (years of school)	Real
Nsibs	Number of siblings of the child	Real
Wmonth	Month the child was weaned	Real
Sfmonth	Month of the child on solid food	Real
Agepn	Time – to – event (age the child in hospital for pneumonia, months)	Real
hHospital	Hospitalization for pneumonia	Binary

Table 22: results on Pneumon dataset

	MKL – CT	SVCR	PSM
AAE on training set	0,06005	0,05885	0,1216
RS on training set	0,8309	0,8529	0,9635
AAE on test set	0,0752	0,0751	0,1777
RS on test set	0,87832	0,8902	0,8630

Table 23: kernel chosen by the MKL – CT and by the SCVCR algorithms (Pneumon dataset).

MKL – CT		
Kernel name	Kernel parameters	Kernel weigth
Polynomial	$d = 1$ $g = 3$ Normalized= false	0.9624
SVCR		
Kernel name	Kernel parameters	Kernel weigth
Gaussian	$\gamma = 10$	0.0376

In this case, the performance of the two kernel methods are highly comparable, while the errors reported by PSM algorithm are clearly higher.

#### 4.4.4. Experiments on Std dataset

Std dataset is related to the study of the recurrence of a particular genitalia infection. It contains eight hundreds seventy seven samples; the training set includes five hundreds cases, the remaining samples form the test set. The presence of censored case is significative: 72% of cases are censored in the training set, and 59% in the test set.

**Table 24: Std dataset description**

Attribute Name	Attribute Description	Values
Race	Race (W=white, B=black)	Real
Marital	Marital status (D=divorced / separated, M=married, S=single)	Discrete
Age	Age	Real
Yschool	Years of schooling	Real
linfct	Initial infection (1= gonorrhea, 2=chlamydia, 3=both)	Discrete
Npartner	Number of partners	Real
Os12m	Oral sex within 12 months (1=yes, 0=no)	Binary
Os30d	Oral sex within 30 days (1=yes, 0=no)	Binary
Rs12m	Rectal sex within 12 months (1=yes, 0=no)	Binary
Rs30d	Rectal sex within 30 days (1=yes, 0=no)	Binary
Abdpain	Presence of abdominal pain (1=yes, 0=no)	Binary
Discharge	Sign of discharge (1=yes, 0=no)	Binary
Dysuria	Sign of dysuria (1=yes, 0=no)	Binary
Condom	Condom use (1=always, 2=sometime, 3=never)	Discrete
Itch	Sign of itch (1=yes, 0=no)	Binary
Lesion	Sign of lesion (1=yes, 0=no)	Binary
Rash	Sign of rash (1=yes, 0=no)	Binary
Lymph	Sign of lymph (1=yes, 0=no)	Binary
Vagina	Involvement vagina at exam (1=yes, 0=no)	Binary
Dchexam	Discharge at exam (1=yes, 0=no)	Binary
Abnode	Abnormal node at exam (1=yes, 0=no)	Binary

Time	Time to reinfection	Real
Rinfct	Reinfection (1=yes, 0=no)	Binary

Table 25: results on Std dataset

	MKL – CT	SVCR	PSM
<b>AAE on training set</b>	205,1019	202,38124	232,0005
<b>RS on training set</b>	0,5461	0,5651	0,5659
<b>AAE on test set</b>	210,0401	207,5616	238,2922
<b>RS on test set</b>	0,59116	0,59455	0,5563

Table 26: kernel chosen by the MKL – CT and by the SCVCR algorithms (Std dataset).

<b>MKL – CT</b>		
<b>Kernel name</b>	<b>Kernel parameters</b>	<b>Kernel weight</b>
Polynomial	$d = 1$ $g = 3$ Normalized= false	0.9624
<b>SVCR</b>		
<b>Kernel name</b>	<b>Kernel parameters</b>	<b>Kernel weight</b>
Gaussian	$\gamma = 10$	0.0376

The Std dataset is the only case where the performances of MKL – CT algorithm are sensibly worse than the performances of SVCR models. So, this case represent a precious counterexample in order to define the limits of MKL – CT algorithm.

#### ***4.5. Comparison of MKL – CT and SVCR algorithms in terms of time spent during the training phase.***

Generally, training a single MKL – CT model requires more time than training a single SVCR model. However, as we explained in paragraph 4.1.1, the parameters optimization phase requires considerably less trials with the MKL – CT algorithm

than with the SVCR algorithm. So, we expect that MKL – CT algorithm will be faster than SVCR during the parameters optimization phase. Table 27 reports the time spent in order to optimize the parameters for each dataset. It is evident that the MKL – CT algorithm allowed a considerable saving of time, being up to twenty times faster. The only exception is given by the Std dataset; it should also be noted that the MKL – CT algorithm provided the worst result just when applied to the Std dataset. Further studies are needed in order to fully understand which are the particularities of the Std dataset that determine the poor performances of MKL – CT algorithm, both in terms of training time and performances.

**Table 27: parameters optimization times (in seconds)**

	<b>MKL – CT</b>	<b>SVCR</b>
<b>Bfeed</b>	560	4794,5
<b>Nwtco</b>	341,6	468,6
<b>Pneumon</b>	361,3	943
<b>Std</b>	2645,8	2013,7
<b>Freid_Left</b>	5390,7	79954,6
<b>Freid_Right</b>	5226	103841,4
<b>Freid_Double</b>	5274,2	107081,3
<b>Freid_Interval</b>	5346,1	107708

#### **4.6. Critical discussion of results.**

At the beginning of the experimentation phase, we planned to check the validity of two important features of MKL – CT algorithm: the automatic kernel selection procedure, and the ability of dealing with heterogeneous types of data.

Regarding the automatic kernel selection procedure, the results indicated that the MKL – CT algorithm provides better results than the single kernel SVCR, *even if not in all cases*. That is, for some dataset the MKL – CT algorithm provides the best results (see Fried\_right, Fried\_double, Nwtco dataset), while for some other datasets the SVCR generates the best results (Fried\_interval, Bfeed, Pneumon, Std datasets). Moreover, both algorithms give approximately the same results at least in one case (Fried\_left dataset).

However, it should be considered that:

1. even when the performance of MKL –CT algorithm are worse than the performances of SVCR models, the loss in terms of generalization capabilities is very small;
2. the MKL – CT automatic kernel selection procedure is highly faster than the enumerative kernel selection procedure usually utilized with single kernel SVM model (like the SVCR).

Then, we can state that *the MKL – CT algorithm is a valid alternative to the SVCR models, because it is able to provide comparable results in less time.*

The only exception is represented by the Std dataset; MKL – CT algorithm provide worse results in more time, when applied to the Std dataset. Further researches could explain the reason of such behaviour.

Regarding the ability of dealing with dataset composed by heterogeneous variables, we can state that the experimentation on Carolei dataset provided a proof of MKL – CT algorithm potential usefulness. In fact, the results of the analysis on Carolei dataset demonstrated that only the genetic variables were relevant for the problem under study. This kind of information is particularly relevant in the analysis of complex phenotypes, because in such analysis it is necessary to evaluate the weight of medical – clinical factors against the influence of the individual genetic profile. Of course, other dataset with heterogeneous covariates should be studied in order to stronger asses the effective capabilities of MKL – CT algorithm.

## Conclusions

In the present thesis work we unified two known SVM approaches, namely the Support Vector Regression for Censored Target and the Multiple Kernel Learning approach. The resulting model, MKL – CT, offers the advantages of both its predecessors, that is the ability of dealing with censored data and the automatic selection of the optimal kernel function. The implementation of MKL – CT was carried out by modifying an open source code of known reliability.

The experimentation phase provided precious information about the effective usefulness of our model. In particular, we compared the performances of MKL – CT algorithm with the performances of the SVCR model (note that MKL – CT can be thought as the “multiple kernel” version of SVCR) and with the results of the Parametric Survival Models, a widely known and used class of statistical methods for the analysis of censored data.

While PSM performances were almost inferior to the performance of the two kernel methods (with a unique, yet significant exception), the performances of SVCR and MKL – CT models were largely comparable.

Thus, given a survival analysis task, we can not “a priori” state which one approach will ensure the best results, among SVCR and MKL – CT.

However, there are two issues that should be considered: firstly, the time employed by the MKL – CT algorithm during the parameters optimization phase is dramatically lower than the time spent by the SVCR models for the same task. Secondly, even when the performances of MKL – CT algorithm are not better than the performances of SVCR models, the differences between the two approaches are minimal.

Thus, we argue that the MKL – CT algorithm is a valid alternative to the use of the common “single kernel” SVCR model, since it can provide comparable results in a lower time. This argument is valid especially when huge datasets must be analyzed or when several (e.g. hundreds) different kernel functions must be tested.

Finally, it must be noted that the MKL – CT algorithm potentially offers great advantages for the analysis of censored dataset with heterogeneous covariates. In the single experiments we performed on a heterogeneous dataset, we found that

the MKL – CT algorithm was able to provide the same performances of SVCR models by using only a part of the whole dataset. In particular, the MKL – CT algorithm indicated that the only relevant variables were the attributes related to the genetic profiles. That is, the MKL – CT algorithm demonstrated that only the genetic variables were significant for the specific problem under study.

Our further researches will deeper investigate the effective usefulness of MKL – CT algorithm for the analysis of heterogeneous datasets, with a particular regard to the potential applications in the study of complex phenotypes.

## Bibliography

- [1] D. R. Cox, “Regression models and life-tables”, *Journal of the Royal Statistical Society*, The Royal Statistical Society, Series B 34, 1972, pp. 187–202.
  
- [2] D. R. Cox and D. Oakes, *Analysis of Survival Data*, Chapman and Hall, London U.K, vol. 21, 1984.
  
- [3] I. James, “Accelerated failure-time models”, in P. Armitage and T. Colton editors, *Encyclopedia of Biostatistics*, John Wiley & Sons, 1998, pp. 26–30.
  
- [4] R. J. Gray, “Flexible methods for analyzing survival data using splines, with applications to breast cancer prognosis”, *Journal of the American Statistical Association*, The American Statistical Association, 87, 1992, pp. 942–951
  
- [5] W. Sauerbrei, and P. Royston, “Building multivariable prognostic and diagnostic models: Transformation of the predictors by using fractional polynomials”, *Journal of the Royal Statistical Society*, The Royal Statistic Society, Series A 162, 1999, pp. 71–94.
  
- [6] D. Ashby, “Bayesian statistics in medicine: A 25 year review”, *Statistics in Medicine*, Wiley InterScience, 25, 2006, pp. 3589–3631
  
- [7] M. R. Segal, “Regression trees for censored data”, *Biometrics*, International Biometric Society, 44, 1988, pp. 35–47.
  
- [8] G. C. Cawley, M.W. Peck, and P. S. Fernández, “A neural model of time to toxin production by nonproteolytic *Clostridium botulinum*,” in Proc. IEEE Int. Joint Conf. Neural Networks, Anchorage, AK, May 1998.



- [9] T. Hothorn, P. Buhlmann, S. Dudoit, A. M. Molinaro, and M. J. van der Laan, "Survival Ensembles", *U.C. Berkeley Division of Biostatistics Working Paper Serie*, U.C. Berkeley Division of Biostatistics, Working Paper 174, 2005.
- [10] E. Bair, T. Hastie, P. Debashis, and R. Tibshirani, "Prediction by Supervised Principal Components", *Journal of the American Statistical Association*, American Statistical Association, Volume 101, Number 473, March 2006 , pp. 119-137.
- [11] P. K. Shivaswamy, W. Chu, and M. Jansche, "A support vector approach to censored targets", in *IEEE International Conference on Data Mining (ICDM-07)*, Omaha, NE (USA), 2007.
- [12] L. Evers and C. M. Messow, "Sparse kernel methods for high-dimensional survival data", *Bioinformatics*, Oxford University Press, Volume 24, Number 14, May 2008, pp. 1632–1638
- [13] V. Van Belle, K. Pelckmans, J. Suykens and S. Van Huffel, "Support Vector Machine for Survival Analysis", *Proc. of the Third International Conference on Computational Intelligence in Medicine and Healthcare (CIMED2007)*, Plymouth, England, Jul. 2007.
- [13] H. Li, and Y. Luan, "Kernel Cox Regression Models for Linking Gene Expression Profiles to Censored Survival Data", *Pacific Symposium on Biocomputing*, 8, 2003, pp. 65-76.
- [14] G. C. Cawley, Member, IEEE, N. L. C. Talbot, G. J. Janacek, and M. W. Peck, "Sparse Bayesian Kernel Survival Analysis for Modelling the Growth Domain of Microbial Pathogens", *IEEE Transaction on Neural Networks*, IEEE Computational Intelligence Society, vol. 17, no. 2, March 2006, pp. 471–481.

- [15] B. Baesens, T. Van Gestel, M. Stepanova, D. Van den Poel and J. Vanthienen, "Neural network survival analysis for personal loan data", *Journal of the Operational Research Society*, Operational Research Society, 56, 2005, pp. 1089–1098.
- [16] E. Biganzoli, P. Boracchi, L. Mariani and E. Marubini, "Feed Forward Neural Networks For The Analysis Of Censored Survival Data: A Partial Logistic Regression Approach", *Statistics in Medicine*, Wiley InterScience, 17, 1998, pp. 1169–1186.
- [17] E. Biganzoli, P. Boracchi, and E. Marubini, "A General Framework for Neural Network Models on Censored Survival Data", *Neural Networks*, Elsevier, 15, 2002, pp. 209–218.
- [18] E. Biganzoli, P. Boracchi, F. Ambrogi and E. Marubini, "Artificial Neural Network for the Joint Modelling of Discrete Cause – Specific Hazard", *Artificial Intelligence in Medicine*, Elsevier, 37, 2006, pp. 119–130.
- [19] F. Ambrogi, N. Lama, P. Boracchia and E. Biganzoli, "Selection of Artificial Neural Network Models for Survival Analysis with Genetic Algorithms", *Computational Statistics & Data Analysis*, Elsevier, 52, 2007, pp. 30–42.
- [20] P. Lapuerta, S. P. Azen and L. La Bree, "Use of neural networks in predicting the risk of coronary artery disease". *Computers and biomedical Research*, Elsevier, 28, 1995, pp. 38–52.
- [21] <http://www.support-vector.net/tutorial.html>
- [22] Scholkopf B. and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, Massachussets, USA, 2002.

- [23] J. Platt, “Fast training of support vector machines using sequential minimal optimization”, in B. Scholkopf, C. Burges, and A. Smola editors, *Advances in Kernel Methods - Support Vector Learning*, MIT Press, Cambridge, MA, USA, 1999, pages 185–208.
- [24] H. Ishwaran and U.B. Kogalur, “Random survival forests for R”, *Rnews*, The R project for Statistical Computing, 7, October 2007, pp.25–31.
- [25] L. Breiman, “Random forests”, *Machine Learning*, Kluwer Academic Publishers, 45, 2001, pp. 5 – 32.
- [26] P. Lisboa, E. Biganzoli, A. Taktak, T. Etchells, I. Jarman, H. Aung and F. Ambrogi, “Assessing flexible models and rule extraction from censored survival data”, Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 2007.
- [27] H. M. Bøvelstad, S. Nyga, H.L. Størvold, M. Aldrin, Ø. Borgan, A. Frigessi and O. C. Lingjærde, “Predicting survival from microarray data — a comparative study”, *Bioinformatics*, Oxford University Press, Volume 23, Number 16, June 2007, pp. 2080–2087
- [28] Shawe-Taylor J. and N. Cristianini, *Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.
- [29] W. M. Campbell, “A Sequence Kernel and its Application to Speaker Recognition”, *Advances in Neural Information Processing Systems*, Neural Information Processing Systems Foundation, 14, 2001, pp. 1157–1163.
- [30] G. R. G Lankriet, N. Cristianini, P. Bartlett, L. El Ghaoui and M. I. Jordan, “Learning the Kernel Matrix with Semidefinite Programming”, *Journal of Machine Learning Research*, MIT Press, 5, 2005, pp. 27–72.

- [31] De Klerk E., *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, Kluwer Academic Publishers, March 2002.
- [32] C. S. Ong, A. J. Smola and R. C. Williamson, “Learning the Kernel with Hyperkernels”, *Journal of Machine Learning Research*, Microtome Publishing, 6, 2005, pp. 1043–1071.
- [33] S. Sonnenburg, G. Ratsch, C. Schafer and B. Scholkopf, “Large Scale Multiple Kernel Learning”, *Journal of Machine Learning Research*, Microtome Publishing, 7, 2006, pp. 1531–1565.
- [34] R. Hettich and K. O. Kortanek. ”Semi-infinite programming: Theory, methods and applications”, *SIAM Review*, Society for Industrial and Applied Mathematics, 3, 1993, pp. 380–429.
- [35] F. R. Bach, G. R. G. Lanckriet and M. I. Jordan, “Multiple Kernel Learning, Conic Duality, and the SMO Algorithm”, in Proceedings of the 21<sup>st</sup> International Conference on Machine Learning, Banff, Canada, 2004.
- [36] G. Rätsch, A. Demiriz, and K. Bennett. ”Sparse regression ensembles in infinite and finite hypothesis spaces”, *Machine Learning*, Kluwer Academic Publishers , 48(1–3), 2002, pp. 193–221.
- [37] T. Joachims, “Making large-Scale SVM Learning Practical”, in B. Scholkopf, C. Burges, and A. Smola editors, *Advances in Kernel Methods - Support Vector Learning*, MIT Press, Cambridge, MA, USA, 1999, pages 169–184.
- [38] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

- [39] M. F. Folstein, S. E. Folstein and P. R. McHugh, "Minimal state. A practical method for grading the cognitive state of patients for the clinician", *Journal of Psychiatric Research*, Elsevier, 12, 1975, pp. 189 – 198.
- [40] J. I. Sheikh and J. A. Yesavage, "Geriatric Depression Scale (GDS): recent evidence and development of a shorter version", in T.L. Brink, editor, *Clinical gerontology: a guide to assessment and intervention*, The Haworth Press, New York, NY, USA, 1986, pp 165-173.
- [41] S. Katz, A. B. Ford, R. W. Moskowitz, B. A. Jackson, M. W. Jaffe Studies of illness in the aged. The index of ADL: a standardized measure of biological and psychosocial function. *Journal of American Medical Association*, The American Medical Association, 21, 1963, pp. 914–919.
- [42] G. Carrieri, M. Bonafe, M. De Luca, G. Rose, O. Varcasia, A. Bruni, R. Maletta, B. Nacmias, S. Sorbi, F. Corsonello, E. Feraco, K. F. Andreev, A. I. Yashin, C. Franceschi, G. De Benedictis, "Mitochondrial DNA haplogroups and APOE4 allele are non-independent variables in sporadic Alzheimer's disease", *Human Genetics*, Springer, 108, 2001, pp.194-198.
- [43] F. De Rango, O. Leone, S. Dato, A. Novelletto, A. C. Bruni, M. Berardelli, V. Mari, E. Feraco, G. Passarino, G. De Benedictis, "Cognitive functioning and survival in the elderly: the SSADH C538T polymorphism". *Annals of Human Genetics*, Blackwell Publishing, 72, 2008, pp. 630 – 635.