UNIVERSITÀ LA SAPIENZA DI ROMA

SCUOLA DI DOTTORATO

Dottorato in Ricerca Operativa – XV ciclo

Tesi di Dottorato

Neural networks, surrogate models and black box algorithms: theory and applications

Vittorio Latorre

Tutore
Prof. Gianni Di Pillo

Coordinatore del corso di dottorato
Prof. Stefano Lucidi

Anno accademico 2011-2012

Summary

In this Ph. D. Thesis we will analyze some of the most used surrogate models, together with a particular type of line search black box strategy. After introducing these powerful tools, we will present the Canonical Duality Theory, the potentiality it has to improve these tools, and some of their applications.

The principal contributes of this Thesis are the reformulation of the Radial Basis Neural Network problem in its canonical dual form in Section 2.2 and the application of the surrogate models and black box algorithms presented in this Thesis on various real world problems reported in Chapter 3.

Acknowledgements

I would like to thank Professor Di Pillo and Professor Lucidi for the useful and interesting things they taught me in the past three years, Professor David Yang Gao for the hospitality he showed me during my period abroad and for the fantastic job we did together, Angelo Ciccazzo for the hours we spent in conference call in order to make algorithms work even if I was at thousands kilometers of distance. Also I would like to thank my co-workers even if they are a little too noisy and my family.

Contents

Chapter 1

Predictive Models

Predictive models are mathematical models that aim to approximate the relation between a vector of independent variables in input and a dependent variable in output. Generally, the kind of problems that these models try to approximate are divided into two classes, depending of the type of output they intend to predict:

- Classification problems: the samples can only belong to a finite number of classes, also called labels. Examples for these problems are characters recognition or the classification of the state of a certain disease for a patient;
- Regression problems: for these problems the mathematical models approximate the function that connects the input variables and the output variables. There are several application for regression problem. One type of application is the prediction of time series, like prediction of sales or meteorological concentrations. Another type of regression is used to approximate complex simulations obtained by costly cpu calculations.

The model creation is based on the use of couples of samples taken from the phenomenon. The set of samples can be described in the following manner:

$$S := \{(x_i,y_i),x_i \in \mathbb{R}^n,y_i \in C), \quad for \quad i = 1,\ldots,l\}, \tag{1.1}$$

where C can be the set of the output classes in the case of classification, or the set $\mathbb{R}$ in the case of regression, n is the dimension of the input and l is the number of set samples.

The knowledge about the phenomenon the model is able to obtain is stored within some parameters of the mathematical model itself, for example the weights of the Neural Network, the Support Vectors of the Support vector machines or the β parameters in the Response Surface Methodologies.

There are two kinds of ways to learn from the data:

- unsupervised learning: the output of the samples are not known or not utilized for defining the parameters of the mathematical models, and the relation among the different samples are defined with clustering methods;

- supervised learning: both the information on the input and the output are used in defining the parameters of the mathematical models.

In this chapter we will describe several mathematical methods that belong to this kind of models and that are utilized in solving real world problems.

1.1 Artificial Neural Networks

1.1.1 Formal Neuron

The structure of a multilayer ANN is inspired by the brain's structure of evolved organisms. Basically, like the brain, it is a network formed by simple units that are linked by connections. The simple unit that forms the network is the formal neuron. This elementary unit performs a transformation of the vector $x \in \mathbb{R}^n$ in the corresponding output $y(x)$. The values of the vectors in input to the neuron are multiplied for the weights that represents the synaptic connections. The weighted sum of the vectors is compared with a threshold, and if the the sum is greater than the threshold the neuron gives 1 as output or -1 otherwise.

As the output $y(x) \in \{-1,1\}$, this is a simple classification model. If we indicate with the vector $w \in \mathbb{R}^n$ the vector of the weights, and with θ the threshold we have:

$$y(x) = g\left(\sum_{j=1}^{n} w_j x_j - \theta\right) = g(w^T x - \theta),$$

where the function $g : \mathbb{R}^{n+1} \to \{-1,1\}$ is called the activation function. In this simple application, we can consider as activation function the sign function $sgn(t)$.

The formal neuron can be considered a linear classifier that gives to a generic input vector x a label with value $y(x) = 1$ or $y(x) = -1$ based on a linear discriminating function $g(w^T x - \theta)$. The values of the weights can be determined by a learning process starting by the set of training samples:

$$T := \{(x_i,y_i), x_i \in \mathbb{R}^n, y_i \in \{-1,1\} \quad for \quad i = 1,\ldots,l\}.$$

The samples in T are considered well classified by the formal neuron if:

$$w^T x - \theta \geq 0 \quad if \quad y_i = 1, \quad w^T x - \theta < 0 \quad if \quad y_i = -1.$$

From a geometrical point of view this mathematical model finds the separation hyperplane $H = x \in \mathbb{R}^n : w^T x = \theta$ that separates the two sets:

$$A = \{x_i : (x_i,y_i) \in T, \quad y_i = 1\} \quad B = \{x_i : (x_i,y_i) \in T, \quad y_i = -1\}.$$

This problem is admissible only if the two sets A, B are linearly separable. The formal neuron is unable of classify non linearly separable sets. It is possible to avoid these problems by performing a transformation of the vectors in input but even this option is subject to a great number of limitations. In order to surmount these limitations, it is possible to use the formal neuron as a simple unit of a more complex system. This system is the neural network.

1.1.2 Multilayer Neural Networks

In order to overcome the limitation of the formal neuron, an architecture of several formal neurons connected among them is needed. This structure is known as multilayer artificial neural network(ANN) [4]. A multilayer ANN is composed of:

- a number of n input units, without elaboration capabilities, that are associated to the n attributes in input to the network;
- a set of N artificial neurons, characterized by activation functions, organized in $L \geq 2$ layers with $L - 1$ hidden layers in which the output of every layer is the input of the successive layer;
- an output layer with $K \geq 1$ neurons that are associated to the outputs of the network;
- a set of oriented and weighted arcs that represent the connections between neurons. We suppose that there are no connections between neurons of the same layer, and that there are only forward connections without feedback ones.

This kind of networks are also known as multilayer feed-forward networks because they only have forward connections in the neurons. This mathematical model has the property of being an universal approximator of continue functions on a compact set, that is, given any continuous function $f : \mathbb{R}^n \to \mathbb{R}$ defined on a compact set $\mathcal{C} \subset \mathbb{R}^n$, it is possible to build a 2-layer network with the property that, for any $\epsilon > 0$ it results

$$\max_{x \in \mathcal{C}} |f(x) - y(x)| < \epsilon.$$

From now on we will only consider neural networks with n neurons in the input layer without elaboration capabilities, an hidden layer with N neurons with an activation function in every neuron, and an output layer with a single neuron that performs

a weighted sum of the outputs of hidden layer neuron. The weights between the neurons of the input layer and the ones in the hidden layer will be indicated as w_{ji}, for $j = 1,\ldots,N$, $i = 1,\ldots,n$, while the weights between the neurons of the hidden layer and the output layer will be indicated as v_j, $j = 1,\ldots,N$. For every neuron there will be a threshold called θ_j. Every formal neuron with an activation function $g : \mathbb{R} \to \mathbb{R}$ performs a weighted sum of its inputs:

$$Y_j(x) = g\left(\sum_{i=1}^{n} w_{ji}x_i - \theta_j\right) \quad j = 1,\ldots,N,$$

so the output function of the neural network $y(x)$ is given by:

$$y(x) = \sum_{j=1}^{N} v_j g\left(\sum_{i=1}^{n} w_{ji}x_i - \theta_j\right),$$

By adding the dummy weight w_{jn+1} for the threshold we can consider the vector $w_j = \{w_{j1},\ldots,w_{jn},w_{jn+1}\}$ and the vector $x = \{x_1,\ldots,x_i,\theta_j\}$ and rewrite the previous formula with these vectors:

$$y(x) = \sum_{i=1}^{N} v_j g(w_j^T x).$$

The knowledge gained by the training is stored in the connections between neurons, in particular it is stored in the weights associated with every connection, including the dummy ones that may represent the thresholds. The learning process of the ANN consists in adjusting w_j,θ_j,v_j, $j = 1,\ldots,N$, in such a way that the output $y(x)$ of the ANN is able to predict the value $f(x)$ produced in a given environment by the input x.

The learning process uses the training set S described in (1.1) with the set C corresponding to the set $\mathbb{R}$. Let us denote by W the $n \times N$ dimensional vector collecting the weights $\{w_j,\ j = 1,\ldots,N\}$, by $\mathbf{v}$ the $N-$vector with components v_j, $j = 1,\ldots,N$, with $\boldsymbol{\theta}$ the vector with components $\{\theta_j,\ j = 1,\ldots,N\}$ and by $y(x_i;W,\boldsymbol{\theta},\mathbf{v})$ the output of the network given the input x_i and the weights $W,\boldsymbol{\theta},\mathbf{v}$. In this case the training is based on the solution of an unconstrained optimization problem of the kind:

$$\min_{W,\boldsymbol{\theta},\mathbf{v}} E(W,\boldsymbol{\theta},\mathbf{v}) = \frac{1}{2}\sum_{i=1}^{l}\left(y\left(x_i;w,\theta,v\right) - y_i\right)^2 + \gamma_1\|w\|^2 + \gamma_2\|\theta\|^2 + \gamma_3\|v\|^2, \qquad (1.2)$$

where $\gamma_1,\gamma_2,\gamma_3 > 0$ and $\|.\|$ denotes the Euclidean norm.

In the function $E(w,\theta,v)$ the first term measures the distance between the output of the network $y(x_i;W,\boldsymbol{\theta},\mathbf{v})$ and the real output y_i. As to the remaining three terms,

they add a penalty on the norm of the weights W,$\boldsymbol{\theta}$,$\mathbf{v}$ that makes compact the level sets of the objective function $E(W,\boldsymbol{\theta},\mathbf{v})$, and regularizes the class of functions realized by the network; the first effect is beneficial for the convergence of the training algorithm, the second one is exploited in cross-validation of the network, as we will mention in the following.

Regarding the activation function $g(\cdot)$, it is supposed that it is differentiable and sigmoidal. The most used ones are the logistic function:

$$g(t) = \frac{1}{1 + e^{-\sigma t}},$$

and the hyperbolic function:

$$g(t) = \frac{1 - e^{-t}}{1 + e^{-t}}.$$

1.1.3 Radial Basis Neural Network

Radial basis functions were introduced as a tool for interpolating multivariate functions. For first we introduce the radial basis function for interpolation problems, then we see how they can be employed in neural networks [4].

Given the set $S := \{(x_i,y_i),x_i \in \mathbb{R}^n,y_i \in \mathbb{R}, \text{ for } i = 1,\dots,l\}$ the interpolation problem consists in finding a function $f : \mathbb{R}^n \to \mathbb{R}$ such as

$$f(x_i) = y_i \quad \forall i = 1,\dots,l.$$

In order to solve this problem the following choice of the function f can be employed:

$$f(x) = \sum_{i=1}^{l} w_i \phi(\|x - x_i\|),$$

where the function $\phi : \mathbb{R}^+ \to \mathbb{R}$ is a continuous function called radial basis function, $\|\cdot\|$ is the euclidean norm in $\mathbb{R}^n$. The resulting function is a linear combination of the ϕ functions that have as argument the distance $\|x - x_i\|$, and the weight of this linear combination are the coefficients w_i, $i = 1,\dots,l$.

In order to calculate the values of the weight w_i the following system of equations must be solved:

$$\Phi \mathbf{w} = y \tag{1.3}$$

where $y = (y_1,\dots,y_l)$, Φ is the $l \times l$ matrix which the single element ϕ_{ij} is given by: $\phi_{ij} = \phi(\|x_j - x_i\|)$ and $\mathbf{w}$ is the vectors with components w_i $i = 1,\dots,l$. Once this system of l variables in l unknowns is solved, it is possible to calculate the function f.

In order to use the radial basis functions in neural networks, techniques from the regularization theory are employed. Suppose that we want to approximate the function $g : \mathbb{R}^n \to \mathbb{R}$ and the set of samples $S := \{(x_i,y_i),\ x_i \in \mathbb{R}^n, y_i \in \mathbb{R}, for \quad i = 1,\ldots,l\}$ is available for the training of the model. Regularization techniques search for the approximating function f by minimizing a functional formed by two terms:

$$\Gamma(x,y) = \frac{1}{2}\sum_{i=1}^{l}[y_i - y(x_i)]^2 + \frac{1}{2}\lambda\|\Psi g\|^2, \tag{1.4}$$

where $\lambda > 0$ is a regularization parameter, Ψ is a differential operator and $\|\cdot\|$ is a norm in the space where Φ belongs.

The first term measures the distance between the approximating function and the terms in the training set, while the second term is comprised by a functional the penalizes the violation on some regularization terms on the function $g(\cdot)$. In other words, the second term depends on some information already known about the function, like for example continuity or differentiability. These informations are contained in the differential operator Ψ.

It is possible to prove that under certain assumption on the differential operator Ψ, the function f that solves the problem (1.4) has the form:

$$f(x) = \sum_{i=1}^{l} w_i\phi(\|x - x_i\|), \tag{1.5}$$

where $\phi : \mathbb{R}^+ \to \mathbb{R}$ is a radial basis function and $\mathbf{w} \in \mathbb{R}^l$ is the solution of the linear system:

$$(\Phi + \lambda I)\mathbf{w} = y,$$

where I is the identity matrix. Equation (1.5) can be seen as the output of a feedforward neural network with an hidden layer.

The neurons in the hidden layers will have as activation function ϕ, which argument will be the distance between the input vector and the center associated with that neuron. In the output layer of this network there will be a single neuron that operates a weighted sum of the outputs of the hidden layers neurons.

These networks have the following approximation property:

Theorem 1. *For every continuous function $g(\cdot)$ on a compact set H, exists a regularized RBF in the form:*

$$f(x) = \sum_{i=1}^{l} w_i\phi(\|x - x_i\|)$$

that for every x belonging to H and for every $\epsilon > 0$:

$$|g(x) - f(x)| < \epsilon.$$

One of the principal problems of regularized radial basis is that there is a neuron in the hidden layer for every element in the training set. In this case, for large datasets, it would become expensive to solve the $P \times P$ system used to calculate the weights w_i.

For this reason generalized RBF were introduced. This kind of neural networks have $N < l$ number of neurons in the hidden layers, and the centers c_i,$i = 1,\ldots,N$ of the RBF do not necessarily coincide with the points x_i belonging to the training set.

The expression of the regression function f is:

$$f(x) = \sum_{i=1}^{N} w_i \phi(\|x - c_i\|),$$

in this equation there are two groups of variables, the weights w_i and the centers c_i. This means that the generalized RBF depends non-linearly from its parameters, differently from the regularized RBF.

This brings to a different formulation of the training problem for generalized RBF. Given the training set S, the number of neuron in the hidden layer N, we indicate with $\mathbf{w}$ the vector of the weights, that is $\mathbf{w} = (w_1,\ldots,w_N)^T \in \mathbb{R}^N$ and with $\mathbf{c}$ the vector of the centers $\mathbf{c} = (c_1,\ldots,c_N)^T \in \mathbb{R}^n$.

The training problem of a generalized RBF neural network can be formulated in a way similar to that of the 2-layer ANN, but it is possible to choose two different strategies in order to perform the training:

- training with supervised weights and non-supervised centers;
- training with supervised centers and weights.

The first strategy is the most simple, and consists in choosing the centers among the points of the training set, keep them fixed to a value and minimize the error function utilizing the vector of weights $\mathbf{w}$. The training consists in solving the unconstrained minimization problem:

$$\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{l} \left(\sum_{j=1}^{N} w_i \phi(\|x_i - c_j\|) - y_i \right)^2 + \frac{\gamma}{2} \|\mathbf{w}\|^2 \tag{1.6}$$

where the term $\gamma > 0$ is the regularization term for the weights. It is possible to easily solve this problem thanks to the least square method. As a matter of facts, it is possible to reformulate problem (1.6) like a least square problem in the following manner:

$$\min_{\mathbf{w}} E(\mathbf{w}) = \left\| \begin{pmatrix} \Phi(\mathbf{c}) \\ \sqrt{\gamma} I \end{pmatrix} \mathbf{w} - \begin{pmatrix} y \\ 0 \end{pmatrix} \right\|^2$$

where $\Phi(\mathbf{c})$ is a $P \times N$ matrix which elements are $\Phi_{ij} = \phi(x_i - c_j)$, $y = (y_1, \ldots, y_l)^T$ and I is the identity matrix.

The principal problem of this formulation is the choice of the vector of centers $\mathbf{c}$ that must remain fixed for the entire training phase. If the centers are not chosen well or their number is too small, the RBF NN could not yield good results.

The second strategy minimizes the error function in respect of both the centers and the weights. Differently from the strategy that optimizes just the weights, this fomulation is non-convex. If we denote by $y(x; \mathbf{w},\mathbf{c})$ the output of the generalized RBF ANN, then the problem becomes:

$$\min_{\mathbf{w},\mathbf{c}} E(\mathbf{w},\mathbf{c}) = \frac{1}{2} \sum_{p=1}^{P} (y(x^p; \mathbf{w},\mathbf{c}) - y^p)^2 + \gamma_1 \|\mathbf{w}\|^2 + \gamma_2 \|\mathbf{c}\|^2.$$

This can be solved with several strategies:

- non-linear optimization algorithms: the algorithm minimizes the function by changing the weight and the centers simultaneously, and in this case algorithms for large non-linear optimization such as the LBFS are suggested;
- decomposition algorithms: this strategy divides the variables in the weights variables block and in the centers variables blocks. For every iteration it fixes one of the two blocks and minimizes the other. It has been proved that under certain assumption, this algorithm converges to a local minimum point of the error function.

As regards the approximation properties of this model, the generalized RBF includes the regularized RBF and then they have the same function approximation properties.

1.2 Support Vector Machines

Support Vector Machines(SVM) [34] are another type of learning machine for the creation of predictive mathematical models. The fundamental model of the SVM was developed by Vapnik [79]. The creation of the basic SVM model takes inspiration from a fundamental problem of learning theory, that is the classification of points belonging to two linearly separable sets.

We are given the following empirical data set:

$$S := \{(x_i, y_i), x_i \in \mathbb{R}^n, y_i \in \{1, -1\}, \quad for \quad i = 1, \ldots, l\},$$

and:

$$A = \{x_i \in S : y_i = 1\}$$

$$B = \{x_i \in S : y_i = -1\}$$

Where A and B are linearly separable, that is it exists a linear hyperplane $H = x \in \mathbb{R}^n : w^T x + b = 0$ that divides $\mathbb{R}^n$ in two semi-spaces with all the points belonging to the set A in one semi-space and all the points belonging to the set B in the other semi-space. We are interested in finding the best linear hyperplane that divides the two sets of points, that is the hyperplane with the maximum minimal distance from the points of the set. This is a max-min problem that can be written in the following way:

$$\max_{w,b} \min \frac{\|w^T x_i + b\|}{\|w\|}, \tag{1.7}$$

where the term $\frac{\|w^T x_i + b\|}{\|w\|}$ is also known as separation margin.

In order to construct the optimal hyperplane with the largest margin we have to solve the following problem:

$$\begin{array}{l} \min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \frac{1}{2}\|w\|^2 \\ w^T x_i + b \geq 1 \quad \forall x_i \in A \\ w^T x_j + b \leq -1 \quad \forall x_j \in B \end{array} \tag{1.8}$$

it is possible to prove that problem (1.8) is equivalent to problem (1.7) and that problem (1.8) has a single global minimum.

By utilizing the labels $y_i = 1$ for $x_i \in A$ and $y_i = -1$ for $x_i \in B$ it is possible to rewrite problem (1.8) in the following compact manner:

$$\begin{array}{l} \min \frac{1}{2}\|w\|^2 \\ y_i[w^T x_i + b] - 1 \geq 0 \quad for \quad i = 1, \ldots, l \end{array} \tag{1.9}$$

Problem (1.9) is a non-linear optimization constrained problem. In order to solve this problem the Karush-Khun-Tucker conditions with the Lagrangian multiplier and the Lagrangian function are used:

$$L(w,b,\lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \lambda_i (y_i[w^T x_i + b] - 1) \tag{1.10}$$

Where λ_i are the Lagrange multipliers, while vector $\lambda \in \mathbb{R}^n$ is the vector of the Langrange multipliers. The variables (w,b) are the primal variables, while the variable λ_i are the dual variables.

To find the optimal point of problem (1.9) the derivatives of problem (1.10) with respect to the primal variables must vanish:

$$\frac{\partial}{b} L(w,b,\lambda) = \frac{\partial}{w} L(w,b,\lambda) = 0$$

that is:

$$\sum_{i=1}^{l} \lambda_i y_i = 0 \tag{1.11}$$

and

$$w = \sum_{i=1}^{l} \lambda_i y_i x_i \tag{1.12}$$

By equation (1.12) the vector of weight is a linear combination of the input vectors x_i with corresponding non zero multiplier λ_i. These vectors are support vector of the mathematical model. Another condition of the KKT is complementarity:

$$\lambda_i(y_i[w^T x_i + b] - 1) = 0 \quad for \quad i = 1, \ldots, l \tag{1.13}$$

With this condition we have that the support vectors, that is the vectors that do not have corresponding multipliers at zero, are those that lie on the separation hyperplane, this means that all the remaining training examples are irrelevant to create the model.

By substituting the values of equation (1.11) and (1.12) in equation (1.10) it is possible to eliminate the primal variables obtaining the following dual optimization problem usually solved in this kind of application:

$$\begin{array}{l} \min_{\lambda \in \mathbb{R}^l} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j (x_i)^T x_j \lambda_i \lambda_j - \sum_{i=1}^{l} \lambda_i \\ \\ \sum_{i=1}^{l} \lambda_i y_i = 0 \\ \lambda_i \geq 0 \quad i = 1, \ldots, l. \end{array} \tag{1.14}$$

Problem (1.14) is a convex quadratic problem with a unique optimal solution. It is possible to prove that the duality gap between the optimal solution of problem (1.9) and (1.14) is zero and that the once the vector λ^* is found it is possible to find the vector w^* by means of the (1.12) and the scalar b by means of the complementarity conditions.

1.2.1 Nonlinearly separable case

In practice the case of linearly separable sets cannot be used because high level noise in the that causes a large overlap in the classes. Given the two sets A e B, let's assume that they are not linearly separable, that is the system:

$$\begin{array}{ll} w^T x_i + b \geq 1 & \forall x_i \in A \\ w^T x_j + b \leq 1 & \forall x_j \in B \end{array} \tag{1.15}$$

Is not solvable. It is possible to add the positive slack variables ξ_k with $k = 1,\ldots,l$ In order to make this system solvable:

$$\begin{aligned} &w^T x_i + b \geq 1 - \xi_i \quad \forall x_i \in A \\ &w^T x_j + b \leq 1 + \xi_j \quad \forall x_j \in B \\ &\xi_k \geq 0 \quad k = 1,\ldots,l \end{aligned} \tag{1.16}$$

If a input vector x_i is not correctly classified, the value of its corresponding ξ_i is greater than 1. In other words the quantity $\sum_{i=1}^{l} \xi_i$ is an upper bound to the number of classification errors. These kinds of models are known as soft margin classifiers. In these mathematical models a good generalization proficiency is obtained by controlling both the classifier capacity through the weights w and the sum $\sum_{i=1}^{l} \xi_i$ in the objective function. The last term is multiplied for a constant term C in the objective function. This term determines a trade-off between the margin maximization and the training error minimization. It is possible to write the corresponding optimization problem in the following form:

$$\begin{aligned} &\min \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i \\ &y_i[w^T x_j + b] - 1 + \xi_i \geq 0 \quad for \quad i = 1,\ldots,l \\ &\xi_i \geq 0 \quad i = 1,\ldots,l \end{aligned} \tag{1.17}$$

like for problem (1.9) it is possible to find the corresponding dual problem of problem (1.17) by using the Lagrangian function:

$$\begin{aligned} &\min_{\lambda \in \mathbb{R}^l} \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j (x_i)^T x_j \lambda_i \lambda_j - \sum_{i=1}^{l} \lambda_i \\ &\sum_{i=1}^{l} \lambda_i y_i = 0 \\ &0 \leq \lambda_i \leq C \quad i = 1,\ldots,l \end{aligned} \tag{1.18}$$

This is a quadratic problem, just like problem (1.14) with the only notable difference being the upper bound C to the Lagrange multipliers λ_i. In this problem the vector w and the scalar b are found in the same way of problem (1.14).

1.2.2 Nonlinear Support Vector Machines

The potentiality and the applicability of the SVM can be further expanded by using Kernel functions.

Definition 1. *given the set $X \subseteq \mathbb{R}^n$, the a function*

$$k : X \times X \to \mathbb{R}$$

is a kernel if it satisfies the following property:

$$k(x,y) = \langle \phi(x), \phi(y) \rangle \quad \forall x,y \in X$$

Where ϕ is a function $\phi : X \to H$, and H is an euclidean space with dot product $\langle \cdot,\cdot \rangle$.

Kernels are symmetric function and are used to create Gram matrices:

Definition 2. *Given a function $k : X \times X \to \mathbb{R}$ and a set of patters $P := \{x_i \in \mathbb{R}^n, i = 1,\dots,l\}$ the $l \times l$ matrix with elements*

$$K_{ij} = k(x_i,x_j)$$

is called the Gram matrix (or Kernel matrix) of k with respect of the set of samples P.

Gram matrices are symmetric and positive semidefinite. Support vector machines with kernels transform the input vectors of training set x_i projecting them from $\mathbb{R}^n$ to an euclidean space of dimension greater than n (or even infinite) called feature space. This projection is performed through the application $\phi : X \to H$ where H is the feature space. This new problem consists in finding the optimal linear hyperplane in the feature space that becomes non-linear when the problem comes back in the input space, $\mathbb{R}^n$.

In order to realize SVM that uses kernel functions it is possible in the (1.18) to substitute the $x_i^T x_j$ in the objective function with their kernel transformation:

$$\begin{aligned} &\min_{\lambda \in \mathbb{R}^l} \tfrac{1}{2} \textstyle\sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j k(x_i,x_j) \lambda_i \lambda_j - \sum_{i=1}^{l} \lambda_i \\ &\textstyle\sum_{i=1}^{l} \lambda_i y_i = 0 \\ &0 \le \lambda_i \le C \quad i = 1,\dots,l. \end{aligned} \tag{1.19}$$

In the objective function it is not necessary to know that explicit expression of the $\phi(\cdot)$ function, but just the kernel function k that is:

$$k(x,y) = \langle \phi(x), \phi(y) \rangle = \phi(x)^T \phi(y).$$

Then, it is possible to write problem (1.19) in the following manner:

$$\begin{aligned} &\min_{\lambda \in \mathbb{R}^l} \tfrac{1}{2} \textstyle\sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \phi(x_i)^T \phi(x_j) \lambda_i \lambda_j - \sum_{i=1}^{l} \lambda_i \\ &\textstyle\sum_{i=1}^{l} \lambda_i y_i = 0 \\ &0 \le \lambda_i \le C \quad i = 1,\dots,l. \end{aligned} \tag{1.20}$$

Examples of most used kernels are:

- Linear kernel: the linear kernel is defined as $k(x_i,x_j) = x_i^T x_j$, that is the dot product between the two vectors, with $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ with $\Phi(x) = x$.

- Polynomial Kernel: it is defined as $k(x_i,x_j) = (ax_i^T y + b)^p$ with $p \geq 1$ $a,b \in \mathbb{R}$. In this case, the euclidean space H and the transformation Φ can vary even if the same kernel is considered.

- Gaussian kernel: it is defined by $k(x_i,x_j) = e^{-\frac{\gamma\|x_i - x_j\|^2}{2\sigma^2}}$. It is a transformation $\Phi : \mathbb{R}^l \to H$ where H is an euclidean space of infinite dimension.

1.2.3 Support Vector Regression

It is possible to expand the mathematical model of the SVM from the field of pattern recognition to that of regression estimation. In this case the training set S will be defined as: $S := \{(x_i,y_i),x_i \in \mathbb{R}^n,y_i \in \mathbb{R}, \text{ for } i = 1,\ldots,l\}$. Differently from the SVM for classification the output values of y_i are real valued.

In order to not lose the mathematical properties of the SVM for regression case, the *ϵ-insensitive loss function* is introduced:

$$|y - f(x)|_\epsilon = \max\{0,|y - f(x)| - \epsilon\}$$

This function does not penalize errors below the threshold $\epsilon \geq 0$. The reasoning behind this choice comes from some properties of the original classification problem. In classification, if a pattern is well classified and far from the margin, it does not contribute to the creation of the margin itself. In other words it is not a support vector, it does not add any further information to the model and it is located in a so called 'insensitive zone'. In the same way, if the output of a pattern is too close to the original output, it must be placed into an insensitive zone.

The objective function that must be minimized is:

$$\min \frac{1}{2}\|w\|^2 + |y - f(x)|_\epsilon. \tag{1.21}$$

The training error is zero if the following system is satisfied:

$$\begin{array}{lll} w^T x_i + b - y_i \geq \epsilon & for & i = 1.\ldots,l \\ y_i - w^T x_i - b \leq \epsilon & for & i = 1.\ldots,l \end{array}$$

like for problem (1.17), it is possible to introduce two kinds for slack variables, for every family of constrains:

$$\begin{array}{lll} w^T x_i + b - y_i \geq \epsilon + \xi_i & for & i = 1.\ldots,l \\ y_i - w^T x_i - b \leq \epsilon + \hat{\xi}_i & for & i = 1.\ldots,l \\ \xi_i,\hat{\xi}_i \geq 0 & & \end{array}$$

the quantity $\sum_{i=1}^{l} \xi_i + \hat{\xi}_i$ is an upper bound to the training error, so it is possible to replace the second term in the objective function (1.21) with it, obtaining the following constrained optimization problem:

$$\begin{array}{l} \min \frac{1}{2}\|w\|^2 + \sum_{i=1}^{l} \xi_i + \hat{\xi}_i \\ w^T x_i + b - y_i \geq \epsilon + \xi_i \quad for \quad i = 1 \ldots, l \\ y_i - w^T x_i - b \leq \epsilon + \hat{\xi}_i \quad for \quad i = 1 \ldots, l \\ \xi_i, \hat{\xi}_i \geq 0. \end{array} \tag{1.22}$$

Problem (1.22) can be reformulated by following the procedure made for problem (1.9) by means of the Lagrangian function:

$$\min_{\lambda \in \mathbb{R}^l} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j k(x_i, x_j)(\hat{\lambda}_i - \lambda_i)(\hat{\lambda}_j - \lambda_j) - \sum_{i=1}^{l} (\hat{\lambda}_i - \lambda_i) y_i + \epsilon \sum_{i=1}^{l} (\hat{\lambda}_i - \lambda_i)$$

$$\begin{array}{l} \sum_{i=1}^{l} (\hat{\lambda}_i - \lambda_i) y_i = 0 \quad i = 1, \ldots, l \\ 0 \leq \lambda_i \leq C \quad i = 1, \ldots, l \\ 0 \leq \hat{\lambda}_i \leq C \quad i = 1, \ldots, l. \end{array} \tag{1.23}$$

In the objective function there is the kernel $k(x_i, x_j)$ so the non-linear generalization is also applicable to regression SVM.

1.3 Kriging methods

Kriging methods are a type of Gaussian surrogate models [33]. There are several types of Kriging models and in this section we will talk about one of its most used mathematical formulation, that is the Ordinary Kriging. The problem consists in: given a random process $F(\cdot)$, we want to create an estimator $\hat{f}$ of the values of a sample path function $f(\cdot)$ of the random process $F(\cdot)$ in a point $\bar{x}$, given the function evaluations in the points $x_1, \ldots, x_l$. The Kriging method is a linear estimator, because it estimates the value of the point $\bar{x}$ by a linear combination of the other l observations:

$$\hat{f}(\bar{x}) = \sum_{i=i}^{l} \lambda_i(\hat{x}) f(x_i)$$

the scalars $\lambda_i(\hat{x})$ are solutions of a linear system and they depend on the choice of the point $\hat{x}$. the weights λ_i must be obtained so that the prediction error:

$$\epsilon(x) = F(x) - \sum_{i=i}^{l} \lambda_i(\hat{x}) F(x_i)$$

must be minimized. For this reason there are different assumptions for every formulation of the Kriging problem. For ordinary Kriging, it is supposed that the expected value $E[F(x)] = \mu$ is constant but unknown and the variogram, defined as $\gamma(x,y) = E[(F(x) - F(x))^2]$, of $F(x)$ is known.

In order to find the best approximation of the point $\bar{x}$ the values of λ_i must be chosen so that the estimator $\hat{f}(\bar{x})$ has the minimum variance:

$$\min \sigma^2 = \min E\left[\left(\hat{f}(\bar{x}) - \sum_{i=1}^{l} \lambda_i F(x_i)\right)^2\right], \tag{1.24}$$

considering the constraint:

$$\sum_{i=1}^{l} \lambda_i = 1. \tag{1.25}$$

This is a constrained optimization problem that can be solved by using the Lagrangian function. For first equation (1.24) is manipulated in the following way using the variograms and the rules of the variance:

$$\sigma^2 = 2\sum_{i=1}^{l} \lambda_i \gamma(\bar{x},x_i) - \gamma(\bar{x},\bar{x}) - \sum_{i=1}^{l}\sum_{j=1}^{l} \lambda_i \lambda_j \gamma(x_i,x_j) \tag{1.26}$$

By differentiating eq. (1.26) and adding the condition on the constraint (1.25) it is possible to obtain the following linear system of equations:

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_l \end{pmatrix} = \begin{pmatrix} \gamma(x_1,x_1) & \dots & \gamma(x_1,x_l) \\ \vdots & \ddots & \vdots \\ \gamma(x_l,x_1) & \dots & \gamma(x_l,x_l) \end{pmatrix}^{-1} \begin{pmatrix} \gamma(x_1,\bar{x}) \\ \vdots \\ \gamma(x_l,\bar{x}) \end{pmatrix} \tag{1.27}$$

after finding the values of $\lambda_i, i = 1,\dots,l$ it is possible to calculate the value of the point $\hat{x}$.

1.4 Response Surface Methodologies

Response surface methodologies(RSMs) [33] are another statistical tool used for regression. This model tries to predict a response y that depends on controllable input variables $x_1,x_2,\dots,x_n$ by approximating the following relationship:

$$y = f(x_1,x_2,\dots,x_n) + \epsilon$$

Where f is unknown and the term ϵ represents the variability not considered in the function f. These variabilities can be measurement errors, noise, other variables not accounted in the analysis and so on. ϵ is normally treated as a statistical error

characterized by normal distribution with mean zero and variance σ^2. In this way the expected value of the prediction y is:

$$E(y) = E(f(x_1,x_2,\ldots,x_n)) + E(\epsilon) = f(x_1,x_2,\ldots,x_n)$$

In the majority of the cases, two types of RSM are used:

- first-order model:
$$y = \beta_0 + \sum_{j=1}^{n} \beta_j x_j + \epsilon \tag{1.28}$$
- second-order model:
$$y = \beta_0 + \sum_{j=1}^{n} \beta_j x_j + \sum_{j=1}^{n} \beta_{jj} x_j^2 + \sum_{j=1}^{n-1} \sum_{i>j}^{n} \beta_{ij} x_i x_j + \epsilon \tag{1.29}$$

Where the β_j and β_{ij} are the parameters of the models. The first-order model generates linear surfaces that are simpler to use than the non-linear surfaces generated by the second order model, but the second order model is more flexible and generates surfaces that can easily adapt to a greater range of applications.

It is possible to represent the second order model with the first order model by performing the following substitutions:

$$x_j^2 = x_{jj}, \quad x_i x_j = x_{ij}$$

and obtain:

$$y = \beta_0 + \sum_{j=1}^{n} \beta_j x_j + \sum_{j=1}^{n} \beta_{jj} x_{jj} + \sum_{j=1}^{n-1} \sum_{i>j}^{n} \beta_{ij} x_{ij} + \epsilon \tag{1.30}$$

that is a linear model. In general any regression model that is linear in the model parameters is a linear regression model, regardless of the shape of the response surface it generates.

Problem (1.28) is a multiple linear regression model with n independent variables and $n+1$ regression coefficients β_i for $i = 0,\ldots,n$. In order to create the regression model, we suppose to have the training set $S := \{(x_{ij},\ldots,x_{in},y_i), x_{ij} \in \mathbb{R}, y_i \in \mathbb{R}$ for $i = 1,\ldots,l\}$ with $l > n$. By using model (1.28) we obtain:

$$y_i = \beta_0 + \sum_{j=1}^{n} \beta_j x_{ij} + \epsilon_i \quad i = 1,\ldots,l. \tag{1.31}$$

The estimation of the model parameters will be performed with the least squared method, that is the parameters β are chosen in order to minimize the sum of square of the errors ϵ_i. The least squared function is:

$$L = \sum_{i=1}^{n} \epsilon^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij} \right)^2 \tag{1.32}$$

Function L must be minimized with respect to β_i,$i = 1,\ldots,n$. In other words the derivative with respect to β_i,$i = 1,\ldots,n$ must satisfy:

$$\frac{\partial L}{\partial \beta_0} = -2\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{n}\beta_j x_{ij}\right) = 0 \tag{1.33}$$

and

$$\frac{\partial L}{\partial \beta_i} = -2\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{n}\beta_j x_{ij}\right) x_{ij} = 0 \quad i = 1,\ldots,n \tag{1.34}$$

the conditions (1.33) and (1.34) can be rewritten as:

$$\begin{array}{c} n\beta_0 + \beta_1 \sum_{j=1}^{n} x_{i1} + \cdots + \beta_k \sum_{j=1}^{n} x_{i1} = \sum_{j=1}^{n} y_i \\ \beta_0 \sum_{j=1}^{n} x_{i1} + \beta_1 \sum_{j=1}^{n} x_{i1}^2 + \cdots + \beta_k \sum_{j=1}^{n} x_{i1}x_{ik} = \sum_{j=1}^{n} x_{i1}y_i \\ \vdots \\ \beta_0 \sum_{j=1}^{n} x_{ik} + \beta_1 \sum_{j=1}^{n} x_{i1}x_{ik} + \cdots + \beta_k \sum_{j=1}^{n} x_{ik}^2 = \sum_{j=1}^{n} x_{ik}y_i \end{array} \tag{1.35}$$

We note that there are $n+1$ equations for $n+1$ unknown that is the model parameters. It is also possible to express equations (1.31), (1.32), (1.33) and (1.34) by matrix notation with:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix},$$

equation (1.31) becomes:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

from this condition the least square function becomes:

$$L = \epsilon^T\epsilon = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$$

by expanding the terms we obtain:

$$L = \mathbf{y}^T\mathbf{y} - \beta^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\beta = \mathbf{y}^T\mathbf{y} - 2\beta^T\mathbf{X}^T\mathbf{y} + \beta^T\mathbf{X}^T\mathbf{X}\beta,$$

since $\beta^T\mathbf{X}^T y$ is a scalar and its transpose $\mathbf{y}^T\mathbf{X}\beta$ is the same scalar. the first order condition for this problem is:

$$\mathbf{X}^T\mathbf{X}b = \mathbf{X}^T\mathbf{y},$$

by multiply both sides of equation for the inverse of the matrix $\mathbf{X}^T\mathbf{X}$ we obtain the value of the estimator β

$$\beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

As said beforehand, any regression model that is linear in the parameters can be represented as a first order regression model. Thanks to this property, it is possible to further expand the model by substituting the vectors x_j in the (1.28) with a basis function $f(x_j)$ where $f : \mathbb{R}^n \to \mathbb{R}$. Several choices can be made for the basis function, the most used ones are:

- Polynomial function in the form:

$$f(x) = \prod_{i=1}^{n_p} p^{j_i}, \quad with \quad 0 \leq j_i \text{ and } \sum_{i=1}^{n_p} \leq n_{poly}$$

  Where n_p is the number of polynomials and n_{poly} is a variable that controls the order of the polynomial. This function generates a response surface that approximates the output well in the case it is smooth over a large range of the inputs. The major problem of this function is that the number of parameters increases exponentially with the order of the polynomials, in other words the higher the polynomial the higher number of samples is needed.

- Radial Basis Functions. Just like the neural networks, it is possible to use the radial basis functions for the response surface models in the form:

$$f(x) = \Phi(\|x\|)$$

  The choice of the Φ function is similar to that in the neural networks. The choice of this function generates a response surface well suited to model functions with significant local changes.

There are also other options for the basis function. For example it is possible to mix an affine term and a radial basis term to obtain a response surface capable of capture the global trend together with local deviation as well, or a polynomial function with a low grade mixed with a radial basis that captures very well the global trend and local deviation.

1.5 Cross Validation

In the previous sections of this chapter we presented several mathematical models for classification and regression. For every model there are two types of parameters. The first type of parameters are the variables of the optimization problem which optimal values are determined during the training phase, like for example the vector

of weights $\mathbf{w}$ in the Neural Networks of the coefficients λ_i in the Support Vector Machines. the second type of parameters are the so called Hyper parameters or model parameters. These parameters are set before the training phase and in a certain sense determine the "model" of the learning machines. Examples of these model parameters are the number of neurons N in the Neural Networks or the penalization parameter C in the Support Vector Machines. Obviously the performances of the mathematical model heavily depend on the choice of these parameters.

Before the training phase, it is possible to set the hyper parameters in order to decide how much the model will fit the training data. For example high values of N influences the NN model so that it tends to fit very precisely the samples in the training set. But this choice also increases the model degrees of freedom. In this way the trained model is very accurate when it comes to predicting the values of the training set, but it fails when it comes to predict the values of new samples taken from the same phenomenon. If such a thing happens, it means that the neural network has extracted so much information on the training set that it also extracted information not relevant to the phenomenon, in other words the Neural network began to model the noise associated with the samples. The same thing can happen for SVM or other mathematical models. This issue of surrogate models is known as "Over-fitting". On the other hand, if these parameters are not chosen well, the surrogate models could fail to extract enough information from the training set.

Because of these issues, normally the error on the training set is not a good indicator of the predictive performances of the surrogate models. To solve this issue, normally a new set of independent samples, know as Validation set, is used to evaluate the performances of the surrogate model.

The strategies that tackle the problem of finding the best hyper parameters for a surrogate model are known as cross validation methodologies. There are several strategies to compute the cross validation error on the validation set. The most simple one, normally used when a large quantity of data is available, is to simply divide the set of samples into the training set and validation set, train different models that differ for the choice of the hyper parameter and choose the one with the lowest error on the validation set. This kind of strategy has the draw back that it can be used only when there are a lot of samples available for the validation set, in order for the validation error to give a good estimate.

A second strategy, used to solve this issue, is the k-fold cross validation. This strategy consists in dividing the training set into k different partitions, and to train the surrogate model using as training set only $k-1$ of these sets, while leaving the last set for calculating the validation error. After the first training, one of the $k-1$ partition is chosen as the new validation set, the old partition that was left out is comprised in the training set and a new training is performed, until all the k partitions are used as validation set. The final validation error will be the average of the validation error on the single partitions.

In the case there are only a few samples available, the Leave One Out cross validation strategy can be used. This strategy consists in training the surrogate model with all the available samples less one, and then calculate the error on this single sample left out. This training strategy is repeated leaving just one sample out of the training set for every sample. The validation error in this case will be the average of the error on every single sample.

One of the major drawbacks of this kind of strategy is the number of training runs, especially if the training itself is quite computationally expensive. Furthermore, if the number of hyper parameters for the surrogate models is large, the number of these trainings could grow exponentially. Therefore there is a great interest in defining efficient automatic techniques for tuning the model parameters.

A simple and widely used technique for computing a good combination of the hyper parameters is to use a grid search with the cross validation technique [28]. The hyper parameters are varied with a fixed step-size (usually on the log scale) in a large set of values and the efficiency of the corresponding surrogate model is evaluated by computing some kind of performance measure. This is a simple procedure, but it can be time demanding if the grid is too dense. In fact in this case this procedure can require a great number of model trainings and hence extremely large computational times. On the other hand if the grid is too sparse, it can fail to find good values for the parameters. An alternative method for cross validation will be presented in the following chapters of this Thesis.

The validation error cannot be considered a genuine measure of the surrogate model performances, because the hyper parameters are chosen according to that value. For this reason, another set of samples, called the Test set, is used to calculate the overall performance of the surrogate model. The Test set is a new set of samples, never used in the training phase nor the validation phase.

1.6 Clustering

Clustering is an application of unsupervised learning [82]. In this section we will introduce basic formulation of the clustering problem, expand it and present a simple algorithm to solve it.

We assume to have a given set A of a finite number of points with $A \subset \mathbb{R}^n$, that is:

$$A = \{x_1, \dots, x_m\}, \quad x_i \in \mathbb{R}^n, \quad i = 1, \dots, m$$

The problem of clustering consists in partitioning the set A in a number of k subsets $A^i, i = 1, \dots, k$ with the property that:

$$A = \bigcup_{i=1}^{k} A^i$$

In the case of hard clustering we also have the following condition:

$$A^i \bigcap A^j = \emptyset$$

The most common approach in clustering is to represent a partition A^i with its cluster center called z_i and to minimize a certain distance $d(.,.)$ of the points of the set A from the center z_i. A point $x_j \in A$ will be assigned to the partition A^i if

$$d(x_j,z_i) = \min_{l=1,\dots,k}\{d(x_j,z_l)\}$$

So the clustering problem seeks to minimize the average of the distances on the entire set A:

$$\min \frac{1}{m} \sum_{x_j \in A} \min_{l=1,\dots,m} d(x_j,z_l) \tag{1.36}$$

This formulation for the clustering problem is know as the "Nonsmooth formulation" because the term $\min_{l=1,\dots,m} d(x_j,z_l)$ is not differentiable.

1.6.1 Smooth Formulation

It is possible to use the support functions [83] to derive an equivalent smooth formulation for problem (1.36):

Definition 3. *for any set $C \subset \mathbb{R}^k$, the function $\sigma_C : \mathbb{R}^k \to [-\infty, +\infty]$ defined by:*

$$\sigma_C(v) := sup\{u^T v | u \in C\},$$

is called support function of C.

Support functions are used to transfer the properties of sets via functions.
It is possible to give the following example of a support function for the unit simplex set $\Delta = \{u \in \mathbb{R}^k | \sum_{j=1}^k u_j = 1, u_j \geq 0, j = 1,\dots,k\}$:

$$\sigma_\Delta(v) = sup\{u^T v | u \in \Delta\} = max_{j=1,\dots,k} v_j$$

In this way we can rewrite the nonsmooth term in (1.36) in the following way:

$$\min_{l=1,\dots,m} d(x_j,z_l) = -\sigma_\Delta(d_i(\mathbf{z})) = \min\{w_{ij} d_i(\mathbf{x}) | w_{ij} \in \Delta\}$$

With:

$$(d_i(\mathbf{z}) := (d(x_i,z_1),\dots,d(x_i,z_k)) \quad \in \mathbb{R}^n$$

and $w_{ij} \in \Delta$ being the "membership variables" associated with the cluster A^i that are equal to one if the point x_i is assigned to the cluster A^j. In this way the clustering problem becomes:

$$\begin{array}{l} \min \sum_{i=1}^k \sum_{j=1}^m w_{ij} d(x_j,z_i) \\ \sum_{i=1}^k w_{ij} = 1, \quad j = 1,\dots,m \\ w_{ij} \in \{0,1\} \quad i = 1,\dots,k \quad j = 1,\dots,m \end{array} \tag{1.37}$$

That is a mixed integer/continuous optimization problem.

1.6.2 Problems correspondence

The proprieties described in this and the following three sections are an expansion of the ones described in [81]. The aim of these sections is to describe the properties of formulation (1.37) and of a basic k-mean algorithm to solve this formulation in a inclusive framework.

Theorem 2. *Given the set* $S \subset \mathbb{R}^n$ *and a set of centers* $z_1,\ldots,z_k \in \mathbb{R}^n$ *with* $Z = [z_1,\ldots,z_k]$*, we have that the problem:*

$$minf(W,Z) = \sum_{i=1}^{k}\sum_{j=1}^{m} w_{ij}D(x_j,z_i) \quad w_{i,j} \in S, \quad z_i \in \mathbb{R}^n \tag{1.38}$$

is equivalent to the problem

$$\min F(W) = \{\min_{Z} f(W,Z) Z \in \mathbb{R}^{nk}\} \quad W \in S \tag{1.39}$$

Where the matrix W *is an* $k \times m$ *matrix that has as elements the weight* w_{ij}

Proof. Hyp: (W^*,Z^*) optimal solution of (1.38)
Thesis: (W^*) optimal solution of (1.39)
from the hypothesis:

$$f(W^*,Z^*) \leq f(W,Z) \quad \forall (W,Z) \in S \times \mathbb{R}^{nk}$$

we have:

$$F(W) = f(W,Z(W))$$

with

$$Z(W) = \arg\min_{Z} f(W,Z), Z \in \mathbb{R}^{nk}$$

we have:

$$f(W^*,Z^*) \leq f(W,Z(W)) = F(W) \quad \forall W \in S$$

Hip: (W^*) optimal solution of (1.39)
thesis: (W^*,Z^*) optimal solution of (1.38)
Proof by contradiction, $W^*,Z(W^*)$ is not a global minimum of problem (1.38), so we have a $(\hat{W},\hat{Z})$ optimal solution of (1.38) such as:

$$f(\hat{W},\hat{Z}) < f(W^*,Z(W^*)) \leq f(W,Z(W)) \quad \forall W \in S$$

in the third member of the inequality we choose $\hat{W}$:

$$f(W^*,Z(W^*)) \leq f(\hat{W},Z(\hat{W}))$$

from the (1.39) we have:

$$f(\hat{W},Z(\hat{W})) \leq f(\hat{W},Z) \quad \forall Z \in \mathbb{R}^{nk}$$

so we have:

$$f(\hat{W},Z(\hat{W})) = f(\hat{W},\hat{Z})$$

that is:

$$f(\hat{W},\hat{Z}) < f(W^*,Z(W^*)) \leq f(\hat{W},\hat{Z})$$

□

Theorem 3. *the vertexes of the problem:*

$$S = \sum_{i=1}^{k} w_{ij} = 1, \quad j = 1,\dots,m \quad w_{ij} \geq 0, i = 1,\dots,k, j = 1,\dots,m \tag{1.40}$$

are feasible for the problem:

$$\sum_{i=1}^{k} w_{ij} = 1, \quad j = 1,\dots,m \quad w_{ij} \in \{0,1\} \quad i = 1,\dots,k \quad j = 1,\dots,m \tag{1.41}$$

that is the vertexes of (1.40) have integer values.

Proof. the point $(\hat{W})$ is a vertex of the set (1.40) if there are km active constraint in that point. There are always m active constraints because of the first family of constraints. Plus the first family of constrains forces at least m constrains to be non zero. We note that if there are exactly m non zero $\hat{w}_i$ they would be forced to 1 by the first family of constraints. By contradiction, we suppose that the vertex $\hat{W}$ has $n > m$ fractional values. There are m active constraints from the first family and $km - n$ active constraints from the second family for a total of $m + km - n < km$ active constraints. Contradiction, $\hat{W}$ is not a vertex for (1.40).

So me have that $\hat{W}$ is a vertex if and only if it has m $\hat{w}_i$ different from zero, that are all set to 1 because of the first family of constraints, while the others are set to zero, in other words $\hat{W}$ is feasible for (1.41). □

Observation: with this we proved that if the objective function is concave, there is no difference in using set (1.40) or set (1.41).
Observation: using the objective function (1.39) with set (1.40) or set (1.41) yields the same results. On the other hand using (1.39) on a given set is equivalent to using (1.38) on the same set, so we have:

Theorem 4. *the problem:*

$$\begin{array}{c} \min f(W,Z) = \sum_{i=1}^{k} \sum_{j=1}^{m} w_{ij} D(x_j,z_i) \\ \sum_{i=1}^{k} w_{ij} = 1, \quad j = 1,\ldots,m \\ w_{ij} \in \{0,1\} \quad i = 1,\ldots,k \quad j = 1,\ldots,m \end{array} \tag{1.42}$$

is equivalent to:

$$\begin{array}{c} \min f(W,Z) = \sum_{i=1}^{k} \sum_{j=1}^{m} w_{ij} D(x_j,z_i) \\ S = \sum_{i=1}^{k} w_{ij} = 1, \quad j = 1,\ldots,m \\ w_{ij} \geq 0, i = 1,\ldots,k, j = 1,\ldots,m \end{array} \tag{1.43}$$

1.6.3 KKT conditions

Here we refer to problem (1.43). Problem (1.43) can be divided into two partial subproblems. in the first, given the vector $\bar{Z}$ we minimize W:

$$min f(W,\bar{Z}) \quad \forall W \in S \tag{1.44}$$

and the problem where we given $\bar{W}$ and then minimize for Z:

$$min f(\bar{W},Z) \quad \forall Z \in \mathbb{R}^{nk} \tag{1.45}$$

Definition 4. *We call the point* (W^*,Z^*) *that solves both the two subproblems:*

$$\begin{array}{c} f(W^*,Z^*) \leq f(W,Z^*) \\ f(W^*,Z^*) \leq f(W^*,Z) \end{array} \tag{1.46}$$

partial solution.

Observation We have that:

$$min f(W,Z^*) = f(W^*,Z^*) = min f(W^*,Z) \tag{1.47}$$

the optimal point of problem (1.44) with Z^* fixed is W^* while the optimal point of problem (1.45) with W^* fixed is Z^*.
We want to characterize the relation between the partial optimal solution and the solution of problem (1.43)

Theorem 5. *the point* (W^*,Z^*) *is a KKT point of problem (1.43) if and only if it is a partial optimal solution*

Proof. let's write the KKT conditions for problem (1.43). First the Lagrangian function L:

$$L(W,Z,\mu,\lambda) = \sum_{i=1}^{k}\sum_{j=1}^{m} w_{ij}D(x_j,z_i) + \sum_{j=1}^{m}\mu_j\left(\sum_{i=1}^{k} w_{ij} - 1\right) - \sum_{i=1}^{k}\sum_{j=1}^{m}\lambda_{ij}w_{ij} \tag{1.48}$$

from this we have the following KKT conditions:

$\frac{\partial L}{\partial w_{ij}} = D(x_j,z_i) + \sum_{j=1}^{m}\mu_j - \lambda_{ij} = 0 \quad \forall i = 1,\dots,k,j = 1,\dots,m$
$\sum_{i=1}^{k} w_{ij} = 1 \quad j = 1,\dots,m \qquad w_{ij} \geq 0, i = 1,\dots,k,j = 1,\dots,m$ (admissibility)
$\lambda_{ij}w_{ij} = 0 \quad \lambda \geq 0 \quad \forall i = 1,\dots,k,j = 1,\dots,m$ (complementarity)
$\frac{\partial L}{\partial z_i} = \frac{\partial D(x_j,z_i)}{\partial z_i} \quad \forall i = 1,\dots,k$

A partial optimal solution is a KKT point of (1.44) and it satisfies the first three KKT conditions of problem (1.43), and the partial optimal solution is also a KKT point of problem (1.45) that satisfies the fourth condition of problem (1.43), obtaining all the four conditions satisfied.
Let's assume that (W^*,Z^*) satisfies the KKT condition for the whole problem. It satisfies the KKT conditions for both problem (1.44) and problem (1.45). So (W^*,Z^*) is a KKT point for the problem (1.43) and a KKT point for problem (1.44) and (1.45). We have that problem (1.44) is linear so KKT conditions are necessary and sufficient for point (W^*,Z^*) to be an optimal solution.

For problem (1.45), we have from (1.47) that its optimal solution has the same value of $minf(W^*,Z)$. it is already proved that $minf(W^*,Z^*) = f(W^*,Z)$ and that (W^*,Z^*) is a KKT point for problem (1.45), so it follows that it is a minimum point of the problem (1.45).
From the first condition of the KKT we obtain:

$$\lambda_{ij} = D(x_j,z_i) + \sum_{j=1}^{m}\mu_j$$

from this we can obtain the same conditions but written in scalar form:

(i) $D(x_j,z_i) + \sum_{j=1}^{m}\mu_j \geq 0 \quad \forall i = 1,\dots,k,j = 1,\dots,m$
(ii) $(D(x_j,z_i) + \sum_{j=1}^{m}\mu_j)w_{ij} = 0 \quad \forall i = 1,\dots,k,j = 1,\dots,m$
(iii) $\sum_{i=1}^{k} w_{ij} = 1 \quad j = 1,\dots,m \qquad w_{ij} \geq 0 \quad i = 1,\dots,k,j = 1,\dots,m$
(iv) $\frac{\partial L}{\partial z_i} = \frac{\partial D(x_j,z_i)}{\partial z_i} \quad \forall i = 1,\dots,k$

□

Now we define a property about the local optimality of problem (1.39). First we define an important property about the derivatives.

Theorem 6. *suppose that:*

1. *V is the convex hull of all the pattens;*
2. *the objective function f and its derivatives in respect to w_{ij} are continuous;*
3. *$A(W^*) = \{Z : Z$ minimizes $f(W^*,Z), Z \in V\}$.*

The directional derivative of F at W^*:

$$F'(W^*;d) = lim_{\alpha \to 0^+}[F(W^*,\alpha d) - F(W^*)]/\alpha$$

Corresponds to

$$F'(W^*;d) = min\{\nabla_w f(W^*,Z)^T d : Z \in A(W^*)\}$$

we have the following general optimality lemma adapted for the (1.39):
Lemma: if W^* is an optimal solution of the problem (1.39) then we have:

$$F'(W^* : d) \geq 0$$

for each feasible direction d at W^*.

Theorem 7. *let (W^*,Z^*) be a given point such that W^* is an extreme point of (1.40) and $Z^* \in A(W^*)$. Then W^* is a local minimum of the problem (1.39) if and only if*

$$F(W^*) = f(W^*,Z^*) \leq min\{f(W,Z); W \in S, Z \in A(W^*)\} \tag{1.49}$$

Proof. Hyp: (1.49) is verified;
Thesis: (W^*,Z^*) is a local minimum, that is $F'(W^* : d) \geq 0$ for every feasible direction d.
From (1.49) maintaining $\hat{Z} \in A(W^*)$ fixed:

$$F(W^*) = f(W^*,\hat{Z}) \leq min\{f(W,\hat{Z}), \forall W \in S\}$$

We have:

$$\nabla_W f(W^*,\hat{Z})^T d \geq 0$$

This property is valid for an arbitrary $Z \in A(W^*)$ that means

$$min\{\nabla_w f(W^*,Z)^T d\} | Z \in A(W^*)\} \geq 0$$

that is:

$$F'(W^*;d) \geq 0$$

for every direction d proving the thesis.

Hyp: (W^*,Z^*) is a local minimum of problem (1.39), that is $F'(W^*;d) \geq 0$ for every feasible direction d.
Thesis: (1.49) is verified
From the thesis $F'(W^*;d) \geq 0$, and from the first theorem:

$$F'(W^*;d) = \nabla_w f(W^*,Z)^T d \geq 0 \quad \forall Z \in A(W^*) \tag{1.50}$$

We choose an arbitrary $\hat{Z} \in A(W^*)$. $f(W,\hat{Z})$ is linear in W. From the linearity of f and the condition (1.50) we have that:

$$f(W^*,\hat{Z}) \leq min\{f(W,\hat{Z})\} \quad W \in S$$

but we choose an arbitrary $\hat{Z} \in A(W^*)$ so we have:

$$f(W^*,Z^*) \leq \min\{f(W,Z); W \in S, Z \in A(W^*)\}$$

proving the thesis.

□

In the case $A(W^*)$ is a singleton a partial optimal solution (W^*,Z^*) of the problem (1.38) is a optimal solution of the problem (1.39). because the (1.39) reduces to:

$$f(W^*,Z^*) \leq min\{f(W,Z^*)W \in S\}$$

that is the definition of W^* as partial solution.

1.6.4 Local optimality

In this section we present the condition for the set $A(W^*)$ to be a singleton. We say that a distance $D(.,.)$ is a Minkowsky metric if:

$$D(x_j,z_i) = (\sum_{l=1}^{m} |x_{jl} - z_{il}|^p)^{1/p}.$$

Our analysis will consider W_i that is the the i-th row of the matrix W. so we have that $f(W,Z) = \sum_{i=1}^{k} f_i(W_i,z_i)$ with $f_i(W_i,z_i) = \sum_{j=1}^{m} w_{ij} D(x_j,z_i)$. Then for every row W_i we define $A_i(W_i^*) = \{z_i = z_i \text{ minimizes } f_i(W_i,z_i), z_i \in V_i\}$, where V_i is a compact center that contain the center Z_i. Obviously we have that $A(W^*)$ is singleton if and only if $A_i(W_i^*)$ is a singleton for $i = 1,\ldots,k$. In the next theorem we give the conditions for $A_i(W_i^*)$ to be a nonsingleton.

Theorem 8. *let $D(x_j,z_i)$ be a Minkowsky distance function. Then the set $A_i(W_i^*)$ is nonsingleton if and only if:*

1. *the points* x_j,$j = 1,\ldots,m$ *are collinear*

2. $\sum_{j=1}^{m} w_{ij}^*$ *is even.*

Proof. Hypothesis: the two conditions holds
Thesis: $A_i(W_i^*)$ is nonsigleton
From the first condition we know that the points x_j can be located on a line, that is: $x_j = a + bt_j$ where a and $b \in \mathbb{R}^n$ are given and $t_j \in \mathbb{R}$. The solution Z_i is in V_i so we know that it lies on the convex hull of the patters x_j forming the cluster. So we have $z_i = a + b\theta_i$. Let $b^T = (b_1,\ldots,b_n)^T$:

$$D(x_j,z_i) = (\sum_{l=1}^{m} |x_{jl} - z_{il}|^p)^{1/p} = (\sum_{l=1}^{m} b_l^p|\theta_i - t_j|^p)^{1/p} = |\theta_i - t_j|(\sum_{l=1}^{m} b_l^p)^{1/p}$$

Substituting this value of D in the function:

$$f_i(W_i^*,z_i) = \sum_{j=1}^{m} w_{ij}D(x_j,z_i) = (\sum_{j=1}^{m} W_{ij}^*|\theta_i - t_j|)(\sum_{l=1}^{m} b_l^p)^{1/p}$$

the problem of finding the center z_i reduced to find the scalar θ_i, that is the problem to locate a point on a straight line. We will assume without loss of generality that $t_1 \leq t_2 \leq \cdots \leq t_m$. Can be shown that the optimal solution is $\theta_i^* = t_r$ where r is the index that:

$$\sum_{j=1}^{r-1} w_{ij}^* < \sum_{j=r}^{m} w_{ij}^* \text{ and } \sum_{j=1}^{r} w_{ij}^* \geq \sum_{j=r+1}^{m} w_{ij}^*$$

that is the r corresponding to the median value of the w_{ij}. Since the w_{ij} are or 0 or 1, we have that $\sum_{j=1}^{m} w_{ij}$ is an integer value and from the second condition we have that $\sum_{j=1}^{m} w_{ij}^*$ is an even number so we have that $\theta_i^* = t_r$ or alternatively $\theta_i^* = t_{r+1}$. Furthermore by the convexity of the function all the points belonging to the segment $[t_r,t_{r+1}]$ are optimal solution. So $A_i(W_i^*)$ is nonsigleton
Hypothesis: $A_i(W_i^*)$ is nonsigleton.
Thesis: the two conditions holds
let's assume for contradiction that $A_i(W_i^*)$ is nonsigleton and the two conditions do not hold. The function $f_i(W_i^*,z_i)$ is a sum of strictly convex functions so f_i is strictly convex and it has a unique minimum which is a contradiction. □

if the quadratic distance function:

$$D(x_j,z_i) = (x_j - z_i)^T(x_j - z_i) \tag{1.51}$$

Is used then we have:

Theorem 9. *Consider problem (1.43) where the quadratic distance function (1.51) is used. Then partial optimal solutions are always local minimum points as shown in Theorem 8.*

Proof. Let (W^*,Z^*) be a partial optimal solution with $Z^* = (z_1^*,\ldots,z_2^*)$, with the values for the single components as:

$$z_i^* = \frac{\sum_{j=1}^n w_{ij}^* x_j}{\sum_{j=1}^n w_{ij}^*}$$

for the first order conditions. It is a unique value, that is $A_i(W_i^*)$ is a singleton for $i = 1,\ldots,k$ so $A(W^*)$ is a singleton. □

1.6.5 A simple algorithm

In this section we will present a simple algorithm for clustering based on formulation (1.43).

Algorithm:

1. Choose an initial point $Z^0 \in \mathbb{R}^{nk}$, and solve P_1 with $Z = Z^0$. Let W^0 be the partial optimal solution at Z^0. Set $r = 0$, for $r = 0,1,\ldots$

2. Solve P_2 with $\hat{W} = W^r$. the solution Z will be Z^{r+1}. If $f(\hat{W},Z^{r+1}) = f(\hat{W},Z^r)$ then stop the optimal solution is $(W^*,Z^*) = (\hat{W},Z^{r+1})$. Otherwise go to the next step

3. Solve P^1 with $\hat{Z} = Z^r$. The solution W will be W^{r+1}. If $f(W^{r+1},\hat{Z}) = f(W^r,\hat{Z})$ then stop the optimal solution is $(W^*,Z^*) = W^{r+1},\hat{Z}$. Otherwise let $r = r + 1$ and to to step 2

Theorem 10. *The algorithm converges to a partial optimal solution of the problem (1.43) in a finite number of iterations.*

Proof. We know that the feasible set has only a finite number of extreme points, so the only thing to prove is that the algorithm visits an extreme point at most once before stopping. let's assume that this is not true and that exists two indexes r_1 and r_2 with $r_1 \neq r_2$ and $W^{r_1} = W^{r_2}$, we get as optimal solution for problem P_2 respectively Z^{r_1+1} and Z^{r_2+1}. Since $W^{r_1} = W^{r_2}$ we have that:

$$f(W^{r_1},Z^{r_1+1}) = f(W^{r_2},Z^{r_1+1}) = f(W^{r_2},Z^{r_2+1})$$

but the sequence $\{f(\cdot)\}$ generated by the algorithm is strictly decreasing(otherwise the stopping criterion would be satisfied) so it is not true that $W_1^r = W_2^r$. □

This algorithm is the base framework of the k-mean algorithm largely used in clustering problems. We note that this problem is easily attracted by shallow local minima and is not able to obtain good solutions. There is another version of this algorithm based on the nonsmooth formulation (1.36):

Algorithm:

1. take any k observations as centers of the first k clusters
2. assign the remaining $m-k$ samples to the k clusters on the basis of the distance function $D(\dots)$
3. after assigning the point to the clusters the center are recomputed and updated
4. if there is almost no observation that change cluster stop, otherwise go to step 1

Let (W^*,Z^*) be a partial optimal solution of problem (1.38). we define the set of its adjacent point as:

$$T(W^*) = \{W \in S : W \text{ an extreme point of } S$$

$$\text{and } W^* \text{ and } W \text{ differ of exactly 2 variables}\}$$

There is the following result

Theorem 11. *Let $\bar{W} \in S$ be such that $F(\bar{W}) \leq F(W)$ for all $W \in T(\bar{W})$, then $\bar{W}$ is a local minimum point of problem (1.39) (note this is the reduced concave problem).*

Proof. Let d be any feasible direction in S at $\bar{W}$ and d^q, $q \in Q$ be the the set of extreme directions in S incident on $\bar{W}$. Extreme directions are those direction that are incident on the extreme points of the set. We have that $d = \sum_{q\in Q} u_q d^q$ for some $u_q \geq 0$. The directional derivate at $\bar{W}$ is:

$$DF(\bar{W};d) = \min\{\nabla'_w f(\bar{W},Z)^T d : Z \in A(\bar{W})\} = \min\left\{\sum_{q\in Q} u_q \nabla'_w f(\bar{W},Z)^T d^q : Z \in A(\bar{W})\right\}$$

$$\sum_{q\in Q} u_q \min\{\nabla'_w f(\bar{W},Z)^T d^q : Z \in A(\bar{W}\} = \sum_{q\in Q} u_q DF(\bar{W};d^q)$$

Since $F(\bar{W}) \leq F(W), \forall W \in T(\bar{W})$ we have that:

$$DF(\bar{W};d^q) \geq 0 \quad \forall q \in Q$$

That means: $DF(\bar{W};d) \geq 0$. So from the lemma and the convexity of S we obtain that $\bar{W}$ is minimum point for the problem. □

This theorem is important in the case the k-mean algorithm at a point (W^*,Z^*) that is not a local minimum for the problem (1.39). It is possible to reach a minimal point for the problem by examining its adjacent extreme points and find the minimum. Then from that point it possible to relaunch the K-mean algorithm.

The operation of examining the adjacent points it is not a complex one, as we explain in the following. The total number of variables w_{ij} is mk. At the extreme point of s, exactly m variables are in base, so the cardinality of T is $m(k-1)$.

Suppose that the point W^* is the point where the algorithm stops and it is not a solution for problem (1.39). Suppose that the variable $w^*_{eg} = 0$ is the one selected to enter in the base and the variable $w^*_{lg} = 1$ is the one selected to exit the base. In this way we will obtain the point W^r with $w^r_{eg} = 1$ and $w^r_{lg} = 0$ and $w^r_{ij} = w^*_{ij}$ for all the others values of w_{ij} varying i and j.

By using the division of the W in rows W_i we have:

$$F(W^r) = \sum_{i=1}^{k} \min_{z_i} f_i(W^r_i,z_i)$$

this is equal to

$$\sum_{i=1,i\neq l,e}^{k} \min_{z_i} f_i(W^r_i,z_i) + \min_{z_l} f_l(W^r_i,z_l) + \min_{z_e} f_e(W^r_i,z_e)$$

We note that

$$W^r_i = W^*_i \quad for \quad i = 1,\ldots,k; i \neq e,l$$

hence:

$$F(W^r) = \sum_{i=1,i\neq l,e}^{k} \min_{z_i} f_i(W^*_i,z_i) + \min_{z_l} f_l(W^r_i,z_l) + \min_{z_e} f_e(W^r_i,z_e)$$

With this we note that only the values linked to two centers (two rows of the W matrix) must be recomputed. We have:

$$\min_{z_l} f_l(W^*_l,z_l) - D(x_g,z^*_l) \geq \min_{z_l} f_l(W^r_l,z_l)$$

note that $D(x_g,z^*_l)$ becomes zero because $w^r_{gl} = 0$. For the cluster e we have:

$$\min_{z_e} f_e(W^*_e,z_e) + D(x_g,z^*_e) \geq \min_{z_e} f_e(W^r_e,z_e)$$

we have that the difference of the objective function between $F(W^r)$ and $F(W^*)$ is:

$$F(W^r) - F(W^*) = \min_{z_l} f_l(W^r_i,z_l) + \min_{z_e} f_e(W^r_i,z_e) - [f_l(W^*_i,z_l) + \min_{z_e} f_e(W^*_i,z_e)] \tag{1.52}$$

from the previous two inequalities we have:

$$\min_{z_l} f_l(W_l^r,z_l) - \min_{z_l} f_l(W_l^*,z_l) \leq -D(x_g,z_l^*)$$
$$\min_{z_l} f_l(W_l^r,z_l) - \min_{z_e} f_e(W_e^*,z_e) \leq D(x_g,z_e^*)$$

By substituting these values in equation (1.52) we obtain:

$$F(W^r) - F(W^*) \leq D(x_g,z_e^*) - D(x_g,z_l^*) \tag{1.53}$$

Because the point W^*,Z^* is solution for the problem P_1 we have that the right hand of the (1.53) is always no negative, and it gets smaller the higher is the probability that $F(W^r) < F(W^*)$. A quick way to improve an extreme point is to calculate the right hand of (1.53) for $e = 1,\ldots,k,g = 1,\ldots,m$ and $w_{eg}^* = 0$, arrange the values in ascending order and finally investigate the points with the highest rankings.

1.6.6 An algorithm to choose the number of clusters

If there is not enough information about the classes of the datasets, there could be the difficulty in determining the right number of clusters k. A simple way to determine this parameter is to begin from a small number of k and then gradually increase it until a certain stopping criterion is satisfied. This means to apply an optimization algorithm several times for finding the global optimum or a good solution of the problem, increasing the number of clusters each time.

The following method creates an algorithm [82] that begins the minimization from a good starting point utilizing the information obtained by the previous steps . The version of this algorithm uses the non-smooth formulation (1.36) and the Algorithm 1.6.5:

algorithm

1. (initialization). Choose a tolerance $\epsilon > 0$ and a positive number k_0 as the initial number of clusters. select a starting point for the centers $Z^0 = (z_1^0,\ldots,z_n^0, \ldots,z_1^{k_0},\ldots,x_n^{k_0}) \in \mathbb{R}^{n\times k_0}$ and solve the problem (1.36). Let Z^{1*} be the optimal solution solution. set $k = k_0$.

2. select a point $z^0 \in \mathbb{R}^n$ and solve the following optimization problem:

$$\min f_k(z) \quad z \in \mathbb{R}^n$$

where

$$f_k(z) = \sum_{a\in A} \min\{||z^{1*} - a||,\ldots,||z^{k*} - a||,||z - a||\}$$

find the point $z^{k+1} \in \mathbb{R}^n$. Note that the solution of this problem is the best point candidate to become a new cluster center, given the vector Z^{1*}

3. add a new cluster to the problem starting from the initial point $z^{k+1,0} = (z^{1*}, \ldots, z^{k*}, z^{k+1})$ and solve problem (1.36) with $k+1$ centers

4. let $x^{k+1,*}$ be the optimal solution found in the previous step, if:

$$\frac{f(z^{k*}) - f(z^{k+1})}{f(z^{1*})} < \epsilon$$

then stop, otherwise set $k = k+1$ and go to step 2.

We note that if this algorithm starts from $k = 1$ the clustering problem is a convex optimization problem and that it is quite possible that the point $z^{k+1,0}$ calculated in the step 2 is not far from the solution of the problem (1.36) so it is possible that it takes a moderate number of iteration for finding the optimal point.
As regards the choice of the tolerance value ϵ, large values of ϵ result in large clusters, while small values of ϵ can produce small and artificial clusters.

Chapter 2

Optimization Tools

In order to widen the fields in which the predictive models presented in the previous chapter can be applied, and in order to improve their mathematical formulation and accuracy, several optimization algorithms and theories are used. In this chapter we present a class of Derivative-Free algorithms, that coupled with the surrogate models permits to obtain approximate solutions for problems where the objective function is not available in closed form. In the second part of this chapter we will also introduce the Canonical Duality Theory and present an application of this theory to the formulation of Radial Basis Function Neural Networks.

2.1 Black Box Optimization Algorithms

Generally, the types of optimization algorithms can be divided according to the method and the informations they use to find the direction to use to move from the current point x_k to the new point x_{k+1}. The definition of the direction d_k is based on creating a local model of the objective function in the current point and the direction itself can be interpreted as the result of relatively simple approximations. In general, the most important distinctions among the methods depends on the information that the algorithm uses:

- methods that use the knowledge on the first order and second order derivatives(Newton Methods);
- methods that utilize only the knowledge on the first order derivatives (Gradient methods);
- derivative free methods that only use the evaluations on the objective function

Also it is important to define the convergence properties of such algorithms. That is, chosen the initial point x_0 we have:

- Global convergence properties, that is the algorithms converges to a stationary point of the objective function no matter the initial point x_0
- Local convergence properties, that is the algorithm converges to a stationary point of the objective function if the initial point x_0 is in a neighborhood Ω of said stationary point.

Among these optimization algorithms we are interested in black box algorithms that only need evaluations of the objective function. On the other hand, we want that these methods to be globally convergent to a stationary point of the objective function.
This kind of methods are becoming more and more popular in the industrial and scientific applications in which the first order derivatives of the objective function f are not available, or in applications in which the objective function is not even available in closed form. The most widespread examples can be the optimization of a complex black box system where there is not complete information on the phenomenon one has to manage, like for example the results from simulations, or other complex systems where only samples of the characterizing function are available. In this case it is possible to extrapolate the function through surrogate models and optimize the surrogate function through these black box methods.
It is really important that not only these methods converge to a stationary point of the black box function, but also that the convergence is fast enough. As a matter of facts, in the case of optimizing a black box function which evaluations are obtained from costly simulations, to obtain a good solution without wasting these function evaluations is important.

An important class of derivative free methods are the so-called direct search methods [23], [24], [25], which base the minimization on the comparison of objective function in suitable trial points. Two of the most used subclasses of such methods are:

- pattern search methods, which have the feature of evaluating the objective function on specific geometric patterns;
- line search methods, which are inspired from gradient-based methods and perform one dimensional minimizations along suitable directions.

Each one of these methods present different and interesting features. The pattern search methods can accurately sample the objective function in a neighborhood of the current point and identify good directions along which the objective function decreases with a good rate. On the other side, the line search algorithms can perform large steps along the search direction and then exploit well a good direction. In other words, if it is possible to combine these approaches it would be possible to create

derivative free algorithms that are able to define good direction where the objective function significantly decreases and to find a suitable step in order to exploit these good directions with the minimum possible function evaluations.

As we said in the introduction of these methods, we are not only interested in methods that quickly converge to a solution, but also to methods that assure the convergence to a stationary point. In general, when it is possible to exploit the information on the function gradient, it is possible to:

- compute and select good descend directions where the objective function has an high decrease rate;
- find, thanks to suitable line search strategies, a suitable step size on the chosen directions in order to exploit these directions and assure a sufficient decrease of the objective function.

Algorithms that use information on the gradient have such good properties because the gradient posses information on the local behavior of the function f in the current point x_k. As a matter of facts the i-th component $\nabla_i f$ of the gradient is the directional derivative along the coordinate direction e^i, while $-\nabla_i f$ is the directional derivative along the coordinate direction $-e^i$. Then gradient vector provides information on the rate of change of the objective function along the $2n$ direction $[e^1, \ldots, e^n, -e^1, \ldots, -e^n]$. This guarantees that the gradient gives accurate information on the local behavior of the objective function in the neighborhood of the point where the derivatives are computed.
In order to get a good direction for the objective function in the point without informations on the gradient a sampling strategy in the neighborhood of the point is needed. This sampling is performed on a certain set of directions, and the properties of these methods change according to the chosen set of directions. These directions should be chosen so that the behavior of the objective function on these directions is sufficiently indicative of its local behavior in the neighborhood of the current point. Thanks to this property it is possible to:

- realize if the current point is a good approximation of a stationarity point of the objective function;
- To find a specific direction along which the objective function decreases.

In order to analyze this class of particular methods, we begin with the unconstrained optimization algorithm presenting the a suitable set of search directions for the objective function sampling and the conditions to have a globally convergent algorithm with these directions. Then we introduce the different variation of this unconstrained method and talk about their properties.

2.1.1 Unconstrained Optimization Method

We consider the problem in the form

$$\min_{x\in\mathbb{R}^n} f(x) \tag{2.1}$$

and make the following assumption:

Assumption 1. *The function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable.*

As said before, in order to overcome the difficulties caused by the absence of information on the gradient, a suitable set of search directions p_k^i,$i = 1,\dots,r$ associated with each point x_k of the sequence generated by the algorithm should be chosen. In order to characterize this set of directions with this particular property, we introduce a new condition. This condition imposes that the distance between the point generated by the algorithm and the set of stationary points of the objective function tends to zero if and only if the directional derivatives of the objective function along these directions tend to assume nonnegative values. Formally this condition can be posed as:

Condition 1. *Given a sequence of points $\{x_k\}$, the sequence directions $\{p_k^i\}$,$i = 1,\dots,r$, are bounded and such that*

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0 \quad \textit{if and only if} \quad \lim_{k\to\infty} \sum_{i=1}^{r} \min\left\{0, \nabla f(x_k)^T p_k^i\right\} = 0.$$

thanks to this condition we can state the following proposition

Proposition 1. *Let $\{x_k\}$ be a bounded sequence of points and let $\{p_k^i\}$,$i = 1,\dots,r$, be sequences of directions which satisfy Condition 1. For every $\eta > 0$, there exist $\gamma > 0$ and $\delta > 0$ such that, for all but finitely many k, if x_k satisfies $\nabla f(x_k) \geq \eta$, then there exists a direction $p_k^{i_k}$, with $i_k \in \{1,\dots,r\}$, for which*

$$f(x_k + \alpha p_k^i) \leq f(x_k) - \gamma \|f(x_k)\| \|p_k^{i_k}\|$$

This proposition assurers that, if the current point is not a stationarity point, it is possible to assure a sufficient decrease of the value of the objective function f by using the set of direction satisfying Condition 1. In other words Condition 1 assures us to find a specific direction along which the objective function decreases, just as we described in the previous section.

One of the most used set of directions that satisfy Condition 1 are the directions on the coordinate of the axes

$$p_k^1 = e^1, \quad p_k^2 = e^2, \quad \dots, \quad p_k^n = e^n.$$

coupled with the directions

$$p_k^{n+1} = -e^1, \quad p_k^{n+2} = -e^2, \quad \dots, \quad p_k^n = -e^{2n}.$$

The global convergence of this class of algorithms can be guaranteed by using some suitable sequences of points along search directions $\{p_k^i\}$,$i = 1,\dots,r$, that satisfy Condition 1. Thanks to Condition 1 we can characterize a stationary point of f with the fact that the objective function does not decrease locally along the chosen directions in points sufficiently close to the current point x_k. This enables us to define new general conditions for the global convergence of the algorithms by means of sequence of points in which the value of the objective function does not decrease for the directions $\{p_k^i\}$,$i = 1,\dots,r$. Also in order to assure the convergence we suppose that the following standard assumption holds:

Assumption 2. *The level set*

$$\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$$

is compact.

Thanks to this assumption we can state the following global convergence condition

Proposition 2. *Let $\{x_k\}$ be a sequence of points; let $\{p_k^i\}$,$i = 1,\dots,r$ be sequences of directions; and suppose that the following conditions hold:*

(I) $f(x_{k+1}) \leq f(x_k)$*;*

(II) $\{p_k^i\}$,$i = 1,\dots,r$ *satisfy Condition 1;*

(III) there exist sequences of points $\{y_i^k\}$ and sequences of positive scalars $\{\xi_k^i\}$, for $i = 1,\dots,r$, such that

$$f(y_k^i + \xi_k^i p_k^i) \geq f(y_k^i) - o(\xi_k^i), \tag{2.2}$$

$$\lim_{k\to\infty} \xi_k^i = 0, \tag{2.3}$$

$$\lim_{k\to\infty} \|x_k - y_k^i\| = 0$$

then,

$$\lim_{k\to\infty} \|\nabla f(x_k) = 0\|$$

From the results obtained until now, we can affirm that Condition 1 is similar to the gradient-related condition employed in gradient based algorithms, as it assures a sufficient decrease of the objective function at every iteration plus the global convergence.

The use of directions that satisfy Condition 1 is a common element of the globally convergent derivative-free algorithms. In the pattern search algorithms the condition (2.2) occurs by requiring only a simple decrease of f, while the (2.3) is satisfied by imposing further restrictions on the search directions and on the step lengths. These two conditions in the line search algorithms are satisfied by enforcing a sufficient decrease of f depending on ξ_i^k and without imposing further restrictions on the search directions.

Since Proposition 2 gives some common theoretical features of pattern and line search approaches, it is suitable for defining algorithms which combine these two approaches. In particular, thanks to the conditions given in that proposition it is possible to propose algorithms that get sufficient information on the local behavior of the objective function f, like in a pattern strategy and exploit the possible knowledge of a good direction, like in a line search strategy.

In the following we report an algorithm that exploits these good features. The basic idea of these algorithms is to sample, at each iteration k, the objective function f along a set $\{p_k^i\}_{i=1}^r$ of search directions. First promising directions are found, then sufficiently large steps are performed along them. Both the sufficient decrease of the objective function and the sufficient step length are realized thanks to a line search approach.

This algorithm produces sequences of points with the property that every limit point is a stationary point of f. This property can be obtained by investigating in detail the behavior of the objective function along the search directions $\{p_k^i\}$,$i = 1,\ldots,r$ and by using a derivative-free line search technique to ensure sufficiently large movements along the good directions identified by the algorithm. The line search is showed in the following pseudo code:

Algorithm 1 line search procedure: **LS**$(\tilde{\alpha}_k^i, y_k^i, p_k^i, \gamma, \delta)$

given $\tilde{\alpha}_k^i > 0, y_k^i, p_k^i, \gamma > 0, \delta \in (0,1)$
while $f(y_k^i + \alpha_k^i p_k^i) \leq f(x_k) - \gamma(\alpha_k^i)^2$ **do**
  $\alpha_k^i = \frac{\alpha_k^i}{\delta}$
  until $f\left(y_k^i + \frac{\alpha_k^i}{\delta} p_k^i\right) < max\left[f(y_k^i + \alpha_k^i p_k^i), f(y_k^i) - \gamma\left(\frac{\alpha_k^i}{\delta}\right)^2\right]$
end while
return α_k^i

The conditions expressed in the While loops correspond to the derivate free line

search conditions. The derivative free algorithm is in the following.

Algorithm 2 Unconstrained optimization algorithm

given $x_0 \in \mathbb{R}^n$,$\tilde{\alpha}_0^i > 0$,$i = 1,\ldots,r$,$\gamma > 0$,δ,$\theta \in (0,1)$
for k=0,1,... **do**
  $y_k^i = x_k$.
  for $i = 1$ **to** r **do**
    if $f(y_k^i + \alpha_k^i p_k^i) \leq f(y_k^i) - \gamma(\alpha_k^i)^2$ **then**
      compute α_k^i with **LS**$(\tilde{\alpha}_k^i, y_k^i, p_k^i, \gamma, \delta)$ and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$
    else
      set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta\tilde{\alpha}_k^i$
    end if
    Set $y_k^{i+1} = y_k^i + \alpha_k^i p_k^i$.
  end for
  Find x_{k+1} such that

$$f(x_{x+1} \leq f(y_k^{r+1}))$$

end for

At each iteration k the algorithm examines the behavior of the objective function along all the search directions $\{p_k^i\}$, $i = 1,\ldots,r$. Once it founds a direction on which it is possible to have a sufficient decrease of the value of the objective function, it produces a sufficiently large movement on that direction thanks to the line search procedure. The new point x_{k+1} can be the point y_k^{r+1} produced by the for loop in k or any other point where the objective function is improved from $f(y_k^{r+1})$. Thanks to this property it is possible to preserve the convergence properties even if the approximation scheme for the objective function is changed in order to improve the efficiency of the algorithm. In this algorithm it is possible to associate to each direction p_k^i a different initial step size $\tilde{\alpha}_k^i$ that is updated on the basis of the behavior of the objective function along p_k^i during the iterations. With the changing step size for every direction it is possible to account the the changes of the objective function f during the iterations even if the same set of direction is used. Notice that the algorithm, similarly to the strong form of pattern search algorithms, has to examine first f along all the r directions, while the current point x_k is updated by means of intermediate points y_k^{i+1} when a sufficient decrease of f is obtained on the directions. It is possible to state the following convergence result on the algorithm:

Proposition 3. *Let $\{x_k\}$ be the sequence produced by Algorithm 2. Suppose that the sequences of directions $p_{ki=1}^{i\,r}$ satisfy Condition 1. Then, Algorithm 2 is well defined and we have*

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

2.1.2 Box Constrained Optimization Method

The method described in the previous section can also be easily expanded to be used to solve box constrained optimization problems. Now we consider the problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && l \leq x \leq u, \end{aligned} \tag{2.4}$$

Where x, l, $u \in \mathbb{R}$, with $l < u$ and $f : \mathbb{R}^n \to \mathbb{R}$. This is a more general case of the one described in the previous paragraph because it is possible to allow both $l^i = -\infty$ and $u^i = \infty$ for some (or all) $i \in \{i, \ldots, n\}$. From the (2.4) we can denote the following feasible set

$$\mathcal{F} = \{x \in \mathbb{R}^n : l \leq x \leq u\}.$$

For the problem we define the following condition for stationarity:

Definition 5. *A feasible point x^* is a stationarity point of problem (2.4) if it satisfies the following first order conditions:*

$$\nabla f(x^*)^T (y - x^*) \geq 0 \quad \textit{for all } y \in \mathcal{F} \tag{2.5}$$

Like in the previous case, we do not have any information on the first order derivatives of the problem, for this reason we use the same strategy adopted for the unconstrained case and perform a sampling around the current point of the algorithm in order to compensate for the lack of information provided by the gradient.

The presence of bound constraints imposes further restrictions on the choice of the directions. As a matter of facts in a non stationarity point the direction d must not be only a descend direction but also a feasible direction, that is it exists a small enough step size along such direction that produces feasible points where the objective function is reduced.

As in the unconstrained case, this method draws inspiration from the gradient based methods in order to assure both global convergence properties and efficiency. In general the main features of the algorithm are:

- an exploitable descent and feasible direction is obtained by investigating the local behavior of the function around the current point by using a set of direction that satisfy Assumption 1. In this particular case, we use the coordinate directions;
- Once an exploitable descend and feasible direction is found along a coordinate, a new point is found by using a derivative free line search similar to the one showed for the unconstrained case

- the informations obtained in every single iteration can be used to build an approximation model of the objective function in order to improve the local behavior of the algorithm.

Similarly to the unconstrained case, for this algorithm the convergence to a stationarity point of the objective function f has been proved as well.

The coordinate directions allow us to cope with the presence of box constraints. As a matter of facts, this can be easily derived from the stationarity conditions in (2.5). If the feasible point $\bar{x}$ is not a stationary point of f, then there must exist for condition (2.5) a feasible point y and an integer $h \in \{1,\dots,n\}$ such that $\nabla f(\bar{x})^T(y-\bar{x})^h<0$. If we define with $\bar{\alpha} = (y-\bar{x})^h > 0$, then by the fact that $\mathcal{F}$ is formed by box constraints, we have

$$\bar{\alpha}\nabla f(\bar{x})e_h < 0, \quad \bar{x} + \bar{\alpha}e_h \in \mathcal{F}.$$

From the continuity of the gradient and the convexity of the feasible set we have that it exists a positive $\bar{\alpha}$ such that:

$$f(\bar{x} + \alpha e_h) < f(\bar{x}), \quad \bar{x} + \alpha e_h \in \mathcal{F},$$

for all $\alpha \in (0,\bar{\alpha})$. For $\bar{\alpha} = (y-\bar{x})^h < 0$ the same conclusion is valid with $-e_h$ instead of e_h. In other words, if $\bar{x}$ is not a stationarity point, there is at least a coordinate direction(or its opposite) on which the objective function decreases in another feasible point. So by choosing a suitable step size, on such direction, it is possible to obtain an improvement of the objective function. If the acceptable step size goes under a certain threshold, the current point can be also considered as a good approximation of a stationarity point. Thanks to this reasoning, the re-

Algorithm 3 line search procedure with expansion step: **LSX**(d_i,α,α_{max},γ)

1: given $d_i = e_i$,α,α_{max},$\gamma > 0$,$\delta \in (0,1)$
2: $\bar{\alpha} = \min\left\{\alpha_{max}, \frac{\alpha}{\delta}\right\}$
3: **while** $f(x_k + \bar{\alpha}d_k) \leq f(x_k) - \gamma\bar{\alpha}^2$ **do**
4: **if** $\alpha = \alpha_{max}$ **then**
5: exit
6: **end if**
7: $\bar{\alpha} = \min\left\{\alpha_{max}, \frac{\alpha}{\delta}\right\}$
8: $\alpha = \bar{\alpha}$
9: **end while**
10: $\alpha_k = \alpha$
11: **return** α_k

ported algorithmic model is able to find coordinate directions on which the objective

function sufficiently decreases by sampling points on such directions. Once a good direction has been detected, a derivative free line search technique is employed for performing a sufficiently large step along it in order to exploit this descent direction as much as possible. Thanks to the sampling on the coordinate directions, we are also able to overcome the the lack of gradient information. For this reason a maximum value for the step α_{max} is determined and the actual step at iteration k is determined by the the expansion step described in Algorithm 3

The maximum step α_{max} must assure that the points in which the algorithm moves are feasible.

In the Algorithm 4, the basic iteration begins with examining the direction d_i, in order to see if it is possible to find a feasible point on such direction where the objective function decreases. In order to understand if the direction d_i is good direction, the maximum feasible step length α_{max} along the d_i starting for the current point x_k is calculated. The trial step size α is determined by choosing the minimum between α_{max} and $\bar{\alpha}_i^k$. The scalar $\bar{\alpha}_i^k$ is computed on the basis of the objective function behavior along the d_i in the previous iterations. in other words the value stored in the scalar $\bar{\alpha}_i^k$ represent the sensitivity of the objective function on the direction d_i, hence representing a promising initial step-size. In the subsequent condition it is verified that the direction is not only feasible but also of descent.

Once it has been verified that the direction is not only feasible, but also of descend, the routine with the line search expansion step is employed. This routine computes a sufficiently good estimate of the minimum of f along d_i without requiring any information on the slope of the objective function. This routine is thought so that the feasible and descent direction is exploited as much as possible. Then the step size chosen by the line search routine is used as $\bar{\alpha}_{k+1}^i$.

If the direction d_i is not of descent, the direction $-d_i$ is investigated. If this direction does not produce a sufficient decrease of the function, α_k is set equal to zero and the scalar α_k^i is reduced by a factor θ.

Once a good direction has been found, the candidate point $\bar{x}_{k+1}$ is generated. In the final part of the algorithm, the new point x_{k+1} is generated, and coordinate direction is selected in order to be analyzed in the next iteration. At each iteration x_{k+1} can be always set equal to the candidate point $\bar{x}_{k+1}$ produced in the previous part of the iteration. The index h_k counts the number of successive iterations in which such thing occurred. If the condition $h_k > n$ is verified, it means that the algorithm generated enough points in the neighborhood and has enough information about the local behavior of the function. In this case the next point x_{k+1} can be generated by minimizing any approximation model of the objective function built by using the information obtained until now. This does not affect the convergence properties, but it can increase the efficiency.

After describing the algorithm we are able to state its convergence properties, that are similar to the unconstrained case.

Algorithm 4 Box Constrained Derivative Free Algorithm

Given $x_0 \in \mathcal{F}$,$\theta \in (0,1)$,$\gamma > 0$,$0 < \bar{\alpha}_0^i < \infty$,$d_i = e_i$ for $i = 1,\ldots,n$
$i = 1$,$h_k = 1$
for $k = 1,\ldots$ **do**
  Compute α_{max} such as $x_k + \alpha_{max} d_i \in \partial\mathcal{F}$ and set $\alpha = \min\{\bar{\alpha}_k^i, \alpha_{max}\}$.
  if $\alpha > 0$ and $f(x_k + \alpha d_i) \leq f(x_k) - \gamma\alpha^2$ **then**
    Call **LSX**$(d_i,\alpha,\alpha_{max},\gamma)$
  else
    Compute α_{max} such as $x_k - \alpha_{max} d_i \in \partial\mathcal{F}$ and set $\alpha = \min\{\bar{\alpha}_k^i, \alpha_{max}\}$.
    if $\alpha > 0$ and $f(x_k - \alpha d_i) \leq f(x_k) - \gamma\alpha^2$ **then**
      set $d_i = -d_i$
      Call **LSX**$(d_i,\alpha,\alpha_{max},\gamma)$
    end if
  else
    set $\alpha_k = 0$,$\bar{\alpha}_{k+1}^i = \theta\alpha$
  end if
  set $\bar{x}_{k+1} x_k + \alpha_k d_i$,$\bar{\alpha}_{k+1}^i = \bar{\alpha}_k^j$, for $j \in \{1,\ldots,n\}$ and $j \neq i$.
  if $h_k \geq n$ **then**
    Find x_{x+1} such that

$$f(x_{k+1}) \leq f(\bar{x}_{k+1}) \quad and \quad x_{k+1} \in \mathcal{F},$$

  else
    set $x_{k+1} = \bar{x}_{k+1}$.
  end if
  if $x_{k+1} \neq \bar{x}_{k+1}$ **then**
    $h_{k+1} = 1$
  else
    set $h_{k+1} = h_k + 1$.
  end if
  Set $i = mod(i,n) + 1$,$k = k + 1$
end for

Proposition 4. *Suppose that f is bounded below on the feasible set $\mathcal{F}$ and let $\{x_k\}$ be the sequence produced by Algorithm 4. Then:*

- *Algorithm 4 is well defined;*
- *every limit point of $\{x_k\}$ belongs to $\mathcal{F}$;*
- *we have*

$$\lim_{k\to\infty} \alpha_k = 0 \tag{2.6}$$
$$\lim_{k\to\infty} \bar{\alpha}_k^i = 0 \qquad for \quad i = 1,\dots,n. \tag{2.7}$$

Proposition 5. *Suppose that f is bounded below on the feasible set $\mathcal{F}$ and let $\{x_k\}$ be the sequence produced by Algorithm 4. Then every limit point of $\{x_k\}$ is a stationary point for problem (2.4).*

2.1.3 Box Constrained Mixed-Integer Optimization Method

In the recent years, more and more problems where the variables could take both continues and integer values arose. In general, these problems are tackled with adaption of continuous optimization methods or discrete optimization methods. In this section we present a further expansion of the two algorithms presented in the previous sections that is able to manage problem with variables that can be both continuous and integer. It is important to define the problem because it comes from two distinct frameworks.

From now on we will consider the following bound constrained mixed variable problem

$$\begin{gathered} \min f(x) \\ l \le x \le u, \\ x_i \in Z \quad i \in I_z \quad x_j \in \mathbb{R} \quad j \in I_c \end{gathered} \tag{2.8}$$

Where $x \in \mathbb{R}^n$,l,$u \in \mathbb{R}^n$,$I_z \in \{1,\dots,n\}$ is the index set of the integer variables, and $I_c \in \{1,\dots,n\}$ is the index set of the continuous variables and $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function with respect to x_i,$i \in I_c$. We also assume that that X is a compact set, so l_i and u_i cannot be infinite.

In order to define the local minimum points of this problem we introduce the two following types of neighborhoods with respect to continuous and discrete variables, that is given a point $\bar{x} \in \mathbb{R}$, we define the two neighborhoods:

$$\mathcal{B}_c(\bar{x},\rho) = \{x \in \mathbb{R}^n : x_z = \bar{x}_z, \|x_c - \bar{x}_c\| \le \rho\}$$

$$\mathcal{N}_z(\bar{x}) = \{x \in \mathbb{R}^n : x_c = \bar{x}_c, \|x_z - \bar{x}_z\| \le 1\}$$

Due to the mixed-integer nature of the problem, different definitions of a local minimum point can be used.

Definition 6. *A point $x^* \in X$ is a local minimum of Problem (2.8) if, for some $\epsilon > 0$,*

$$\begin{aligned} f(x^*) \leq f(x), \quad \forall x \in \mathcal{B}_c(x^*,\epsilon) \cap X, \\ f(x^*) \leq f(x), \quad \forall x \in \mathcal{N}_z(x^*) \cap X, \end{aligned} \tag{2.9}$$

and, every point $\bar{x} \in \mathcal{N}_z(x^*) \cap X$ such that $f(\bar{x}) = f(x^*)$ satisfies (2.9) for some $\bar{\epsilon} > 0$

Now let's characterize the critical point.

Proposition 6. *Let $x^* \in X \cap Z$ be a local minimum of Problem (2.8). Then*

$$\begin{aligned} \nabla_c f(x^*)^T (x - x^*)_c \geq 0, \text{ for all } x \in X. \\ f(x^*) \leq f(x) \text{ for all } x \in \mathcal{N}_z(x^*) \cap X. \end{aligned} \tag{2.10}$$

Note that a subset of the conditions is the same as the constrained derivative free case. In other words a point x^* is a critical point if it is stationary with respect to the continuous variables and with respect to the discrete variables, that is it must be a local minimum within the discrete neighborhood $\mathcal{N}_z(x^*)$.

The reported algorithm called DFL(Derivative-Free Linesearch), explores the coordinate directions and updates the iterate whenever a sufficient reduction of the objective function is found. Hence it performs a minimization distributed along all the variables.

The basic ingredients of the method are the Continuous search and Discrete search procedures. They are needed to explore the coordinate directions associated with, respectively, continuous and discrete variables. The current point is updated as soon as a sufficient reduction of the objective function is achieved by one of the procedures. The continuous search is a modification of one reported for the constrained case, that is Algorithm 3. The Discrete search procedure 6 is similar to the Continuous search but the sufficient reduction is governed by a control parameter ξ , which is reduced during the optimization process. This parameter is reduced when any of the discrete variables is updated by the Discrete search procedure and the current steps for the current variables are equal to one.

Every search direction d_i, for $i = 1,\ldots,n$ is characterized by a tentative step along that direction called $\bar{\alpha}$ just like the continuous constrained case. In order to reduce the step when it is needed, a constant factor $\theta \in (0,1)$ is adopted. The initial point for the algorithm is x_0. At every iteration k the Algorithm DFL explores, starting from the current iterate x_k, all the coordinate directions and produces the

Algorithm 5 line search procedure for continuous variables: **ContSear**($\bar{\alpha}$,y,d,α)

given $\gamma > 0$,$\delta \in (0,1)$
if $\alpha < 0$ **then**
  $\alpha = 0$
  return α
else
  if $f(y + \alpha d) \leq f(y) - \gamma\alpha^2$ **then**
    search the largest α_{max} so that $y + \bar{\alpha}d \in X \cap Z$. Set $\alpha = \min\{\alpha_{max},\bar{\alpha}\}$
  else
    if $f(y - \alpha d) \leq f(y) - \gamma\alpha^2$ **then**
      search the largest α_{max} so that $y - \bar{\alpha}d \in X \cap Z$. Set $\alpha = \min\{\alpha_{max},\bar{\alpha}\}$
      set $d = -d$
    end if
  end if
end if
while $\alpha < \bar{\alpha}$ and $f\left(y + \frac{\alpha}{\delta}d\right) \leq f(y) - \gamma\frac{\alpha^2}{\delta^2}$ **do**
  $\alpha = \frac{\alpha}{\delta}$
end while
$\alpha = \min\{\alpha,\bar{\alpha}\}$
return α

Algorithm 6 line search procedure for integer variables: **DiscrSear**($\bar{\alpha}$,y,d,ξ,α)

if $\alpha < 0$ **then**
  $\alpha = 0$
  return α
else
  if $f(y + \alpha d) \leq f(y) - \xi$ **then**
    search the largest α_{max} so that $y + \bar{\alpha}d \in X \cap Z$. Set $\alpha = \min\{\alpha_{max},\bar{\alpha}\}$
  else
    if $f(y - \alpha d) \leq f(y) - \xi$ **then**
      search the largest α_{max} so that $y - \bar{\alpha}d \in X \cap Z$. Set $\alpha = \min\{\alpha_{max},\bar{\alpha}\}$
      set $d = -d$
    end if
  end if
end if
while $\alpha < \bar{\alpha}$ and $f(y + 2\alpha d) \leq f(y) - \xi$ **do**
  $\alpha = 2\alpha$
end while
$\alpha = \min\{\alpha,\bar{\alpha}\}$
return α

intermediate points y_k^i,$i = 1,\ldots,n$. When $i \in I_c$ that is a continuous variables is analyzed, the actual steps α_k^i are computed in the same way as described for the unconstrained case. If $i \in I_z$ the algorithm performs the Discrete search, similar to the Continuous search except for the fact that the sufficient reduction is governed by the parameter ξ_k. The updating formula of tentative steps $\bar{\alpha}_k^i$ is such that $1 \leq \bar{\alpha}_k^i \in Z$.

Algorithm 7 Algorithm DFL

Given $\theta \in (0,1)$,$\xi_0 > 0$,$x_0 \in X \cap Z$,$\bar{\alpha}_0^i > 0$,$i \in I_c$,$\bar{\alpha}_0^i = 1$, $\in I_z$, and set $d_0^i = e^i$
for $k = 1,\ldots$ **do**
  set $y_k^1 = x_k$.
  for $i = 1,\ldots$ **do**
    if $i \in I_c$ **then**
      call **ContSear**($\bar{\alpha}$,y,d,α) and find α
      if α=0 **then**
        set $\alpha_k^i = 0$ and $\bar{\alpha}_{k+1}^i = \theta\bar{\alpha}_k^i$.
      else
        set $\alpha_k^i = \alpha$ and $\bar{\alpha}_{k+1}^i = \alpha$
      end if
    else
      Call **DiscrSear**($\bar{\alpha}$,y,d,ξ,α)
      if α=0 **then**
        set $\alpha_k^i = 0$ and $\bar{\alpha}_{k+1}^i = \max\{1, \lfloor \bar{\alpha}_k^i/2 \rfloor\}$.
      else
        set $\alpha_k^i = \alpha$ and $\bar{\alpha}_{k+1}^i = \alpha$
      end if
    end if
    set $y_k^{i+1} = y_k^i + \alpha_k^i d_i^k$ and $d_{k+1}^i = d_k^i$
  end for
  if $(y_k^{n+1})_z = (x_k)_z$ and $\bar{\alpha}_k^i = 1$,$i \in I_z$ **then**
    set $\xi_{k+1} = \theta\xi$
  else
    set $\xi_{k+1} = \xi$
  end if
  find $x_k + 1 \in X \cap Z$ such that $f(x_k + 1 \leq f(y_k^{n+1}))$
end for

We now report the results for the convergence of the algorithm.

Proposition 7. *Let* $\{x_k\}$,$\{\xi_k\}$,$\{y_k^i\}$,$\{\alpha_k^i\}$,$\{\bar{\alpha}_k^i\}$,$i = 1,\ldots,n$ *be the sequences produced by Algorithm DFL. Then,*

1. *Algorithm DFL is well-defined;*

2. *for all* $i \in I_c$

$$\lim_{k\to\infty} \alpha_k^i = 0$$

$$\lim_{k\to\infty} \bar{\alpha}_k^i = 0$$

3.

$$\lim_{k\to\infty} \xi_k = 0$$

Proposition 8. *Let* $\{x_k\}$ *be the sequence of points produced by Algorithm DFL. Let* $H \subseteq 1,2,\ldots$ *be defined as in Proposition 7 and* x^* *be any accumulation point of* $\{x_k\}_H$*, then*

$$\nabla_c f(x^*)^T (x - x^*)_c \geq 0, \text{ for all } x \in X,$$

$$f(x^*) \leq f(\bar{x}), \text{ for all } \bar{x} \in \mathcal{N}_z(x^*) \cap X.$$

Theorem 12. *Let* $\{x_k\}$ *be the sequence of points generated by Algorithm DFL. Let* $H \subseteq 1,2,\ldots$ *be defined as in Proposition 7. Then,*

1. *a limit point of* $\{x_k\}_H$ *exists;*

2. *every limit point* x^* *of* $\{x_k\}_H$ *is a stationary point of Problem (2.8).*

2.1.4 Derivative Free Black Box Robust Optimization

In the previous sections, we started from a general unconstrained derivate free method and then enlarged its applications to constrained optimization and mix integer optimization. Another field where there is a great interest of research is robust optimization.

Robust optimization tries to make the solution "robust" in the case of uncertainty on the parameters of the problem. For many optimization problems, if the parameters are slightly changed, the optimal solution can become infeasible or sub-optimal. Robust optimization's aim is to find a solution that is optimal in respect of any realization of uncertainty in a given set.
The general formulation of the problem is:

$$\begin{aligned} \text{minimize} \quad & f(x,u_i) \\ \text{subject to} \quad & h_i(x,u_i) \quad \forall u_i \in \mathcal{U}_i, i = 1,\ldots,m. \end{aligned} \tag{2.11}$$

Where $u_i \in \mathbb{R}^n$ are the disturbance vectors or parameters uncertainties and $\mathcal{U}_i \subset \mathbb{R}^k$ are the uncertainty sets.

For most formulations a min-max approach is used, that is the problem consists in minimizing(or maximizing) the objective function with maximizing(or minimizing) the disturbance vectors or parameters uncertainties in the objective function or in the constraints. The obtained solution gives the best design against the worst possible realization of parameters.

One of the principal issues of robust optimization is tractability. Given a class of nominal problems, for example linear programming problems or quadratic problems, and a structured uncertainty set, what is the complexity of the corresponding robust problem? Is this problem mathematically tractable and solvable?
Even with the advancements in the last years, most current methods are restricted to convex problems such as linear, convex quadratic, conic quadratic, linear discrete problems [1], [2], [3]. So other class of problems like robust formulations of general non-linear problems cannot be easily solved. Also there are no proposed methods to solve general black box robust optimization problems, like for example the problems coming from circuit simulations or surrogate model functions.

In the following we present a black box optimization approach to robust optimization.

The problem we want to solve is:

$$\min_{v \in V \subset \mathbb{R}^{n_1}} f(v) = \min_{v \in V \subset \mathbb{R}^{n_1}} \max_{w \in W \subset \mathbb{R}^{n_2}} g(v,w) \tag{2.12}$$

where $f : \mathbb{R}^{n_1} \to \mathbb{R}$, $g : \mathbb{R}^{n_1+n_2} \to \mathbb{R}$, some of the v variables are constrained to assume integer values whereas the w variables are estimation of uncertain data or implementation parameters. Let

$$\begin{aligned} V &= \{v \in \mathbb{R}^{n_1} : l_v \leq v \leq u_v\} \\ W &= \{w \in \mathbb{R}^{n_2} : l_w \leq w \leq u_w\}, \end{aligned}$$

where the variable can v and w can be both continuous or integer. Furthermore, let us assume that V and W are both compact. In the following we will consider the variable v as the original variables of the problem, while the w as the disturbance vectors and parameters uncertainties.

The philosophy we want to follow in order to solve the problem is to find a solution that is a stationarity point in respect to the worst realization of the parameters for that solution.

At every iteration k of the black box algorithm, the value of the objective function is $\max_{w \in W \subset \mathbb{R}^{n_2}} g(v,w)$. In order to calculate even an approximation of this maximum, the values of the variables v are set to their current values v_k and a black box maximization in respect to the uncertainty parameters is performed on the function $g(v_k,w)$.

In other words we realize a double level black box optimization in which:

- at the outer level we perform a standard black box optimization in respect to the variables $v \in V$;

- once we decide to calculate the objective function in a certain point v_k a black box maximization in performed in respect to the function $g(v_k,w)$.

So basically the framework is similar to the one of the Algorithm 7, but the fundamental difference is that every time it is necessary to calculate the value of the objective function, another black box optimization method is called.

This kind of strategy is also applicable to other black box strategies. As a matter of facts at the inner level it is possible not only to use a generic local black box optimization method, but also a general global optimization method in order to find the solution.

2.2 Canonical Duality Theory

Duality is generally utilized for linear and quadratic problems. This procedures consists in reformulating the original problem, also known as primal problem, into a different but equivalent problem also called dual problem. The equivalence of the two problems is given by the fact that the solution of the primal problem corresponds to the solution of the dual problem and there is no duality gap between these solutions, that is the value of the solution for the primal objective function is equal to the value of the dual solution for the dual objective function.

The principal drawback of this theory is that this correspondence between the primal and the dual formulation exists only for convex problems. For non-convex problems and general complex systems, the duality gap exists, and even if it is possible to reformulate a dual problem starting from the primal, these two problems are not equivalent.

With the canonical duality theory developed in [6], it is possible to formulate a perfect dual problem in the sense that there is no duality gap and the associated triality theory can be used to identify both global and local optimal solutions. This theory is composed by a canonical dual transformation methodology, a complementarity dual principle and a triality theory. The canonical dual transformation is a versatile method which can be used to formulate perfect dual problems without duality gap; the complementary-dual principle presents a unified analytic solution form for general problems in continuous and discrete systems; the triality theory, whose components comprise a saddle min/max duality and two pairs of double-min, double-max dualities, can be used to identify both global and local extrema, and to develop effective canonical dual algorithms for solving a wide class of nonconvex/nonsmooth/discrete optimization/variational problems.

In order to demonstrate the application of this theory, let us consider the following non-convex minimization problem:

$$(P): \quad \min_{\mathbf{z}\in\mathcal{Z}_{\mathrm{a}}} \left\{ P(\mathbf{z}) = \frac{1}{2}\langle \mathbf{z}, A\mathbf{z}\rangle - \langle \mathbf{z}, \mathbf{f}\rangle + W(\mathbf{z}) \right\}, \tag{2.13}$$

where $A \in \mathbb{R}^{n\times n}$ is a given symmetric indefinite matrix, $\mathbf{f} \in \mathbb{R}^n$ is a given vector, $\langle \mathbf{v}, \mathbf{v}^*\rangle$ denotes the bilinear form between $\mathbf{v}$ and its dual variable $\mathbf{v}^*$, $W(\mathbf{z})$ is a general non convex function, and $\mathcal{Z}_{\mathrm{a}} \subseteq \mathbb{R}^{\mathrm{n}}$ is the feasible space for the vector $\mathbf{z}$.

Problem (2.13) represents the general format for an unconstrained optimization problem and even for certain constrained problems as $W(\mathbf{z})$ can be considered as an indicator of certain constraint set.

The key step of the canonical dual transformation is to choose a nonlinear operator $\xi = \Lambda(\mathbf{z}) : \mathcal{Z}_{\mathrm{a}} \to \mathcal{E}_{\mathrm{a}}$, where $\mathcal{E}_a$ is the domain where ξ is defined, and a canonical function $V : \mathcal{E}_{\mathrm{a}} \to \mathbb{R}$ such that the nonconvex function $W(\mathbf{z})$ can be expressed in the canonical form $W(\mathbf{z}) = V(\Lambda(\mathbf{z}))$. The function $V(\Lambda(\mathbf{z}))$ is convex in respect of Λ. A typical example for the nonconvex function $W(\mathbf{z})$ could be the so-called double-well energy:

$$W(\mathbf{z}) = \frac{1}{2}\left(\frac{1}{2}\|\mathbf{z}\|^2 - \lambda\right)^2, \tag{2.14}$$

which has extensive applications in mathematical physics [6] and network optimization (see [9]). In this case the primal problem is

$$P(\mathbf{z}) = \frac{1}{2}\left(\frac{1}{2}\|\mathbf{z}\|^2 - \lambda\right)^2 + \frac{1}{2}\mathbf{z}^T A\mathbf{z} - \mathbf{f}^T\mathbf{z}. \tag{2.15}$$

For this example, we can simply choose $\xi = \frac{1}{2}\|\mathbf{z}\|^2$ and $V(\xi) = \frac{1}{2}(\xi - \lambda)^2$.

Once the nonlinear operator and the canonical functions has been chosen, the canonical function $V(\Lambda(\mathbf{z}))$ is said to be a canonical function on its domain if the the duality mapping $\sigma = V'(\xi)$ from $\mathcal{E}_{\mathrm{a}} = \{\xi \in \mathbb{R} |\ \xi \geq 0\}$ to its range $\mathcal{E}_a^* = \{\sigma \in \mathbb{R} |\ \sigma \geq -\lambda\}$ is invertible and the conjugate function $V^*(\sigma)$ can be defined by the Legendre transformation

$$V^*(\sigma) = \mathrm{sta}\left\{\langle \xi, \sigma\rangle - V(\xi) | \xi \in \mathcal{E}_{\mathrm{a}}\right\}, \tag{2.16}$$

where the notation $\mathrm{sta}\{g(\xi)|\xi \in \mathcal{E}_{\mathrm{a}}\}$ stands for stationary point of the function $g(\xi)$ on $\mathcal{E}_{\mathrm{a}}$. As the function $V(\xi)$ is convex in ξ, the solution is unique and the conjugate function can be easily found. In the double well example the Legendre conjugate is equal to

$$V^*(\sigma) = \frac{1}{2}\sigma^2 + \lambda\sigma$$

Therefore, by using the equality $W(\mathbf{z}) = \langle \Lambda(\mathbf{z}), \sigma\rangle - V^*(\sigma)$, the nonconvex function $P(\mathbf{z})$ can be written in the form of the so-called total complementarity function

$\Xi(\mathbf{z},\sigma)$ (see [6])

$$\Xi(\mathbf{z},\sigma) = \langle \Lambda(\mathbf{z}),\sigma\rangle - V^*(\sigma) + \mathbf{z}^T A\mathbf{z} - \mathbf{f}^T\mathbf{z} \tag{2.17}$$

Function $\Xi(\mathbf{z},\sigma)$ can also be regarded as the extended or nonlinear Lagrangian. As a matter of facts if the mathematical operator ξ is chosen linear, the total complementarity function corresponds to the standard Lagrangian form. In general, it is possible to collect the linear terms in of the primal variable $\mathbf{z}$ to create the term $F(\sigma)$ and the quadratic terms to create the term $G(\sigma)$ and rewrite the total complementarity function in the following way

$$\Xi(\mathbf{z},\sigma) = \frac{1}{2}\mathbf{z}^T G(\sigma)\mathbf{z} - F(\sigma)^T\mathbf{z} - V^*(\sigma)$$

From this total complementary function, the canonical dual function can be defined by

$$P^d(\sigma) = \mathrm{sta}\{\Xi(\mathbf{z},\sigma)|\mathbf{z} \in \mathcal{Z}_\mathrm{a}\}. \tag{2.18}$$

By the stationary condition $\nabla_\mathbf{z}\Xi = G(\sigma)\mathbf{z} - \mathbf{f} = 0$, then on the dual feasible space

$$\mathcal{S}_a = \{\sigma \in \mathcal{E}_a^* |\ \det G(\sigma) \neq 0\},$$

the problem dual to (P) can be formulated as

$$(P^d): \quad \max\left\{P^d(\sigma) = -\frac{1}{2}F(\sigma)^T G(\sigma)^{-1}F((\sigma) - V^*(\sigma) \mid \sigma \in \mathcal{S}_a\right\}. \tag{2.19}$$

For the double well total function, the total complementarity function is:

$$\Xi(\mathbf{z},\sigma) = \frac{1}{2}\mathbf{z}^T G(\sigma)\mathbf{z} - \frac{1}{2}\sigma^2 - \lambda\sigma - \mathbf{f}^T\mathbf{z}$$

Where $G(\sigma) = \sigma I + A$ and the term $F(\sigma) = \mathbf{f}$. For this particular case the linear term does not depend from the dual variable σ. The first order conditions of the problem are

$$\mathbf{z} = G(\sigma)^{-1}\mathbf{f}$$

and the dual problem is

$$P^d = \frac{1}{2}\mathbf{f}^T G(\sigma)^{-1}\mathbf{f} - V^*(\sigma)$$

Now we describe the general properties of this dual formulation.

Theorem 13. *(Complementarity-dual principle [6]). Problem (P^d) is canonically dual to (P) in the sense that if $(\bar{\boldsymbol{z}},\bar{\sigma})$ is a critical point of $\Xi(\boldsymbol{z},\sigma)$, then $\bar{\boldsymbol{z}}$ is a feasible solution of (P), $\bar{\sigma}$ is a feasible solution of (P^d) and*

$$P(\bar{\boldsymbol{z}}) = \Xi(\bar{\boldsymbol{z}},\bar{\sigma}) = P^d(\bar{\sigma}). \tag{2.20}$$

This theorem indicates that there is no duality gap between the critical point of the dual problem and the critical points of the primal problem.

Theorem 14. *(Analytic solution to (P) [6]). If $\bar{\sigma} \in \mathcal{S}_{\mathrm{a}}$ is a solution of (P^d), then*

$$\bar{\boldsymbol{z}} = G(\bar{\sigma})^{-1} F(\sigma) \tag{2.21}$$

is a feasible solution of (P) and $P(\bar{\boldsymbol{z}}) = P^d(\bar{\sigma})$. Conversely if $\bar{\boldsymbol{z}}$ is a solution of (P), it must be in the form of (2.21) for certain solutions $\bar{\sigma}$ of (P^d).

This Theorem gives the relation that connects the critical points in the primal problem to the critical points in the dual problem. It also suggest a relatively easy way to find the solution of the primal problem once the solution of the dual has been found.

In order to study extremalty conditions of the general analytic solution, we define the two following subsets of $\mathcal{S}_{\mathrm{a}}$:

$$\mathcal{S}_{\mathrm{a}}^{+} = \{\sigma \in \mathcal{S}_{\mathrm{a}} | \mathrm{G}(\sigma) \succ 0\}, \ \mathcal{S}_{\mathrm{a}}^{-} = \{\sigma \in \mathcal{S}_{\mathrm{a}} | \mathrm{G}(\sigma) \prec 0\}.$$

Theorem 15. *(Triality theory [6, 11]). Suppose that $\bar{\sigma}$ is a critical point of P^d and that $\bar{\boldsymbol{z}} = G(\bar{\sigma})^{-1}\boldsymbol{f}$. If $\bar{\sigma} \in \mathcal{S}_a^{+}$, then $\bar{\boldsymbol{z}}$ is a global minimizer of (P), $\bar{\sigma}$ is a global maximizer of (P^d), and*

$$\min_{\boldsymbol{z} \in \mathcal{Z}_{\mathrm{a}}} P(\bar{\boldsymbol{z}}) = \Xi(\bar{\boldsymbol{z}}, \bar{\sigma}) = \max_{\sigma \in \mathcal{S}_{\mathrm{a}}^{+}} P^d(\bar{\sigma}) \tag{2.22}$$

If $\bar{\sigma} \in \mathcal{S}_a^{-}$ and Problem (P) has the same dimension as (P^d), then on a neighborhood $\mathcal{Z}_0 \times \mathcal{S}_0 \subset \mathcal{Z}_{\mathrm{a}} \times \mathcal{S}_{\mathrm{a}}^{-}$ of $(\bar{\boldsymbol{z}}, \bar{\sigma})$, we have either

$$\min_{\boldsymbol{z} \in \mathcal{Z}_0} P(\bar{\boldsymbol{z}}) = \Xi(\bar{\boldsymbol{z}}, \bar{\sigma}) = \min_{\sigma \in \mathcal{S}_0} P^d(\bar{\sigma}) \tag{2.23}$$

or

$$\max_{\boldsymbol{z} \in \mathcal{Z}_0} P(\bar{\boldsymbol{z}}) = \Xi(\bar{\boldsymbol{z}}, \bar{\sigma}) = \max_{\sigma \in \mathcal{S}_0} P^d(\bar{\sigma}). \tag{2.24}$$

This triality theory gives a subset of the feasible set for the dual variable where to search for the global solution of the problem. It can be used to identify both global and local extrema of the nonconvex problem (P). Extensive applications of the canonical duality theory have been given in fields of computational biology [13], mathematical physics [12], and discrete and network optimization [7, 10].

In Order to show the potentiality of the Canonical Dual Theory, we briefly show in Figure 2.1 a one dimensional example of the double well energy function. It is possible to notice that all the three critical points in the primal have corresponding points in the dual problem and they have the same value of their respective objective functions. On the right side of the figure there is $\mathcal{S}_a^{+}$. The global maximum of the dual in $\mathcal{S}_a^{+}$ corresponds to the global minimum in the dual problem. All the three theorems previously reported are satisfied and the global solution of the problem is easily found in the dual problem.

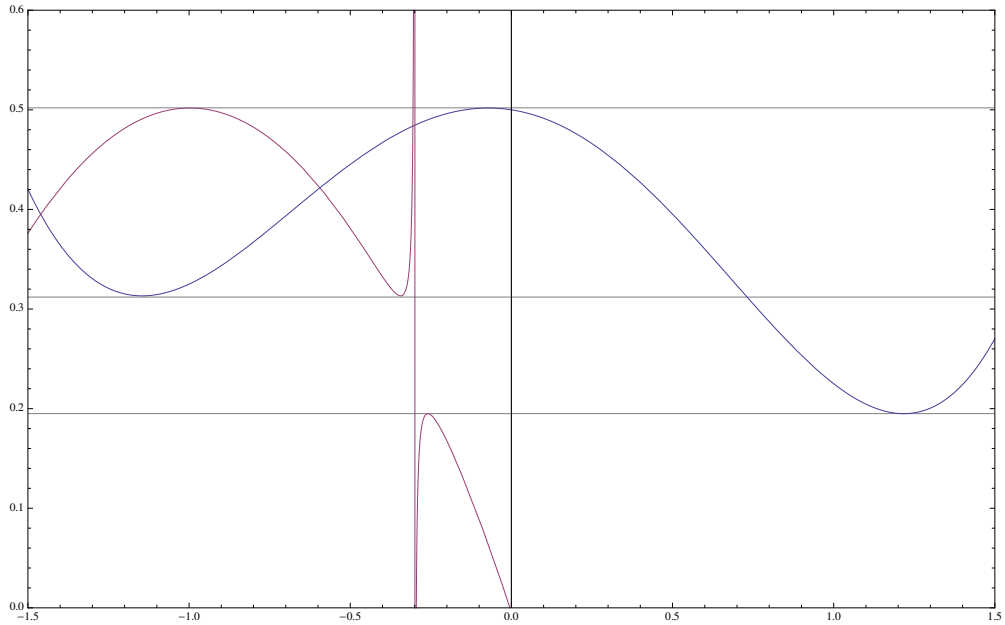

Figure 2.1. Comparison between the primal problem (in blue) and the dual problem (in red) for the one dimensional double well function.

Some Issues and Warnings about the use of Canonical Duality

There are some issues about the use of canonical duality theory. Triality theory was originally discovered in nonconvex mechanics [16]. The general warning that is given about the target function is that it must be objective. Objectivity in physics means that the objective function does not depend on the coordinates but only certain measures as for example the l_p norms. The function with such property can be measured in several ways, but their values is always the same.

Definition 7. *let*

$$\mathcal{Q} = \{Q \in \mathbb{R}^{m\times m} | Q^T = Q^{-1}, \det Q = 1\}$$

be a proper orthogonal rotation group. A subset $\mathcal{Y}_a \subset \mathbb{R}^m$ is said to be objective if $Q\mathbf{y} \in \mathcal{Y}_\mathbf{a}$ $\forall \mathbf{y} \in \mathcal{Y}_\mathbf{a}$ and $\forall Q \in \mathcal{Q}$. A real-valued function $T : \mathcal{Y}_a \to \mathbb{R}$ is said to be objective if its domain is objective and

$$T(Q\mathbf{y}) = T(\mathbf{y}) \quad \forall \mathbf{y} \in \mathcal{Y}_a \text{ and } \forall Q \in \mathcal{Q}$$

In other words the objective function does not depend on rotation. The canonical duality theory was developed from physics where the target functions are objective in order to be measurable, so if canonical duality theory is applied to non-objective

functions it is not guaranteed to work well. Also for the dual canonical transformation, geometrical operators are used. Without objectivity, the use of geometrical operators could not be theoretically well justified.

Another important issue is to destroy the symmetry of the problem, that is to assure that there is some kind of internal or external input that destroys the state of equilibrium in which the problem is. For example in the double well function, if the linear term **f** is reduced we have the result showed in Figure 2.2. By lowering

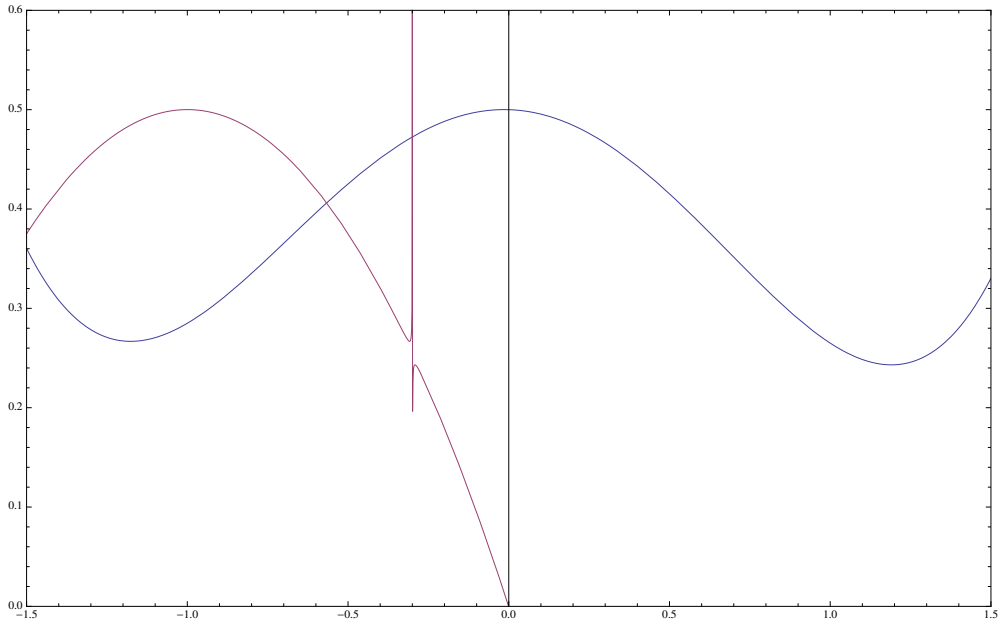

Figure 2.2. Double well function problem with a low value for the external input, the primal is in blue and the dual in red.

the value of the external input **f**, the two minima assume the same value of the primal objective function and the corresponding critical points in the dual get closer and closer. In Figure 2.3 the external input is zero. With this the symmetry is restored, and the two critical points compensate each other making the singularity in the dual disappear. Because of this compensation, there is only a critical point in the dual that corresponds to the maximum in the primal. Because of this, if the tryality theorem is applied to this problem and the global maximum is accepted as the solution in the dual, the worst possible critical point, the maximum in the primal problem, is chosen as optimal. It is possible to prove, for a great array of applications, that with a small enough perturbation the resulting problem is still equivalent to the original one.

Finally it is important to define, step by step, the domains of the primal and the dual problem. As a matter of facts the general solution described in Theorem 15

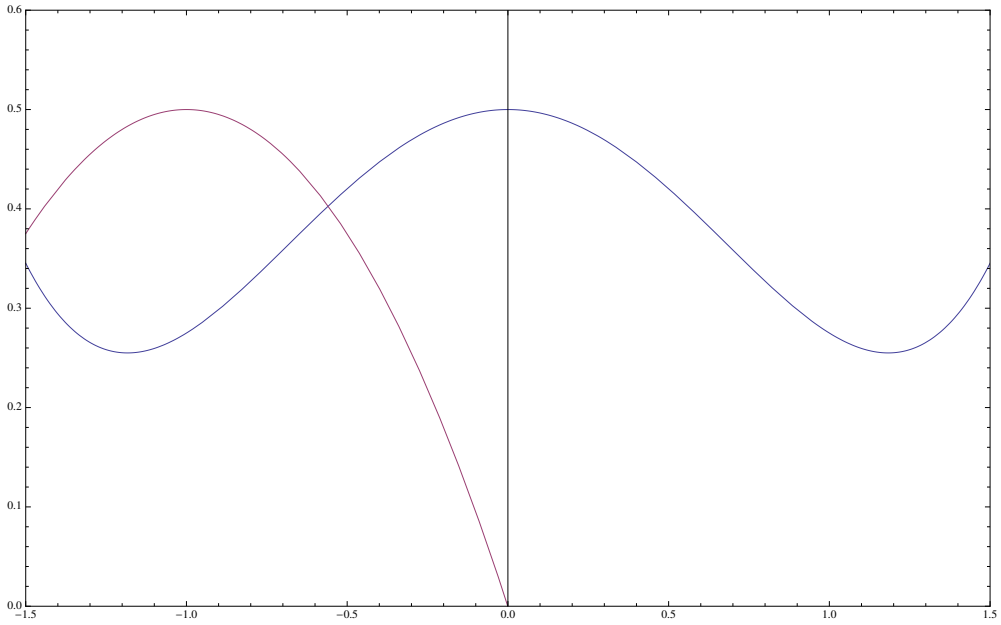

Figure 2.3. Effect of the symmetry on the double well function, the only critical point in the dual (in red) is the maximum.

can be easily found once the set $\mathcal{S}_a^+$ is defined. For more complex problems it can be convenient to define other subsets in order to classify the solution.

2.2.1 Canonical Dual Radial Basis Neural Networks

We apply canonical duality theory on radial basis neural networks among the different surrogate models we introduced in the previous chapter on the predictive models. In detail we decided to apply this theory on the formulation that considers both the weights and the centers as variables of the problem. We report the unconstrained optimization problem once again here for convenience:

$$E(\mathbf{w},\mathbf{c}) = \frac{1}{2}\sum_{p=1}^{P}\sum_{i=1}^{N}(w_i\phi(\mathbf{c}_i) - y^p)^2 + \frac{1}{2}\beta_w\|\mathbf{w}\|^2 + \frac{1}{2}\beta\sum_{i=1}^{N}\sum_{j=1}^{n}c_{ji}^2. \tag{2.25}$$

This problem is non-convex, but from empirical experiments in [20] it emerged that it generally yields neural networks with an higher precision than the ones trained with strategy that uses only the weights as variables. However, due to the non-convexity of the problem (2.25), there are some fundamental difficulties to find the global minimum of the problem and to characterize local minima. Indeed, the problem (2.25) is considered to be NP-hard even if the radial basis function $\phi(\mathbf{c})$ is a quadratic function and $n = 1$ [17, 18]. Another issue that characterizes this problem,

as we explained when we introduced this surrogate model for the first time, is the choice of the regularization parameters β_w and β_2. In general a cross-validation strategy is applied in order to find these regularization parameters. Until now it was not possible to find a closed form for the optimal values of these parameters in the general case. If it is possible to find at least an upper bound for these parameters, the time needed to perform a cross validation would greatly decrease.

In this section we study the canonical duality theory for solving the general Radial Basis Neural Networks optimization problem (2.25) and mainly analyze one-dimensional case in order to find properties and intuitions that can be useful for the multidimensional cases that will be analyzed in the following sections.

Primal problem for general Radial Basis Functions(RBF) and Dual Formulation

The general one dimensional non-convex function to be addressed in this paper can be proposed in the following form:

$$P(c) = W(c) + \frac{1}{2}\beta c^2 - fc,$$

where β is the regularization coefficient and f is a positive scalar close to zero. The term $-fc$ is not comprised in the original Radial Basis Neural Networks formulation but we consider it for the general mathematical case. The non-convex function $W(c)$ depends on the choice of the radial basis function $\phi(\cdot)$:

$$W(c) = \frac{1}{2}\left(w\phi(\|x-c\|^2) - y\right)^2, \tag{2.26}$$

where x, y and w belong to $\mathbb{R}$. In applications the parameter w is also a variable, but the original problem (2.25) is convex in w while non-convex in respect to the center of the radial basis function c. Therefore, the one-dimensional non-convex primal problem can be formulated as

$$(\mathcal{P}): \quad \min\left\{P(c) = \frac{1}{2}\left(w\phi(\|x-c\|^2) - y\right)^2 + \frac{1}{2}\beta c^2 - fc \quad | \ \forall c \in \mathbb{R}\right\}. \tag{2.27}$$

In order to apply the canonical duality theory to solve this problem, we need to choose the geometrically nonlinear operator:

$$\xi = \Lambda(c) = w\phi(\|x-c\|^2): \quad \mathbb{R} \to \mathcal{E}_{\mathrm{a}}. \tag{2.28}$$

Clearly, this is a nonlinear map from $\mathbb{R}$ to a subsapce $\mathcal{E}_{\mathrm{a}} \in \mathbb{R}$, which depends on the choice of the Radial Basis Function $\phi(\cdot)$. The *canonical function* associated with this geometrical operator is

$$V(\xi(c)) = \frac{1}{2}(\xi(c) - y)^2 = W(\Lambda(c)). \tag{2.29}$$

By the definition introduced in [15], $V : \mathcal{E}_{\rm a} \to \mathbb{R}$ is said to be canonical function on $\mathcal{E}_{\rm a}$ if for any given $\xi \in \mathcal{E}_{\rm a}$, the duality relation

$$\sigma = V'(\xi) = \{\xi - y\} : \mathcal{E}_{\rm a} \to \mathcal{S}_{\rm a} \tag{2.30}$$

is invertible, where $\mathcal{S}_{\rm a}$ is the range of the duality mapping $\sigma = \partial V(\xi)/\partial\xi$, which depends on the choice of the Radial Basis Function $\phi(\cdot)$. The couple (ξ,σ) forms a canonical duality pair on $\mathcal{E}_{\rm a} \times \mathcal{S}_{\rm a}$ with the Legendre conjugate $V^*(\sigma)$ defined by

$$V^*(\sigma) = \{\xi\sigma - V(\xi)|\sigma = V'(\xi)\} = \left(\frac{1}{2}\sigma^2 + y\sigma\right). \tag{2.31}$$

By considering that $W(c) = \Lambda(c)\sigma - V^*(\sigma)$, the primal function $P(c)$ can be reformulated as the so-called *total complementarity function* defined by

$$\begin{aligned} \Xi(c,\sigma) &= \Lambda(w,c)\sigma - V^*(\sigma) + \frac{1}{2}\beta c^2 - fc \\ &= w\phi(\|x-c\|^2)\sigma - \left(\frac{1}{2}\sigma^2 + \sigma y\right) + \frac{1}{2}\beta c^2 - fc. \end{aligned} \tag{2.32}$$

The function $\phi(\cdot)$ can be a non convex function just like $W(c)$. For this reason we have to perform a sequential canonical dual transformation for the nonlinear operator $\Lambda(c)$. To this aim we choose a second nonlinear operator:

$$\epsilon = \Lambda_2(c) = \|x - c\|^2 \tag{2.33}$$

which is a map from $\mathbb{R}$ to $\mathcal{E}_{\rm b} = \{\epsilon \in \mathbb{R} | \epsilon \geq 0\}$. In terms of ϵ, the first level operator $\xi = \Lambda(c)$ can be written as

$$\xi = U(\epsilon) = w\phi(\epsilon). \tag{2.34}$$

We assume that $U(\epsilon)$ is a convex function on $\mathcal{E}_{\rm b}$ such that the second-level duality relation

$$\tau = U'(\epsilon) = w\phi'(\epsilon) \tag{2.35}$$

is invertible, i.e.,

$$\epsilon = \left(\phi'\left(\frac{\tau}{w}\right)\right)^{-1}, \tag{2.36}$$

where the term $\left(\phi'\left(\frac{\tau}{w}\right)\right)^{-1}$ is the inverse of the function $\phi'(\epsilon)$. Thus, the Legendre conjugate of U can be obtained uniquely by

$$U^*(\tau) = \tau\left(\phi'\left(\frac{\tau}{w}\right)\right)^{-1} - w\phi\left(\left(\phi'\left(\frac{\tau}{w}\right)\right)^{-1}\right). \tag{2.37}$$

We notice that $\xi = w\phi(\epsilon)$. By substituting the value of ϵ given by (2.36) we find a relation that connects the first level primal variable ξ with the second level dual variable τ:

$$\xi = w\phi\left(\left(\phi'\left(\frac{\tau}{w}\right)\right)^{-1}\right). \tag{2.38}$$

By plugging this in (2.30) we obtain

$$\sigma = w\phi\left(\left(\phi'\left(\frac{\tau}{w}\right)\right)^{-1}\right) - y.$$

Generally speaking, it is possible, for certain functions ϕ, to use the canonical dual transformation to find the relation between the first level dual variable σ and the second level dual variable τ by means of the derivatives of $\phi(\cdot)$ and the first primal variable ξ. In general this relation is:

$$\tau = w\phi'\left(\phi^{-1}\left(\frac{\sigma + y}{w}\right)\right). \tag{2.39}$$

Therefore, replacing $U(\xi) = \Lambda(c)$ by its Legendre conjugate U^*, the total complementarity function becomes

$$\Xi(c,\sigma,\tau) = \left(\|x_p - c_i\|^2\tau - U^*(\tau)\right)\sigma - V^*(\sigma) + \frac{1}{2}\beta c^2 - fc. \tag{2.40}$$

It is also possible to rewrite the total complementary function (2.40) in the following form:

$$\Xi(c,\sigma,\tau) = \frac{1}{2}c^2(2\tau\sigma + \beta) - c(2\tau\sigma x + f) - U^*(\tau)\sigma - V^*(\sigma) + x^2\tau\sigma. \tag{2.41}$$

By the criticality condition $\partial\Xi(c,\sigma,\tau)/\partial c = 0$ we obtain

$$c(\tau,\sigma) = \frac{2\tau x\sigma + f}{2\tau\sigma + \beta}. \tag{2.42}$$

Clearly, if $2\tau\sigma + \beta \neq 0$, the general solution of (2.42) is

$$c = \frac{2\tau x\sigma + f}{2\tau\sigma + \beta} \quad \forall(\sigma,\tau) \in \mathcal{S}_{\mathrm{a}} = \{\sigma,\tau |\ 2\tau\sigma + \beta \neq 0\} \tag{2.43}$$

and the canonical dual function of $P(c)$ can be presented as

$$P^d(\sigma,\tau) = -\frac{1}{2}\frac{(2\tau x\sigma + f)^2}{2\tau\sigma + \beta} - U^*(\tau)\sigma - V^*(\sigma) + x^2\tau\sigma. \tag{2.44}$$

By considering dual relation given in (2.39) we can write the total complementarity function in terms of c and σ only

$$\Xi(c,\sigma) = \frac{1}{2}c^2G(\sigma) - cF(\sigma) - U^*(\sigma)\sigma - V^*(\sigma) + x^2w\phi'\left(\phi^{-1}\left(\frac{\sigma+y}{w}\right)\right)\sigma \quad (2.45)$$

where

$$\begin{aligned} G(\sigma) &= 2w\phi'\left(\phi^{-1}\left(\frac{\sigma+y}{w}\right)\right)\sigma + \beta, \\ F(\sigma) &= 2w\phi'\left(\phi^{-1}\left(\frac{\sigma+y}{w}\right)\right)x\sigma + f, \\ U^*(\sigma) &= w\phi'\left(\phi^{-1}\left(\frac{\sigma+y}{w}\right)\right)\phi^{-1}\left(\frac{\sigma+y}{w}\right) - (\sigma+y). \end{aligned}$$

Therefore, in terms of σ only, the canonical dual function can be written as

$$P^d(\sigma) = -\frac{1}{2}\frac{F(\sigma)^2}{G(\sigma)} - U^*(\sigma)\sigma - V^*(\sigma) + x^2w\phi'\left(\phi^{-1}\left(\frac{\sigma+y}{w}\right)\right)\sigma. \quad (2.46)$$

Complementary-Dual Principle

In this section we present a theorem that is a particular case for Theorems 13 and 14, with its proof.

Theorem 16. *If $\bar{\sigma}$ is a critical point of (P^d) and the term:*

$$G'(\bar{\sigma}) = \left[w\phi'\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right) + \sigma\phi''\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right)\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right)'\right] \neq 0 \quad (2.47)$$

then the point

$$\bar{c} = \frac{F(\bar{\sigma})}{G(\bar{\sigma})} \quad (2.48)$$

is a critical point of P and $P(\bar{c}) = P^d(\bar{\sigma})$

Proof. Suppose that $\bar{\sigma}$ is a critical point of P^d then we have

$$P^d(\bar{\sigma})' = \left[\bar{c}^2 - 2xc + x^2 - \phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right]G'(\bar{\sigma}) - \quad (2.49)$$

$$\sigma\left[\phi'\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right)\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right)' - 1\right] = 0$$

Notice that

$$\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right)' = \frac{1}{\phi'(\epsilon)} = \frac{1}{\phi'\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right)},$$

The third term in (2.49) is zero. The term $G'(\bar{\sigma})$ is not zero from the hypothesis, so we obtain

$$(x-\bar{c})^2 - \phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right) = 0, \tag{2.50}$$

that is

$$\bar{\sigma} = w\phi\left((x-\bar{c})^2\right) - y. \tag{2.51}$$

The critical point condition for the primal problem $P(c)$ is

$$P'(c) = 0 \rightarrow -2w(x-c)\phi'(\|x-c\|^2)(w\phi(\|x-c\|^2) - y) + \beta c - f = 0.$$

By considering that $\phi'(\|x-c\|^2) = \phi'\left(\phi^{-1}\left(\frac{\sigma+y}{w}\right)\right)$ and $\sigma = w\phi\left((x-c)^2\right) - y$ we obtain

$$2w(x-c)\phi'\left(\phi^{-1}\left(\frac{\sigma+y}{w}\right)\right)\sigma + \beta c - f = 0, \tag{2.52}$$

that is

$$c = \frac{2\phi'\left(\phi^{-1}\left(\frac{\sigma+y}{w}\right)\right)\sigma + f}{2\phi'\left(\phi^{-1}\left(\frac{\sigma+y}{w}\right)\right)\sigma + \beta}. \tag{2.53}$$

By setting $\sigma = \bar{\sigma}$ in (2.53) we obtain (2.43) proving that $\bar{c}$ is a critical point of $P(c)$. For the correspondence of the function values we start from the dual function

$$P^d(\bar{\sigma}) = -\frac{1}{2}\frac{F^2(\bar{\sigma})}{G(\bar{\sigma})} - U^*(\bar{\sigma})\bar{\sigma} - V^*(\bar{\sigma}) + x^2 w\phi'\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right)\bar{\sigma}$$

add and subtract the term $\frac{1}{2}\frac{F^2(\bar{\sigma})}{G(\bar{\sigma})}$ and substitute the value of $\bar{c}$

$$\frac{1}{2}\bar{c}^2 G(\bar{\sigma}) - \bar{c}F(\bar{\sigma}) - U^*(\bar{\sigma})\bar{\sigma} - V^*(\bar{\sigma}) + x^2 w\phi'\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right)\bar{\sigma}$$

by reordering the terms we obtain

$$\left(\|x-\bar{c}\|^2 w\phi'\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right) - U^*(\bar{\sigma})\right)\bar{\sigma} - V^*(\bar{\sigma}) + \frac{1}{2}\beta\bar{c}^2 - f\bar{c},$$

Considering the (2.30), setting $\bar{\epsilon} = \|x-\bar{c}\|^2$ and $\phi'\left(\phi^{-1}\left(\frac{\bar{\sigma}+y}{w}\right)\right) = \phi'(\bar{\epsilon})$ we obtain:

$$[w\phi'(\bar{\epsilon})\bar{\epsilon} - w\phi'(\bar{\epsilon})\bar{\epsilon} + w\phi(\bar{\epsilon})][w\phi(\bar{\epsilon}) - y] - \left[\frac{1}{2}(w\phi(\bar{\epsilon}) - y)^2 + y(w\phi(\bar{\epsilon}) - y)\right] +$$

$$\frac{1}{2}\beta\bar{c}^2 - f\bar{c} = w^2\phi(\bar{\epsilon})^2 - yw\phi(\bar{\epsilon}) - \frac{1}{2}(w\phi(\bar{\epsilon}) - y)^2 - yw\phi(\bar{\epsilon}) + y^2 + \frac{1}{2}\beta\bar{c}^2 - f\bar{c}$$

by collecting the terms we obtain:

$$(w\phi(\bar{\epsilon}) - y)^2 - \frac{1}{2}(w\phi(\bar{\epsilon}) - y)^2 + \frac{1}{2}\beta\bar{c}^2 - f\bar{c},$$

that is

$$\frac{1}{2}\left(w\phi(\|x-\bar{c}\|^2)-y\right)^2+\frac{1}{2}\beta\bar{c}^2-f\bar{c}=P(\bar{c}).$$

that proves the theorem □

Theorem 16 shows that the problem $(\mathcal{P}^d)$ is canonically dual to the primal $(\mathcal{P})$ in the sense that the duality gap is zero.

Gaussian function

One of the most used RBF is the Gaussian function. In this section we will analyze the problem with $\phi(\|x-c\|^2)=\exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\}$, where α is a parameter that represents the standard deviation of the Gaussian function. In the Radial Basis formulation normally there is not the linear term fc. The primal problem is:

$$\min P(c)=\frac{1}{2}\left(w\exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\}-y\right)^2+\frac{1}{2}\beta c^2\quad\forall c\in\mathbb{R}. \tag{2.54}$$

For this problem, the nonlinear operator $\xi:\mathbb{R}\to\mathcal{E}_{\mathrm{a}}$ from (2.28) becomes

$$\xi=w\exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\}.$$

The expressions that define σ, V and V^* are the same as the general problem that is:

- $V(\xi(c))=\frac{1}{2}(\xi-y)^2$;
- $\sigma=\xi-y$;
- $V^*(\sigma)=\left(\frac{1}{2}\sigma^2+y\sigma\right)$.

The second order operator $\Lambda_2(c):\mathbb{R}\to\mathcal{E}_{\mathrm{b}}$ is

$$\epsilon=\Lambda_2(c)=\|x-c\|^2=\epsilon \tag{2.55}$$

The second level canonical function becomes

$$U(\epsilon)=w\exp\left\{-\frac{\epsilon}{2\alpha^2}\right\}.$$

And the second order duality mapping τ is

$$\tau=w\phi'(\epsilon)=-\frac{w}{2\alpha^2}\exp\left\{-\frac{\epsilon}{2\alpha^2}\right\}.$$

So the Lagrende conjugate $U^* : \mathcal{S}'_{\mathrm{b}} \to \mathbb{R}$ is

$$U^*(\tau) = \tau\left(\phi^{-1}\left(\frac{\tau}{w}\right)\right)' - w\phi\left(\phi^{-1}\left(\frac{\tau}{w}\right)\right)' = -2\alpha^2\tau\left(log\left(\frac{-2\alpha^2\tau}{w}\right) - 1\right).$$

The derivative of the exponential function is the exponential function itself. This simplifies the relation (2.38) between ξ and τ making it linear, that is $\xi = -\frac{\tau}{2\alpha^2}$. The relation between σ and τ is:

$$\tau = -\frac{(\sigma + y)}{2\alpha^2}$$

that is also linear. The total complementarity function becomes:

$$\Xi(c,\sigma) = \frac{1}{2}c^2G(\sigma) - cF(\sigma) - U^*(\sigma)\sigma - V^*(\sigma) - \frac{x^2(\sigma^2 + y\sigma)}{2\alpha^2}$$

where:

- $G(\sigma) = \beta - \frac{\sigma^2+y\sigma}{\alpha^2}$
- $F(\sigma) = -\frac{x\sigma^2+xy\sigma}{\alpha^2}$
- $U^*(\sigma) = (\sigma + y)\left(log\left(\frac{\sigma+y}{w}\right) - 1\right)$

The dual problem is

$$P^d(\sigma) = -\frac{1}{2}\frac{\left(-\frac{x\sigma^2+xy\sigma}{\alpha^2}\right)^2}{\beta - \frac{\sigma^2+y\sigma}{\alpha^2}} - \left(log\left(\frac{\sigma + y}{w}\right)\right)\left(\sigma^2 + y\sigma\right) + \frac{1}{2}\sigma^2 - \frac{x^2(\sigma^2 + y\sigma)}{2\alpha^2} \quad (2.56)$$

The domains of the variables in the primal and dual problems are:

- $\mathcal{E}_{\mathrm{b}} = \{\epsilon \in \mathbb{R} | \epsilon \geq 0\}$
- $\mathcal{S}_{\mathrm{b}} = \{\tau \in \mathbb{R} | -\infty < \tau < 0\}$ if $w > 0$, $\mathcal{S}_{\mathrm{b}} = \{\tau \in \mathbb{R} | -\infty < \tau < 0\}$ if $w < 0$
- $\mathcal{E}_{\mathrm{a}} = \{\xi \in \mathbb{R} | 0 \leq \xi \leq \mathrm{w}\}$
- $\mathcal{S}_{\mathrm{a}} = \{\sigma \in \mathbb{R} | -\mathrm{y} \leq \sigma \leq \mathrm{w} - \mathrm{y}\}$ if $w > 0$, $\mathcal{S}_{\mathrm{a}} = \{\sigma \in \mathbb{R} | \mathrm{w} - \mathrm{y} \leq \sigma \leq -\mathrm{y}\}$ if $w < 0$

Remark 2.2.1. *Parameters β, x, y, and w play an important role in solving the non-convex problem (P). In the original problem (2.27) one searches for the value of c that brings the term $w \exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\}$ as closer as possible to y, that is $\sigma = w \exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\} - y = 0$.*

If $y < 0$ and $w > 0$ or $y > 0$ and $w < 0$ we will have that $|\sigma| > 0$. This means that in the case of the exponential function, it would be better to choose c as bigger as possible in order to make the exponential go to zero, but the result would never be satisfactory as the error committed by the approximation would go close to $-y$ as c goes to infinity. The value $-y$ is not a good value for the error as it is far from zero. On the other hand if y and w have the same sign and $|y| > |w|$ the value of c will be x in order to have the exponential equal to 1 and to have the lowest value for $\sigma = w \exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\} - y$.
In order to have a realistic problem we will consider the case with y and w with the same sign, and with $|y| < |w|$. The cases with y,$w > 0$ and y,$w < 0$ are equivalent, so we will suppose that both y and w are positive without losing generality.

Theorem 17. *Suppose that $\bar{\sigma} \in \mathcal{S}_a$ is a critical point of the dual problem (2.56) with the corresponding $\bar{c} = \frac{F(\bar{\sigma})}{G(\bar{\sigma})} \in \mathbb{R}$ and that $\bar{\sigma} \neq \frac{y}{2}$. Then $\bar{c}$ is a critical point of the primal problem and:*

$$P^d(\bar{\sigma}) = P(\bar{c}). \tag{2.57}$$

moreover, there are the following relations between the critical points of the primal problem and the dual problem:

1. *If $(2\bar{\sigma} + y) > 0$ and $G(\bar{\sigma}) \geq 0$ or $(2\bar{\sigma} + y) < 0$ and $G(\bar{\sigma}) \leq 0$ then if $\bar{\sigma}$ is a local minimum of the dual problem, the corresponding $\bar{c}$ is a local maximum of the primal problem; if $\bar{\sigma}$ is a local maximum of the dual problem the corresponding $\bar{c}$ is a local minimum of the primal problem;*

2. *If $(2\bar{\sigma} + y) > 0$ and $G(\bar{\sigma}) \leq 0$ or $(2\bar{\sigma} + y) < 0$ and $G(\bar{\sigma}) \geq 0$ then if $\bar{\sigma}$ is a local minimum of the dual problem the corresponding $\bar{c}$ is a local minimum of the primal problem; if $\bar{\sigma}$ is a local maximum of the dual problem the corresponding $\bar{c}$ is a local maximum of the primal problem.*

Let $x_o = \sqrt{-2\alpha^2 Log\left(\frac{y}{2w}\right)}$. If $\bar{\sigma} = -\frac{y}{2}$, then there is a corresponding critical point to $\bar{\sigma}$ in the primal problem if and only if the parameters x, y, β and w satisfy one of the two following conditions:

$$\begin{aligned} \beta x + \left(\beta + \tfrac{y^2}{4\alpha^2}\right) x_o = 0 \\ \beta x - \left(\beta + \tfrac{y^2}{4\alpha^2}\right) x_o = 0 \end{aligned} \tag{2.58}$$

and the corresponding critical point $\bar{c}$ in the primal problem is always a local minimum. If neither of conditions (2.58) is satisfied, $\bar{\sigma} = -\frac{y}{2}$ is always a critical point of the dual problem, but it does not have any corresponding critical point in the primal problem.

Proof. The first order derivative for the dual problem is:

$$P^d(\bar{\sigma})' = -\left[\left(x - \frac{-\frac{\bar{\sigma}^2 x+\bar{\sigma}xy}{\alpha^2}}{\beta - \frac{\bar{\sigma}^2+\bar{\sigma}y}{2\alpha^2}}\right)^2 \frac{1}{2\alpha^2} + log\left(\frac{\bar{\sigma}+y}{w}\right)\right][2\bar{\sigma}+y] = 0 \tag{2.59}$$

so the term (2.47) is equal to $2\bar{\sigma}+y$. If $\bar{\sigma} \neq -\frac{y}{2}$, the critical point equivalency and condition (2.57) are consequences of Theorem 16.
To prove statements 1 and 2 we use the second order derivatives of the problems $P(c)$ and $P^d(\sigma)$

$$P(c)'' = \beta - \frac{1}{\alpha^2} w \exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\}\left(w\exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\} - y\right) + \tag{2.60}$$

$$\frac{(x-c)^2}{\alpha^4}\exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\}\left(2w\exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\} - y\right)$$

$$P^d(\sigma)'' = -\frac{2\sigma+y}{\sigma+y} - 2log\left(\frac{\sigma+y}{w}\right) - \frac{1}{\alpha^2}\left(x - \frac{-\frac{x\sigma^2+xy\sigma}{\alpha^2}}{\beta - \frac{\sigma^2+y\sigma}{\alpha^2}}\right)^2\left(1 + \frac{(2\sigma+y)^2}{\alpha^2(\beta - \frac{\sigma^2+y\sigma}{\alpha^2})}\right). \tag{2.61}$$

Since $\bar{\sigma}$ is a critical point of the dual, we have that $P^d(\sigma)' = 0$. Therefore when $\bar{\sigma} \neq -\frac{y}{2}$:

$$\left(x - \frac{-\frac{\bar{\sigma}^2 x+\bar{\sigma}xy}{\alpha^2}}{\beta - \frac{\bar{\sigma}^2+\bar{\sigma}y}{2\alpha^2}}\right)^2 = -2\alpha^2 log\left(\frac{\bar{\sigma}+y}{w}\right) \tag{2.62}$$

By using condition (2.62) in (2.61) we obtain:

$$P^d(\bar{\sigma})'' = (2\bar{\sigma}+y)\left(\frac{2log\left(\frac{\bar{\sigma}+y}{w}\right)(2\bar{\sigma}+y)}{\alpha^2(\beta - \frac{\bar{\sigma}^2-\bar{\sigma}y}{\alpha^2})} - \frac{1}{\bar{\sigma}+y}\right). \tag{2.63}$$

Noticing $\sigma = w\exp\left\{-\frac{\|x-c\|^2}{2\alpha^2}\right\} - y$, it is possible to rewrite $P(\bar{c})''$ in terms of $\bar{\sigma}$, i. e.:

$$P(c(\bar{\sigma}))'' = \beta - \frac{\bar{\sigma}^2 - \bar{\sigma}y}{\alpha^2} + \frac{2}{\alpha^2}(\bar{\sigma}+y)(2\bar{\sigma}+y)\left(x - \frac{-\frac{x\sigma^2+xy\sigma}{\alpha^2}}{\beta - \frac{\sigma^2+y\sigma}{\alpha^2}}\right)^2. \tag{2.64}$$

by using again condition (2.62) we obtain:

$$P(c(\bar{\sigma}))'' = \frac{1}{\alpha^2}\left[\alpha^2\left(\beta - \frac{\bar{\sigma}^2 - \bar{\sigma}y}{\alpha^2}\right) - 2(\bar{\sigma}+y)(2\bar{\sigma}+y)log\left(\frac{\bar{\sigma}+y}{w}\right)\right]. \tag{2.65}$$

It is also possible to rewrite equation (2.63) in the following form:

$$P^d(\bar{\sigma})'' = -(2\bar{\sigma}+y)\left(\frac{\alpha^2(\beta-\frac{\bar{\sigma}^2-\bar{\sigma}y}{\alpha^2})-2(2\bar{\sigma}+y)(\bar{\sigma}+y)log\left(\frac{\bar{\sigma}+y}{w}\right)}{\alpha^2(\beta-\frac{\bar{\sigma}^2-\bar{\sigma}y}{\alpha^2})(\bar{\sigma}+y)}\right). \tag{2.66}$$

Due to the fact that $G(\sigma)=\beta-\frac{\sigma^2-\sigma y}{\alpha^2}$ we have:

$$P^d(\bar{\sigma})'' = -\frac{2\bar{\sigma}+y}{G(\bar{\sigma})(\bar{\sigma}+y)}P(c(\bar{\sigma}))''. \tag{2.67}$$

From this relation we have four possible cases:

$(2\bar{\sigma}+y)$	$G(\bar{\sigma})$	P	P^d
>0	>0	$\pm$	$\mp$
>0	<0	$\pm$	$\pm$
<0	<0	$\pm$	$\mp$
<0	>0	$\pm$	$\pm$

Table 2.1. Relations between the second order derivatives of the primal problem and the second order derivatives of the dual problem

From Table 1, we obtain:

1. if $(2\bar{\sigma}+y)>0$ and $G(\bar{\sigma})\geq 0$ or $(2\bar{\sigma}+y)<0$ and $G(\bar{\sigma})\leq 0$ then the second order derivate of the primal problem and the second order derivate of the dual problem have opposite sign at their critical points;

2. if $(2\bar{\sigma}+y)>0$ and $G(\bar{\sigma})\leq 0$ or $(2\bar{\sigma}+y)<0$ and $G(\bar{\sigma})\geq 0$ then the second order derivate of the primal problem and the second order derivate of the dual problem have the same sign at their critical points.

This proves statements 1 and 2.
The point $\bar{\sigma}=-\frac{y}{2}$ is a critical point of P^d according to the second part of the (2.59). The point $\bar{c}$ corresponding to $\bar{\sigma}=-\frac{y}{2}$ is a critical point of the primal problem if and only if $P'(\bar{c})=0$. We can use the (2.30) to find the relation between $\bar{\sigma}$ and $\bar{c}$ that is:

$$\bar{\sigma}=\bar{\xi}-y \rightarrow \bar{\sigma}=we^{-\frac{(x-\bar{c})^2}{2\alpha^2}}-y$$

$$\bar{c}=x\pm\sqrt{-2\alpha^2 Log\left(\frac{\bar{\sigma}+y}{w}\right)}.$$

For $\bar{\sigma}=-\frac{y}{2}$ we obtain:

$$\bar{c}=x\pm x_o. \tag{2.68}$$

Substituting these values in the first order derivative of the primal problem:

$$P'(\bar{c}) = \frac{(x-\bar{c})}{\alpha^2} w e^{-\frac{(x-\bar{c})^2}{2\alpha^2}} \left(w e^{-\frac{(x-\bar{c})^2}{2\alpha^2}} - y \right) + \beta \bar{c} \tag{2.69}$$

and considering that $w \exp\left\{-\frac{(x-\bar{c})^2}{2\alpha^2}\right\} = \bar{\sigma} + y = \frac{y}{2}$ and $w \exp\left\{-\frac{(x-\bar{c})^2}{2\alpha^2}\right\} - y = \bar{\sigma} = -\frac{y}{2}$ we obtain that the primal problem has a critical point at $\bar{c}$ corresponding to the critical $\bar{\sigma} = -\frac{y}{2}$ if and only if:

$$\beta x \pm \left(\beta + \frac{y^2}{4\alpha^2} \right) x_o = 0. \tag{2.70}$$

This happens only for a particular configuration of the parameters w, β, x and y that makes one of the roots the first term of the derivative (2.59):

$$- \left[\left(x - \frac{-\frac{\bar{\sigma}^2 x + \bar{\sigma} x y}{\alpha^2}}{\beta - \frac{\bar{\sigma}^2 + \bar{\sigma} y}{2\alpha^2}} \right)^2 \frac{1}{2\alpha^2} + log\left(\frac{\bar{\sigma} + y}{w} \right) \right] = 0$$

be in $\bar{\sigma} = -\frac{y}{2}$.
To prove that at $\bar{\sigma} = -\frac{y}{2}$ the critical point of the dual problem corresponds to a minimum point of the primal problem we plug the value of $\bar{\sigma} = -\frac{y}{2}$ in the (2.64) and obtain

$$P''(\bar{\sigma}) = \beta + \frac{y^2}{4\alpha^2},$$

which is always a positive value. □

Remark 2.2.2. *From now on we will refer to the critical point $\sigma_f = -\frac{y}{2}$ as pseudo dual critical point as it is a critical point of the dual problem that generally does not have a corresponding critical point for the primal problem.*

Choice of the critical point

In order to find the best solution among the critical points of problem (2.54) we introduce the following feasible spaces:

$$\mathcal{S}_{\mathrm{a}}^{+} = \{\sigma \in \mathcal{S}_{\mathrm{a}} | \mathrm{G}(\sigma) > 0\} \tag{2.71}$$

$$\mathcal{S}_{\mathrm{a}}^{-} = \{\sigma \in \mathcal{S}_{\mathrm{a}} | \mathrm{G}(\sigma) < 0\} \tag{2.72}$$

The following theorem explains the relations between the critical points:

Theorem 18. *Suppose that the point $\bar{\sigma}_1 \in \mathcal{S}_{\mathrm{a}}^+$ and $\bar{\sigma}_2 \in \mathcal{S}_{\mathrm{a}}^-$ are critical points of the dual problem, that $\bar{\sigma}_i \neq -\frac{y}{2}$ for $i = 1,2$ and that $\bar{c}_1$ and $\bar{c}_2$ are the corresponding critical points of the primal problem. Then if both $\bar{c}_1$ and $\bar{c}_2$ are local minima or local maxima of the primal problem, the following relation always holds:*

$$P(\bar{c}_1) = P^d(\bar{\sigma}_1) < P(\bar{c}_2) = P^d(\bar{\sigma}_2) \tag{2.73}$$

Proof. This theorem is a consequence of the first theorem in triality theory [6]. □

Remark 2.2.3. *The pseudo critical point $\sigma_f = -\frac{y}{2}$ will always be in S_a^+ as*

$$\mathcal{S}_{\mathrm{a}}^+ := \left\{ \sigma \in \mathbb{R} \frac{-\mathrm{y} - \sqrt{\mathrm{y} + 4\alpha^2\beta}}{2} < \sigma < \frac{-\mathrm{y} + \sqrt{\mathrm{y} + 4\alpha^2\beta}}{2} \right\}.$$

From the results in theorem 18 it is always better to search for the dual critical point in $\mathcal{S}_{\mathrm{a}}^+$ that corresponds to a minimum in the primal problem. In order to characterize the solutions in $\mathcal{S}_{\mathrm{a}}^+$ and the domains in which search for the best solution, two theorems are proposed in the following:

Theorem 19. *Let $\sigma_f = -\frac{y}{2}$ be the pseudo critical point of the dual problem, $x_o = \sqrt{-2\alpha^2 Log\left(\frac{y}{2w}\right)}$, x positive. Then:*

- *if $x \in (0,x_o)$ then σ_f is always a local minimum of $P^d(\sigma)$;*
- *if $x > x_o$ then:*
  1. *if $\beta > 0$ and $\beta < \frac{y^2 x_o}{4\alpha^2(x-x_o)}$, σ_f is a local minimum for the dual problem;*
  2. *if $\beta > 0$ and $\beta > \frac{y^2 x_o}{4\alpha^2(x-x_o)}$, σ_f is a local maximum for the dual problem;*
  3. *if $\beta > 0$, $\beta = \frac{y^2 x_o}{4\alpha^2(x-x_o)}$, σ_f is an inflection point in which the first order derivative is zero and that corresponds to a a local minimum of the primal problem.*

Proof. In order to understand that $\sigma_f = -\frac{y}{2}$ is a minimum or a maximum for the dual we have to plug its value in the second order derivative of $P^d(\sigma)$ that is equation (2.61) and analyze its sign. After the substitution we obtain

$$P^d(\sigma_f) = -\left[2log\left(-\frac{y}{2w}\right) + \frac{1}{\alpha^2}\left(\frac{x\beta}{\beta + \frac{y^2}{4\alpha^2}}\right)^2 \right]. \tag{2.74}$$

The first order derivate in β of (2.74) is $-\frac{2x\beta^2}{\alpha^2\left(\beta+\frac{y^2}{4\alpha^2}\right)^2}$, that is the function is monotonic decreasing in β. The value of (2.74) in $\beta = 0$ is $-log\left(-\frac{y}{2w}\right)$ that is positive. If we make β go to $+\infty$ we obtain:

$$\lim_{\beta\to+\infty} -\left[2log\left(-\frac{y}{2w}\right)+\frac{1}{\alpha^2}\left(\frac{x\beta}{\beta+\frac{y^2}{4\alpha^2}}\right)^2\right] = -2log\left(-\frac{y}{2w}\right)+\frac{x^2}{\alpha^2}$$

that is the second order derivative of $P^d(\sigma)$ in σ_f is non negative for any value of $\beta > 0$ if

$$x \in [-x_o,x_o]$$

If x does not satisfy this condition, from the (2.74) we have that the second order derivative of the dual problem is positive in σ_f if β satisfies:

$$\beta > \frac{-y^2x_o}{4\alpha^2\left(x+x_o\right)} \text{ and } \beta < \frac{y^2x_o}{4\alpha^2\left(x-x_o\right)}. \tag{2.75}$$

On the other hand if:

$$\beta < \frac{-y^2x_o}{4\alpha^2\left(x+x_o\right)} \text{ or } \beta > \frac{y^2x_o}{4\alpha^2\left(x-x_o\right)} \tag{2.76}$$

there will be a local maximum in σ_f. As x is considered positive, the term $\frac{-y^2x_o}{4\alpha^2(x+x_o)}$ is always negative, so β will always be greater than it.
If the condition $\beta = \frac{y^2x_o}{4\alpha^2(x-x_o)}$ is satisfied, the critical point σ_f is an inflection point that also satisfies the first order condition and it has a corresponding minimum point in the primal problem for theorem 17. □

Remark 2.2.4. *In the case of x negative, the conditions are changed in the following way:*

- *if $x \in (-x_o,0)$ then σ_f is always a local minimum of $P^d(\sigma)$*
- *if $x < -x_o$ then:*
  1. *if $\beta > 0$ and $\beta < \frac{-y^2x_o}{4\alpha^2(x+x_o)}$, σ_f is a local minimum for the dual problem;*
  2. *if $\beta > 0$ and $\beta > \frac{-y^2x_o}{4\alpha^2(x+x_o)}$, σ_f is a local maximum for the dual problem;*
  3. *if $\beta > 0$, $\beta = \frac{-y^2x_o}{4\alpha^2(x+x_o)}$, σ_f is an inflection point in which the first order derivative is zero and that corresponds to a a local minimum of the primal problem.*

The proof of these statement is similar to that of theorem 19 and can be omitted.

Remark 2.2.5. *Theorem 19 shows the effects of the parameter β on the pseudo critical point σ_f. Similar effects can also be obtained in respect to y, x, α, and w. The reason we choose β is because it is an hyper-parameter that can be chosen by the practitioner before performing the optimization.*

For the next theorem, we introduce the two following subsets of $\mathcal{S}_{\mathrm{a}}^{+}$:

$$\mathcal{S}_{\sharp}^{+} = \left\{\sigma \in \mathcal{S}_{a}^{+} | \sigma > -\frac{y}{2}\right\} \tag{2.77}$$

$$\mathcal{S}_{\flat}^{+} = \left\{\sigma \in \mathcal{S}_{a}^{+} | \sigma < -\frac{y}{2}\right\} \tag{2.78}$$

Theorem 20. *Let $\sigma_f = -\frac{y}{2}$ be the pseudo critical point in the dual problem and let the primal problem have a maximum of five critical points. Then*

- *if σ_f is a local minimum for the dual function, there will be a local maximum in $\mathcal{S}_{\sharp}^{+}$ that corresponds to a minimum of the primal problem.*
- *if σ_f is a local maximum then:*
  1. *there are no critical points in $\mathcal{S}_{\sharp}^{+}$;*
  2. *there is at least one critical point in $\mathcal{S}_{\flat}^{+}$*

Proof. In the dual problem there must be a singularity point in $G(\sigma) = 0$ that goes to $-\infty$, so if σ_f is a local minimum, there must be a local maximum in $\mathcal{S}_{\sharp}^{+}$.
If σ_f is a local maximum, we prove condition 1 by negating the thesis and suppose that there is a least one critical point in $\mathcal{S}_{\sharp}^{+}$. As $P^d(\sigma)$ goes to $-\infty$ if $G(\sigma) \to 0$, there will be no one, but two critical points in $\mathcal{S}_{\sharp}^{+}$, a local minimum σ_1 and a local maximum σ_2 with the relation $P^d(\sigma_1) < P^d(\sigma_2)$. For theorems 17 and 18, σ_1 corresponds to the second highest local maximum of the primal function c_1, and σ_2 corresponds to the lowest or second lowest local minimum of the primal function c_2, that is the relation $P(c_2) < P(c_1)$ is satisfied. By Theorem 16 we have:

$$P^d(\sigma_1) < P^d(\sigma_2) = P(c_2) < P(c_1) = P^d(\sigma_1)$$

that is a contradiction.
To prove condition 2, it is sufficient to notice that if there are no critical points in $\mathcal{S}_{\sharp}^{+}$, for the triality theory there must be at least one critical point corresponding to the global minimum in $\mathcal{S}_{\mathrm{a}}^{+}$ and this point will be in $\mathcal{S}_{\flat}^{+}$. □

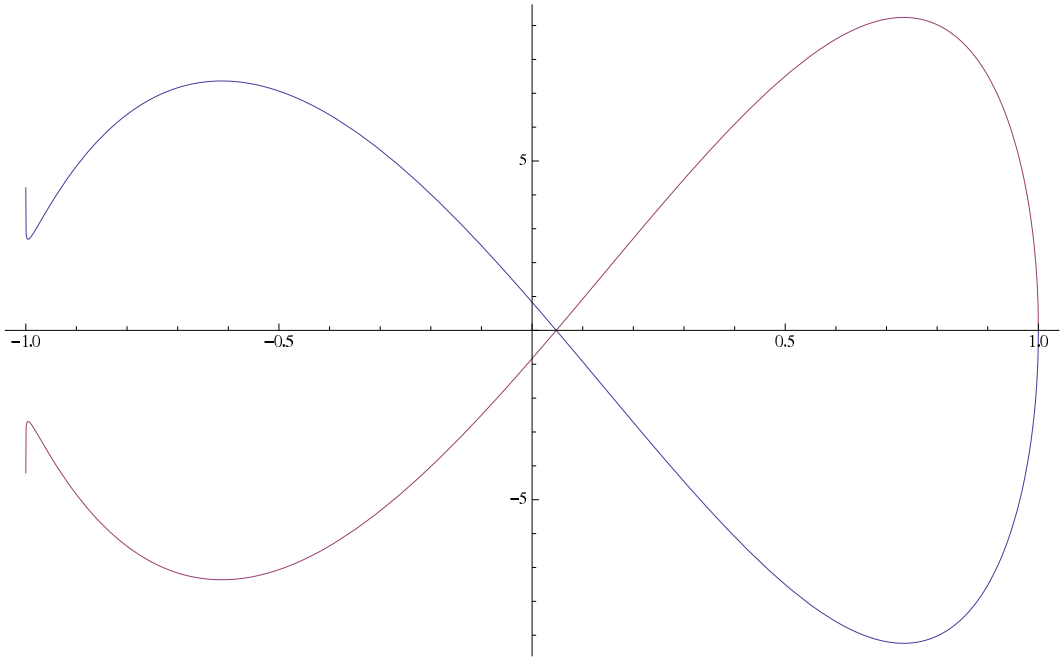

Figure 2.4. Dual algebraic curves with $y = 1$, $w = 2$, $\alpha = \frac{\sqrt{2}}{2}$ and $\beta = 0.1$ in respect to the internal input x

Depending on the parameters, the primal problem (2.54) can have at most five critical points. There are several cases:

Case 1: Three critical points for $P(c)$ and four critical points for $P^d(\sigma)$, two critical point in $\mathcal{S}_{\mathrm{a}}^{+}$ and 2 critical points in $\mathcal{S}_{\mathrm{a}}^{-}$, with σ_f as local minimum. The values of the parameters are $y = 1$, $x = 1$, $w = 2$, $\alpha = \frac{\sqrt{2}}{2}$, $\beta = 0.1$ (see Figure 2.5)). This case can be easily solved with the general canonical duality framework [6], as the local maximum in $\mathcal{S}_{\mathrm{a}}^{+}$ corresponds to the global minimum of the problem, and the local minimum and maximum in $\mathcal{S}_{\mathrm{a}}^{-}$ correspond to the local minimum and maximum in the primal problem.

Case 2: Five critical points for $P(c)$, six critical points for $P^d(\sigma)$. The values of the parameters are $y = 1$, $x = 4$, $w = 2$, $\alpha = \frac{\sqrt{2}}{2}$. The only parameter that has changed in respect to Case 1 is x. With these parameters the problem becomes multi-welled. By referring to Figure 2.6, the two critical points with the lowest values of the objective function are c_1 and c_3, which belong to the same double well and their corresponding dual critical points are in $\mathcal{S}_{\mathrm{a}}^{+}$. The critical point $c_1 \simeq 0$ has its corresponding critical point σ_1 close to the boundary of $\mathcal{S}_{\mathfrak{b}}^{+}$ which is visible in Figure 2.7 that is an enlargement of Figure 2.6 around the boundary of $\mathcal{S}_{\mathfrak{b}}^{+}$, near $-y$.

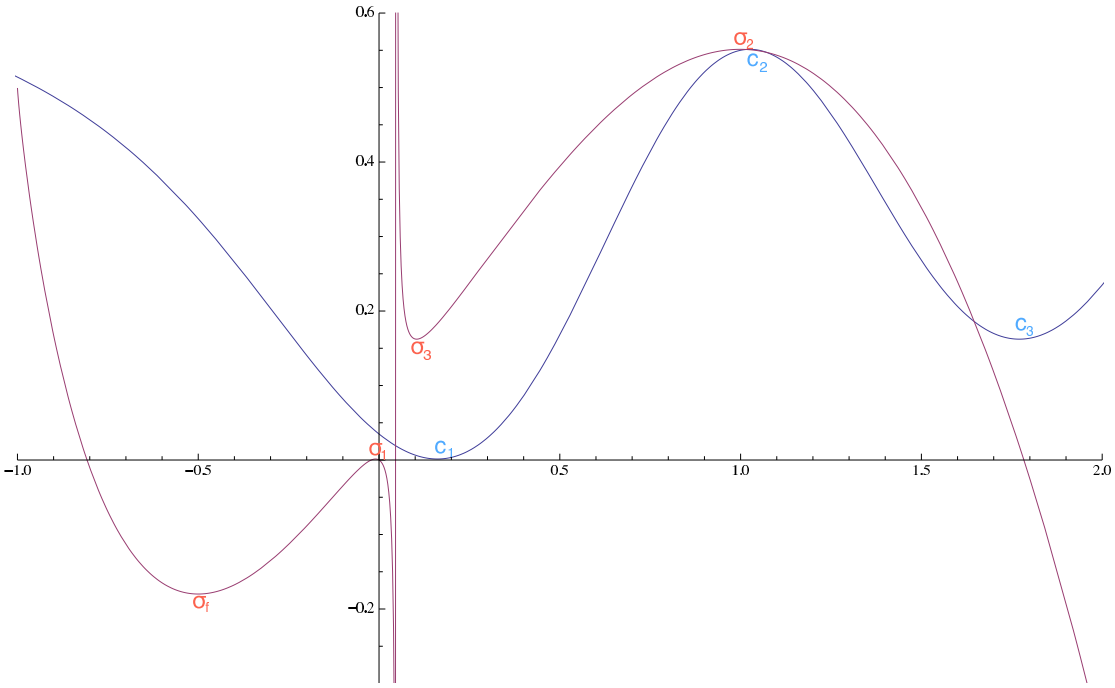

Figure 2.5. Primal(in blue) and dual(in red) functions for Case 1 with 3 critical points

For this case, we discover the following interesting phenomena:

- For small values of β (see Figure 2.6 with $\beta = 0.1$), the global minimizer of $P(c)$ is c_3 which is also a desirable solution to the original problem, and the corresponding dual solution σ_3 is in $\mathcal{S}_\sharp^+$.

- For big values of β (see Figure 2.8 with $\beta = 0.12$), the global minimizer of $P(c)$ is $c_1 \approx 0$, which is not a desired solution to the original problem. The corresponding dual solution σ_1 is in $\mathcal{S}_\flat^+$. In this case, the local minimizer c_3 is the desirable solution with the corresponding $\sigma_3 \in \mathcal{S}_\sharp^+$.

Detailed explanation on this discovery is needed. By the fact that the critical point ($c_1 \simeq 0$) of $P(c)$ is generated by the term $\frac{1}{2}\beta c^2$, which is the regularization used to make the objective function coercive and more regular, we understand that this critical point is not a desired solution to the original problem. The corresponding dual variable $\sigma_1 \in \mathcal{S}_\flat^+$ is always near the boundary, because as c gets close to zero, σ gets close to $-y$. Since $\sigma = w \exp\left\{\frac{(x-c)^2}{2\alpha^2}\right\} - y$ is a measure of the prediction error, the critical point c_1 corresponding to this $\sigma_1 \in \mathcal{S}_\flat^+$ has a high value of error and should not be considered as a good solution.

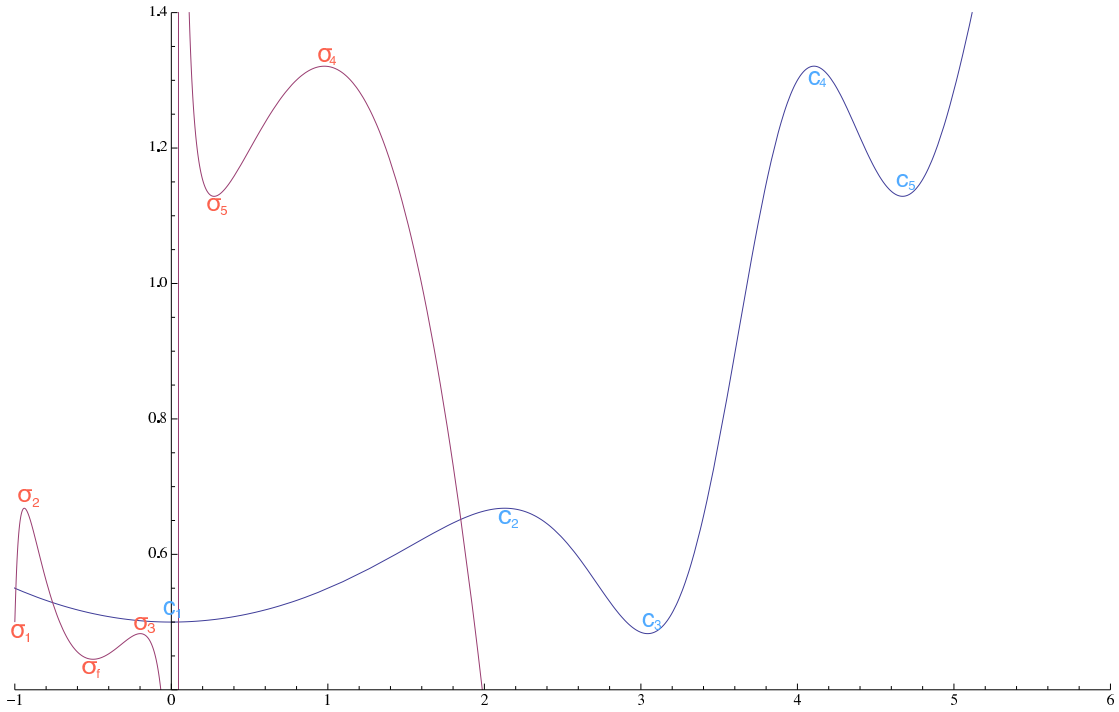

Figure 2.6. Primal(in blue) and dual(in red) functions for case 2 with 5 critical points in the primal and 6 critical points in the dual.

On the other hand, the stationary point $\sigma_3 \in \mathcal{S}_\sharp^+ \subset \mathcal{S}_a^+$ corresponding to a minimum of the primal, has a lower absolute value of σ than the one near the boundary of $\mathcal{S}_\flat^+ \subset \mathcal{S}_a^+$. This means that the solution corresponding to $\sigma_3 \simeq 0$ should have a better final prediction.

From Theorem 19 we know that by having a lower value of β, the pseudo critical point σ_f is usually a local minimum such that the canonical dual feasible set $\mathcal{S}_\sharp^+$ contains a local maximizer of $P^d(\sigma)$, which is corresponding to as a global minimum of the primal problem. However, in high dimensional problems, we don't know the best regulation value β and the solution that corresponding to critical point in $\mathcal{S}_\sharp^+$ could be a local minimum. Therefore, it is possible that the global minimum of the problem may not be the desired solution.

Case 3: Three critical points for $P(c)$ and four critical points for $P^d(\sigma)$, all belonging to $\mathcal{S}_{\mathrm{a}}^+$. The values of the parameters are $y = 1$, $x = 4$, $w = 2$, $\alpha = \frac{\sqrt{2}}{2}$ and $\beta = 0.22$ (see Figure 2.9). This case is similar to the previous one, and the solution of the dual problem should be the critical point that corresponds to a minimum in the primal problem with the value of σ closer to zero.

Case 4: Three critical points in the primal and four critical points in the dual, but with two critical points in $\mathcal{S}_{\mathrm{a}}^+$, two critical points in $\mathcal{S}_{\mathrm{a}}^-$ and σ_f as local maximum.

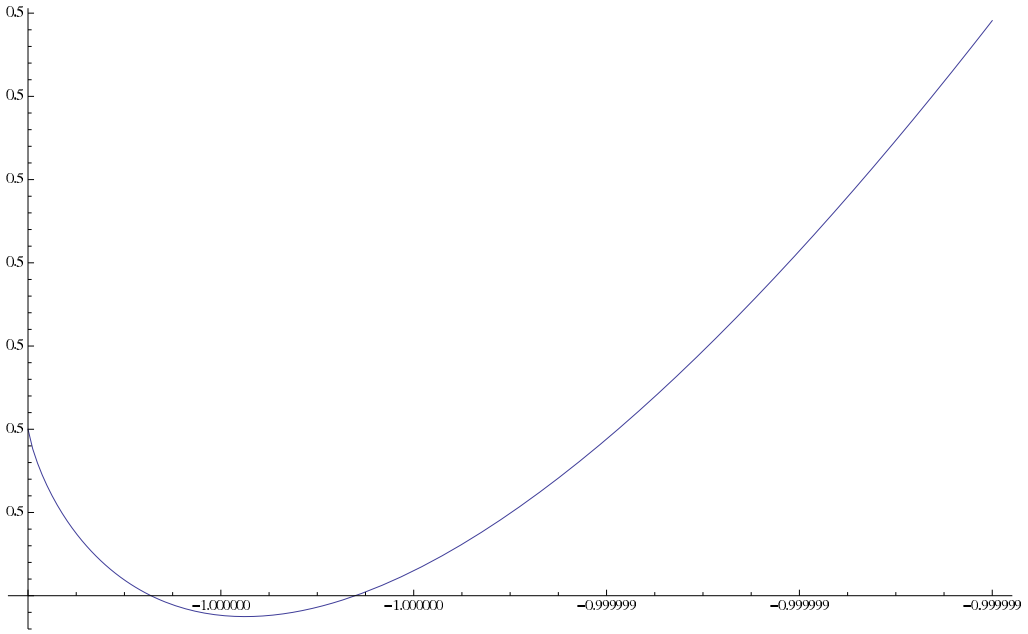

Figure 2.7. Critical point on the boundary of the dual function feasible set for case 2.

The values of the parameters are $y = 1$, $x = 8$, $w = 2$, $\alpha = \frac{\sqrt{2}}{2}$ and $\beta = 0.1$ (see Figure 2.10). If the value of the hyper parameter β is reduced it is possible to make σ_f into a local minimum and return in one of the previous cases.

Case 5: One critical point in the primal problem and two critical points in the dual problem. This case occurs when the quadratic term with beta dominates the error function $W(x)$. If this case occurs, it means that the value of β is too big and the problem is not related with the original anymore, so one should choose a smaller value of β to have a problem related to the original.

Based on the study of these cases, we can obtain the general idea to find the best solution, i. e. the hyper parameter β should be set to a value that satisfies condition (2.75) in order to have σ_f as a local minimum, then search for the critical point in the domain $\mathcal{S}_\sharp^+$. By using condition (2.75) we can impose an upper bound to the value of the parameter β simplifying the issue of the cross validation.

This research leads to the following important results:

1. Global minimum of a nonconvex problem in complex systems may not be the desirable solution to the problem considered.
2. In order to find the optimal solution for the original problem, we should find the critical point of $P^d(\sigma)$ in $\mathcal{S}_\sharp^+ \subset \mathcal{S}_a^+$ even if the corresponding primal solution is not a global minimum.

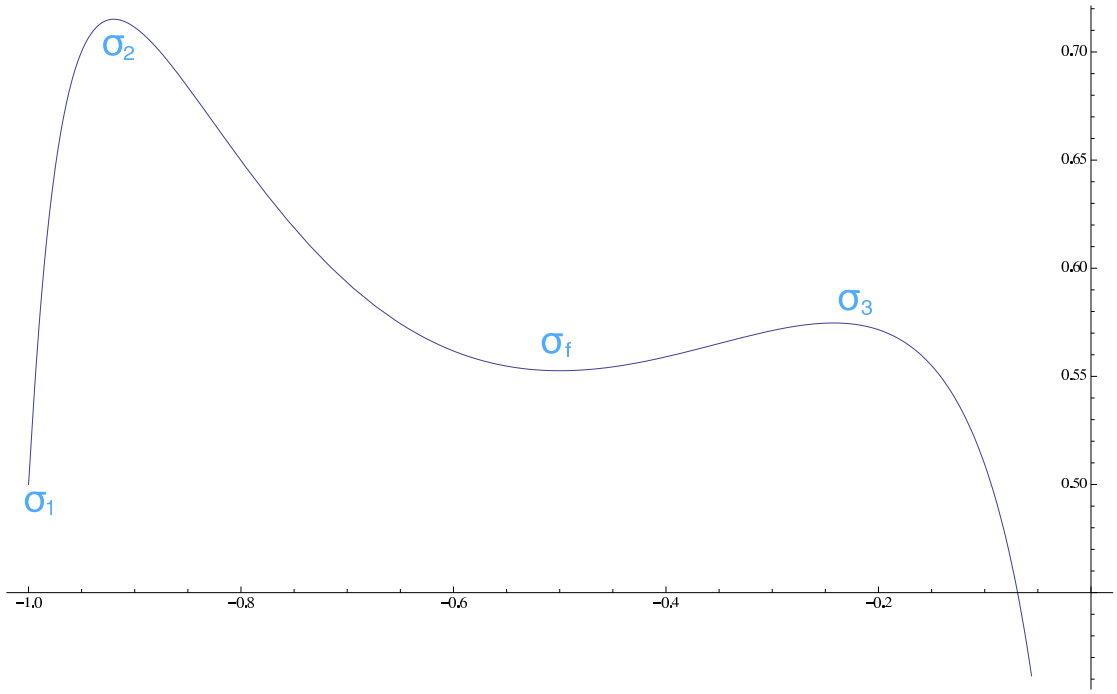

Figure 2.8. $\mathcal{S}_{\mathrm{a}}^{+}$ of the dual problem in the case of $\beta = 0.12$. The minimum near the boundary σ_1 is a global minimum.

2.2.2 Multidimensional Case

After introducing and analyzing throughly the one-dimensional case, we briefly analyze the multidimensional case and give the a complementarity dual principle theorem. Like for the one-dimensional we apply a dual sequential transformation, but this time to vectors, so the analysis is performed once again step by step. As this case is more complex, we analyze the neural network with Gaussian type of radial basis function.

The primal problem is:

$$P(\mathbf{c}) = \frac{1}{2}\sum_{p=1}^{P}\left(\sum_{i=1}^{N} w_i e^{-\frac{\|x_p - \mathbf{c}_i\|^2}{2\alpha^2}} - y_p\right)^2 + \frac{1}{2}\beta\|\mathbf{c}\|^2 - f\|\mathbf{c}\| \tag{2.79}$$

Where $\mathbf{w} \in \mathbb{R}^N$ is the vector of the weights, $\mathbf{y} \in \mathbb{R}^P$ is the vector of the samples output, $\mathbf{c}$ is a matrix in $\mathbb{R}^{n\times N}$ with $\mathbf{c}_i$ as the i-th column of the matrix. We choose the following geometrical nonlinear operator in order to apply the canonical duality theory to this problem:

$$\xi_p = \Lambda_p(\mathbf{c}) = \sum_{i=1}^{N} w_i e^{-\frac{\|x_p - c_i\|^2}{2\alpha^2}} \quad p = 1, \ldots, P \tag{2.80}$$

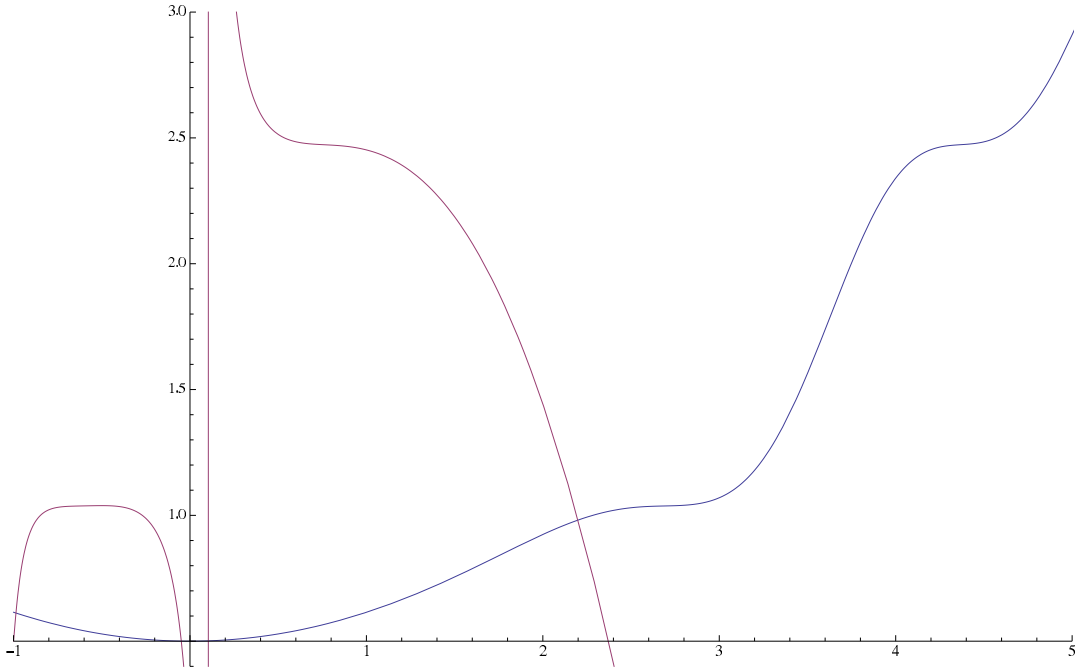

Figure 2.9. Primal(in blue) and dual(in red) functions for the case 3 with three critical points in the primal and 4 critical points in $\mathcal{S}_{\mathrm{a}}^{+}$.

with $\mathbf{\Lambda} : \mathbb{R}^N \to \mathcal{E}_{\mathrm{a}} \in \mathbb{R}^{\mathrm{P}}$. $\mathbf{\Lambda}$ is a nonlinear map from $\mathbb{R}^N$ to a sub space $\mathcal{E}_{\mathrm{a}}$ of $\mathbb{R}^P$. The canonical function $V : \mathcal{E}_{\mathrm{a}} \to \mathbb{R}$ associated with this nonlinear operator is:

$$V(\boldsymbol{\xi}(\mathbf{c})) = \frac{1}{2}\sum_{p=1}^{P}(\xi_p - y_p)^2 \tag{2.81}$$

Where $\boldsymbol{\xi}$ is the vector in $\mathbb{R}^p$ which elements are the $\xi_p = \sum_{i=1}^{N} w_i e^{-\frac{\|x_p - \mathbf{c}_i\|^2}{2\alpha^2}}$. The function $V : \mathcal{E}_{\mathrm{a}} \to \mathbb{R}$ is a canonical dual function on $\mathcal{E}_{\mathrm{a}}$ if for any given $\epsilon \in \mathcal{E}_{\mathrm{a}}$ the following duality relation:

$$\sigma_p = \frac{\partial V(\boldsymbol{\xi})}{\partial \xi_p} = \{\xi_p - y_p\}, for \quad i = 1, \ldots, P. \tag{2.82}$$

is invertible for all $p = 1, \ldots, P$. The variables σ_p are the duality mapping of the problem and they are defined on the range $\mathcal{S}_{\mathrm{a}}$. The couples (ξ_p, σ_p) for $p = 1, \ldots, P$ form a canonical duality pair on $\mathcal{E}_{\mathrm{a}} \times \mathcal{S}_{\mathrm{a}}$ with the Legendre conjugate $\mathbf{V}^*(\sigma)$ defined by

$$V^*(\boldsymbol{\sigma}) = \left\{ \sum_{p=1}^{P} \xi_p \sigma_p - V(\boldsymbol{\xi}) | \sigma_p = \frac{\partial V(\boldsymbol{\xi})}{\partial \xi_p} \ \forall p \right\} = \sum_{p=1}^{P} \left(\frac{1}{2}\sigma_p^2 + y_p \sigma_p \right). \tag{2.83}$$

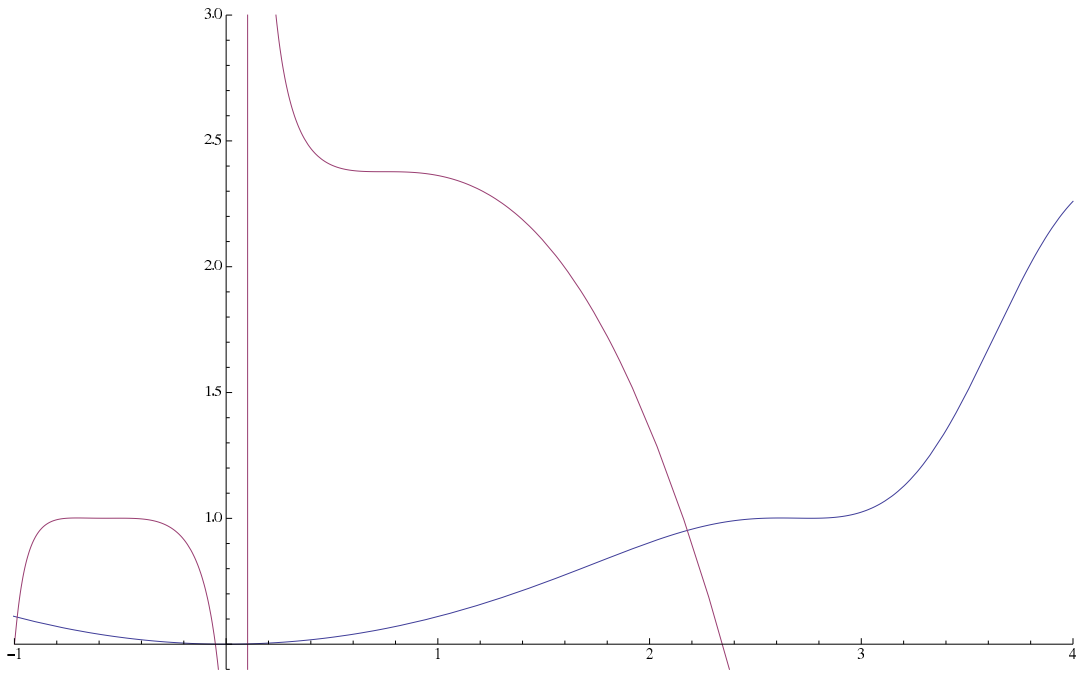

Figure 2.10. Primal(in blue) and dual(in red) functions for the case 4 with 3 critical points in the primal and 2 critical points in $\mathcal{S}_{\mathrm{a}}^{+}$ and 2 critical points in $\mathcal{S}_{\mathrm{a}}^{-}$ and σ_f as a local maximum.

The term $W(\mathbf{c})$ can be replaced by the term $\langle \mathbf{\Lambda}(c), \boldsymbol{\sigma} \rangle - V^*(\boldsymbol{\sigma})$ and the primal function $P(\mathbf{c})$ can be reformulated as the so-called *total complementarity function* defined by:

$$\Xi(\mathbf{c}, \boldsymbol{\sigma}) = \langle \mathbf{\Lambda}(\mathbf{c}), \boldsymbol{\sigma} \rangle - V^*(\boldsymbol{\sigma}) + \frac{1}{2}\beta\|\mathbf{c}\|^2 - f\|\mathbf{c}\| = \tag{2.84}$$

$$\sum_{p=1}^{P} \sigma_p \left(\sum_{i=1}^{N} \left(w_i exp\left\{ -\frac{\|x_p - \mathbf{c}_i\|^2}{2\alpha^2} \right\} \right) - \frac{1}{2}\sigma_p^2 - y_p\sigma_p \right) + \frac{1}{2}\beta\|\mathbf{c}\|^2 - f\|\mathbf{c}\|$$

The radial basis function $exp\left\{ -\frac{\|x_p - c_i\|^2}{2\alpha^2} \right\}$ also creates nonlinearities in the problem. In order to eliminate these nonlinearities, we have to perform a second sequential canonical dual transformation for the nonlinear operator $\mathbf{\Lambda}(\mathbf{c})$. The second level nonlinear operator we choose :

$$\epsilon_{pi}(\mathbf{c}_i) = \|x_p - \mathbf{c}_i\|^2 \tag{2.85}$$

The first level geometrical operator becomes:

$$\Gamma_{pi}(\mathbf{c}) = w_i exp\left\{ -\frac{\epsilon_{pi}}{2\alpha^2} \right\} = U_{pi}(\boldsymbol{\epsilon}) \tag{2.86}$$

with $\mathbf{U} : \mathbb{R}^N \to \mathcal{E}_{\mathrm{b}} \in \mathbb{R}^{\mathrm{P}\times\mathrm{N}}$. In term of ϵ_{pi} the first level operator can be written as

$$\xi_p = \sum_{i=1}^{N} U_{pi}(\boldsymbol{\epsilon}) = \sum_{i=1}^{N} w_i exp\left\{-\frac{\epsilon_{pi}}{2\alpha^2}\right\} \quad p = 1,\ldots,P. \tag{2.87}$$

Like for the first level dual transformation, we assume that the function U is convex in $\boldsymbol{\epsilon}$. With such property the second-level duality relation:

$$\tau_{pi} = \frac{\partial U}{\partial \epsilon_{pi}} = -\frac{w_i}{2\alpha^2} exp\left\{-\frac{\epsilon_{pi}}{2\alpha^2}\right\} \tag{2.88}$$

is invertible for every $p = 1,\ldots,P$ and $i = 1,\ldots,N$, that is the variable $\boldsymbol{\epsilon}$ can be expressed in respect to $\boldsymbol{\tau}$ as

$$\epsilon_{pi} = -2\alpha^2 Log\left(\frac{2\alpha^2\tau_{pi}}{w_i}\right)$$

for every $p = 1,\ldots,P$ and $i = 1,\ldots,N$. The Legendre conjugate is:

$$U^*_{pi} = -2\alpha^2\tau_{pi}\left[\ln\left(\frac{-2\alpha^2\tau_{pi}}{w_i}\right) - 1\right] \quad p = 1,\ldots,P, i = 1,\ldots,N. \tag{2.89}$$

As we have that $U(c) = \sum_{p=1}^{P}\sum_{i=1}^{N}\epsilon_{pi}\tau_{pi} + 2\alpha^2\tau_{pi}\left[\ln\left(\frac{-2\alpha^2\tau_{pi}}{w_i}\right) - 1\right]$, the total complementarity function can be rewritten from this form:

$$\Xi(c,\sigma,\tau) = \langle U(c),\sigma\rangle - V^*(\sigma) + \frac{1}{2}\beta\|c\|^2 - fc \tag{2.90}$$

to the following form:

$$\Xi(c,\sigma,\tau) = \sum_{p=1}^{P}\sigma_p\left(\sum_{i=1}^{N}\left(\|(x_p - c_i\|^2\tau_{pi} + 2\alpha^2\tau_{pi}\left[\ln\left(\frac{-2\alpha^2\tau_{pi}}{w_i}\right) - 1\right]\right) - \frac{1}{2}\sigma_p^2 - y_p\sigma_p\right) \tag{2.91}$$

$$+ \frac{21}{\beta}\|c\|_{2_{\mathrm{pi}}}$$

X

$$F_{ji}(\boldsymbol{\tau},\boldsymbol{\sigma}) = \left(f_{ji} + 2\sum_{p=1}^{P} x_{jp}\tau_{pi}\sigma_p \right) \quad j = 1,\ldots,n \quad i = 1,\ldots,N \tag{2.94}$$

and

$$F_i(\boldsymbol{\tau},\boldsymbol{\sigma}) = \sum_{j=1}^{n} F_{ji} = \sum_{j=1}^{n} \left(f_{ji} + 2\sum_{p=1}^{P} x_{jp}\tau_{pi}\sigma_p \right) \tag{2.95}$$

By reordering the terms in (2.91) according to the values of $G(\boldsymbol{\tau},\boldsymbol{\sigma})$ and $F_{ji}(\boldsymbol{\tau},\boldsymbol{\sigma})$ in (2.93) and (2.94) we obtain

$$\Xi(c,\sigma,\tau) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{n}\left[c_{ji}^2\left(\beta + 2\sum_{p=1}^{P}\tau_{pi}\sigma_p\right) - c_{ji}\left(f_{ji} + 2\sum_{p=1}^{P} x_{jp}\tau_{pi}\sigma_p\right) + \sum_{p=1}^{P} x_{jp}^2\tau_{pi}\sigma_p \right] \tag{2.96}$$
$$-\frac{1}{2}\sum_{p=1}^{P}\sigma_p^2 - \sum_{p=1}^{P} y_p\sigma_p + 2\alpha^2\sum_{p=1}^{P}\sigma_p\sum_{i=1}^{N}\left(ln\left(\frac{-2\alpha^2\tau_{pi}}{w_i}\right) - 1\right)\tau_{pi}$$

by using the first order derivative in c_{ji}, from the 2.96 we have that

$$\frac{\partial\Xi(c,\sigma,\tau)}{\partial c_{ji}} = c_{ji}G_i(\boldsymbol{\tau},\boldsymbol{\sigma}) - F_{ji}(\boldsymbol{\tau},\boldsymbol{\sigma}) \rightarrow c_{ji} = F_{ji}(\boldsymbol{\tau},\boldsymbol{\sigma})G_i^{-1}(\boldsymbol{\tau},\boldsymbol{\sigma}) \tag{2.97}$$

by substituting the value of c_{ji} for $j = 1,\ldots,n$ and $i = 1,\ldots,N$ found in (2.97) we find the dual problem in σ and τ

$$P^d(\sigma,\tau) = -\frac{1}{2}\sum_{i=1}^{N}\left[\frac{\left(\sum_{j=1}^{n}\left(f_{ij} + 2\sum_{p=1}^{P} x_{jp}\tau_{pi}\sigma_p\right)\right)^2}{\beta + 2\sum_{p=1}^{P}\tau_{pi}\sigma_p} + \sum_{j=1}^{n} x_{jp}^2\tau_{pi}\sigma_p \right] - U^*(\tau,\sigma) - V^*(\sigma). \tag{2.98}$$

from equation (2.88) we have that $\xi_p = -\frac{1}{2\alpha^2}\sum_{i=1}^{N}\tau_{pi}$. In this way we obtain that

$$\sigma_p = -\left(y_p + 2\alpha^2\sum_{i=1}^{N}\tau_{pi}\right) \tag{2.99}$$

by using this relation, it is possible to write the dual problem only in respect to the variable $\boldsymbol{\tau}$

$$P^d(\tau) = -\frac{1}{2}\sum_{i=1}^{N}\left[\frac{\left(\sum_{j=1}^{n}\left(f_{ij} - 2\sum_{p=1}^{P} x_{jp}\tau_{pi}\left(y_p + 2\alpha^2\sum_{k=1}^{N}\tau_{pk}\right)\right)\right)^2}{\beta - 2\sum_{p=1}^{P}\tau_{pi}\left(y_p + 2\alpha^2\sum_{k=1}^{N}\tau_{pk}\right)} \right] -$$

$$\sum_{p=1}^{P}\sum_{j=1}^{n} x_{jp}^2\sum_{i=1}^{N}\tau_{pi}\left(y_p + 2\alpha^2\sum_{k=1}^{N}\tau_{pk}\right) - 2\alpha^2\sum_{p=1}^{P}\left(y_p + 2\alpha^2\sum_{k=1}^{N}\tau_{pk}\right)\sum_{i=1}^{N}\tau_{pi}\left(ln\left(\frac{-2\alpha^2\tau_{pi}}{w_i}\right) - 1\right)$$

$$-\frac{1}{2}\sum_{p=1}^{P}\left(y_p+2\alpha^2\sum_{i=1}^{N}\tau_{pi}\right)^2+\sum_{p=1}^{P}y_p\left(y_p+2\alpha^2\sum_{i=1}^{N}\tau_{pi}\right). \tag{2.100}$$

Remark 2.2.6. *In the one dimensional case, the dual was written in respect of the first level dual variable σ. In the multidimensional case such thing is not possible as there are some terms, like for example the logarithms, where it is impossible to substitute τ with σ.*

if we define:

$$\|x_m-c_l(\boldsymbol{\tau})\|^2=\left[\sum_{j=1}^{n}\left(x_{jm}-\frac{f_{jl}-2\sum_{p=1}^{P}x_{jp}\tau_{pl}\left(y_p+2\alpha^2\sum_{k=1}^{N}\tau_{pk}\right)}{\beta-2\sum_{p=1}^{P}\tau_{pl}\left(y_p+2\alpha^2\sum_{k=1}^{N}\tau_{pk}\right)}\right)^2\right] \tag{2.101}$$

First order derivative in τ_{ip} is:

$$\frac{\partial P^d(\boldsymbol{\tau})}{\partial\tau_{ml}}=-\left(y_m+2\alpha^2\sum_{k=1}^{N}\tau_{mk}\right)\left(\|x_m-c_l\|^2+2\alpha^2\ln\left(\frac{-2\alpha^2\tau_{ml}}{w_l}\right)\right) \tag{2.102}$$

$$-\delta_{ll}2\alpha^2\sum_{i=1}^{N}\tau_{mi}\left[\|x_m-c_i\|^2+2\alpha^2\ln\left(\frac{-2\alpha^2\tau_{mi}}{w_i}\right)\right]\quad m=1,\dots,P\quad l=1,\dots,N$$

Where δ_{ll} is defined as

$$\delta_{ij}=\begin{cases}1, & \text{if } i=j\\ 0 & \text{if } i\neq j\end{cases}$$

Now that we have the second order derivatives for the dual, we can prove the complementarity dual principal in the multidimensional case.

Theorem 21. *if $\bar{\boldsymbol{\tau}}$ is a critical point of (P^d) and $y_p+2\alpha^2\sum_{k=1}^{N}\bar{\tau}_{kp}\neq 0$ for all $p=1,\dots,P$, then the point $\bar{\mathbf{c}}\in\mathbb{R}^{n\times N}$ defined as:*

$$\bar{c}_{ji}=\left\{\frac{f_{ji}-2\sum_{p=1}^{P}x_{jp}\bar{\tau}_{ip}\left(y_p+2\alpha^2\sum_{k=1}^{N}\bar{\tau}_{kp}\right)}{\beta-2\sum_{p=1}^{P}\bar{\tau}_{ip}\left(y_p+2\alpha^2\sum_{k=1}^{N}\bar{\tau}_{kp}\right)}\right\}_{ji} \tag{2.103}$$

is a critical point of $P(\mathbf{c})$ and $P(\bar{\mathbf{c}})=P^d(\bar{\boldsymbol{\tau}})$

Proof. From the first order conditions in (2.102) it must be true that the following relation:

$$-\left[\left(y_m+2\alpha^2\sum_{k=1}^{N}\tau_{mk}\right)\|x_m-c_l\|^2+\sum_{i=1}^{N}2\alpha^2\tau_{mi}\|x_m-c_i\|^2\right]= \tag{2.104}$$

$$-2\alpha^2\left[\left(y_m+2\alpha^2\sum_{k=1}^{N}\tau_{mk}\right)\ln\left(\frac{-2\alpha^2\tau_{ml}}{w_l}\right)+2\alpha^2\sum_{i=1}^{N}\tau_{mi}\left(\ln\left(\frac{-2\alpha^2\tau_{mi}}{w_i}\right)\right)\right]$$

must be true for all $m=1,\ldots,P$ and $l=1,\ldots,N$. In other words it must be satisfied the condition:

$$-||x_m-\bar{c}_l||^2-2\alpha^2\ln\left(\frac{-2\alpha^2\tau_{ml}}{w_i}\right)=0\quad m=1,\ldots,P\quad l=1,\ldots,N$$

that is:

$$\bar{\tau}_{ml}=-\frac{w_l}{2\alpha^2}e^{-\frac{||x_m-\bar{c}_l||^2}{2\alpha^2}}\quad m=1,\ldots,P\quad l=1,\ldots,N$$

the first order conditions for the primal problem are:

$$\frac{\partial P(c)}{\partial c_{hl}}=\sum_{p=1}^{P}\left[\frac{x_{hp}-c_{hl}}{\alpha^2}w_l e^{-\frac{||x_p-c_l||^2}{2\alpha^2}}\sum_{i=1}^{N}\left(w_i e^{-\frac{||x_p-c_l||^2}{2\alpha^2}}-y_p\right)\right]+\beta c_{hl}-f_{hl}=0 \quad (2.105)$$

by noticing that $\sum_{k=1}^{N}w_k e^{-\frac{||x_m-c_k||^2}{2\alpha^2}}-y_m=\sigma_m=-\left(y_m+2\alpha^2\sum_{k=1}^{N}\tau_{mk}\right)$ and from the (2.88) we obtain:

$$c_{hl}=\frac{f_{hl}-2\sum_{p=1}^{P}x_{hp}\tau_{pl}\left(y_p+2\alpha^2\sum_{k=1}^{N}\tau_{pk}\right)}{\beta-2\sum_{p=1}^{P}\tau_{pl}\left(y_p+2\alpha^2\sum_{k=1}^{N}\tau_{pk}\right)} \quad (2.106)$$

by substituting the value of the $\bar{\tau}_{ml}$ in the (2.106) we prove that $\bar{c}_{ji}$ for $j=1,\ldots,n$ $i=1,\ldots,N$ is a critical point for the primal problem.

For the functions value equivalence, we start from the dual problem (2.100) in $\bar{\boldsymbol{\tau}}$ perform the due substitutions and obtain:

$$P^d(\bar{\boldsymbol{\tau}})=\left(\sum_{p=1}^{P}\sum_{i=1}^{N}||x_p-\bar{c}_i||^2\bar{\tau}_{ip}-U^*(\bar{\boldsymbol{\tau}})\right)\left(y_p+2\alpha^2\sum_{k=1}^{N}\bar{\tau}_{kp}\right)-V^*(\bar{\boldsymbol{\tau}})+\frac{1}{2}\beta||\bar{\mathbf{c}}||-\sum_{i=1}^{N}\sum_{j=1}^{n}f_{ji}\bar{c}_{ji}$$

by considering that

$$\sum_{p=1}^{P}\sum_{i=1}^{N}||x_p-\bar{c}_i||^2\bar{\tau}_{ip}-U^*(\bar{\boldsymbol{\tau}})=\sum_{p=1}^{P}\sum_{i=1}^{N}w_i e^{-\frac{||x_p-\bar{c}_i||^2}{2\alpha^2}}$$

and that

$$\left(\sum_{p=1}^{P}\sum_{i=1}^{N}w_i e^{-\frac{||x_p-\bar{c}_i||^2}{2\alpha^2}}\right)\left(y_p+2\alpha^2\sum_{k=1}^{N}\bar{\tau}_{kp}\right)-V^*(\bar{\boldsymbol{\tau}})=\sum_{p=1}^{P}\left(\sum_{i=1}^{N}w_i e^{-\frac{||x_p-\bar{c}_i||^2}{2\alpha^2}}-y_p\right)$$

we obtain that:

$$P^d(\bar{\boldsymbol{\tau}})=\sum_{p=1}^{P}\left(\sum_{i=1}^{N}w_i e^{-\frac{||x_p-\bar{c}_i||^2}{2\alpha^2}}-y_p\right)+\frac{1}{2}\beta||\bar{\mathbf{c}}||-\sum_{i=1}^{N}\sum_{j=1}^{n}f_{ji}\bar{c}_{ji}=P(\bar{\mathbf{c}})$$

that proves the theorem. □

Remark 2.2.7. *If* $y_p + 2\alpha^2 \sum_{k=1}^{N} \bar{\tau}_{kp} = 0$ *for at least one* p *then we are in a pseudo critical point, and as we have seen in Theorem 17, this point does not have any corresponding point in the primal problem.*

In conclusion, this application of canonical duality theory to radial basis neural networks shows the potentially of this theory to simplify the non-convex primal problem into a simpler problem. The multidimensional dual formulation given by (2.100) is still under study as it presents even more issues than the one-dimensional case, but solving these issues could bring a good contribute to the neural network research community.

Chapter 3

Applications

In this chapter we show the different applications on which the models and tools presented in the previous chapters are utilized. Most of these problems are direct applications of surrogate models to the regression of functions in some practical context, while other application heavily integrate the black box methods and the surrogate models in order to solve complex optimization problems.

3.1 Black Box Algorithm for Cross Validation

Like we said in Section 1.5, cross validation is an important phase to determine the best hyper parameters of the model in order to get the best prediction as possible. Also, always in Section 1.5, we reported one of the most simple and most used strategy for cross validation, that is the grid search. As we already said, this simple procedure can be too time demanding if the grid is to dense or not accurate enough if the grid is too sparse.

To avoid the fore mentioned shortcomings, in literature the grid search has been replaced by different optimization techniques. Most of such techniques can be divided into two classes. In the first class, gradient based methods are used to determine the model parameters that minimize a continuously differentiable estimate of a generalization error such as leave one out error or k-fold cross validation error. These methods have a fast convergence but suffer from the fact that they are local methods that can be stuck into a stationary point of the problem and from the fact that the goodness of the obtained points is affected by the error of the estimates used as objective functions.

The second class of methods are based on global stochastic optimization techniques, such as genetics algorithms , simulated annealing or swarm optimization. These methods have a high probability of finding a good approximation of the global minimum of a problem, but at the price of using a large number of function

evaluations. Therefore when the number of function evaluation is limited by a practical stopping criterion (as in the case of determining the model parameter of SVM), the obtained solution can be far from the global optimum and, often, from a local minimum point.

In this research we follow the approach proposed in [35] and test the SVM surrogate model. The main features of this approach are:

- The objective function to be minimized is a generalization error of the SVM and not an approximation; therefore the quality of the obtained solutions does not depend on the error of the approximation function.
- The use of derivative free methods for minimizing the black box objective function consisting in generalization error of the SVM; this class of methods have the advantages over global algorithms to get interesting points after a few number of function evaluations; compared to gradient based algorithms the derivative free methods are less attracted by stationary points.

In [35] the proposed method was tested on a few data sets of small and medium size. The obtained results showed that the proposed approach seems to be interesting.

The aim of this research is to confirm the interest of using a derivative free strategy in choosing the model parameters of a SVM. In particular instead of using a pattern search approach (as in [35]), we use a line search derivative free method for boxed constrained problems proposed in [23] and reported in Chapter 2. The motivation of this choice follows from the fact that the line search approach seems to be efficient in saving the number of function evaluation(see for example the numerical results reported in [36]). In order to evaluate better the possible potentialities of the derivative free approach, we perform a wider numerical experimentation both on classification problems and regression problems.

3.1.1 Black Box Optimization Problem

Given a particular choice of the model parameters, the prediction capability of the SVM is validated on a new set of samples, different from those in the training set. This set is denoted by $V := \{(\hat{x}_i,\hat{y}_i),\hat{x}_i \in \mathbb{R}^n,\hat{y}_i \in H$, for $i = 1,\ldots,mv\}$ with $H = \{-1,1\}$ for the classification problems, and $H = \mathbb{R}$ for the regression problems. The values of these errors heavily depend on the choice of the model parameters. Therefore all the SVM training procedures can be considered as a black box function that, given a set of values of the model parameters, returns the error on the validation set of the trained SVM. However the values of this black box function heavily depend on the training set S and validation set V. In order to have a black box less depending on the particular choice of the sets S and V

the k-fold cross validation is usually used. We described the k-fold cross validation strategy in Section 1.5. After this procedure, it is possible to evaluate the goodness of the particular choice for the model parameters by computing the mean of the k different MSE. In our experimentation we choose $k = 10$ as usually done in most application. All this procedure can be considered a reliable black box function $\phi(\cdot)$ whose minimization should allow to determine good values of the model parameters. In particular in the following we consider two black box optimization problems.

2 variables minimization (DF 2)

In this case we consider the model parameters C and σ. In particular we use the black box function described before to define the following box constrained problem:

$$\begin{aligned} &\min \phi(C,\sigma) \\ &\quad l_C \leq C \leq u_C \\ &\quad l_\sigma \leq \sigma \leq u_\sigma. \end{aligned} \tag{3.1}$$

As regards the bounds on the variables of the black box function and the starting point for the derivative free algorithm, we followed the choices suggested in [28]. In particular the pair (C,σ) were constrained by the following box constrains:

$$\begin{aligned} l_C &= 2^{-5} \qquad & u_C &= 2^{15} \\ l_\sigma &= 2^{-15} \qquad & u_\sigma &= 2^{3}. \end{aligned}$$

and the starting point (C_0,σ_0) suggested is:

$$C_0 = 1 \qquad \sigma_0 = 1/num\_features = 1/12$$

3 variables minimization (DF 3)

In the regression problem, in order to have an higher level of precision, we considered as variable also the parameter ϵ. The derivative free algorithm for regression minimizes the black box function $\phi(C,\sigma,\epsilon)$ in the box constrained optimization problem:

$$\begin{aligned} &\min \phi(C,\sigma,\epsilon) \\ &\quad l_C \leq C \leq u_C \\ &\quad l_\sigma \leq \sigma \leq u_\sigma \\ &\quad l_\epsilon \leq \epsilon \leq u_\epsilon. \end{aligned} \tag{3.2}$$

For the lower and the upper bound values we used for C and σ the same values of the 2 variables optimization, while for ϵ we decided to set:

$$l_\epsilon = 10^{-5} \qquad u_\epsilon = 10^5.$$

For the starting point, the values of C and σ are the same of the 2 variables optimization, while for the initial value of ϵ we refereed to ([28]) setting $\epsilon_0 = 0.1$

We solved Problems (3.1), (3.2) by using the derivative-free algorithm for bound constrained optimization described in Chapter 2.

3.1.2 results

In this section we report the numerical results obtained by the methods described in Section 3.1.1 on several datasets taken from the literature. In particular the approach that optimizes two parameters was tested on 10 problems of classification and 21 problems of regression. The approach that optimizes three parameters was used only on the 21 regression problems.

Before the training, the samples in the data are randomly shuffled and normalized according to the mean and the variance of the samples in the training set.

After choosing the SVM parameters according to the $k - fold$ validation error, the reported results will be on a separate test set not utilized for the cross validation.

The aim of this numerical experimentation is twofold:

- First to understand if it is worthwhile to use an optimization procedure to compute the value of the parameter of the SVM instead of using values dictated from the experience.
- Second to understand the different features of the analyzed methods.

Classification

In the first part of our numerical experience we have considered a set classification problems. In Table 3.1 we report:

- in the first column the name of the dataset;
- in the second column the number of features for that dataset;
- in the third and fourth columns the number of instances used for cross validation and test phases.

In our experimentation we use the percentage of success for the classification, namely:

$$E_c = m_s/m_t$$

where m_s is the number of success and m_t is the total number of samples in the test set.

Dataset	Features	Train	Test
A1A	123	18500	12456
A2A	123	18100	12196
A3A	123	17600	11776
Clean	166	4200	2874
Faults	32	1100	841
Magic	10	11400	7620
Mush	112	4800	3324
RNA	8	357	23835
Spam	57	2700	1901
Svmguide	4	4200	2889

Table 3.1. Division of the classification data sets: 60% for cross validation and 40% for test

For the partition of the data in cross validation and the test set, we decided to divide them with a ration of 60% for the $k - fold$ and the remaining 40% for the test set. The results obtained in our experimentation are summarized in Table 3.2,

Dataset	Error D.	Error GS	Error DF 2	T G-S	T DF	G-S/DF
A1A	0.841	0.845	0.845	424307	29073	0.07
A2A	0.843	0.846	0.847	410245	31279	0.8
A3A	0.846	0.849	0.848	402262	29825	0.07
Clean	0.965	0.995	0.992	19414	2563	0.13
Faults	0.864	0.881	0.843	1450	350	0.24
Magic	0.871	0.875	0.875	189589	4218	0.02
Mush	0.999	1	0.999	18316	2239	0.12
RNA	0.887	0.904	0.955	7566	13262	1.75
Spam	0.921	0.932	0.926	4489	567	0.13
Svmguide	0.971	0.974	0.97	1695	184	0.11

Table 3.2. Results for the classification datasets

where we have reported:

- in the first column the name of the sets;
- in the second column the percentage of success on the test set obtained by training the SVM on the training set by the default settings:

$$C = 1, \qquad \sigma = \frac{1}{nfeatures}; \tag{3.3}$$

- In the third column the test error obtained by the SVM were the parameters C and σ were obtained by the procedure of the Grid Search;
- In the fourth column the test error obtained by the SVM where the parameters C and σ were obtained by the procedure of the black box optimization that optimizes two parameters;
- in the fifth column the total time in seconds needed by the grid search for analyzing the whole grid.
- in the sixth column the total time in seconds needed for the black box optimization to determine the optimal parameters;
- in the seventh column the ratio between the time needed for the grid search method and the derivative free method.

First we notice that even in the default setting the precision of the SVM is quite high. For this reason the two cross validation strategies in the majority of cases are able to improve the prediction of just a small ratio in comparison to the original solution. The only two cases were the cross validation is able to create models with a largely better prediction is for the datasets "Clean" and "RNA". These results indicate that in general for classification it is always a good practice to apply cross validation strategies in order to improve the prediction.

For the comparison of the two strategies it is possible to see that they are able to yield similar results, with the grid search being sightly better. The biggest difference is in the comparison of the time needed to get the solution. Just in one case, "RNA", the DF algorithm takes more time than the Grid Search to find a solution, that in this case is quite better than the other method. For the remaining cases, the DF method takes from a fourth to less than 1/50 times than the Grid Search method to find a solution.

Regression

The second part of our numerical experimentation is centred on regression problems. In Table 3.3 we report the same kind of data reported in Table 3.1 but for the regression set. The tests this time were performed on 21 datasets. In Table 3.4 we report the result in the following fashion:

- In the first column the name of the datasets;
- in the second, third, fourth and fifth columns the results for the initial point of the minimization, the Grid Search, DF2 and DF3 methods respectively. The

Dataset	Features	Train	Test
Abalone	10	2500	1677
Bank	32	4900	3292
Boston	13	300	206
Cadata	8	12400	8240
Cal-h	8	12200	8260
Cpu-a	21	4900	3292
Cpu	12	4900	3292
Dati	12	4950	3298
Delta-a5	5	4300	2829
Delta-e6	6	4300	2829
Elevators	18	10000	6599
F16	40	8250	5500
Fried	10	24460	16308
House	16	13670	9114
Kinematic	8	4890	3263
MG	6	830	555
Pol	48	9000	6000
Puma8	8	4900	3292
Puma32	32	4900	3200
Space	6	1860	1247
Wisconsin	32	110	84

Table 3.3. Division of the regression data sets: 60% for cross validation and 40% for test

performances are measured with the MSE error:

$$MSE := \frac{1}{P}\sum_{p=1}^{P}(f_p - y_p)^2,$$

Where P is the number of samples in input, f_p are the results of the surrogate models, and y_p are the real outputs;

- in the sixth, seventh and eighth columns the time needed for the three methods to find the solution

- in the ninth column the ratio between the time needed for the grid search method and the DF2 method, while in the 10-th column the ratio between the time needed for the grid search method and the DF3 method.

Dataset	Mse D.	Mse G-S	Mse DF2	Mse DF3	T G-S	T DF2	T DF3	G-S/DF2	G-S/DF3
Abalone	2.65E-01	2.72E-01	2.67E-01	2.93E-01	9511	814	604	0.09	0.06
Bank	5.01E-01	4.71E-01	4.72E-01	4.66E-01	26626	3494	5129	0.13	0.19
Boston	1.97E-01	1.62E-01	1.43E-01	1.43E-01	769	243	387	0.32	0.50
Cadata	2.60E-01	2.45E-01	2.47E-01	2.46E-01	423723	25072	25681	0.06	0.06
Cal-h	3.42E-01	2.45E-01	3.50E-01	3.43E-01	379495	5006	7871	0.01	0.02
Cpu-a	6.50E-02	3.45E-01	2.83E-02	3.02E-02	20592	9563	14384	0.46	0.70
Cpu	5.03E-02	2.76E-02	2.82E-02	2.82E-02	25603	6990	13899	0.27	0.54
Dati	2.58E-01	1.71E-01	1.83E-01	1.83E-01	30149	10510	15287	0.35	0.51
Delta-a5	1.11E-03	3.84E-04	4.04E-04	2.61E-04	1121	104	2340	0.09	2.09
Delta-e6	2.17E-02	2.14E-02	2.13E-02	2.13E-02	36287	640	824	0.02	0.02
Elevators	1.49E-01	6.17E-02	6.37E-02	6.41E-02	257862	122193	114397	0.47	0.44
F16	2.96E-03	1.03E-03	9.93E-04	2.39E-04	5162	1001	11966	0.19	2.32
Fried	4.66E-02	4.35E-02	4.31E-02	4.31E-02	45 days	6 days	8 days	0.13	0.18
House	4.94E-01	4.00E-01	4.09E-01	4.10E-01	471245	90555	157347	0.19	0.33
Kinematic	1.07E-01	8.17E-02	7.96E-02	7.96E-02	39955	8747	11978	0.22	0.30
MG	2.63E-01	2.74E-01	2.63E-01	3.00E-01	1216	255	448	0.21	0.37
Pol	1.53E-01	3.82E-02	3.91E-02	3.63E-02	120741	40592	120701	0.34	1.00
Puma8	3.33E-01	3.28E-01	3.31E-01	3.28E-01	34927	2676	4149	0.08	0.12
Puma32	8.10E-01	6.71E-01	7.17E-01	7.12E-01	20817	10243	13630	0.49	0.65
Space	3.66E-01	3.58E-01	3.29E-01	2.91E-01	5475	1560	2321	0.28	0.42
Wisconsin	1.20E+00	1.21E+00	1.31E+00	1.35E+00	527	208	215	0.39	0.41

Table 3.4. Division of the classification data stress test

From these results, we see that three times on 21 the G-S method is not able to improve from the default point, while the DF2 and DF3 methods are not able to improve the results on four datasets on 21, three of these datasets are shared among the three methods. In other words these datasets seem to have some particular issues even if the data samples are randomly shuffled. In any case, when the cross validation method improves the prediction, this improvement is substantial. For the results the G-S method yields the best results on nine datasets while gives equal results to DF3 on one dataset. On a close majority of the datasets the derivative free methods are able to give better results than the grid search method, but in general the two strategies seem to be equivalent. It is important to notice that even if in the DF3 strategy the ϵ hyper parameter is optimized together with C and σ, this method is not able to always give better results than DF2. In fact the derivative free optimization that optimizes just two hyper parameters sometimes is superior. From the point of view of the results DF2 and DF3 can be considered on an equal standing.

For the time needed to find the solution, the same behavior for the analysis performed on the classification datasets can be observed. The derivative free methods are able to obtain comparable results with the G-S method even in a 1/50 of the time. The DF3 strategy, because it has more variables to optimize, takes more time than the DF2 to converge to a solution.

As conclusion, we can say that applying a cross validation strategy is in the majority of cases a convenient choice to improve the results. From our analyses the reported methods are able to yield very similar results, but the derivative free methods have the upper hand from the point of view of the time needed to find a solution. Also the results seems to indicate that not always it is a good choice to take more computational time to optimize all the three hyper parameters.

3.2 Ozone Forecasting

Air pollution is one of the principal problems that affect high-density urban areas such as Rome. One of the most important background pollutant in cities characterized by Mediterranean climate is the ozone. Ozone is a secondary pollutant, which levels are influenced by the chemical reaction from pollutants directly emitted from anthropogenic sources, usually defined as primary pollutants. A survey of the complex photochemical reactions chains can be found in different works ([38], [39]). A high concentration of Ozone usually causes respiratory problems and the World Health Organization declared ozone as a pollutant that can cause relevant effects on human health [40]. In this context, models to forecast short-term Ozone levels can be used to plan a health warning system. Enlarging the period prediction allows the authorities to adopt better measures to help the population. Therefore an increment of the forecast period is an important research argument. Excluding the case of ideal atmosphere composition, chemical reactions cannot be described by simple models because they are characterized by complex and nonlinear relations among pollutants and by some typical feedback behaviours of the NO2 and NO. These reactions depend directly or indirectly by measurable variables like:

- The ones linked to long range transport of ozone;
- The incoming solar radiation;
- Turbulence conditions;
- The main compounds involving the reactions.

Therefore the ozone photochemical production can be described by high non-linear relations and this justifies the use of non-linear mathematical models to simulate all the phenomena connected to the Ozone production. Typical linear models, such as

the statistical regression methods, are often used to approximate linear relations between variables and, therefore, they can be unsuitable to tackle such high non-linear phenomena. Instead SVM are more suitable to approximate non-linear relations and, hence, they should work better in respect to classical regression models to forecast the Ozone levels. In literature several works successfully used Learning machines in the field of ozone forecasting. For example [41] is centered on comparisons of results between learning machines and regression models in forecasting daily ozone pollution. Other authors [42] use Neural Networks (NN) for forecasting daily peaks of several pollutants in a forecast period of 48 hours, ozone included, while [43] makes a 24-hour prediction on daily ozone peaks. Only few papers have considered the use of SVM in pollutant forecasting [44], despite their interesting theoretical properties. In this paper we have considered the data collected in the urban area of Rome in the calendar year 2005. Such Data regard pollutants and meteorological variables. Our main aims are to forecast hourly ozone levels in the short term (24/48 hours) as best as possible and to get reasonable forecasts in the medium term (10 days). Our approach is based on:

- The use of accurate optimization techniques both for training the learning machines and for determining the parameters that define the SVM.
- A complete analysis of the input variables in order to extend the prediction period.

3.2.1 Considered problem and data

As we said in the introduction, ozone is a secondary pollutant. Then a suitable forecasting technique must convey enough information on the complex chemical reactions that happen in the atmosphere. A possible procedure is to use a learning machine that extracts the needed information from a suitable data set. In this work we use the data set obtained by a monitoring station of the ARPA LAZIO (Regional Agency for Environmental Protection in Lazio) network in the urban centre of Rome (the station of Largo Magna Grecia), which recorded hourly data throughout the calendar year 2005. The data consists of both pollutant and meteorological information. The pollutant data are: Carbon monoxide, Nitrogen oxide, Nitrogen dioxide and Ozone. The Meteorological data are: Air Temperature, Global Solar Radiation, Relative Humidity and Pressure. These data are widely considered the most significant in ozone prediction by classical deterministic models [45] [46] [47]. A learning machine approach has the advantage of easily consider new type of information and to evaluate if they are significant or not. As matter of facts the complex chain of reactions that determines the ozone levels in the atmosphere is influenced by phenomena not strictly connected to pollutant or meteorological information.

Examples of such phenomena are anthropogenic emission or seasonality. So we conjecture that an improvement of ozone level prediction can be obtained by exploiting information about these phenomena. For this reason we consider exogenous data and for our experimentation we add four additional integer type of data to the meteorological and pollutant ones. Because of these considerations we utilize for our learning procedures a model with pollutant, meteorological and exogenous variables for a total of 12 variables in input. These input data are used to train different learning machines to predict ozone levels with time lags increasing from 1 day to 10 days after the moment the samples in input are measured.

3.2.2 Prediction of the ozone pollutant

In this section we investigate the use of SVM as a tool to forecast ozone pollutant and to point out hidden relations between ozone levels and exogenous input variables at lag 0. Our analysis begins with investigating the data set at 24 hours and continues analysing the data sets with enlarged prediction period. We do not only use the MSE error to evaluate the goodness of the results, but also the R^2 error, that is

$$R^2 := \frac{\sigma_{f,y}^2}{\sigma_f * \sigma_y},$$

where $\sigma_{f,y}$ is the covariance between the calculated output f and the real output y, σ_f and σ_y are the variances of the calculated output and the real output respectively. This coefficient varies in the interval [0,1] with 1 being the best possible value achievable, it does not depend on the scale of the output and it indicates how much well the models fit the original data.

Prediction at 24h

In this subsection we use the SVM to forecast the ozone levels in the short term (24h). First we consider as input parameters of the SVM variables in Table 3.5 that are the pollutant and meteorological variables usually used in literature. After utilizing the cross validation procedure DF2 the SVM yields the following results:

$$MSE = 0.237 \qquad R^2 = 0.768$$

In order to have an idea of the effectiveness of the SVM in this particular class of problems we compare its prediction with the one obtained with a standard statistical technique. In particular we use the multiple regression [48] on the same data set. We choose as the set of variables in input of the Multilevel Regression (MR) the same set of input variables used for the SVM. The MR gives the following results:

$$MSE = 0.327 \qquad R^2 = 0.667$$

CO	Carbon monoxide
NO	Nitrogen Oxide
NO2	Nitrogen Dioxide
O3	Ozone
T	Temperature
SolRad	Solar Radiation
RHm	Relative Humidity
P	Pressure

Table 3.5. the eight standard variables

Comparing these results, we note that SVM prediction is more accurate. This is probably due to the high non-linearity that characterizes the relations between the input variables and the output. However these eight standard variables probably do not represent all the possible variables that can influence the complex chemical and physical phenomena in the atmosphere. For this reason we try to identify some other variables that could influence directly or indirectly these phenomena. Since we do not have more information besides the ones in the data set, we choose four additional variables, which can be easily obtained, because they are recorded together with the measures. These variables are reported in Table 3.6. With these 12 variables the

h	hour of the day
DW	Day of Week
DM	Day of Month
M	Month

Table 3.6. the four additional variables

SVM gives the following results:

$$MSE = 0.165 \qquad R^2 = 0.828$$

We note that adding these four additional variables to the input gives a better prediction. In fact both the MSE and the R^2 improve significantly. This seems to indicate that these four additional variables are strictly connected with the average turbulence conditions during the ozone detection. The multiple regression with the set of 12 inputs variables gives the following results:

$$MSE = 0.324 \qquad R^2 = 0.674$$

By comparing the results obtained by the MR using eight and 12 parameters in input, we note that the input parameter set with the additional variables does not

produce significant improvements. Comparing the effects of the new four inputs on the prediction of the SVM and the multiple regression, we note that, unlike the SVM, the multiple regression is not able to reproduce the further information given by these additional variables. The previous results seem to point out that SVM are capable to represent better the possible non linear relations between these four new variables and the ozone levels.

Enlarging the Period of Prediction

In this subsection we enlarge the prediction period. As we said in the introduction, this is an important challenge in the field of pollution forecasting. In fact the standard methods are not reliable in forecasting beyond 24/48 hours. Similarly to the case of the 24h prediction, we start by using the standard eight meteorological variables and we predict the ozone values with increasing forecasting periods. In particular our period of prediction goes from the standard 24h (1 day) to 240h period (10 days). In table 3.7 we reported the MSE and the R^2 obtained by the SVM and the multiple regression. We note that the results obtained by the SVM

Days	MSE SVM 8i	MSE MR 8i	R^2 SVM 8i	R^2 MR 8i
1	0.237	0.327	0.768	0.667
2	0.25	0.439	0.749	0.548
3	0.31	0.472	0.691	0.504
4	0.297	0.497	0.708	0.496
5	0.271	0.5	0.726	0.503
6	0.264	0.503	0.735	0.49
7	0.274	0.513	0.726	0.494
8	0.266	0.526	0.737	0.475
9	0.273	0.553	0.73	0.462
10	0.274	0.539	0.724	0.442

Table 3.7. Behaviour of the MSE error and R^2 by using eight input variables

outperform the ones obtained by the multiple regression, both in term of MSE and R^2. As regards the behaviour of the prediction error of the SVM (see Figure 3.1) we note that the MSE error worsens from the first day (24h) to second day (48h), it also gets worse from the second day to the third day, and after the third day it becomes almost constant around the value 0.26/0.27. The value of the MSE obtained by the SVM in the period going from the third day to tenth day, is better than the value of the error obtained by MR at 24h (1 day).

For the MSE of the MR (see Figure 3.1) there is a strong worsening from the first day to the third day and after that day, the error steadily continues to get worse.

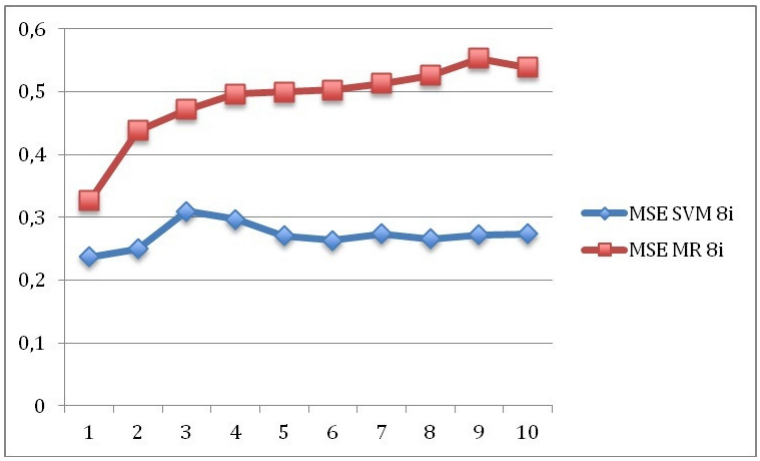

Figure 3.1. Behaviour of the MSE error in SVM and MR with eight inputs.

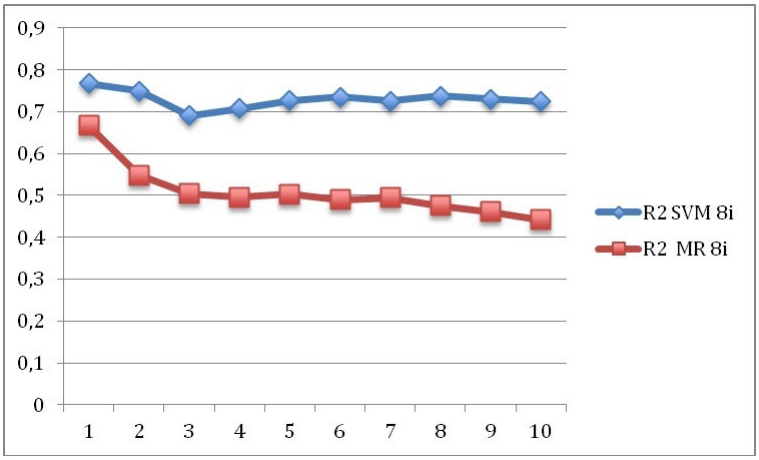

Figure 3.2. Behaviour of the R^2 error in SVM and MR with eight inputs.

As regards the R^2 error (see Figure 3.2) it has a symmetric behaviour in respect to the one of MSE. In particular for SVM we have a worsening of the R^2 from the first day to the third day and after the third day it becomes constant around 0.73, that is better than the best value of the MR in the first day. The R^2 error of the MR (see Figure 3.2) after a strong worsening from the first day to the third day, it continues to get worse in the following days. As second step in our attempt to obtain good predictions for longer periods, we increase the number of the inputs variables. We add to the eight standard meteorological variables (described in Table 3.5) the four additional variables concerning the human activities and seasonal variability

(described in Table 3.6). As regards the multiple regression results, we note that

Days	MSE SVM 12i	MSE MR 12i	R^2 SVM 12i	R^2 MR 12i
1	0.165	0.324	0.828	0.674
2	0.18	0.437	0.818	0.555
3	0.185	0.466	0.818	0.517
4	0.179	0.489	0.82	0.51
5	0.181	0.487	0.821	0.522
6	0.167	0.491	0.833	0.509
7	0.167	0.494	0.835	0.52
8	0.172	0.513	0.827	0.495
9	0.178	0.533	0.821	0.488
10	0.175	0.528	0.825	0.461

Table 3.8. Behaviour of the MSE error and R^2 by using 12 input variables

there are no significant differences between the results reported in Table 3.7 and in Table 3.8. This shows again that the MR seems to not be able to exploit the information contained in the four additional input variables.

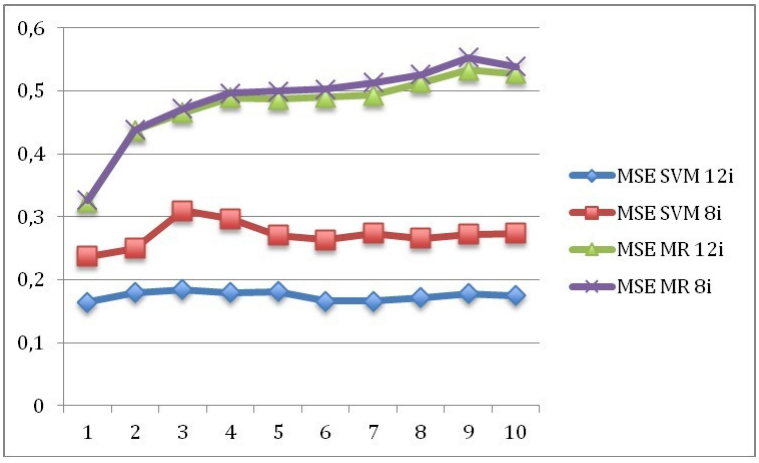

Figure 3.3. Behaviour of the MSE error in SVM and MR with 12 inputs.

In Figure 3.3 and Figure 3.4 we report the values of the MSE and R^2 errors at different days in the cases of SVM and MR with eight and 12 inputs. From Figure 3.3 is evident that the behaviours of the MSE for the MR eight inputs and MR 12 inputs are almost the same. While the behaviour of the MSE for the SVM eight inputs and SVM 12 inputs is different and the improvement of the error is conspicuous. Same

behaviour can be seen for the R^2 in Figure 3.3. In order to better understand the

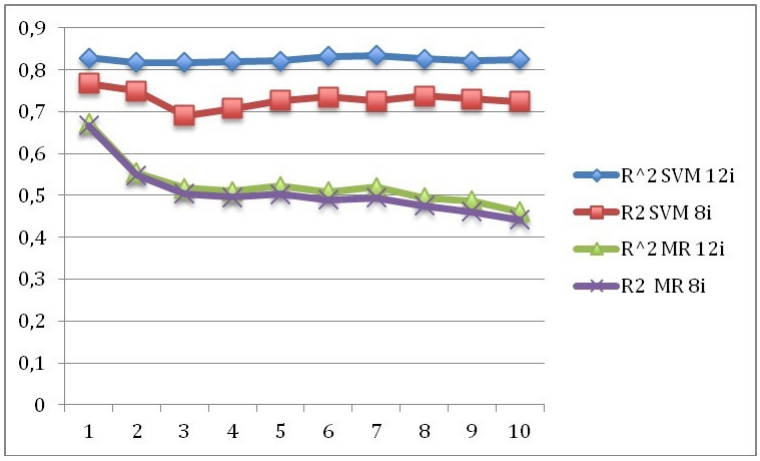

Figure 3.4. Behaviour of the R^2 error in SVM and MR with 12 inputs.

role of the four additional variables described in Table 3.6 in the prediction of ozone pollutant levels, we report in Table 3.9 the R^2 of the SVM model trained by using as input only the four additional variables with the same DF2 strategy. We see in

Days	R^2	Days	R^2
1	0.7293	6	0.7184
2	0.741	7	0.7281
3	0.7398	8	0.7234
4	0.7361	9	0.7378
5	0.7118	10	0.7303

Table 3.9. Behaviour of the MSE error and R^2 by using the four additional variables

Table 3.9 that the the R^2 remains in an interval between 0.7118 and 0.7410. The R^2 begins with a relative good value and does not deteriorate itself from the first day to the tenth day, and sometimes it even gets better. Comparing the results of the R^2 error in Table 3.9 with those in Table 3.7 and Table 3.8, we note that using the set of eight inputs variables (see Table 3.5) in the first day of prediction the R^2 error is better than the one obtained in Table 3.9, while the R^2 from the second day of prediction to the tenth day are comparable. This indicates that, as known, the meteorological and pollutant variables have a significant role for the forecast in the short term and their effect is reduced with the enlargement of the forecasting period. These different behaviors obtained with eight variables and four variables point out two interesting facts:

- The results seem to confirm what is reported in literature: the meteorological and pollutant variables are significant when the prediction is in the short period.
- The four additional variables, that convey mainly with the seasonal variability, are able to represent the common patterns of similar hours in similar days in similar months.

The obtained results indicate that the best predictions can be obtained utilizing all the 12 variables. Even if the set of eight variables gives a significant contribute in the short term prediction, it also improves the forecast for longer periods and the set of the four additional variables improve significantly the prediction in the first day as well.

To summary we can say that The first part of the experimentation done in this work is aimed to the attempt to obtain the ozone forecasts by using eight standard inputs variables that represent the meteorological and pollutant data. The SVM that uses these input variables produces a good forecast for the first day and reasonable forecasts from the second day to the tenth day. The results obtained by the SVM are compared with the ones obtained by a multiple regression technique. This comparison seems to indicate that, at least for this particular problem, the SVM has a better capacity to represent the non-linear relations between the input variables and the output ones.

In the second part we investigated the possibility to improve the performance of the SVM by exploiting further information not conveyed in the eight standard input variables. To this aim, four additional inputs variables are added to the SVM: hour of the day, day of the week, day of the month, month. In some sense these four inputs should represent the human activity and the seasonal variability. These new variables produce no changes in the forecasts obtained by the multilevel regression technique, but improve significantly the forecasts of the SVM both in the short term prediction and in the mid term prediction.

In conclusion the reported results indicate that a possible way to get reasonable forecasts is to use optimized SVMs and suitable input sets composed by standard variables and additional ones.

3.3 Sales Forecasting

This work is concerned with sales forecasting in a retail store of large distribution. In past times managers of such stores normally used their experience to predict the daily sales and to decide the resupply quantities. In more recent years, with the development of computer aided decision making, especially in the bigger firms, the use of mathematical methods has became more and more widespread. In years 70s

and 80s the principal methods used were statistical methods based on time series autoregressive models, like the ARIMA method, the Box-Jenkins method and the Winter modification of the exponential smoothing method (see e.g. [65]). The data used by these methods are taken from the same time series that one wants to forecast, and that can be therefore considered as an output series.

The main concern of this work is how the amount of sales of a given commodity depends on different suitable input attributes. The principal aim of this work is to assess the relative effectiveness in sales forecasting of the three kinds of learning machines: multilayer neural network, radial basis function neural network and support vector machines. The second aim is the comparisons with time series based methods, using the real data of a retail store. A distinguishing features of the paper is that it focuses on the effects of an abnormal input attribute, that is occurrence of promotions on sales.

There are several works in literature that deal with these issues. One of the first works dating to the 90s, [66] showed the superiority of ANN on the ARIMA method in sales forecasting. A state of the art on the use of ANN in forecasting is provided in [67]. In [68] several comparisons are made between learning machines and statistical methods, showing from empirical results that learning machines have an edge on statistical methods especially in periods of volatile economic conditions. Sales forecasts on a weekly basis using different inputs are obtained in [74] and [75], proving again the efficacy of ANN. As concerns SVM, their potential application in sales forecasting is dealt with in [69]. Other works focus on the flexibility of learning machines. For example in [70] fuzzy neural networks, and in [71] both fuzzy neural networks and clustering methods are used to improve neural networks results. In [72] and [76] particular optimization procedures are used, like genetic algorithms or swarm optimization, to improve the forecast and to obtain better results than the statistical methods. In a more general framework, see [73] and [78], the authors use learning methods in the economical context of marketing for predicting consumer's future choices.

3.3.1 Experimental environment

In this section we describe how the learning machines have been used for sales forecasting. In our application, we use two input-output time series, taken from two different retail stores of the same chain of large distribution. As concerns the output y we are interested in the daily sales of a particular kind of pasta of a popular brand; as concerns the input vector x, we will describe below which attributes have been taken into account. In particular we are interested in capturing the effects of promotion policies on the sales. The input-output samples used for training, validation cover four years: 2007, 2008, 2009 and 2010, while the dataset used for testing is the year 2011.

The first time series is taken from the retail store #1, which is characterized by a bad storage management, so that stockout occurs often. This brings the difficulty of not knowing whether an output sample is zero because there was no demand or because there was stockout. We could choose to eliminate the samples with zero output in order to make this dataset more reliable. But we chose to leave the set as it is because in real application, such a thing often happens, and we are also interested in see how the predictive models would behave. The second series is taken from store #2, which has a good storage management, so stockout happens rarely.

In our forecasting we will use as input attributes subsets of the following set of 13 attributes:

- 9 calendar attributes, linked to the specific day in which the output is given: month, day of the month and day of the week. The day of the week is represented by 7 mutually exclusive boolean attributes. These attributes bring into consideration typical human behaviors and customs. For example on Saturday it is expected to sell more than in the other days of the week.

- 4 problem specific attributes: one boolean attribute whose value is one if there is promotion of the product in that day, zero otherwise, number of hours the store is open that day and the daily price of the product; moreover the overall number of receipts released that day in the store, which accounts for the overall volume of sales.

As concerns the last attribute listed before, that is the number of receipts released in the same day for which the forecast is done, we point out that of course its value is not known. Therefore we implement a SVM for forecasting the number of receipts per day. This SVM uses the years 2007, 2008, 2009 and 2010 for training and for validation with a k-mean strategy. Then we use this SVM to produce a forecast of the number of receipts in the 2011. This SVM uses in input 11 attributes: the 9 calendar attributes also used in forecasting sales, the number of hours the store is open and a last attribute that indicates if in that day are expected high or low sales. This attribute is 0 in normal days, 1 in days before festivities, -1 when the store is open on Sundays and 2 on the day of Christmas eve and new year eve. A forecasted attribute can be considered a risky choice for the robustness of the final predictive model. However as we already said, we consider this attribute very important in the prediction and it also can be used in place of the calendar attributes in order to avoid the curse of dimensionality.
We realized several experiments changing the attributes in input:

- in the first experiment we use 4 inputs: promotion, number of opening hours, price of the product and number of daily receipts (forecast);

- in the second experiment we use 12 inputs: promotion, number of opening hours, price of the product and the nine calendar attributes;

In the 4 inputs experiment we test the the goodness of final prediction with the forecasted attribute. In the 12 inputs experiment we test the goodness of the prediction with the calendar attributes, but without the forecasted number of receipts.
The forecasting is executed by adopting the *sliding window* method often used in this kind of applications. After eliminating the days in which the store is closed, we divide the year of test, the 2011, into several intervals with four weeks worth of data, for an overall numbers of 13 prediction periods. We use the samples of the whole year 2011 for forecasting and testing.

the 13 sets of samples used for testing are denoted by $\mathcal{S}_i$,$i = 1,\ldots,13$. The remaining samples in the training set are the set $\mathcal{T}$.
For different learning machines, belonging to the three classes of Multilayer ANN, RBF ANN, SVM, we first perform the k-fold cross validation procedure using the set $\mathcal{T}$ to select the best performing hyper parameters for the learning machine; then we use the selected machine to perform the forecast of the output samples in the test set $\mathcal{S}_1$, and we measure the quality of the forecast by the $MSE(\mathcal{S}_1)$ value. Then we add the set $\mathcal{S}_1$ to the training set, repeat the training and cross validation phase and we take the set $\mathcal{S}_2$ as new test set, and then measure the $MSE(\mathcal{S}_2)$ value. The procedure is repeated, until the last interval of the year 2011 is reached.

We adopt this strategy in order to simulate the typical behavior that a practitioner would use in order to compute a prediction. In general a year-long prediction is not applied, while a month(four weeks) long prediction is more realistic. It is also realistic to put the most updated data when the prediction is performed. This is the reason why we used this sliding windows methodology.

3.3.2 Computational results

In this section we report the results obtained in forecasting the sales during 2011, making use of the different Learning Machines, and we make a comparison with the forecasts provided by traditional statistical methods. In particular, for each store we run nine computations, six for the three different learning machines by using the two different configuration of input attributes, and three for statistical methods, the first method being ARIMA, the second one being the exponential smoothing (ES) and the third one being the Holt-Winter variation of exponential smoothing(HWES).

Forecast of Daily Receipts

Preliminarily we show the results obtained using a SVM for forecasting the number of daily receipts in the 2011, used as input attribute. As already said, we used the

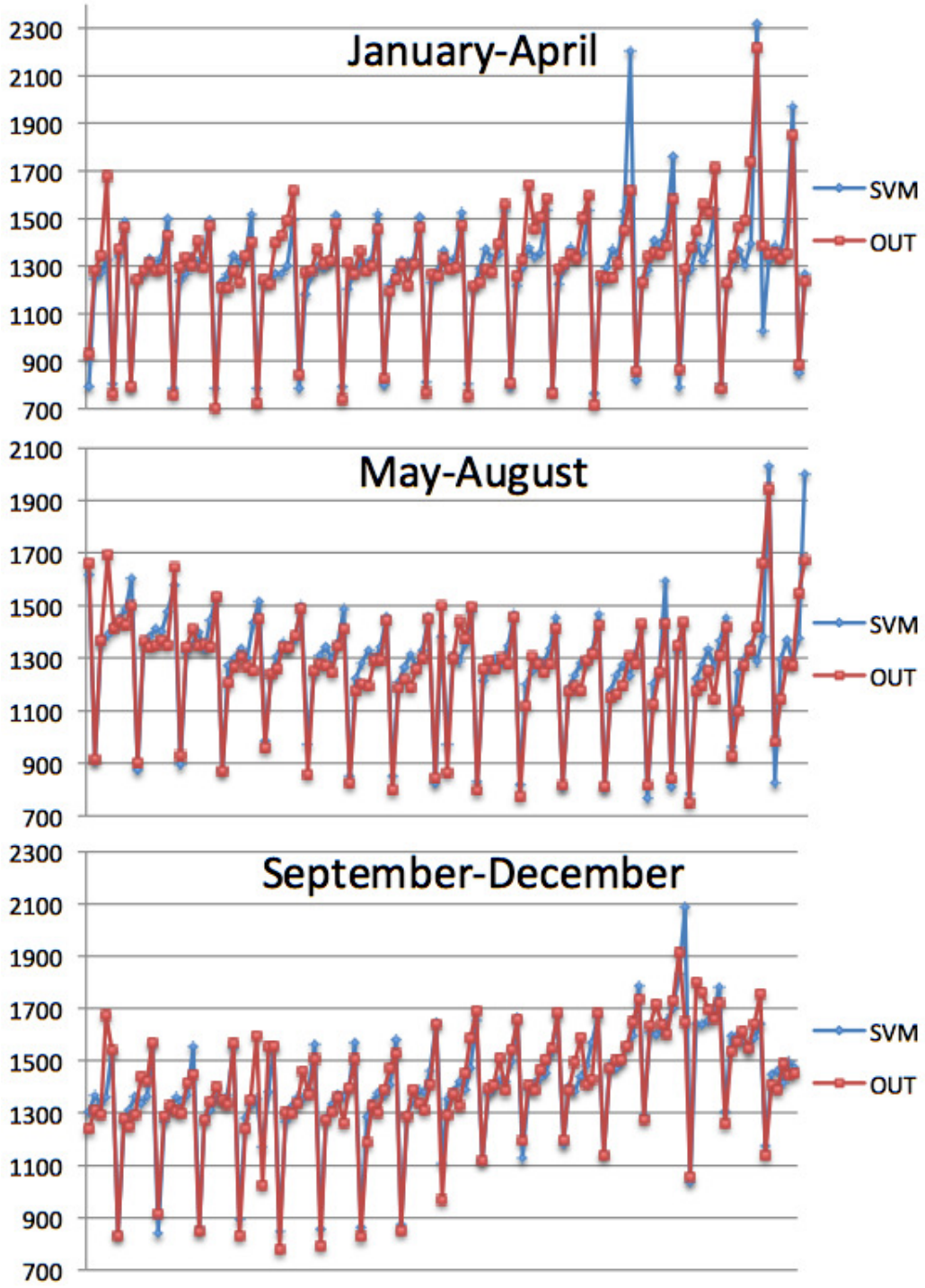

Figure 3.5. Prevision of the receipts for the store # 1

samples of the previous years for training validation, with the 11 input attributes listed in Section 3.3.1.

In Figures 3.5, 3.6 it is possible to see the results for the receipts forecasting. From the results we are able to notice that not only the SVM are able to follow the weekly thread of the output, but also to understand when there are particularly high values of the samples. In order to measure the goodness of the results, we used the R^2 error for this experimentation. We have:

$$R_1^2 = 0.84 \qquad R_2^2 = 0.70$$

Both the values of the of the R^2 are high and indicate a good prediction rate of the surrogate model.

Sales forecast in store #1

The results obtained after training with cross-validation the three kind of learning machines, each one with two different configuration of input attributes, denoted by $4i$ and $12i$ are given in terms of $MSE(\mathcal{S}_i)$,$i = 1, \ldots, 13$ in Table 3.10. In the same table are given the $MSE(\mathcal{S}_i)$ values obtained using the three statistical methods. All the reported results are on the test set not used for training. We notice that the

PERIOD	SVM 4i	SVM12i	RBF 4i	RBF 12i	MULTY 4i	MULTY 12i	EXP	EXP HW	ARIMA
1	44.689	37.223	52.667	82.269	51.429	82.646	74.045	70.494	73.624
2	49.287	47.200	57.308	35.431	53.063	35.431	38.459	38.453	55.592
3	40.579	37.932	40.099	35.093	40.383	35.093	39.275	38.955	36.969
4	37.823	34.913	36.793	42.521	39.393	42.521	46.911	48.804	35.938
5	141.230	172.032	560.366	576.813	296.166	576.813	174.839	167.022	167.428
6	31.656	25.636	32.153	23.947	29.324	23.947	26.498	27.320	27.772
7	80.293	88.650	135.142	436.711	155.888	436.711	107.967	100.863	116.649
8	77.818	44.933	85.364	48.743	72.307	48.743	73.705	73.611	57.188
9	52.446	67.243	54.355	63.090	53.751	63.090	47.940	54.611	54.614
10	101.804	171.564	146.306	221.657	328.535	221.657	86.232	89.004	72.997
11	30.401	26.771	29.843	29.345	37.023	29.345	28.187	28.478	31.291
12	69.175	82.413	80.364	218.250	79.591	218.250	63.411	64.316	53.365
13	44.057	36.842	45.230	35.614	49.374	35.614	81.798	84.176	50.250
MEAN	61.751	67.355	104.677	142.967	99.248	142.997	68.406	68.134	64.310

Table 3.10. MSE results for all the 13 periods and all the predictive models used for store # 1

SVM are able to get a better prediction than the other learning machines because

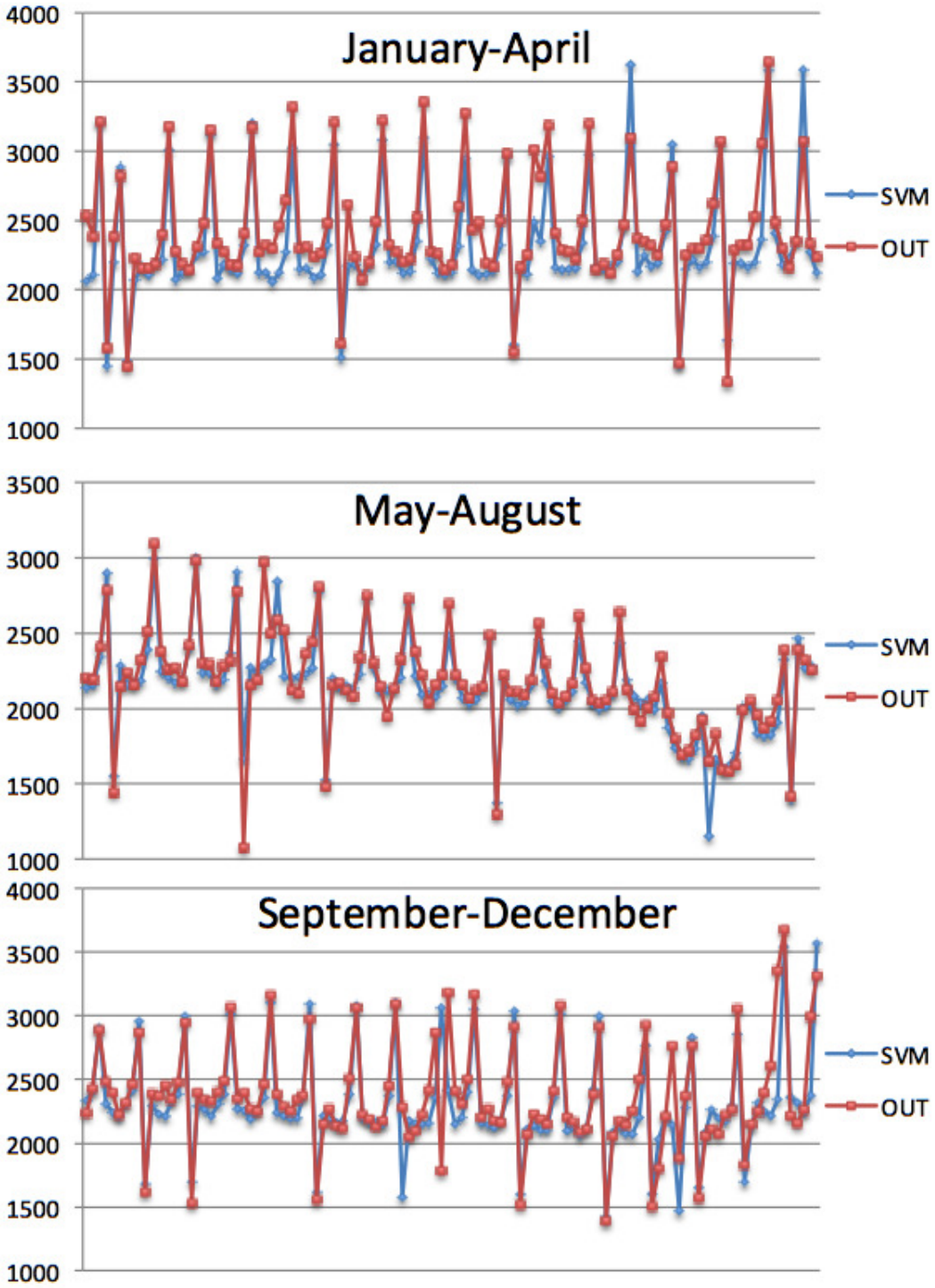

Figure 3.6. Prevision of the receipts for the store # 2

they have in average the best behavior. Also all the learning machines with four inputs behave better than their 12 inputs counterparts. If we take a look to the results period by period, the SVM 4i is able to get the best prediction for five periods on 13, while the SVM 12i is able to get the best prediction for four periods on 13 and the RBF 12i gives the best results for four periods on 13. The statistical methods are never able to get the best prediction for a period. For the certain periods of promotion that is period 5, 7 and 10 there is a substantial increase of the error. The SVM are able to fare better than the other methods because their error increases less. In figure 3.7 there are the results in the promotion periods

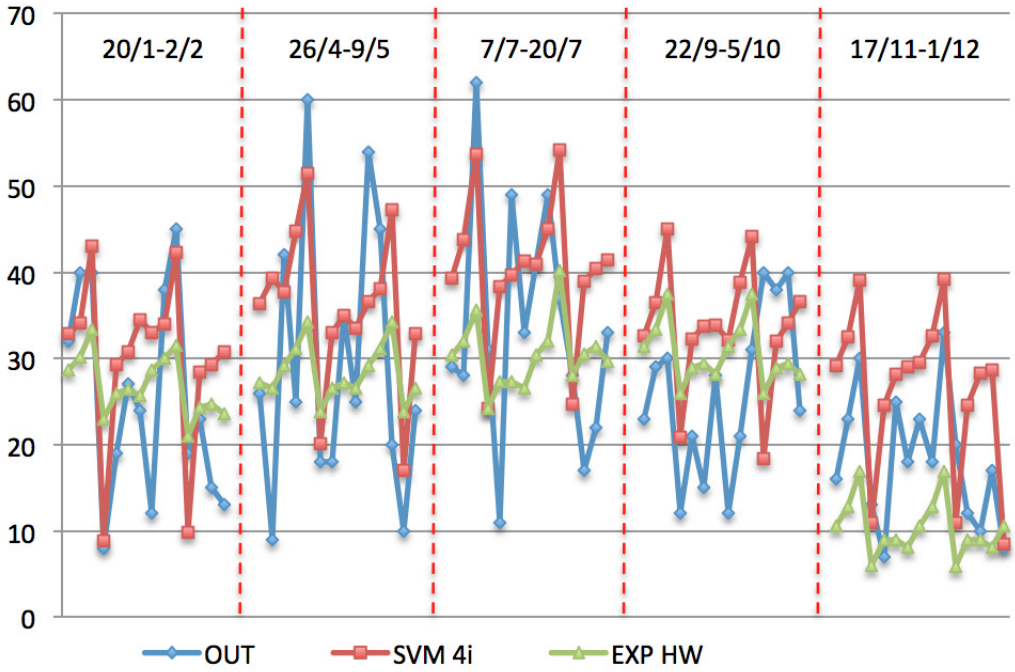

Figure 3.7. Periods of promotion for the store # 1

for the best learning machine configuration, SVM 4i, and the best statistical model EXP HW, compared with the output. First we notice, especially for the periods of promotion going from 26/4-9/5, 7/7-20/7 and 22/9-5/10, is that the weekly trend is totally disrupted by the promotion effect. The output assumes values difficult to predict and with an high variability. The statistical method, even in this periods of promotion, follows its weekly trend, while the SVM are able to catch a part of this consistent variability. From the results, the other learning machines have an high value of the error because their prediction have an higher variability than the one of the output causing them to have very high or really low values.

Sales forecast in store #2

The results for the second store prediction are reported in Table 3.11. This store

PERIOD	SVM 4i	SVM12i	RBF 4i	RBF 12i	MULTY 4i	MULTY 12i	EXP	EXP HW	ARIMA
1	883.66	1232.07	763.75	1630.49	852.83	4948.46	1354.73	1390.98	1383.15
2	295.33	296.26	314.21	388.25	284.21	444.86	271.18	302.91	270.31
3	213.98	202.82	244.40	183.79	230.87	179.07	180.51	149.12	167.61
4	70.49	54.72	70.77	56.48	62.40	49.64	56.58	68.92	55.53
5	364.84	405.70	357.89	323.32	342.93	799.31	617.09	614.62	574.86
6	58.11	72.57	64.25	70.25	55.68	171.86	49.07	48.73	42.81
7	506.85	586.05	682.59	536.77	559.03	713.07	862.46	828.62	853.50
8	133.93	121.03	139.66	109.84	156.32	140.58	130.26	139.43	134.73
9	31.07	60.66	44.61	52.87	30.32	49.14	50.86	57.77	45.72
10	687.71	496.86	790.20	468.41	603.22	349.52	1262.27	1220.03	1177.41
11	100.98	129.08	116.97	132.41	113.56	133.06	79.54	107.40	65.85
12	185.67	264.01	207.97	185.78	426.98	210.32	805.75	843.36	817.82
13	139.92	109.50	131.89	109.41	137.75	131.58	123.48	164.07	120.10
MEAN	281.21	283.56	301.76	324.56	281.21	308.54	454.74	462.85	444.64

Table 3.11. MSE results for all the 13 periods and all the predictive models used for store # 2

has a more regular behavior with a more marked weekly trend. For this reason the statistical methods are able to get the best prediction for four periods on 13. The SVM 4i, even if it has the best prediction only on two periods on 13 has the best average behaviour. Like for the previous series, during the period of promotion the error committed by the predictive models greatly increases. The learning machine models, especially those with four inputs, are able to preserve the error relatively low, while the statistical models show poor performances if compared with the learning machines. As a matter of facts they are able to give a better prediction than the learning machines only in cases of particular regularity. In Figure 3.8 it is possible to observe the behavior of the best configuration among the learning machine, in the periods of promotion. For this series the statistical methods still has its weekly trend that is not changed even in the periods of promotion. As a matter of facts the values of the output are considerably lower than the SVM. On the other hand, the support vector machine is able to follow the erratic behavior of the output better.

As conclusion, we can say that the learning machine have comparable, if not even better, performances to the statistical methods when it come to forecast periods with regularity with some kind of weekly trend while the surrogate models shows to have the upper hand in periods of promotion.

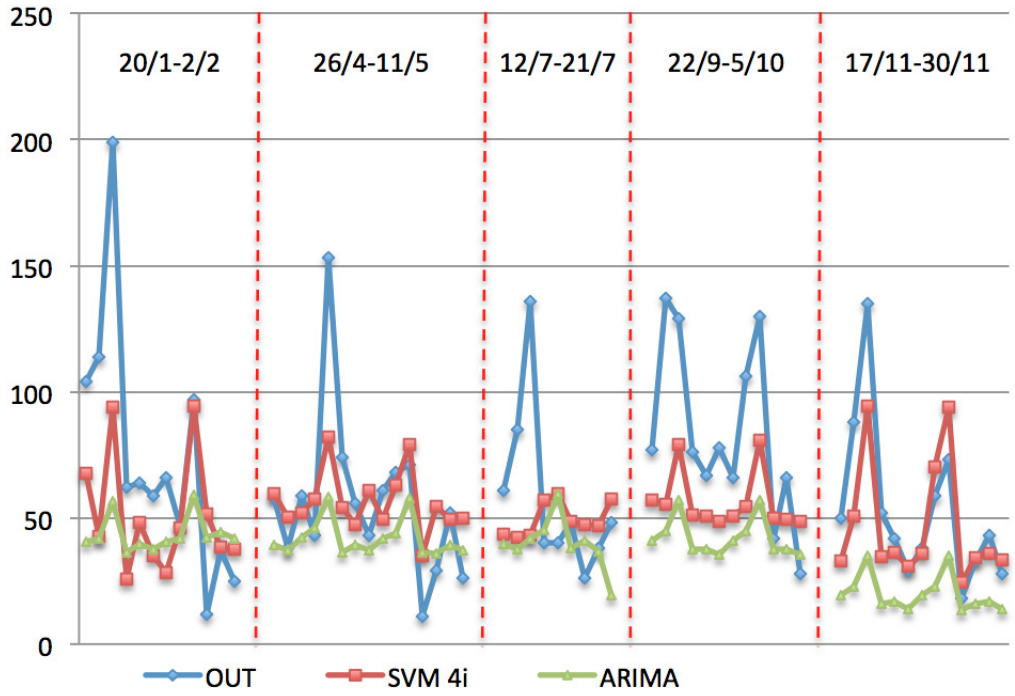

Figure 3.8. Periods of promotion store # 2

From the point of view of the inputs to choose in order to train the model, the 4i configuration shows to have, in average, a better behavior than the 12i configuration. This seems to indicate that the reduction of inputs in the model greatly helped the prediction process.

3.4 Surrogate Modeling for Electronic Circuits

Circuit design is a phase of circuit manufacturing process. In the design development a circuit must be tested in several ways in order to be ready for massive production. These tests determine if a circuit satisfies its specifications in different possible scenarios. If the circuit does not satisfy its specifications, it can bring to circuit malfunctions or circuit failures. These tests are performed with computer simulations. The computer simulates the behavior of the circuit for the time needed to calculate its performances.

The time needed to perform a simulation changes according to the complexity of the circuit, it can take from a minute to several days to complete it. Generally speaking, these simulations can be considered costly. In industrial design there are strict temporal limitation due to the time to market. This motivates the effort to

make faster the phase of circuit analysis by resorting to the use of surrogate models rather than to simulations. Surrogate models are mathematical models that use couples of input-output samples to create a regression function that approximates the relation between the input parameters and the output performances.

In the literature, several works about surrogate models applied to circuit simulations are proposed. For example in [49] the authors use Kriging models to find the effects of time-based degradation on circuit performances. In [50] the authors apply Kriging models to LNA(Low Noise Amplifier) circuits with different optimization strategies in order to define the best hyper-parameters of the model. The same authors in [51] compare several surrogate models based on samples taken from a first-order analytic model of the LNA circuit. In [52] Kriging models are compared with a simple response surface methodology, obtaining good results. Another paper [53] proposes quadratic polynomial methods in an application aimed to find the design parameters able to maximize the lifetime of a circuit.

The more samples a surrogate model has in input, the more the model will be accurate. Every sample in input is generated through a costly CPU simulation. So there is a trade-off between the time needed to generate the set of samples used for the model generation and the accuracy of the model itself. We will perform experiments to see how the surrogate models behave when the number of samples given in input decreases. In literature there are several papers on this trade-off problem, for example in [54] several strategies and surrogate models are presented and compared.

In this experimentation we use the Support Vector Machine(SVM) to create surrogate models based on computer simulations and then compare them with one of the most used surrogate model for this kind of applications, the Response Surface Methodology(RSM) [55], realized in the commercial software WiCkeD [56], widely used in the industrial sector . WiCkeD is a suite for circuit analysis, modeling, sizing, optimization and surrogate model generation. The simulations are based on actual circuits designed and produced by ST-Microelectronics. These circuits will become components of actual consumer technology devices. SVM is a relatively new tool for surrogate modeling and shows good performances for nonlinear regression approximations ([57], [58], [59]). The relations between the input parameters and the output performances of circuit simulations are characterized by a high degree of nonlinearity, and the SVM can be a valuable tool in this kind of applications. Moreover, the training process of the SVM is relatively simple, as it consists in solving a convex optimization problem. This feature enables practitioners to obtain the global solution of the optimization problem and to complete the training of the SVM in a short time.

The principal aim of this work is to understand if the SVM are useful in modeling actual industrial circuits, as in our future work we intend to use them in several application in order to shorten the time needed to perform a circuit analysis. To

this aim the SVM are tested on real circuits and the results are compared with the RSM given by the software suite Wicked. This software platform implements four types of basis functions: Polynomial, Radial Basis, Affine plus Radial Basis and Polynomial plus Radial Basis (for further details see [60] Chapter 6). WiCkeD selects automatically the basis function $g(x)$ for the RSM as the basis function that better fits the data given in the training set. After choosing the basis function, Wicked solves the least square error problem for the RSM, generates the surrogate model and tests it on the test set, that is the same of the SVM.

As we are also interested in using as few simulations as possible to obtain a reliable model, we perform other tests reducing the number of samples in the training in order to understand how many simulations are needed to obtain a reasonable precision.

3.4.1 Experimental Set Up

In this section we explain how the surrogate models have been implemented.

In circuit design there are three types of parameters:

- Design parameters: these parameters are concerned with the sizing process of the circuit, and are given by the geometrical dimensions of the devices that the circuit designers will assemble into the actual circuit;
- Process parameters: these parameters represents statistical variations due for example to the fluctuations in the manufacturing process of the circuit. The process parameters are modeled with a Gaussian or with an uniform probability density function;
- Operating parameters: typical examples of these parameters are those that model the operating conditions of the circuit, like supply voltage and temperature.

The parameters are characterized by box constraints that determine the range in which they can vary. These parameters are normally given in input to the simulator together with a "netlist" that explains the connections in the circuit. A high number of parameters in input causes the so called "curse of dimensionality", so it is convenient to use the less parameters as possible without losing information on the phenomenon under examination. In order to handle this problem, the sensitivity analysis method has been used in this work.

Sensitivities are calculated by finite differences. In its simplest form, a parameter of interest z is altered by a small amount Δz and the resulting variation for the $i-th$ performance v_i, $\Delta v_i = v_i(z + \Delta z) - v_i(z)$, is used to calculate the sensitivity. If the variation is under a certain threshold for every performance g_i, the input parameter is discarded.

The performances of the circuit can be given by the delay between two waveforms, the duration of an impulse, the variation of the voltage due to this impulse, the slope of the change of voltage etc. These are the quantities that the SVM and RSM are required to approximate, given the values of the input parameters. For every performance we will report the interval of its specifications, in order to give a feeling about the required precision.

After selecting the significant parameters, the sets in input to the surrogate models are generated. Following many works presented in literature [77] we choose the Latin Hypercube Sampling as Design Of Experiment scheme for generating the points in the training and test set for both the SVM and WiCkeD RSM. The performances corresponding to these design points are calculated using the simulation software called ELDO. After generation, the samples are normalized in order to be

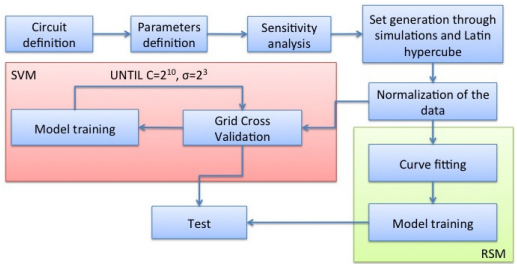

Figure 3.9. Flow chart for the experimental set up

ready for the training phase. Normalization helps the mathematical model that generates the regression function to better understand the importance of the variation of a single parameter in respect of the output performance.

As concerns the software realization of the surrogate models, SVM are implemented in the C programming language starting from the software available at [28]. The RSM are integrated within WiCkeD's platform. An advantage of the SVM is that we have direct access to the source code, thus adding flexibility in our implementation.

In order to determine the number of samples needed to obtain the desired precision, we perform an initial training of the surrogate models with a considerable number of samples in the training set. After the first test, we decrease the number of samples in order to see how the two models would fare. In other words we try to stress the structure of the surrogate models to find a reasonable lower limitation to the number of samples in the training set. This is an important test, because, as we said in the introduction, the less simulations are used to realize a surrogate models, the best it is from a design point of view. Both the SVM and RSM are trained on the same training sets and every generated model is tested on the same test set,

comprised of 500 samples not utilized in the in the training and model selection phase.

The reported results on the test set will not only take into account the MSE on the denormalized performances, but also the coefficient of determination R^2:

3.4.2 Results of the Surrogate Models

In this section we report the results on two circuits designed by ST-Microelectronics. First we present the circuit description and performances with their bounds, then we apply a screening of the parameters. We also report the results and see the effect the lowering of the number of samples has on the prediction. To get a feeling on the desired degree of precision, we label as good the models with an R^2 coefficient that exceeds the 0.9, as satisfying the models that have R^2 between 0.9 and 0.8, as reasonable the models that have R^2 between 0.8 and 0.6, and as bad the models with R^2 lower than 0.6.

Digital to Analog Converter

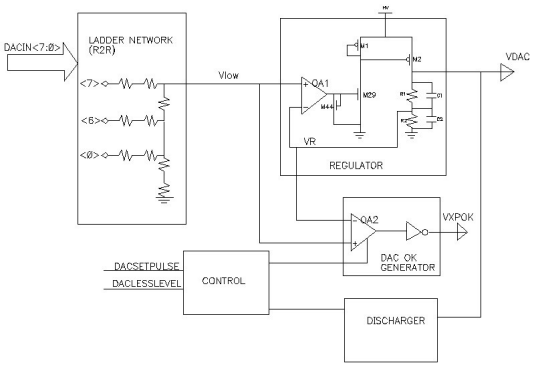

Figure 3.10. DAC architecture

First we present the results regarding a digital to analog converter circuit used for voltage generation and timing in input of cells of flash memory products. Flash memory products need different voltage levels to be applied to the memory cells during operations (reading, programming, erasing). These voltages must follow specific levels and timing for each operation. A DAC circuit controlled by a memory micro-controller is commonly implemented for voltage generation. The architecture of the DAC circuit is shown in Figure 3.10.

This circuit has in input 54 parameters. After a screening of the parameters performed with a sensitivity analysis, the number of significant parameters is reduced

to 28. The performances and their allowable bounds are reported in Table 3.12.

Performance	Lower	Upper
GAIN	0.5	2
SLOPE	1	3
VXP-3O3	1	5
VXP-6	5	7
VXPOK2	5.88	6.12
VXP-9	7	10

Table 3.12. Performance for the DAC circuit

In Table 3.13 we report the results with 28 parameters in input, using 1000 samples for training and 500 samples for testing for both the SVM and the RSM.

Per.	SVM	SVM	RSM	RSM
	MSE	R^2	MSE	R^2
GAIN	5.1E-06	0.860	8.3E-06	0.773
SLOPE	1.8E-05	0.94	2.5E-05	0.909
VXP-3O3	2.7E-04	0.947	2.4E-04	0.952
VXP-6	4.9E-05	0.987	3.4E-05	0.991
VXP-9	1.6E-04	0.963	1.7E-04	0.960
VXPOK2	5.5E-03	0.933	6.0E-03	0.926

Table 3.13. Results for the 6 performances with 1000 samples in the training and validation set and 500 in the test set.

On the overall the two methods yield similar results; in particular the SVM has an edge in the prediction of the performances GAIN and SLOPE while the RSM gives better results for the VXP-6. We also note that according to the values of R^2 the two surrogate models can be classified as good with respect to all performances, with the only exception of the performance GAIN, to which nevertheless corresponds a value of R^2 to be classified as satisfying.

In Table 3.14 we report the results on the same test set of 500 samples used in the previous experiment for the surrogate models obtained decreasing the number of samples in the training set.

Overall the two methods have comparable results with the SVM being a little better, especially when it comes to predicting the following performances: GAIN, SLOPE, VXP-303 and VXPOK2, while the RSM is still better in the prediction of the VXP-6.

500	SVM	SVM	RSM	RSM
	MSE	R^2	MSE	R^2
GAIN	7.66E-06	0.79	1.18E-05	0.67
SLOPE	2.44E-05	0.91	3.46E-05	0.87
VXP-3O3	3.11E-04	0.94	3.06E-04	0.94
VXP-6	5.43E-05	0.99	5.80E-05	0.99
VXP-9	2.13E-04	0.95	1.72E-04	0.96
VXPOK2	8.34E-03	0.90	1.00E-02	0.88
200				
GAIN	1.36E-05	0.62	1.89E-05	0.49
SLOPE	4.25E-05	0.84	5.46E-05	0.81
VXP-3O3	3.40E-04	0.93	4.70E-04	0.91
VXP-6	1.01E-04	0.97	8.62E-05	0.98
VXP-9	4.19E-04	0.90	4.21E-04	0.90
VXPOK2	1.29E-02	0.84	1.39E-02	0.83
100				
GAIN	1.74E-05	0.52	2.07E-05	0.46
SLOPE	4.80E-05	0.82	6.13E-05	0.79
VXP-3O3	3.90E-04	0.93	4.70E-04	0.91
VXP-6	1.32E-04	0.97	1.03E-04	0.97
VXP-9	5.62E-04	0.87	5.55E-04	0.87
VXPOK2	1.82E-02	0.79	1.93E-02	0.76
50				
GAIN	2.11E-05	0.41	2.26E-05	0.39
SLOPE	6.66E-05	0.76	8.57E-05	0.70
VXP-3O3	6.22E-04	0.88	8.48E-04	0.83
VXP-6	2.04E-04	0.95	1.97E-04	0.95
VXP-9	8.72E-04	0.80	9.03E-04	0.79
VXPOK2	2.18E-02	0.75	2.69E-02	0.69

Table 3.14. MSE and R^2 results for the six performances of the DAC diminishing the number of samples in the training of the training on the same test set of 500 samples

The two models still have a good prediction capability even decreasing the number of samples in the training set. We note a steadily decrease of the surrogate models precision. In particular if we take as example the results obtained with only 50 samples in input, we note that the surrogate models for the performance VXP-6 still yield good results, while there are still satisfying results for other two performances, reasonable results for another performance and bad results for the

remaining two.

From these results we can conclude that the surrogate models are able to give, in the majority of cases, more than reasonable results even with a limited number of samples in the training set, with the SVM yielding sightly better results than the RSM.

DC-DC Converter

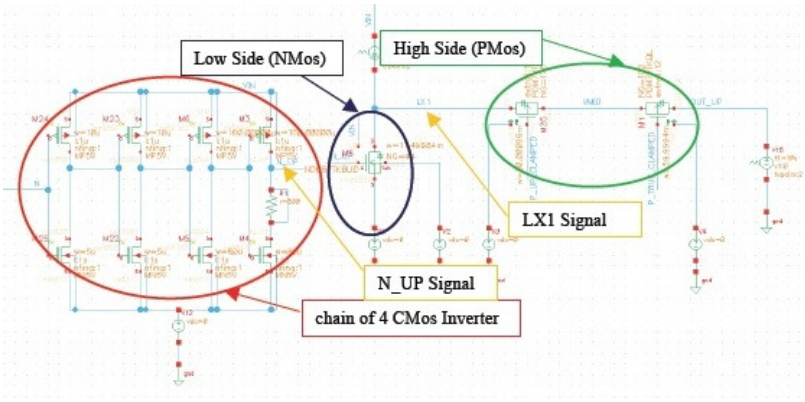

Figure 3.11. DC-DC Converter architecture.

Now we present the results regarding a DC-DC converter for AMOLED display panels. It is important in this circuit the delays between the time the signal to turn off/on the circuit is sent and the time the circuit is actually turned off/on. The longer the difference between these delays is, the greater the unwanted diode re-circulating phase is, hence increasing power losses. In Figure 3.11 a representation of the circuit is shown, with its principal components:

- A chain of 4 CMOS inverter ;
- the High Side (PMOS) and the Low Side (NMOS) output stage;
- the driving signals.

The initial number of input circuit parameters is 84, but after performing a sensitivity analysis only 19 inputs remain. In table 3.15 the performances of the circuit are displayed, with their bounds. Delay1 indicates the delay between the time the signal to turn on the circuit is sent and the time the circuit is actually turned on, Delay2 is the same measure when the signal to turn off the circuit is sent. Delay symmetry indicates the difference between Delay1 and the Delay2.

Performance	Lower	Upper
Delay one	0	20E-9
Delay simmetry	-3E-9	3E-9
Delay two	0	20E-9

Table 3.15. Performance for the DC-DC circuit.

In table 3.16 the results obtained by the SVM and RSM using training sets of different sizes are reported. The test set is given by 500 samples and it is the same for all the experiments.

1000	SVM	SVM	RSM	RSM
	MSE	R^2	MSE	R^2
D1	3.36E-18	0.93	2.94E-18	0.94
D2	1.17E-18	0.97	1.08E-18	0.97
DS	1.44E-18	0.95	1.29E-18	0.95
500				
D1	4.64E-18	0.90	4.85E-18	0.90
D2	1.83E-18	0.96	1.71E-18	0.96
DS	2.13E-18	0.93	1.96E-18	0.93
200				
D1	9.37E-18	0.81	8.35E-18	0.82
D2	4.78E-18	0.89	4.56E-18	0.89
DS	4.22E-18	0.85	4.08E-18	0.86
100				
D1	1.14E-17	0.76	1.19E-17	0.75
D2	6.38E-18	0.85	7.70E-18	0.83
DS	6.49E-18	0.78	5.99E-18	0.80
50				
D1	1.76E-17	0.65	1.59E-17	0.66
D2	9.08E-18	0.78	1.03E-17	0.78
DS	9.80E-18	0.69	8.97E-18	0.70

Table 3.16. MSE and R^2 results for the three performances of the DC-DC Converter diminishing the number of samples in the training on the same test set of 500 samples.

The two surrogate models yield similar results for the DC-DC converter as well, but differently from the DAC, this time the RSM perform slightly better. Overall the two methods are able to give good results when there is a high enough number of samples in the training set. The two surrogate models begin to have problems

when the number of samples in the training goes below the 100 samples. In any case even with only 35 samples in the training set, the two models are able to produce reasonable predictions.

Conclusions

The aim of this work is to apply the surrogate model given by the Support Vector Machine to real industrial circuits, in order to find a method of analyzing these circuits without resorting to a heavy use of costly circuit simulations. To this aim we are also interested in investigating what would happen if a limited number of samples would have been given in input to the surrogate models. We confronted the SVM with the benchmark normally used in this kind of industrial applications, the model obtained by the Response Surface Methodology and obtained comparable results. The obtained results can be reputed satisfying, especially when more than 500 samples are used for the training of the surrogate models. An interesting observation is that the surrogate models are able to give reasonable results for such a complex problem of electronic circuit design even with a limited number of samples in the training set.

In conclusion our experimentation indicates that the SVM is a valid surrogate model for real industrial electronic circuits and it can be considered a valuable alternative for applications in electronic circuit design.

3.5 Yield Optimization

A circuit must have a certain behavior in order to work well. This behavior is generally represented by the circuit performances and their constraints. If the circuit performances satisfy the constraints, the circuit is considered to behave well. There is a random variability on the circuit manufacturing process, and this variability determines variations into the device parameters that cause the circuit performances to be stochastic distributed, that is their values change from circuit to circuit. This phenomenon is unavoidable, but in any case, the values of the circuit performances must satisfy the constraints set by the designers no matter the random variability. In the case the specification are not satisfied, the circuits must be discarded. In other words a production with an high probability that the circuits satisfy the specifications is a production with less costs and more economic benefit.

This problem becomes more and more intense with manufacturing processes that scale down the size of the circuit components. The more the components are small, the more a statistical variations affect the performances.

A measure that expresses the probability that the circuits satisfy the specifications is the yield, that is the percentage of circuits that satisfy all the performance

specifications at the same time. Generally the yield of a circuit can be expressed as a multivariate integral, that can be approximated through a Monte Carlo(MC) analysis. The MC analysis is a flexible and practical tool that can be applied to various circuits without decreasing their number of parameters or simplifying their stochastic distributions. The greatest drawback of the MC analysis is that it requires a great number of evaluations in order to be performed. These evaluations grow with the number of circuit parameters. Every evaluation is performed through a computer simulation. Circuit simulations can be costly, then performing a MC analysis even on circuits of medium size, can be considered time expensive. In order to decrease the cost of the analysis, the two principal strategies can be adopted: change the type of sampling [63], or use surrogate methods, such as learning machines, to approximate the circuit evaluations [64], [61]. In this work we propose an approach that combines the Support Vector Machines(SVM) and a derivative free mix-integer black box algorithm in order to solve the circuit yield optimization problem with a relatively low number of circuit simulations. Our method is compared with the results of the the commercial software WiCkeD [56], widely used in the industrial sector. The simulations are based on an actual circuit designed and produced by ST-Microelectronics.

3.5.1 Problem Description

As we already said in the previous section, there are three kinds of parameters in a circuit:

- Design parameters
- Process parameters
- Operating parameters

The objective of the yield optimization is to determine the design parameters that maximize the yield in spite of the stochastic variations of the process parameters and of the different possible values of the operating parameters. Let p_d,p_p,p_o be respectively the design, process and operational parameters vectors, let f_i,$i = 1,\ldots,r$ be the performances and let f_i^l,f_i^u be lower and upper bounds on the performances. The yield corresponding to a given value of design parameter p_d, under operating conditions given by p_o is determined by the probability that the performance $f_i(p_d,p_p,p_o)$ satisfies the bounds:

$$f_i^l \leq f_i(p_d,p_p,p_o) \leq f_i^u \quad i = 1,\ldots,r.$$

As said in the introduction, in order to approximate the probability of the realization of $f_i(p_d,p_p,p_o)$, a Monte Carlo analysis is performed.

3.5.2 Methodology

The base of our approach is to use the derivative free algorithms presented in Chapter 2 together with the SVM in order to find a good solution to the yield optimization problem in a limited amount of time.

In the previous section, we presented the application of the SVMs as surrogate model for circuit simulations. The SVM showed to have the potentiality to be applied in this kind of application. In the previous section we presented SVMs that approximated the entire phenomenon, modeling all the three kind of parameters with a great variability. In order to reduce the variability that the SVMs have to manage, we decided to create several local methods that manage less complexity.

In order to reduce the complexity we exploited some characteristics of the problem. First we decided to create SVMs that analyze the phenomenon only for the worst case values of the operational parameters instead of all their possible values. in order to find the worst cases of the operational parameters the worst case distances are determined for each parameter. The worst case distances determine the worst possible values of the operational parameters for the circuit when it is working. In general, there are two worst case operational points for every performance, one is the worst case for the upper bound of the performance, and the other is the worst case for the lower bound of the performance. In general it is assumed that if the specifications are satisfied at the worst cases, they are satisfied for the cases that are not the worst ones. A SVM is created for every possible worst case for every performance. In this way, by just doubling the number of the created surrogate models for every performance, we greatly decrease the complexity that the model has to manage.

Another strategy we use to decrease even further the complexity is to integrate the training of the SVMs in the derivative free algorithm. The design parameters are the only parameters that can be chosen by the designer. For this reason the design parameters are the variables of the black box optimization algorithm. Every time the algorithm needs to evaluate the objective function it sets the design parameters and creates a MC analysis around the current design point by varying the process parameters. It is possible to create a SVM that for those values of the design points, creates a Monte Carlo of the process parameters for a certain performance in a certain worst case of that performance. If we define with l the number worst cases points, a total of $r \times l$ SVMs trained at every iteration, it possible to obtain an accurate model of the behaviour of circuits simulations varying the process parameters. As the variability that has to be modeled is low, less simulations than the 1000 utilized in the previous application has to be used in order to create the various SVM models. A summary of the strategy is showed in Figure 3.12. In order

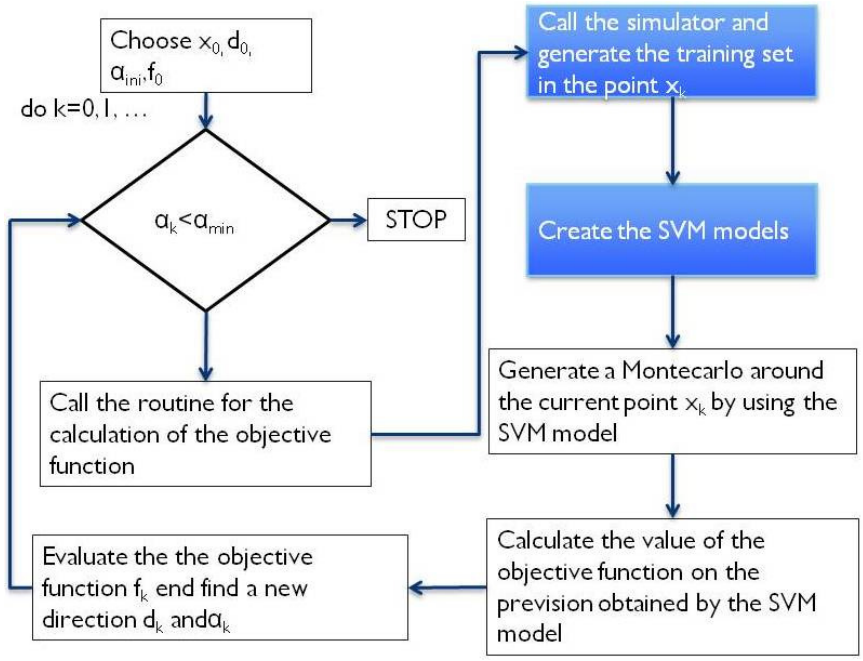

Figure 3.12. Strategy for the yield optimization.

to evaluate the yield, we utilize a zero-norm type of objective function:

$$\sum_{j=1}^{l}\sum_{i=1}^{r}\sum_{k=1}^{m}\log\left(\max\{0, f_{ijk}-f_i^u\}+\epsilon\right)+\log\left(\max\{0, f_i^l-f_{ijk}\}+\epsilon\right)$$

Where m is the number of points in the Monte Carlo, f_{ijk} is the the value of the $i-th$ performance for the $j-th$ operative point of the $k-th$ sample of the Monte Carlo and ϵ is a positive number close to zero.

3.5.3 Results

SVM Precision

In order to have a sufficient precision for the surrogate models we decide that the surrogate models must have a relative average error under the 3%. The expession of the relative error is:

$$RE = \frac{1}{N}\sum_{i=1}^{P_t}\frac{f_i - y_i}{y_i}$$

Where P_t is the number of elements in the test set, f_i are the values calculated by the SVM and y_i is the real output. We applied this strategy on the DC-DC

converter for five cases of the operating parameters, a typical case and four worst cases. There are only four worst cases instead of six because the worst cases of the delay 1 and delay 2, both at the lower bound and upper bound, are the same. In a first stage we used as training set 50 points for every analyzed case utilizing the same strategy described in Section 3.4 for a total of 250 simulations used for the training. This dataset covers the range the points around the average values to five times the variance σ. The results of these surrogate models are tested on a test set

Dataset	Delay 1	Delay 2	Delay S
50P-5_5sig.50K	1.10%	0.97%	33.12%

Table 3.17. RE for the delay 1, delay 2 and delay S with 50 points in the training set for every analyzed case with a range of 5 σ and 10000 points in the test set for every analyzed case.

of 10000 points for every analyzed case for a total of 50000 points. The values of the delay symmetry varies between $[-3.15 \cdot 10^{-9}, 3.15 \cdot 10^{-9}]$, so a lot of values are near zero. This means that for values near zero a non significant error like for example an error of 10^{-11} greatly increases the average relative error. Nevertheless, the error is still quite low for the delay symmetry.

Even if only 250 samples were used for estimating 500000 points, the results satisfy our requests on the precision with a large margin. Probably the strategy of intensely reducing the model variability created a phenomenon with low variation that is easy to predict for the SVMs.

As the precision is higher than expected, we decrease the number of points in the training sets until we don't observe a strong decrease of the performances. In Table 3.18 are reported the results of the SVMs decreasing the points in the training set. Even if such a decrease of the points in the training set is performed the prediction is

Dataset	Delay 1	Delay 2	Delay S
40P-5_5sig.50K	1.18 %	0.83 %	52.67 %
30P-5_5sig.50K	0.98 %	1.12 %	32.77 %
20P-5_5sig.50K	1.34 %	1.26 %	37.17 %
15P-5_5sig.50K	1.59 %	1.31 %	40.34 %
10P-5_5sig.50K	3.06 %	2.05 %	66.94 %

Table 3.18. Results for the delay 1, delay 2 and delay S with decreasing points in the training set for every analyzed case with a range of 5 σ and 10000 points in the test set for every analyzed case.

still good. But we notice a strong decrease of the performance between the prediction

with 15 points in the training set and 10 points in the training set. Plus the delay 1 error is over the 3% threshold.

Black Box Optimization Results

the optimization procedure is performed For every configuration of the training set when the SVMs must be created in order to approximate the MC analysis, starting from 50 points and arriving to 10 points. For every run of the optimization procedure the optimal point is the same with the same number of iterations. This seems to indicate that, even if there is a decrease in the precision of the surrogate models, this decrease does not affect the measurements of the objective function and the sequence of the points generated by the algorithm. In Table 3.19 the results between

Specification	Yield Wicked	Yield SVM
Delay 1 Lower	100%	100%
Delay 1 Upper	94.30%	89.50%
Delay 2 Lower	99.20%	100%
Delay 2 Upper	96.20%	94.90%
Delay S Lower	100%	100.00%
Delay S Upper	96.60%	100.00%
Total Yield	91.40%	89.30%

Table 3.19. Comparisons of the results between the proposed method and Wicked.

the black box method and the optimization tool used by wicked are reported. Both the two optimization routines begin from the same same starting point. Wicked has better results when it comes to making the delay 1 and the delay 2 satisfy the specifications, while it has troubles for the delay symmetry. On the other hand, the proposed method is able to find a design point that completely satisfies the constraints for the delay symmetry at the cost of losing feasibility for the delay 1 and 2. The total yield is calculated as the percentage of circuits that satisfy all the three performances at the same time. In general we can state that the two methods are able to get comparable results, with Wicked being sightly superior.

From the point of view of computational complexity, the proposed method generates 75 circuits simulation every iteration in order to train the SVMs and the solution is found after 86 iterations, for a total of 6450 simulations. On the other hand Wicked takes 11000 simulation to find the solution.

In conclusion, from these preliminary results, the proposed method shows the ability to find a comparable solution with the benchmark with only the 60% of computational cost.

Robust Optimization

The same problem was also solved with the robust optimization strategy described in Chapter 2. In this case the optimization algorithm was directly connected to the simulator. In this optimization strategy, at the outer level the design parameters are modified minimizing the violation of the performances in respect to the bounds. In the inner level the process parameters are modified maximizing the violation of the performances in respect to the bounds. At every iteration of the outer level, the value of the objective function corresponds to the process parameters that violate the most the conditions on the performances. In Table 3.20 it is possible to observe the results of this method. The robust optimization algorithm, at the cost of losing

Specification	Yield Wicked	Yield SVM	Yield Robust
Delay 1 Lower	100%	100%	100%
Delay 1 Upper	94.30%	89.50%	93.80%
Delay 2 Lower	99.20%	100%	100%
Delay 2 Upper	96.20%	94.90%	96.00%
Delay S Lower	100%	100.00%	100.00%
Delay S Upper	96.60%	100.00%	100.00%
Total Yield	91.40%	89.30%	93.30%

Table 3.20. Comparisons of the results between the proposed method and Wicked.

a little percentage of feasibility for the delay 1 and delay 2, is able to get the 100% of delay symmetry. The optimal point found can be considered superior to the optimal point provided by the benchmark and the one found by the support vector machines. The only draw back is the computational expense to get a solution. As a matter of facts 18910 simulations are needed to find the optimal point.

The future research should be centered on testing the proposed methods on other circuits and integrating the SVM in the robust optimization strategy in order to get the best possible solutions in less time.

Conclusions

In this Ph. D. Thesis several surrogate models were presented, reformulated and applied on real world problems. For the future research we intend to continue developing the surrogate model approach, especially in robust optimization, and to create a complete algorithm where it is possible to achieve a reliable yield optimization on real consumer circuits.

From a more theoretical point of view, we intend to continue developing the canonical dual formulation for the multidimensional case of the Radial Basis Neural Networks and also develop a fast algorithm to find a solution. As we consider the canonical duality theory as a powerful tool for finding the global minimum of complex non-convex problems, our aim is also to expand this approach to problems that are difficult to solve with their current formulation. One of these example is the clustering problem, where the application of canonical duality could give birth to a faster a more reliable formulation and strategy to find a global solution of the problem.

Bibliography

[1] Ben-Tal, A, Nemirovski A. (1998) Robust convex optimization. Math. Oper. Res. 23, 769.

[2] Bertsimas D, Sim M. (2006) Tractable approximations to Robust conic optimization. Math. Progr. 107:5-36.

[3] Bertsimas D, Sim M. (2003) Robust discrete optimization and network flows. Math. Progr. 98, 49-71.

[4] Haykin S. (1999) Neural Networks, a Comprehensive Foundation, Prentice-Hall.

[5] Buzzi C, Grippo L, Sciandrone M. (2001) Convergent decomposition techniques for training RBF neural networks, Neural Computatio, 13:1891-1920.

[6] Gao DY (2000), Duality Principles in Nonconvex Systems: Theory, Methods, and Applications, Nonconvex Optimization and Its Applications, Kluwer Academic Publishers.

[7] Gao DY. (2009), Canonical duality theory: theory, method, and applications in global optimization. Comput. Chem. 33:1964-1972.

[8] Latorre V and Gao, DY. (2012) Dual Canonical Theory for one dimensional Radial Basis Neural Network Problems. to be submitted.

[9] Gao DY, Ruan N, Pardalos, P M. (2011). Canonical dual solutions to sum of fourth-order polynomials minimization problems with applications to sensor network localization, in Sensors: Theory, Algorithms and Applications, Springer.

[10] Gao, D Y, Watson L T, Easterling D R, Thacker W I, Billups S C. (2011) Solving the canonical dual of box- and integer-constrained nonconvex quadratic programs via a deterministic direct search algorithm, Optimization Methods & Software, DOI: 10.1080/10556788.2011.641125

[11] Gao D Y, Wu, C Z. (2012) On the triality theory for a quartic polynomial optimization problem, J. Industrial and Management Optimization, 8: 229-242.

[12] Santos H A F A, Gao D Y. (2012) Canonical dual finite element method for solving post-buckling problems of a large deformation elastic beam. Int. J. Nonlinear Mechanics, 47:240-247, doi:10.1016/j.ijnonlinmec.2011.05.012

[13] Zhang J, Gao, D Y, Yearwood J. (2011) A novel canonical dual computational approach for prion AGAAAAGA amyloid fibril molecular modeling. Journal of Theoretical Biology, Vol. 284:149-157. doi:10.1016/j.jtbi.2011.06.024

[14] Bruzzone L, Prieto D. (1998) Supervised training techniques for radial basis function neural networks. Electronic Letters, 34(11):1115-1116.

[15] Gao D Y. (2000) Canonical dual transformation method and generalized triality theory in nonsmooth global optimization. J. Glob. Optim. 17(1/4):127-160.

[16] Gao, D Y. (1997) Dual extremum principles in finite deformation theory with applications to post-buckling analysis of extended nonlinear beam theory. Applied Mechanics Reviews, 50, 11:S64-S71

[17] Moré J J, Wu Z J. (1997) Global continuation for distance geometry problems, SIAM Journal on Optimization, 7(3):814-836.

[18] Saxe J. (1979) Embeddability of weighted graphs in k-space is strongly NP-hard, in Proc. 17th Allerton Conference in Communications, Control, and Computing, Monticello, IL, 480-489.

[19] Wang Z B, Fang S C , Gao D Y , and Xing W X. Canonical dual approach to solving the maximum cut problem, to appear in J. Glob. Optim.

[20] Wettschereck D, Dietterich T. (1992), Improving the Performances of Radial Basis Functions Networks by Learning Center Locations Advances in Neural Information Processing Systems.

[21] S. Lucidi , V. Latorre , DFSVM: an interface for Derivate Free optimization of Support Vector Machines parameters in LIBSVM, to be submitted.

[22] Fasano G, Lucidi S.(2009) A nonmonotone trucated Newton-Krylov method exploiting negative curvature directions, for large scale unconstrained optimization, Optimization Letter, 3:521-535, .

[23] Sciandrone M, Lucidi S.(2002) A Derivative-free algorithm for bound constrained optimization, Computational Optimization and applications, 21:119-142.

[24] Sciandrone M, Lucidi S.(2002) On the global convergence of derivative-free methods for unconstrained optimization, SIAM J. Optim. , 13:97-116.

[25] Liuzzi G, Lucidi S, Rinaldi F (2012) Derivative-free methods for bound constrained mixed-integer optimization, Computational Optimization and applications, 53:505-526.

[26] Chang C-C, Hsu C-W, Lin C-J(2000) The analysis of decomposition methods for support vector machines, IEEE Transactions on Neural Networks, 11:1003-1008.

[27] Lucidi S, Palagi L, Risi A, Sciandrone M.(2009) A convergent hybrid decomposition algorithm model for SVM training. IEEE Trans. on Neural Networks, 20(5):1055-1060, .

[28] Chang C-C, Lin C-J (2001) LIBSVM: a library for support vector machines, Software available at `http://www.csie.ntu.edu.tw/~ cjlin/libsvm`.

[29] Lin C-J. (2001) On the convergence of the decomposition method for Support Vector Machines, IEEE Transactions on Neural Networks, 12:1288-1298.

[30] Lin C-J. (2002) Asymptotic convergence of an SMO algorithm without any assumptions, IEEE Transactions on Neural Networks, 13:248–250.

[31] Palagi L, Sciandrone M. (2005) On the convergence of a modified version of SVM^{light} algorithm, Optimization Methods and Software, 20:311–328.

[32] Serafini T., Zanni L. (2006) Parallel software for training large scale SVM on multiprocessor system, Journal of Machine Learning Research, 7:1467-1492.

[33] Freedman D A. (2005) Statistical Models: Theory and Practice, Cambridge University Press.

[34] Schölkopf B, Smola A (2002) Learning with Kernels, Support Vector Machines, Regularization, Optimization and beyond, The MIT Press.

[35] Momma M, Bennett K P. (2002)A Pattern Search Method for Model Selection of Support Vector Regression , Proceedings of SIAM Conference on Data Mining.

[36] Liuzzi G, Lucidi S, Sciandrone M. (2010) Sequential penalty derivative-free methods for nonlinear constrained optimization , SIAM Journal of optimization, vol 20(5):2614-2635.

[37] Lewis R M, Torczon V. (1999) Pattern Search Algorithms for Bound Constrained Minimization. SIAM J. on Optimization 9(4):1082-1099.

[38] Finlayson-Pitt J B, Pitts W J. (1986) Fundamental and Experimental Techniques. Atmospheric chemistry. John Wiley and Sons, Inc.,New York, Brisbane, Toronto, Singapore.

[39] Seinfeld J H. (1986) "Atmospheric Chemistry and Physics of Air Pollution", A Wiley-Interscience publication- John Wiley & sons.

[40] World Health Organisation, Health aspects of air pollution with particulate matter, ozone and nitrogen dioxide. Report on a WHO Working Group. Bonn, Germany, 13-15 January 2003.

[41] Comrie A C. (1997) Comparing Neural Networks and Regression Models for Ozone Forecasting, Air & Waste Manage. Assoc. 47:653–663, .

[42] Brunelli U, Piazza V, Pignato L, Sorbello F, Vitabile S. (2007) Two-days ahead prediction of daily maximum concentrations of SO_2,O_3,$PM10$,NO_2, CO in the urban area of Palermo, Italy, Atmospheric Environment, 41:2967-2995,

[43] Dutot L, Rynkiewicz J, Steiner F E, Rude J. (2007) A 24-h forecast of ozone peaks and exceedance levels using neural classifiers and weather predictions, Environmental Modelling & software, Vol 22:1261-1269.

[44] Lu W Z, Wang W J. (2005) Potential assessment of the "support vector machine" method in forecasting ambient air polluant trends, Chemosphere, 59:693-701.

[45] Derwent D G, Davies T J. (1994) Modelling the impact of NOx or hydrocarbon control on photochemical ozone in Europe. Atmospheric Environment 28(12):2039-2052.

[46] Hubbard M C, Cobourn W G. (1998) Development of a regression model to

forecast ground-level ozone concentration in Louisville, KY. Atmospheric Environment 32(14-15):2637-2647.

[47] Abdul-Wahab S A , Al-Alawi S.M. (2002) Assessment and prediction of tropospheric ozone concentration levels using artificial neural networks. Environmental Modelling & Software, 17:219-228.

[48] Frank E. Harrell. (2001) Regression Modelling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis, Springer-Verlag New York.

[49] Yelten M, Franzon P, Steer M (2011) Surrogate-Model-Based Analysis of Analog Circuits - Part I: Variability Analysis. IEEE Transactions on Device and Materials Reliability 11:458-465.

[50] Gorissen D, De Tommasi L, Hendrickx W, Croon J, Dhaene T. (2008) RF circuit block modelling via Kriging surrogates. In: 17th International Conference on Microwaves, Radar and Wireless Communications, MIKON 2008, pp. 1-4.

[51] Gorissen D, De Tommasi L, Crombecq K, Dhaene T (2009) Sequential modelling of a low noise amplifier with neural networks and active learning. Neural Comput. Appl. 18:485–494.

[52] You H, Yang M, Wang D, Jia X (2009) Kriging Model combined with latin hypercube sampling for surrogate modelling of analog integrated circuit performance. In: Proceedings of the 10th international symposium on Quality of Electronic Design, ISQED 2009, pp. 554–558.

[53] Sun W, Tay A, Vedantam S (2005) Simulation-based design optimization of solder joint reliability of wafer level copper column interconnects. In: Proceedings of 7th Electronic Packaging Technology Conference, EPTC 2005, 2:444–450.

[54] Liu B, Zhao D, Reynaert P, Gielen G (2011) Synthesis of Integrated Passive Components for High-Frequency RF ICs Based on Evolutionary Computation and Machine Learning Techniques. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 30:1458–1468.

[55] Myers R, Montgomery D, Anderson-Cook C (2009) Response surface methodology: process and product optimization using designed experiments. Wiley series in probability and statistics, Wiley, New York.

[56] MunEDA inc (2012) WiCkeD, a Tool Suite for Nominal and Statistical Custom IC Design. www.muneda.com/Products/

[57] Suykens J (2001) Nonlinear modelling and support vector machines. In: Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, IMTC 2001, 1:287–294.

[58] Mukherjee S, Osuna E, Girosi F (1997) Nonlinear prediction of chaotic time series using support vector machines. In: Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing VII, pp. 511–520.

[59] Müller K-R, Smola A, Rätsch G, Schölkopf B, Kohlmorgen J, Vapnik V (1997)

Predicting time series with support vector machines. In: Artificial Neural Networks, Lecture Notes in Computer Science 1327:999–1004, Springer, Berlin.

[60] MunEDA inc, WiCkeD Manual 6.4-3. (2010-2011).

[61] Boolchandani D, Garg L, Khandelwal S., Sahula V. (2010) Variability aware yield optimal sizing of analog circuits using svm-genetic approach. In Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD), XIth International Workshop on, pages 1 –6.

[62] Ciccazzo A, Dipillo G, Latorre V. Support vector machines for surrogate modeling of electronic circuits. To be published.

[63] Jing M E, Hao Y., Zhang J F, Ma P. J. (2005) Efficient parametric yield optimization of vlsi circuit by uniform design sampling method. Microelectronics Reliability, 45(1):155-162.

[64] Ilumoka A A. (1998) A modular neural network approach to microelectronic circuit yield optimization. Microelectronics Reliability, 38(4):571-580.

[65] Makridakis S, Wheelwright S C, Hyndman R J. (1998) Forecasting: Methods and Applications, John Wiley and Sons.

[66] Ansuj A P, Camargo M E, Radharamanan R, Petry D G. (1996) Sales forecasting using time series and neural networks, Computers Industrial Engineering, 31:421-424.

[67] Zhang G, Patuwo B E, Hu M Y. (1997) Forecasting with artificial neural networks: the state of the art, International Journal of Forecasting, 14:35-62.

[68] Alon I, Qi M, Sadowski R J. (2001) Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods, Retailing and Consumer Services, 8:147-156, .

[69] Levis A A, Papageorgiou L G. (2005) Customer demand forecasting via support vector regression analysis, Chemical Engineering and Design,83:1009-1018.

[70] Kuo R J, Xue K C. (1998) A decision support system for sales forecasting through fuzzy neural networks with asymmetric fuzzy weights, Decision Support Systems, 24:105-126.

[71] Chang P C, Liu C H, Fan C Y. (2009) Data clustering and fuzzy neural network for sales forecasting: a case study in printed circuit board industry, Knowledge-Based Systems, 22:344-355.

[72] Kuo R J, Hu T L, Chen Z Y. (2009) Application of radial basis function neural network for sales forecasting, Inter. Asia Conf. on Informatics in Control, Automation and Robotics, 325-328.

[73] Cui D, Curry D. (2005) Prediction in marketing using the support vector machine, Marketing Science, 24:595-615.

[74] Thiesing F M, Middelberg U, Vornberger O. (1995) Short term prediction of sales in supermarkets, IEEE Inter. Conf. on Neural Network Proceedings, 2:1028–1031.

[75] Thiesing F M, Vornberger O. (1997) Sales forecasting using neural networks, IEEE Inter. Conf. on Neural Network Proceedings, 4:2125-2128, .

[76] Wu Q, Yan H S, Yang H B. (2008) A forecasting model based on support vector machine and particle swarm optimization, Workshop on Power Electronics and Intelligent Transportation System, PEITS'08 Proceedings, 218-222.

[77] Zhang P, Breitkopf P, Knopf-Lenoir C, Zhang W. (2011) Diffuse response surface model based on moving Latin hypercube patterns for reliability-based design optimization of ultra-high strength steel NC milling parameters. Structural and Multidisciplinary Optimization 44:613-628.

[78] West P M, L. Brockett P, Golden L L. (1997) A comparative analysis of neural networks and statistical methods for predicting consumer choice, Marketing Science, 16: 370-391.

[79] Vapnik V, Cortes C. (1995) Support-vector networks, Machine Learning, 20:273-297.

[80] Bertsimas D, Nohadani O, Teo K M. (2010) Robust optimization for unconstrained simulation-based problems, Operation Research, 58:161-178.

[81] Selim S Z, Ismail M A. (1984) K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. IEEE trans. pattern analysis and machine intelligence PAMI-6:81-87.

[82] Bagirov A M, Rubinov A M, Soukhoroukova N V, Yearwood J. (2003) Unsupervised and Supervised Data Classification via Nonsmooth and Global Optimization, TOP 11:1-93.

[83] Teboulle M. (2007) A Unified Continuous Optimization Framework for Center-Based Clustering Methods, Journal of Machine Learning Research, 8:65–102.