Università degli Studi di Udine
Dipartimento di Matematica e Informatica
Dottorato di Ricerca in Informatica Ciclo xxv

Ph.D. Thesis

The PLS regression model: algorithms and application to chemometric data

Candidate:	Supervisor:
Del Zotto Stefania	prof. Roberto Vito

Anno Accademico 2012-2013

Author's e-mail: stefania.delzotto@uniud.it

Author's address:

Dipartimento di Matematica e Informatica
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine
Italia

To all the kinds of learners.

Abstract

Core argument of the Ph.D. Thesis is Partial Least Squares (PLS), a class of techniques for modelling relations between sets of observed quantities through latent variables. With this trick, PLS can extract useful informations also from huge data and it manages computational complexity brought on such tall and fat datasets. Indeed, the main strength of PLS is that it performs accurately also with more variables than instances and in presence of collinearity.
Aim of the thesis is to give an incremental and complete description of PLS, together with its tasks, advantages and drawbacks, starting from an overview of the discipline where PLS takes place up to the application of PLS to a real dataset, moving through a critical comparison with alternative techniques. For this reason, after a brief introduction of both Machine Learning and a corresponding general working procedure, first Chapters explain PLS theory and present concurrent methods. Then, PLS regression is computed on a measured dataset and concrete results are evaluated. Conclusions are made with respect to both theoretical and practical topics and future perspectives are highlighted.

The first part starts describing Machine Learning, its topics, approaches, and designed techniques. Then, a general working procedure for dealing with such problems is presented with its three main phases.
Secondly, a systematic overview of the PLS methods is given, together with mathematical assumptions and algorithmic features. In its general form PLS, developed by Wold and co-workers, defines new components by maximizing covariance between two different blocks of original variables. Then, measured data are projected onto the more compact latent space in which the model is built. The classical computation of the PLS is based on the Nonlinear Iterative PArtial Least Squares (NIPALS) algorithm; but, different variants have been proposed. The PLS tasks include regression and classification, as well as dimension reduction. Nonlinear PLS can also be specified. Since PLS works successfully with large numbers of strictly linked variables with a noticeable computational simplicity, PLS results in a powerful versatile tool that can be used in many areas of research and industrial applications.
Thirdly, PLS can be compared with concurrent choices. As far as regression is concerned, PLS is connected with Ordinary Least Squares (OLS), Ridge Regression (RR), Canonical Correlation Analysis (CCA). PLS can also be related with Prin-

cipal Component Analysis and Regression (PCA and PCR). Moreover, Canonical Ridge Analysis is a unique approach that collects all the previous methods as special cases. In addition, all these techniques can be cast under a unifying approach called continuum regression. Some solutions, as OLS, are the most common estimation procedures and have desirable features but at the same time they require strict constrains to be applied successfully. Since measured data of real applications do not usually comply needed conditions, alternatives should be used. PLS, RR, CCA and PCR try to solve this problem but PLS performance overcomes the others. As regards classification and nonlinearity, PLS can be linked with Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM). LDA as PLS is a supervised classification and dimensionality reduction technique but LDA has a limited representation ability and sets some assumptions that real data usually do not fulfill causing computation fails. SVM has some interesting properties that make it a competitor with respect to PLS but its advantages and drawbacks should be carefully analyzed. SVM allows to expand PLS approach comprising nonlinearity.

In the second part PLS is applied to a chemometric problem. The aim is to define a model able to predict the fumonisins content in maize from Near InfraRed (NIR) spectra.

A dataset collects NIR spectra of maize and their fumonisins (a group of mycotoxins) content. The NIR wavelengths are almost a thousand strictly collinear variables; observations are of the order of hundreds. A model is computed using PLS with full cross validation, after identifying the outliers among both the instances and the wavelengths. The results are promising as regards the correlation between measured and fitted values with respect to both calibration and validation. In addition, the relation between measured quantities is represented by a simpler model with about twenty latent variables and a decreasing residual variance. Moreover, the model assures screening ability with respect to the most important European legal limits, fixed as tolerable daily intake for fumonisins consumption of maize-based food. Furthermore, it can also correctly classify other groups of contaminated maize.

In conclusion, maize as well as maize-based feed and food are a common dietary staple for both cattle and people. However, fungi infection of maize is usually certified throughout the world. Moreover, fungi may produce mycotoxins and consumption of contaminated maize causes serious diseases. As a consequence, procedures to measure as early as possible fumonisins content in maize batches are highly needed in order to split up safely dangerous maize from healthy one. As a matter of fact, traditional techniques for measurements of mycotoxins are expensive, long, destructive, and require trained personnel, whereas NIR spectroscopy is cheaper, faster, simpler, and produce no waste materials. Therefore, PLS modelling techniques grounded on NIR spectroscopy data provide a quite promising methodology for fumonisins detection in maize. As usual, further systematic analyses are required in order to confirm conclusions and improve results.

Acknowledgments

During these years at the Department of Mathematics and Computer Science of the University of Udine I met people without whom my work would not have been possible. I would like to thank all of them for their insightful guidance and help.

I would like to express my sincere gratitude to my supervisor prof. Vito Roberto for his continuous support, precious advice and welcome suggestions that allowed me to improve myself and my studies, capabilities and results.

Words cannot describe how lucky I feel for the chance to work with prof. Giacomo Della Riccia. At the Research Center "Norbert Wiener" we developed together the statistical analysis and his enthusiasm and profound experience were essential to overcome any obstacle. He is a very nice person who delighted me with his incredible and amazing stories, memories of a life, that make my thoughts travel from Alexandria to Moscow, from Paris to Boston.

I also would like to thank my reviewers, professors Corrado Lagazio, Carlo Lauro and Ely Merzbach as well as professors Ruggero Bellio, Francesco Fabris, Hans Lenz and Paolo Vidoni for their invaluable feedback, encouragement and useful discussions.

I am grateful to the director of Ph.D. Course in Computer Science prof. Alberto Policriti, who always answered my questions kindly and carefully followed my student career step by step.

Furthermore, I would like to thank every professor of the courses I followed, not only were they helpful in lecturing they also gave excellent suggestions for final reports.

The described case study is a part of "Micosafe", a project funded by the Natural, Agricultural and Forestry Resources Office of Friuli Venezia Giulia Region, proposed and developed by the Department of Agricultural and Environmental Sciences of the University of Udine in cooperation with the International Research Center for the Mountain (CirMont) and the Breeders Association of FVG. I thank professors Bruno Stefanon and Giuseppe Firrao for choosing the Department of Mathematics and Computer Science as their partner for statistical analysis and for making data available. Brigitta Gaspardo, Emanuela Torelli and Sirio Cividino are the researchers with whom I collaborated. I would like to express my gratitude to them.

The fruitful and enjoyable experience of my studies could not have been accomplished without the daily presence of the people who love and live everyday with me, listening to me and sharing thoughts and feelings. I take this occasion to express to each of them my thanks.

My parents, Laura and Renato, sustain me and highlight my shortcomings to make me a better woman. Their life, their behaviour and their choices are the first visible example and model for me.

My brothers, Simone e Michele, fill my life with their "disorder": they make me happy when I am sad or angry, they speak aloud when I am too silent, they ignore my recurring useless comments.

Simone shares his life and time with me in all the possible ways. He understands me even without words, he sees deep inside me and he respects with endless perseverance me and my moody moments.

The presence of my extended family is also essential: grandparents, aunts and uncles, cousins and other relatives make me feel glad and we stay joyfully together.

Claudia and her group of teenagers warm my heart with their smiles, their hugs and their jokes.

The presence of many other people enriches my days. My friend Cristina spent many hours near me, always listening to my questions and trying to help me. My colleague Omar puts up my grumbles and still asks me to join him for a coffee break.

With Elena I have the most restful talks where she gives me the best advice for every kind of situation.

Furthermore, there are many other people that I cannot list here but whose friendship is essential for me, with whom I spend pleasant moments of spare time and who cannot ever be forgotten.

Contents

List of Figures

List of Tables

Introduction

Data collection and analysis are two key activities for understanding real phenomenons. Indeed, more data are available more knowledge about the system can be gained. However, this kind of tasks is usually expensive and time consuming, so that only a restricted number of quantities is often measured on a limited set of observations. Thus, it is a challenging problem extracting from such datasets useful informations whose content is effective and meaningful not only for those data but that also holds in a more general way for all the subjects involved in the phenomenon.

In order to manage successfully such critical situations, it is helpful to consider the twofold nature of real datasets. Indeed, data have usually an underlying regularity but they are also characterized by a degree of uncertainty. In particular, individual observations are corrupted by random noise, that arises from intrinsically stochastic processes but more typically it is due to sources of variability that are themselves unobserved. So, noise is caused by finite size of datasets as well as by measurements.

With this premises, the most efficient strategies and tools are supplied by disciplines as Statistics and Decision Theory, both grounded on Probability Theory. Probability Theory provides a coherent framework for expressing data uncertainty in a precise and quantitative manner, defining suitable probability distributions. Secondly, Statistics gives the most complete description of the probability distribution that generates both measured and future unknown instances trough several methods based on data analysis. In particular, statistical inference assumes that the phenomenon is governed by a random process and that the observations of the dataset are extracted through random sampling from the whole set of subjects. As a consequence, the data can be seen as a realization of the random process, or with other words that measured observations are independent draws from a hidden and fixed probability distribution. Thirdly, Decision Theory exploits this probabilistic representation in order to make choices that are optimal according to suitable criteria, given all available information, even though it is incomplete or ambiguous. Indeed, Decision Theory provides some useful methods that are able to decide either to do or do not an action given the appropriate probabilities considering that decision should be optimal in some appropriate sense.

Applying assumptions, concepts and techniques of the quoted domains, probabilistic algorithms can be designed to draw knowledge from data, with which subsequent actions may be done. Probabilistic algorithms find the best output for a given input, also including probability of result, after defining an explicit underlying probability model. They indeed produce a confidence value associated with their choice.

Core argument of the Ph.D. Thesis is Partial Least Squares (PLS) that belongs to the family of probabilistic algorithms. PLS was developed in order to extract useful information from large datasets characterized by an high number of strictly linked variables. In particular, PLS allows to describe qualitatively and quantitatively asymmetric relations with collinear quantities by means of new latent variables built from data. Since nowadays a wide range of tools tend to measure highly correlated quantities in several application domains, real datasets are frequently characterized by both collinearity and a number of observations lower than the cardinality of variables. As a consequence, PLS becomes more and more attractive and can be often helpful in many common concrete situations. In particular, PLS can be used alone but it can also be combined with traditional techniques. PLS indeed works firstly as a reduction dimensionality procedure, building a smaller set of uncorrelated latent variables, then it is usefully applied as regression or classification method. In this way, PLS involves computational simplicity and quickness. At the same time, accuracy of results is also assured. PLS could also manage nonlinearity covering all kind of possible request and becoming a complete and flexible tool.
Aim of the thesis is to describe PLS and to provide a review of some alternative competitors belonging to different fields. Before, an overview of Machine Learning is given presenting its approaches and corresponding techniques, together with the description of a whole working procedure. After, PLS performance is verified on a challenging dataset and its advantages and drawbacks are listed. Conclusions on PLS applicability to realistic problems are drawn, as well as reasons of the increasing interest towards PLS are highlighted and future perspectives outlined.
Regarding the evaluation of PLS applied to the considered case study, a preliminary analysis on the real dataset and its partial results are described in Gaspardo et al. (Gaspardo, 2012 [26]). Then, since those outcomes seemed to be promising and additional observations completed the dataset, further researches were done and corresponding results presented in an article submitted to Food Chemistry (Della Riccia and Del Zotto [17]). Comparing these two works, the last one improves the most important parameters that assess goodness and qualities of a model, assuring that the final result satisfies screening, classification and rather accurate prediction.
Since the described work is part of a project developed in collaboration with Friuli Venezia Giulia Region and its breeders association, researchers give prominence to promote results and good practices also in favour of people directly involved into the productive process. For this reason, they took part to local conferences and wrote specific guidelines for making aware as many farmers and breeders as possible (Gaspardo and Del Zotto, 2011 [25]).

The first part of the thesis opens with an overview of Machine Learning that defines its topics and lists the most common techniques developed in the corresponding approaches. Then, focus is made on supervised learning, the particular setting where PLS takes place. Thus, a general supervised working procedure is explored, con-

sidering all its phases. Model fitting, model selection and decision stage are indeed illustrated following an intuitive incremental explanation.
Successively, a formal description of PLS method is provided including its mathematical structure, algorithmic features and geometric representation. PLS tasks are then evaluated, so that PLS and its behaviour in several situations can be compared with alternative techniques in the following. Some PLS applications in different domains are also presented as well as a review of the main softwares that include tools for PLS analysis.
Thirdly, some statistical methods, that have the same goals and tasks of PLS, are briefly explained. The aim is to bring all these techniques together into a common framework attempting to identify their similarities and differences. First of all, Ordinary Least Squares is described, since it is the traditional estimation procedure used for linear regression and because it can be seen as the benchmark for all the following techniques. Then, PLS is related to Canonical Correlation Analysis where latent vectors with maximal correlation are extracted. Moreover, Ridge Regression, Principal Component Analysis and Principal Component Regression are presented. As regression procedures, all these techniques can be cast under a unifying approach called continuum regression. Furthermore, as regards classification, there is a close connection between PLS and Linear Discriminant Analysis.
The first part of the thesis ends with a Chapter concerning Support Vector Machine (SVM) that presents intuitive ideas behind its algorithmic aspects, summarize briefly its historical development, and lists some applications. General theory of SVM is described in order to define the most important concepts that are involved in each form of SVM algorithm. Even if the chosen approach considers the simplest case of binary output, it can be easily generalized to more complex situations. Three specific procedures are illustrated step by step. Distinction between linear and nonlinear classification is done. Moreover, two different possibilities for linear classificators are analyzed, i.e. when they are applied on linearly separable data and on nonlinearly separable data. Finally, examples of nonlinear kernel PLS are given. They complete both approaches of nonlinear PLS regression with kernel-based technique of SVM.

The second part of the thesis starts with a preliminary description of the main concepts involved in the considered PLS application on real data. A gradual but complete overview of applicability area is indeed reported, introducing all topics and showing reasons that are behind this kind of problems and that support further analyses. Then, issues of interest are presented, illustrating in a simple way fungi and mycotoxins, their specific features together with the most troublesome factors related to their development and their management strategies. Details are given explaining core arguments, Fusarium and fumonisins, and their critical aspects. Finally, applied measurement processes are described that include both traditional reference analytical methods and some alternatives based on spectroscopic techniques and chemometric analysis.

After learning this background, the specific PLS application can be presented. In this case, PLS is iteratively computed in order to define the relationship between fumonisins content in maize and its Near InfraRed (NIR) spectra. Goal of the research is indeed the discovery of an accurate, fast, cheap, easy and nondestructive method able to detect fumonisins contamination in maize. First of all, data are described explaining procedures of data collection, that involve sampling and applied measurements methods. Variable of interest, that consists of fumonisins contamination level of maize, is also analyzed. The final result is shown and presents the best model obtained during regression analysis. Both numerical values and graphical outputs are given and discussed. Finally, some comments are made in order to highlight the most important aspects that should be more accurately taken into account in future.

The thesis ends with a critical review of the main developed concepts that allows to understand which future perspectives can be expected in the mid or long period for described methods. In particular, conclusions present core arguments of every previous Chapter, explaining requirements that they need to be correctly applied, problems they solve, strengths and weaknesses that characterize them. Moreover, conclusions remind the most interesting and successful applications, developed until now, of defined techniques and methodologies, and which changes, improvements, further studies and new ideas can be applied in the future in order to reach better results. This guide lines hold both for the theoretical background and the real case study.

I

First Part

1
Preliminaries

This Chapter consists of two parts that make an overview of Machine Learning, the discipline where Partial Least Squares (PLS) the core argument of the thesis take place, introducing relative terms, questions and assumptions. First of all, Machine Learning is defined presenting topics developed in this field and listing the techniques designed to solve corresponding problems (Section 1.1). In particular, focus is made on the two most common approaches of Machine Learning, thus unsupervised and supervised learning are described with more details. Secondly, a general working procedure for dealing with supervised learning problems is summarized step by step, explaining in a simple way the most important concepts that are involved with (Section 1.2). Linear regression is then chosen as the reference setting to work with, even if classification is also taken into account. So, three main phases of every typical supervised learning procedure are presented: model fitting, model selection and decision stage are indeed illustrated following an intuitive approach that would be as more helpful as possible.

Incremental explanation followed in this Chapter allows to understand gradually how to satisfy ultimate purpose of supervised learning that provides the general background for the thesis. For this reason, topics and results of this Chapter are recalled in the following where references to concepts here described are given on footnotes.

1.1 Machine Learning Taxonomy

Machine Learning belongs to Computer Science where it was born as a branch of Artificial Intelligence[1] during sixties. It has then undergone substantial development over the past twenty years, so that its approach is nowadays easily and widely

[1] Artificial Intelligence was defined as “the science and engineering of making intelligent machines” (J. McCarthy that coined the term in 1956 [43]). Indeed, it is based on the idea that human intelligence can be precisely described and simulated by a computer. As a consequence, it studies how to make machines behave like people, reproducing human activities and brain processes. For this reason, it deals with intelligent agent, a system that perceives its environment and takes decisions, in order to do actions that maximize its chances of success. Since learning is a kind of human activity, Machine Learning can be considered as belonging to Artificial Intelligence.

adopted in many applications in order to obtain far better results. Machine Learning is concerned with "design and development of systems able to automatically learn from experience with respect to a task and a performance measure" (T. M. Mitchell, 1997). Indeed, it aims to define computer programs, called learners, whose performance at defined task improves with experience. In particular, a learner should be able to process past observations, called instances and collected in the training set, with a twofold ability: to describe data in some meaningful way (representation) and to perform accurately on new, unseen examples (generalization).

Considering this simple and general definition, Machine Learning algorithms and methods turn out as tools based on data collection and analysis. As a consequence, Machine Learning can be strictly related with both Statistics and Data Mining[2], whose techniques seek to summarize and explain key features of the data. Indeed, core arguments of all these disciplines usually overlap significantly. For instance, modern neural networks are a typical Machine Learning technique, but at the same time they can be seen also as a non-linear statistical data modeling tool. Moreover, clustering is maybe the most interesting example since it is i) a method of unsupervised learning, ii) an iterative process of exploratory Data Mining, and iii) a common technique for statistical data analysis. Similarly, unsupervised learning, that usually improves learner accuracy, is closely related to the problems dealt in descriptive Statistics but is also based on Data Mining methods used to preprocess data. However, although problems are often formally equivalent, different aspects of solutions are emphasized in the specific fields. This leads sometimes to misunderstandings between researchers, because they employ the same terms, algorithms, and methods but have slightly different goals in mind.

This awkward situation across disciplines, characterized by words that mean different techniques or by the same method called with many names, should be carefully managed in order to be clear and to avoid errors. For this reason, Machine Learning is taken as reference in the following except where otherwise noted.

With these premises, the wide variety of Machine Learning approaches, tasks and successful applications can now be investigated. In particular, Machine Learning algorithms can be organized focusing on their outcome or on the type of data available during training the machine. Following this idea, five areas can be highlighted: unsupervised learning, supervised learning, semi-supervised learning, reinforcement learning and learning to learn.

The last one induces general functions from previous experience. Indeed, this type of algorithms can change the way they generalize.

In reinforcement learning[3], the machine interacts with its environment by producing

[2] Data Mining is the analysis step of Knowledge Discovery in Databases and deals with the extraction of unknown properties from the data. On the contrary, Machine Learning focuses on prediction, based on properties learned from the training data (so in some sense known).

[3] Reinforcement learning (term coined by Sutton and Barto in 1998 [66]) is closely related to the fields of both Decision Theory, that belongs to Statistics and Management Science, and Control

actions. These actions affect the state of the environment, which in turn provides feedback to the machine in the form of scalar rewards (or punishments) that guides the learning algorithm. The goal of the machine is to learn to act in a way that maximizes the future rewards it receives (or minimizes the punishments) over its lifetime.

Semi-supervised learning combines both instances characterized by supervised properties and data with unsupervised nature to generate an appropriate function or classifier.

Supervised learning is the task of producing a function from labelled training data. Indeed, here the dataset collects for every instance a pair consisting of both an input and an output. An input is a vector that records the sequence of measurements done on the individual properties of the observation. Each property is usually termed feature[4], also known in Statistics as explanatory variable (or independent variable, regressor, predictor). An output or target stands for a category or a continuous real value and is called label, because it is often provided by human experts hand-labelling the training examples. In Statistics, labels are known as outcomes which are considered to be possible values of the dependent or response variable. So, supervised learning algorithms process labelled training data in order to produce a function able to accurately assign a label to a given input for both observed and future instances.

On the other hand, unsupervised learning refers to the problem of trying to find hidden structures in unlabelled data and includes any method that seeks to explain key properties of the data. In this case, data record only input.

Unsupervised and supervised learning are the most common and broad Machine Learning subfields. Thus, for their prominence, they will be explained thoroughly in the following where the corresponding taxonomy of the algorithms is described with more details and is also most clearly shown in the tables.

1.1.1 Unsupervised learning

Unsupervised learning includes two tasks that consist of cluster analysis and dimension reduction (Table 1.1).

Cluster analysis, also called clustering, aims to group sets of similar objects in the same cluster. So, cluster is a group of akin data objects built in such a way that objects in the same cluster are more similar, in some sense, to each other than to those in other groups. In this way, both the notion of cluster and of similarity have a

Theory, a branch of Engineering.

[4] The term "feature" is also used in feature extraction, a set of techniques that follow an unsupervised approach. In this case, features denote new quantities computed from original measurements. These two slightly different concepts should not be confused. For this reason, in the thesis feature is only used referring to feature extraction whereas variable is preferred in the more general dissertation.

general definition. As a consequence, different cluster models can be employed and many clustering algorithms exist. However, some of them have often some properties in common since they are developed following the same criteria. Taking these characteristics as guide lines, subsets of clustering algorithms can be identified as described hereinafter.

First of all, according to the strategy, clustering can be distinguished in bottom-up (agglomerative) or top-down (divisive) techniques. The last assumes that at the beginning all the elements belong to the same cluster. Then, algorithm breaks this group in more subsets in order to obtain more homogeneous clusters. The algorithm ends when it satisfies a stopping rule, usually when it reaches the previously defined number of clusters. The former starts supposing that every element is a cluster. Then, the algorithm joins near clusters until it satisfies the stopping rule.

Secondly, the relationship of membership between object and cluster can be taken into account. Exclusive clustering, also called hard clustering, assigns each object to a single cluster, so that resulting clusters cannot have elements in common. On the contrary, overlapping clustering (also called soft clustering) reflects the fact that an object can simultaneously belong to more than one group. It is often used when, for example, an object is between two or more clusters and could reasonably be assigned to any of these clusters. In fuzzy clustering each object belongs to each cluster with a certain degree that is between 0 (it absolutely does not belong) and 1 (it absolutely belongs). Usually an additional constraints assures that the sum of weights for each object must equal one. The Fuzzy c-means (FCM) is the most common algorithm of this kind. This approach can be converted to an exclusive one by assigning each object to the cluster in which its weight is the highest.

Thirdly, considering the covering of the dataset, the complete clustering assigns every object to a cluster, whereas a partial clustering does not because some observations may not belong to well-defined groups.

However, the most important types of algorithms can be characterized according to the procedure they follow for defining to which cluster an element belongs. In this case centroid based, hierarchical and probabilistic clustering can be recognized. The former is based on the use of the distance between the object and a representative point of the cluster, having prefixed the number of clusters. This method generalizes the most famous algorithm of clustering called k-means. On the other hand, the second builds a series of clusters according to a statistic that quantifies how far apart (or similar) two cases are. Then, a method for forming the groups should be selected and the number of clusters decided. Hierarchical clustering can be illustrated by a dendrogram and it can be either agglomerative or divisive. The most used algorithm of this type is exactly called hierarchical algorithm. The last technique is based on a completely probabilistic approach that assumes a model-based strategy, which consists in using certain models for clusters and attempting to optimize the fit between the data and the model. In practice, each cluster can be mathematically represented by a parametric distribution, either continuous or discrete. The entire data set is therefore modelled by a mixture of these distributions. An individual distribution used to model a specific cluster is often referred to as a component distribution.

The other task of unsupervised learning, dimension reduction, is the process of shrinking the number of variables under consideration. Dimension reduction is required when performing analysis of complex data, because in this case one of the major problems stems just from the number of properties. Indeed, a dataset with many variables generally involves either a large amount of memory and computation power, or an algorithm which overfits the training set and generalizes poorly to new observations. Dimension reduction gets around these problems assuring that data analysis can be done more accurately in the reduced space than in the original one. In particular, it simplifies the amount of resources required for a large dataset while still describing it properly. As a consequence, it implies a shorter training time, it improves model interpretability, and it enhances generalization by reducing overfitting. Moreover, it is also useful as part of the data analysis process, showing which variables are important for prediction, and how these variables are related. Dimension reduction is a general term that includes several methods and it covers both Variable Subset Selection (VSS) and feature extraction. The former returns a subset of the original variables, whereas the last transforms the data from the original high-dimensional space to a space of fewer dimensions, called features.
VSS is also called variable selection or attribute selection, but feature selection is used, too. In this case feature stands for original variable and has a different meaning with respect to its use in feature extraction. A VSS algorithm is a combination of a search technique for proposing new variables subsets, together with an evaluation measure which scores the different subsets. The evaluation metric heavily influences the algorithm and it distinguishes between the three main categories of VSS algorithms: wrappers, filters and embedded methods. The former uses a predictive model to score variable subsets. Each new subset is used to train a model, which is tested on a hold-out set. Counting the number of mistakes made on that hold-out set (the error rate of the model) gives the score for that subset. As wrapper methods train a new model for each subset, they are very computationally intensive, and have the risk of overfitting, but usually provide the best performing feature set for that particular type of model. The second uses a proxy measure to score a variable subset. This measure is chosen to be fast to compute, whilst still capturing the usefulness of the variable set. Filters are usually less computationally expensive than wrappers, but they produce a variable set which is not tuned to a specific type of predictive model. Many filters provide a variable ranking rather than an explicit best variable subset, and the cut off point in the ranking is chosen via cross-validation. The last is a wide group of techniques which perform variable selection as part of the model construction process. The exemplar of this approach is the Lasso method for constructing a linear model, which penalizes the regression coefficients, shrinking many of them to zero. Any variable which have non-zero regression coefficients are selected by the Lasso algorithm. These approaches tend to be between filters

and wrappers in terms of computational complexity. In Statistics, the most popular form of variable selection is stepwise regression. It is a greedy algorithm that adds the best variable (or deletes the worst one) at each round. The main control issue is deciding when to stop the algorithm and this is done optimizing some criteria or, in machine learning, with cross-validation. Since this leads to the inherent problem of nesting, more robust methods have been explored, such as branch and bound and piecewise linear network.

As previously said, feature extraction builds new features from functions of the original variables. Indeed, when the input data is too large to be processed and it is suspected to be redundant or irrelevant, then the input data can be transformed into a reduced representation set of features. If the features extracted are carefully chosen, it is expected that the features set will extract the relevant information from the input data in order to perform successfully the desired task using this reduced representation instead of the full size input. The data transformation is usually linear, but many nonlinear techniques also exist, as the nonlinear dimensionality reduction. Principal Component Analysis (PCA) is the main linear technique for dimensionality reduction. PCA is strictly related to the statistical method of factor analysis even if they are not identical because the last uses regression modelling techniques (supervised approach), while the former belongs to descriptive Statistics (unsupervised approach). Factor analysis seeks to discover if the observed variables can be explained largely or entirely in terms of a much smaller number of latent factors. Indeed, when the variations in the observed variables mainly reflect the variations in fewer unobserved factors, variability among observed correlated variables can be described in terms of a potentially lower number of unobserved factors. In particular, factor analysis searches for such joint variations in response to unobserved latent factors. The observed variables are so modeled as linear combinations of the potential factors, plus error terms. Then, the information gained about the interdependencies between observed variables can be used later to reduce the set of variables in the dataset. Computationally, this technique is equivalent to low rank approximation of the matrix of observed variables. Among other tasks, Partial Least Squares (PLS) can be used also as a linear feature extraction technique that, as factor analysis, follows a supervised approach.

Table 1.1: A taxonomy for unsupervised Machine Learning methods. Synonyms and acronyms are written between brackets; the most used algorithms are included after an arrow.

Task	Criterium	Technique
Cluster Analysis (clustering)	Strategy	Bottom-up (agglomerative) Top-Down (divisive)
	Membership	Exclusive (hard) Overlapping (soft) Fuzzy → Fuzzy c-means (FCM)
	Covering	Complete Partial
	Procedure	Centroid-based → k-means Hierarchical → hierarchical Probabilistic → mixture of Gaussians
Dimension reduction	Variable Subset Selection[1] (VSS)	Filter Wrapper Embedded methods → Lasso
	Feature extraction[2]	Linear → Principal Component Analysis (PCA) Nonlinear → Nonlinear dimensionality reduction

[1] In Statistics, the most popular form of VSS is stepwise regression. However, since it leads to the inherent problem of nesting, more robust methods have been explored, such as branch and bound and piecewise linear network.

[2] Factor analysis is a statistical method for feature extraction even if it does not follow an unsupervised approach. Similarly, also PLS can be used as a supervised linear feature extraction method.

1.1.2 Supervised learning

As previously said, supervised learning is based on the assumption that input data are labelled. Depending on the type of label, that can represent one of a finite number of discrete categories or that can be a continuous real value, supervised learning consists respectively of classification and regression (Table 1.2).

Classification is the problem of identifying to which of a set of classes a new instance belongs, on the basis of a training set containing observations whose category membership is known. Indeed, in Machine Learning the possible categories are called classes and an algorithm that implements classification is known as a classifier. However, the term classifier sometimes also refers to the mathematical function, implemented by the algorithm, that maps input data to a category.
First of all, classification can be thought of as two separate problems: binary classification and multiclass classification. In binary classification, a better understood task, there are only two classes, whereas multiclass classification involves assigning an object to one of several classes. Since many classification methods have been developed specifically for binary classification, multiclass classification often requires the combined use of multiple binary classifiers.
Secondly, a large number of algorithms for classification can be phrased in terms of a linear function that assigns a score to each possible category by combining the input vector of an instance with a vector of weights, using a dot product. The predicted category is the one with the highest score. This type of score function is known as a linear predictor function. Algorithms with this basic setup are known as linear classifiers. The most widely used linear classifiers are Support Vector Machines (SVM), perceptron, which is one of the simpler machine learning algorithms, and neural network that is a multi-layer perceptron. On the other hand, nonlinear classification is defined by algorithms with a nonlinear predictor function.
Thirdly, a common subset of classifiers is composed by probabilistic algorithms. This classification uses statistical inference to find the best class for a given instance. Unlike other algorithms, which simply output a class, probabilistic one output a probability of the instance being a member of each of the possible classes. Then, the best class is normally selected as the one with the highest probability. Probabilistic algorithms have numerous advantages over nonprobabilistic classifiers. Indeed, it can output a confidence value associated with its choice (in general, a classifier that can do this is known as a confidence-weighted classifier). Correspondingly, it can abstain when its confidence of choosing any particular output is too low. At the end, because of the probabilities output, probabilistic classifiers can be more effectively incorporated into larger machine-learning tasks, in a way that partially or completely avoids the problem of error propagation. The opposite group is called nonprobabilistic classification.
Fourth, classification algorithms can be parametric or nonparametric. A technique is nonparametric when it does not make any assumptions on the underlying data

distribution. This is pretty useful, as in the real world, most of the practical data does not obey the typical theoretical assumptions made. The k-Nearest Neighbour (KNN) is one of the most simple and widely used Machine Learning algorithms and it is a nonparametric learning algorithm.
In Statistics, logistic regression, that assumes a Logit Model, is the most common classification technique for binary outputs. It links the outcome's probability to the linear predictor function through the logit. The logit, named also log-odds, is the inverse of the sigmoidal logistic function and it is equal to the natural logarithm of the odds. The odds are defined as the probability of an output divided by the probability of the other label. When the dependent variables take ordinal multiple outputs, logistic regression is extended in Ordered Logit (or ordered logistic regression). Similarly, a Probit Model is a classification method for binary dependent variables that uses a probit link function. The probit function is defined as the cumulative distribution function of a standard normal (Gaussian) random variable. Ordered Probit is the corresponding generalization to the case of more than two classes of an ordinal dependent variable. Moreover, Linear Discriminant Analysis (LDA) and Fisher Discriminant Analysis (FDA) can also be applied. The terms LDA and FDA are often used interchangeably, although Fisher's original article actually describes a slightly different method, which does not make some of the assumptions of LDA such as normally distributed classes or equal class covariances (Fisher, 1936 [23]).

Regression is the supervised learning task that should be applied when labels are continuous real values. In this case, supervised learning algorithms process the training data to produce a function that assigns the correct continuous label to any valid input for both observed and future instances. The function is called regression function. Regression is maybe the task where the disciplines of Machine Learning and Statistics overlap more clearly. Indeed, here supervised algorithms coincides with statistical methods. As a consequence, for this specific approach, language used and concepts described encompass both fields.
A large body of techniques for carrying out regression analysis has been developed and there are several criteria for distinguishing them. First of all, regression methods can be parametric or nonparametric. The former assumes that the regression function is defined in terms of a finite number of unknown parameters that are estimated from the data. On the contrary, the last refers to techniques that allow the regression function to lie in a specified set of functions, which may be infinite-dimensional. Indeed, in nonparametric regression the functional form is not known and so can not be parameterized in terms of any function. Since the model does not take a predetermined form, it is constructed according to information derived from the data. As a consequence, it requires a large number of observations, because the data must supply the model structure as well as the model estimates, and is computationally intensive. Two of the most commonly used approaches to nonparametric regression

are smoothing splines and kernel regression. Smoothing splines minimize the sum of squared residuals plus a term which penalizes the roughness of the fit, whereas kernel regression involves making smooth composites by applying a weighted filter to the data. Kernel regression indeed estimates the continuous dependent variable from a limited set of data points by convolving the data points' locations with a kernel function.

Then, considering parametric techniques and focusing on the type of regression function, linear and nonlinear regression can be distinguished. The former assumes that the relationship between output and input is a linear combination of the parameters. The methods that belong to this family are called General Linear Models (GLM) and are the most common approach for regression. On the contrary, in nonlinear regression observational data are modeled by a function which is a nonlinear combination of the model parameters. The data are then fitted by a method of successive approximations.

Thirdly, there are different regression methods considering the nature of independent variables. For example, if the input vector collects categorical values, Analysis of Variance (ANOVA) should be applied. In its simplest form, ANOVA provides a statistical test of whether or not the means of several groups are all equal, and therefore generalizes t-test to more than two groups.

Furthermore, regression techniques can relate the response variable to one single explicative, this is the case of simple regression, or to more than one predictors in the case of multiple regression. Combining this last property with linearity a very wide family of techniques can be obtained called Multiple Linear Regression (MLR).

Similarly, if there is only one output, univariate regression is defined; whereas in presence of two or more response, multivariate regression is used.

Finally, regression techniques can be distinguished considering the applied estimation procedure. Usually the parameters of a linear regression model are estimated using the method of Ordinary Least Squares (OLS). But this method is based on the computation of the inverse of the explicative covariance matrix. Thus, input data are required to be linearly independent, i.e. neither variables nor instances should be equal to a linear combination of some other variables or instances, so that both the regressor and the covariance matrix have full rank and the inverse exists. Moreover, OLS assumes that the error terms should have null mean, constant variance (homoscedasticity) across observations and should be uncorrelated between them. These conditions are usually fulfilled looking for independent and identically distributed (i.i.d.) data. This means that all observations are taken from a random sample which makes all the assumptions listed earlier simpler and easier to interpret. Therefore, if and only if data abide these rather strong assumptions, OLS can be applied and parameters estimated assuring their good properties. But often real data do not comply OLS requirements, so that OLS cannot be applied expecting robust estimates. In this case, some other technique should be chosen. First of all, if the outcomes are uncorrelated but do not have constant variance (heteroscedastic data), weighted least squares should be used. They minimize a sum of weighted squared residuals where each weight should ideally be equal to the inverse of the variance of the observation, but weights may be also recomputed on each iteration, in an iteratively weighted least squares algorithm. Then, if the predictors or the instances are not linearly independent, dataset is characterized by collinearity. Traditional solution to this problem are Ridge Regression (RR) that includes a regularization term in the minimization problem. But some other methods exist that try to solve the fitting problem in presence of collinearity. These are for example PLS, PCR, CCA or Lasso. All these methods are described with more details in the thesis.

Decision tree is a very common tool used in both Machine Learning and Statistics and Data Mining in order to describe data. It can be applied for both classification and regression so that it is also called classification tree or regression tree, respectively.

This technique uses a dendrogram as a model in order to predict outcomes by starting at the root of the tree and moving through it until a leaf node. Indeed, in the tree structure, each interior node corresponds to one of the input variables. Then, there are edges to children for each of the possible values of that input variable. So, each leaf represents a label given the values of the input variables represented by the path from the root to the leaf.

Decision tree has various advantages. Firstly, it is simple to understand and to apply after a brief explanation. Secondly, it requires little data preparation without needing data normalization, dummy variables and removal of blank values. Thirdly, it is able to handle both numerical and categorical data, whereas other techniques usually work with only one type of variable. Then, decision tree can be validated using statistical tests, so that the reliability of the model can be accounted. It is also robust, performing well even if its assumptions are somewhat violated by the true model from which the data were generated. Moreover, it works well with huge datasets in a short time, because large amounts can be analyzed using standard computing resources. However, decision tree has also some limitations. The problem of learning an optimal decision tree is known to be NP-complete even for simple concepts. Consequently, practical decision tree are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node. But such algorithms cannot guarantee to return the global optimum. Moreover, decision tree can create over-complex trees that do not generalize the data well. So, decision tree can be characterized by overfitting. In this case, mechanisms such as pruning are necessary. In addition, there exist concepts that are hard to learn because decision trees do not express them easily. In such cases, the decision tree becomes prohibitively large. Solutions involve either changing the representation of the domain or using learning algorithms based on more expressive representations. Furthermore, for data including categorical variables with different numbers of levels, information is biased in favour of those attributes with more levels.

Table 1.2: A taxonomy for supervised Machine Learning methods. Synonyms and acronyms are written between brackets; the most used algorithms are included after an arrow.

Task	Criterium	Technique
Classification[1;2] (categorical label)	N. of classes	Binary Multiclass
	Function	Linear classifier → Support Vector Mach. (SVM) → Perceptron → Neural network Nonlinear classifier
	Strategy	Probabilistic Nonprobabilistic
	Distribution	Parametric Nonparametric → k-Nearest Neighbours (KNN)
Regression[2] (continuous label)	Form	Parametric Nonparametric → Smoothing splines → Kernel regression
	Function	Linear → General Liner Model (GLM) Nonlinear
	Input	Categorical → Analysis of Variance (ANOVA) Continuous
	N. of inputs	Simple Multiple → Multiple Linear Regression (MLR)
	N. of outputs	Univariate regression Multivariate regression
	Estimation	i.i.d. → Ordinary Least Squares (OLS) Heteroscedasticity → Weighted least squares → Ridge Regression (RR) → Principal Component (PCR) Collinearity → Partial Least Squares (PLS-R) → Canonical Corr. Analysis (CCA) → Lasso

[1] In Statistics, classification for binary dependent variables is done using Logit and Probit models. In the case of ordered multiclass labels, Ordered Logit and Ordered Probit models can be applied. Moreover, Linear Discriminant Analysis (LDA) and Fisher Discriminant Analysis (FDA) can also be alternative methods.

[2] Decision trees are powerful and popular tools for both classification and prediction.

1.2 Working methodology

After the overall description of Machine Learning with its definition, its relations with other disciplines, its subfields and their techniques, this Section focuses on supervised learning. Here indeed every step of a complete procedure for dealing with such problems is explained gradually in order to define more specifically supervised learning and to better understand its tasks and aims.

Supervised learning, as previously said, consists of analyzing labelled training data to learn a function able to accurately relate a label to a given input for both observed and future instances. Usually, it is assumed that the training set collects n instances where every entry is denoted by a k-dimensional vector $\{\mathbf{x}_i\}_{i=1}^n$ that carries on input. Then, in the most general case, labels are expressed using a d-dimensional vector $\{\mathbf{y}_i\}_{i=1}^n$. Since the final result and the whole algorithm depend on the data, an analysis of their nature should be made firstly.

Real datasets are characterized by a twofold nature that makes supervised learning someway more challenging. Indeed, data have usually an underlying regularity but individual observations are also corrupted by random noise. First, regularity of data has to be learned. In the simplest case, considering a single continuous target variable and a parametric approach[5], this can be done by directly constructing an appropriate function

$$\mathbf{y} = f(\mathbf{X}, \mathbf{b}) \tag{1.1}$$

where $\mathbf{y}$ is the n-dimensional vector that collects values of n training observations with respect to a single target variable, $\mathbf{X}$ is the $(n \times k)$ matrix of input values and $\mathbf{b}$ is the k-dimensional vector of coefficients.

Second, noise is caused by finite size of datasets as well as by measurements, since it arises from intrinsically stochastic processes but more typically it is due to sources of variability that are themselves unobserved.

Considering both elements, one can see data as generated from a structural component with random noise included in target values. So, for a given $\{\mathbf{x}_i\}_{i=1}^n$ there is uncertainty about the appropriate value for $\{y_i\}_{i=1}^n$. This means that previous result of Equation 1.1 is incomplete and should be widen adding a term in the following way

$$\mathbf{y} = f(\mathbf{X}, \mathbf{b}) + \mathbf{e} \tag{1.2}$$

where $\mathbf{e}$ is a random variable that represents noise. Equation 1.2 is called adaptive model whereas curve fitting consists of defining the structural component, choosing the form of the function and tuning its parameters, that include also some unknown amounts related to random noise. When Equation 1.2 shares the property of being a linear function of the adjustable parameters $\mathbf{b}$, the resulting broad class is defined linear regression models. All the following results refer to this specific case. Their

[5] With these assumption the following discussion refers and is restricted to univariate regression, unless otherwise specified.

simplest form involves a linear combination of input variables $\{\mathbf{x}_j\}_{j=1}^{k}$

$$\mathbf{y} = b_0 + b_1\mathbf{x}_1 + \cdots + b_j\mathbf{x}_j + \cdots + b_k\mathbf{x}_k + \mathbf{e}$$

so that they are also linear functions of input quantities. However, a much more useful class can be obtained by taking linear combinations of a fixed set of nonlinear functions of input variables, known as basis functions and denoted with $\phi_h(\mathbf{X})$. By using nonlinear basis functions, model is a nonlinear function of input vector. Nevertheless, such models are linear functions of parameters, which gives them simple analytical properties. Although it also leads to some significant limitations as practical techniques for Machine Learning, particularly for problems involving input spaces of high dimensionality, linearity in parameters will greatly simplify analysis of this class of models, that form the foundation for more sophisticated results. Such models are generally formalized as

$$\mathbf{y} = b_0 + \sum_{h=1}^{m-1} b_h\phi_h(\mathbf{X}) + \mathbf{e}$$

where, by denoting the maximum value of index h by $m-1$, total number of parameters in this model will be m. Parameter b_0 allows for any fixed offset in data. It is often convenient to define an additional dummy basis function $\phi_0(\mathbf{X}) = 1$ so that

$$\mathbf{y} = \sum_{h=0}^{m-1} b_h\phi_h(\mathbf{X}) = \phi(\mathbf{X})\mathbf{b} + \mathbf{e} \qquad (1.3)$$

As regards input variables that enter into a model, some form of fixed preprocessing are usually applied to original predictors for most practical applications of Machine Learning. Indeed, data are preprocessed to transform them into some new space of variables where, it is hoped, Machine Learning problem will be easier to solve. For instance, data are typically translated and scaled and this greatly reduces variability within data. When location and scale of all data are the same, subsequent Machine Learning algorithm distinguishes different labels in a simpler way. Moreover, preprocessing stage might also be performed in order to speed up computation and in this case it involves dimensionality reduction. For example computer must handle huge numbers of data per second but presenting these directly to a complex Machine Learning algorithm may be computationally infeasible. Instead, the aim is to find a reduced set of useful variables that are fast to compute, and yet that also preserve useful discriminatory information enabling data to be analyzed. These variables can be expressed in terms of basis functions $\phi_h(\mathbf{X})$ and are then used as inputs to Machine Learning algorithm. However, care must be taken during preprocessing because often information is discarded. If this information is important to solution of the problem then, overall accuracy of system can suffer. Furthermore, new test data must be preprocessed using the same steps as training data.

Finally, many practical applications of Machine Learning usually deal with spaces of high dimensionality, since they comprise a lot of input variables. The severe difficulty that can arise in spaces of many dimensions is sometimes called the curse of dimensionality (term coined by Bellman in 1961 [5]). This poses some serious challenges and is an important factor influencing the design of Machine Learning techniques. However, effective methods applicable to high-dimensional spaces can be found. The reason is twofold. First, real data are often confined to a region of space having lower dimensionality. As a consequence, the directions over which important variations in the target variables occur may be confined. Second, datasets typically exhibit some smoothness properties, at least locally. As a consequence, for the most part small changes in the input variables produce small changes in the target variables. So, local interpolation-like techniques can be exploited to make label predictions for new instances. Successful Machine Learning techniques use one or both of these properties.

In conclusion, supervised learning processes labelled data characterized by uncertainty in order to describe their regularity (representation) and to perform accurately on new observations (generalization). For this reason, the algorithms aim to define the adaptive model that automatically relates input and label of both observed and future instances. Thus, supervised learning should perform curve fitting during the training phase, also known as learning phase. The following three Subsections will give more concrete details about this task explaining both model fitting (Subsection 1.2.1) and model selection (Subsection 1.2.2) and reasons that support their criteria (Subsection 1.2.3).

In order to satisfy all these goals, learning algorithms look to both Statistics and Decision Theory that are ground on Probability Theory. Probability Theory provides a consistent framework for expressing data uncertainty in a precise and quantitative manner. Then, Statistics allows to give the most complete probabilistic description of data trough several methods. Thus, Decision Theory exploits this probabilistic representation in order to take decisions that are optimal according to appropriate criteria, given all available information, even though that they may be incomplete or ambiguous.

1.2.1 Model fitting

Curve fitting from a finite dataset with instances corrupted by random noise is intrinsically a difficult problem. But variability is a key concept in the field of Probability Theory that expresses uncertainty over a variable using a probability distribution. Formally, probability distribution of a univariate random variable $\mathbf{y}$ is equal to $P_{\mathbf{y}}(A) = P(y \in A)$ and denotes the probability that random variable

$\mathbf{y}$ assumes values that belong to A, a subset of its support[6]. However, in order to avoid a rather cumbersome notation in the following a simple writing $P(\mathbf{y})$ denotes the distribution over the random variable $\mathbf{y}$.

Then, modelling distributions from data lies at the heart of statistical inference. Here, the parametric approach of density modelling uses probability distributions with a specific functional form governed by a small number of parameters, whose values should be determined from a dataset. In particular, parametric supervised learning assumes that observations are drawn independently from the same distribution. This means that data are realizations of independent and identically distributed variables (often abbreviated to i.i.d.). As a consequence, training set can be characterized by a joint probability distribution $P(\mathbf{X}, \mathbf{y})$ that records all the available information of observed data and provides a complete summary of the uncertainty associated with these variables. Determination of $P(\mathbf{X}, \mathbf{y})$ from a set of training data gives the most complete probabilistic description of the situation and is an example of inference that is typically a very difficult problem.

From this joint distribution other quantities can be computed, as marginal distribution $P(\mathbf{X})$ or conditional distributions $P(\mathbf{y}|\mathbf{X})$ and $P(\mathbf{X}|\mathbf{y})$ that are linked each other by the following relation

$$P(\mathbf{y}|\mathbf{X}) = \frac{P(\mathbf{X}|\mathbf{y})P(\mathbf{y})}{P(\mathbf{X})}$$

that come from the fundamental result of Probability Theory called Bayes theorem

$$P(\mathbf{y}|\mathbf{X}) = \frac{P(\mathbf{y}, \mathbf{X})}{P(\mathbf{X})}$$

Since supervised learning goal is to assign a label y_i given a new input $\mathbf{x}_i$ on the basis of the training set, the most interesting quantity is intuitively[7] the distribution of $\mathbf{y}$ given the corresponding value of $\mathbf{X}$, i.e. $P(\mathbf{y}|\mathbf{X})$. So, from a probabilistic perspective, aim is to model this predictive distribution that expresses uncertainty about the value of $\mathbf{y}$ given values of $\mathbf{X}$.

In agreement with model of Equation 1.3, it is usually assumed that target variable $\mathbf{y}$ is given by a deterministic function with additive Gaussian noise. This means that noise is a random variable that follows a Gaussian distribution, usually assumed with zero mean, unknown variance σ^2 and with uncorrelated realizations. Thus, conditional distribution is also Gaussian

$$P(\mathbf{y}|\mathbf{X}, \mu, \sigma) = \mathcal{N}(\mu, \sigma^2)$$

[6] The support of a random variable is the set of all possible values that random variable can assume.

[7] In Subsection 1.2.3 paragraph concerned with regression explains more formally reasons of this statement introducing concept of loss function and describing assumptions of Decision Theory that support conditional mean, that is equivalent to regression function $\phi(\mathbf{x}_i)^T\mathbf{b}$, as the optimal solution for a regression setting (Equation 1.15).

In this case the conditional mean will be simply

$$\mu = E[\mathbf{y}|\mathbf{X}] = E[\phi(\mathbf{X})\mathbf{b} + \mathbf{e}] = \phi(\mathbf{X})\mathbf{b} \tag{1.4}$$

because of properties of average: mean of a sum is equal to sum of means and average of a constant is equal to the same constant. Indeed, if relation of Equation 1.3 holds, $\phi(\mathbf{X})\mathbf{b}$ given $\mathbf{X}$ can be seen as a constant and noise has null mean. So, it is supposed that data are drawn independently from

$$\begin{aligned} P(\mathbf{y}|\mathbf{X}, \mathbf{b}, \sigma) &= \mathcal{N}(\phi(\mathbf{X})\mathbf{b}, \sigma^2) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} exp\{-\frac{1}{2\sigma^2}[\mathbf{y} - \phi(\mathbf{X})^T\mathbf{b}]^2\} \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\{-\frac{1}{2\sigma^2}[y_i - \phi(\mathbf{x}_i)^T\mathbf{b}]^2\} \end{aligned} \tag{1.5}$$

a Gaussian distribution whose mean (Equation 1.4) and variance σ^2 are unknown and should be computed from training set. The last condition holds since probability of two independent events is given by product of marginal probabilities for each event separately, so that conditional distribution is equivalent to product of marginal probabilities $P(y_i|\mathbf{x}_i, \mathbf{b}, \sigma)$. Note that the Gaussian noise assumption implies that the conditional distribution of $\mathbf{y}$ given $\mathbf{X}$ is unimodal, which may be inappropriate for some applications.

Parameters are unfortunately unknown, so that the problem is to compute them by fitting the function of Equation 1.3 to training data $\{(y_i, \mathbf{x}_i)\}_{i=1}^{n}$. There are several tools to determine parameters in a probability distribution of training set. An intuitive approach suggest to minimizing an error function that measures misfit between $\phi(\mathbf{x}_i)^T\mathbf{b}$, values of function of Equation 1.3 for any given value of $\mathbf{b}$, and y_i training set data values [8]. One simple choice of error function, which is widely used, is given by sum of squares of errors between $\phi(\mathbf{x}_i)^T\mathbf{b}$ for each data point $\mathbf{x}_i$ and the corresponding target value y_i, so that we minimize

$$RSS(\mathbf{b}) = \frac{1}{2}\sum_{i=1}^{n}[\phi(\mathbf{x}_i)^T\mathbf{b} - y_i]^2 \tag{1.6}$$

where the factor of 1/2 is included for later convenience. This function is called residual sum of squares and it is a nonnegative quantity that would be zero if, and only if, the function $\mathbf{y} = \phi(\mathbf{X})\mathbf{b}$ was to pass exactly through each training data point. So, curve fitting problem can be solved by choosing the value of $\mathbf{b}$ for which $RSS(\mathbf{b})$

[8] In Subsection 1.2.3 paragraph concerned with regression explains more formally reasons of this statement introducing concept of loss function and describing assumptions of Decision Theory that support conditional mean, that is equivalent to regression function $\phi(\mathbf{x}_i)^T\mathbf{b}$, as the optimal solution for a regression setting (Equation 1.15).

is as small as possible. Because error function is a quadratic function of coefficients $\mathbf{b}$, its derivatives with respect to coefficients will be linear in elements of $\mathbf{b}$, and so minimization of error function has a unique solution, called fitted parameters and denoted by a circumflex $\hat{\mathbf{b}}$, which can be found in closed form from normal equations as

$$\begin{aligned} \phi(\mathbf{X})^T\phi(\mathbf{X})\mathbf{b} =& \phi(\mathbf{X})^T\mathbf{y} \\ \hat{\mathbf{b}} =& [\phi(\mathbf{X})^T\phi(\mathbf{X})]^{-1}\phi(\mathbf{X})^T\mathbf{y} \end{aligned} \tag{1.7}$$

The resulting fitted function is then given by $\hat{\mathbf{y}} = \phi(\mathbf{X})\hat{\mathbf{b}}$.
It is sometimes more convenient to use the root mean square error defined by

$$RMSE(\mathbf{b}) = \sqrt{\frac{2RSS}{n}}$$

in which division by n allows to compare different sizes of datasets on an equal footing, and square root ensures that RMSE is measured on the same scale (and in the same units) as the target variable $\mathbf{y}$.

However, the most common criterion for determining parameters in a probability distribution using an observed dataset is to find parameter values that maximize the likelihood function. Likelihood function is the conditional probability of dataset $P(\mathbf{y}|\mathbf{X}, \mathbf{b}, \sigma)$ when seen as a function of its unknown parameters $\mathbf{b}$ and σ. This might seem like a strange criterion because it would seem more natural to maximize probability of parameters given the data, not probability of data given parameters. However, these two criteria are strictly related.
With previous assumption, as Gaussian noise and i.i.d. data, likelihood is equal to

$$L(\mathbf{b}, \sigma, |\mathbf{y}, \mathbf{X}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\{-\frac{1}{2\sigma^2}[y_i - \phi(\mathbf{x}_i)^T\mathbf{b}]^2\}$$

In practice, it is usually more convenient to maximize the log of the likelihood function. Because logarithm is a monotonically increasing function of its argument, maximization of log of a function is equivalent to maximization of the function itself.

$$\ln L(\mathbf{b}, \sigma, |\mathbf{y}, \mathbf{X}) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}[y_i - \phi(\mathbf{x}_i)^T\mathbf{b}]^2 - \frac{n}{2}\ln\sigma^2 - \frac{n}{2}\ln(2\pi) \tag{1.8}$$

Taking the log not only simplifies the subsequent mathematical analysis, but it also helps numerically because the product of a large number of small probabilities can easily underflow the numerical precision of the computer, and this is solved by computing instead the sum of the log probabilities.

Computing first derivatives with respect to $\mu = \phi(\mathbf{X})\mathbf{b}$ and σ and putting them equal to zero the following results are obtained

$$\begin{aligned}\hat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} y_i \\ &= \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{x}_i)^T\mathbf{b}\end{aligned} \tag{1.9}$$

$$\begin{aligned}\hat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu})^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}[\phi(\mathbf{x}_i)^T\mathbf{b} - \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{x}_i)^T\mathbf{b}]^2\end{aligned} \tag{1.10}$$

that are respectively sample mean and sample variance measured with respect to the sample mean. Usually a joint maximization of Equation 1.8 should be performed with respect to μ and σ^2, but in the case of Gaussian distribution solution for μ decouples from that for σ^2 so that Equation 1.9 can be first solved and then subsequently its result used to evaluate Equation 1.10. However both solutions depend on parameter $\mathbf{b}$ that is still unknown. It can be estimated following the same procedure, computing derivative of likelihood with respect to $\mathbf{b}$ and putting it equal to zero, but this time vectorial formula of likelihood, that is equal to second relation of Equation 1.5, is considered as starting point. As a consequence, $\hat{\mathbf{b}}$ is equal to

$$\hat{b} = [\phi(\mathbf{X})^T\phi(\mathbf{X})]^{-1}\phi(\mathbf{X})^T\mathbf{y} \tag{1.11}$$

that becomes in the simplest linear regression model, i.e. model that is linear also in variables, the famous formula

$$\hat{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{1.12}$$

Once computed $\hat{\mathbf{b}}$ it can be substituted in previous equations. Maximum likelihood approach has significant drawbacks, for instance if data follow a Gaussian distribution, it systematically underestimates variance[9]. This is an example of a phenomenon called bias[10] that is related to problem of overfitting.

Instead of maximizing the log likelihood, negative log likelihood can be equivalently

[9] Mean of maximum likelihood solution for variance is equal to $E[\hat{\sigma^2}] = \frac{n-1}{n}\sigma^2$, so that every estimated value is less than true value of σ^2 by a factor of $\frac{n-1}{n}$.

[10] Bias is defined as the difference between mean of estimator and true value of parameter. When mean of estimator is equal to true value of parameter, so that difference is null, estimator is said unbiased or correct. On the contrary, for example sample variance is a biased estimator of variance. However, bias becomes less significant as number n of data points increases, and in the limit $n \to +\infty$ estimator equals true variance of distribution that generated data. In practice, for anything other than small n, this will not prove to be a serious problem and sample variance is defined asymptotically correct.

minimized. As a consequence, under the assumption of a Gaussian noise distribution, maximizing likelihood is equivalent, so far as determining $\mathbf{b}$ is concerned, to minimizing the sum of squares error function defined by Equation 1.6. Indeed, result of Equation 1.11 is equal to that of Equation 1.7. Thus the sum of squares error function has arisen as a consequence of maximizing likelihood.

However, an important limitation of parametric modelling approach is that the chosen density might be a poor model of distribution that generates data, which can result in poor predictive performance. Nonparametric approaches to density estimation exist and they make few assumptions about form of distribution. Histogram is for instance the most simple frequentist method for modelling probability distribution using an observed dataset. Standard histograms simply partition observed values into distinct bins of width Δ_i and then count the number n_i of observations falling in bin i. In order to turn this count into a normalized probability density, it should be simply divided by the total number n of observations and by the width Δ_i of bins, so that probability values for each bin are obtained as

$$p_i = \frac{n_i}{n\Delta_i}$$

This gives a model for density $p(x)$ that is constant over the width of each bin, and often bins are chosen to have the same width $\Delta_i = \Delta$. In principle, a histogram density model is also dependent on choice of edge location for bins, though this is typically much less significant than value of Δ. Histogram method has the property that, once the histogram has been computed, the dataset itself can be discarded, which can be advantageous if the dataset is large. Also, histogram approach is easily applied if data points are arriving sequentially. In practice, histogram technique can be useful for obtaining a quick visualization of data in one or two dimensions but is unsuited to most density estimation applications. One obvious problem is that estimated density has discontinuities that are due to bin edges rather than any property of underlying distribution that generated data. Another major limitation of histogram approach is its scaling with dimensionality. If we divide each variable in a k-dimensional space into M bins, then total number of bins will be Mk. This exponential scaling with k is an example of curse of dimensionality. In a space of high dimensionality, quantity of data needed to provide meaningful estimates of local probability density would be prohibitive.

1.2.2 Model selection

After model training, based on likelihood approach or equivalently minimizing RSS criterion, label can be assigned also to new inputs. Indeed, once parameters are fitted, the whole probability distribution of data is known and values of target variables for new inputs can be obtained. In this case, labels are denoted by $\hat{y}_i$ and are called predictions. They are formally defined as the conditional mean $E[y_i|\mathbf{x}_i; \hat{\mathbf{b}}]$ that, as previously shown in Equation 1.4, is equal to $\phi(\mathbf{x}_i^T\hat{\mathbf{b}})$ that is known as the regression function[11].

Supervised learning applications require accurate predictions for target variable of coming data. However, good predictions are provided if and only if supervised learning assures a satisfactory generalization performance. This means that the model learned by algorithms should be able to predict accurately next observations beyond describing those given. With other words, it should correctly represent a generative function of both observed and future data. Learned model has indeed a twofold aim since it should provide both the description of known dataset and at the same time the predictions of new instances. These goals are however opposite making model fitting a challenging problem. For example, an excellent fit to training data, where fitted curve passes exactly through each data point and $RSS = 0$, can give a perfect description of observed data but a very poor representation of the real function, so that prediction of future values will be seriously damaged and wrong. This behaviour is known as overfitting.

In order to control overfitting and to assure a good prediction ability, some quantitative insights into dependence of generalization performance on model are needed, in particular on its form and number of parameters. This issue is called model complexity, model comparison or model selection, and it is concerned with choosing the order of computed function by making a trade off between fit to data and representation of function that generates data.

First of all, increase of dataset size reduces the overfitting problem. As a consequence, the larger the dataset the more complex and more flexible can be the model. However, sampling is usually expensive and size of dataset can not always be changed. So, there are many other techniques to manage this problem.

One approach that is often used to control overfitting phenomenon is that of regularization, which involves adding a penalty term to error function (Equation 1.6) in order to discourage coefficients from reaching large values. The simplest such penalty term takes the form of a sum of squares of all of the coefficients, leading to a modified error function of the form

$$RSS_r = \frac{1}{2}\sum_{i=1}^{n}[\phi(\mathbf{x}_i)^T\mathbf{b} - y_i]^2 + \frac{\eta}{2}||\mathbf{b}||^2 \tag{1.13}$$

where $||\mathbf{b}||^2 = \mathbf{b}^T\mathbf{b}$ is the squared norm of parameters vector $\mathbf{b}$ and coefficient η governs the relative importance of regularization term compared with sum of squares error term and the effective complexity of the model. Often coefficient $\mathbf{b}_0$ is omitted, because its inclusion causes results to depend on choice of the origin for target

[11] In Subsection 1.2.3, the paragraph concerned with regression explains more formally reasons of this statement introducing concept of loss function and describing assumptions of Decision Theory that support conditional mean, that is equivalent to regression function $\phi(\mathbf{x}_i)^T\mathbf{b}$, as the optimal solution for a regression setting (Equation 1.15).

variable. Otherwise it may be included, but with its own regularization coefficient. Again, error function of Equation 1.13 can be minimized exactly in closed form. Such techniques are known in Statistics literature as shrinkage methods because they reduce value of coefficients towards zero. The particular choice of quadratic regularizer is called ridge regression[12]. In the context of Machine Learning literature, this approach is known as weight decay because it encourages weight values to decay towards zero, unless supported by the data.

In maximum likelihood approach, results on training set are not a good indicator of predictive performance on unseen data due to problem of overfitting. So, a simple way to determine a suitable value for model complexity is to consider a separate test set of independent data for which trained model can be evaluated. If data is plentiful, a simple method suggests to take available data and partitioning it into a training set and a separate test set, called validation or hold-out set. The former is used to train a range of models determining their parameters, or a given model with a range of values for its complexity parameters; the last allows to optimize the model complexity, comparing models and selecting the one having the best predictive performance. A similar method can be applied with error function approach, where residual sum of squares is computed for training data as in Equation 1.6 but also on test set since test set error is a measure of prediction ability. In many applications, however, this will prove to be too wasteful of valuable training data, and a more sophisticated techniques should be sought. Indeed, supply of data for training and testing can be limited, and in order to build good models, its better to use as much of available data as possible for training. Moreover, if the model design is iterated many times using a limited size dataset, then some overfitting to validation data can occur and so, it may be necessary to keep aside a third test set on which performance of selected model is finally evaluated. Furthermore, if validation set is small, it will give a relatively noisy estimate of predictive performance.

One solution to this dilemma is to use cross validation (Stone, 1974 [64]). This allows a proportion $(S-1)/S$ of available data to be used for training while making use of all of the data to assess performance. When data is particularly scarce, it may be appropriate to consider the case $S = n$, where n is the total number of data points, which gives the leave-one-out technique, called also full cross validation. In this case the number of parameters of the best model is fitted as

$$\hat{m} = \underset{0 \leq m \leq rk(\mathbf{X}^T\mathbf{X})}{\operatorname{argmin}} \sum_{i=1}^{n} (Y_i - \hat{Y}_{m\backslash i})$$

where $\hat{y}_{m\backslash i}$ is the m-th model computed on the sample data with the i-th observation removed. In Chemometrics, model selection is nearly always done through ordinary cross validation. One major drawback of cross validation is that number of training runs that must be performed is increased by a factor of S, and this can

[12] Ridge Regression is described more widely in Section 3.4 whereas some details are added in Section 3.6.

be problematic for models in which training is itself computationally expensive. A further problem with such techniques, that use separate data to assess performance as cross validation, is that there might be multiple complexity parameters for a single model. Exploring combinations of settings for such parameters could, in the worst case, require a number of training runs that is exponential in the number of parameters.

Ideally, a better approach should rely only on training data and should allow multiple hyperparameters and model types to be compared in a single training run. A measure of performance should so be defined which depends only on training data and which does not suffer from bias due to overfitting. Historically various “information criteria” have been proposed that attempt to correct for bias of maximum likelihood by addition of a penalty term to compensate for overfitting of more complex models. For example, the Akaike Information Criterion, or AIC (Akaike, 1974 [1]), chooses the model for which the quantity

$$AIC = -2 \ln L(\hat{\mathbf{b}}, \hat{\sigma} | \mathbf{y}, \mathbf{X}) + 2m$$

is largest, where $L(\hat{\mathbf{b}}, \hat{\sigma}, | \mathbf{y}, \mathbf{X})$ is the best fit likelihood, and m is the number of adjustable parameters $\mathbf{b}$ in the model. A variant of this quantity is called the Bayesian Information Criterion, or BIC (Schwarz, 1978)

$$BIC = -2 \ln L(\hat{\mathbf{b}}, \hat{\sigma} | \mathbf{y}, \mathbf{X}) + m \ln n$$

However, such criteria do not take account of uncertainty in model parameters, and in practice they tend to favour overly simple models.

1.2.3 Decision making

Definition of a joint probability distribution from a set of training data is an example of inference and, even if it is typically a very difficult problem, it assures a complete summary of uncertainty associated with variables. Moreover, once the whole probability distribution of data is known, specific predictions $\hat{y}_i$ for target variable given new input $\mathbf{x}_i$ can be done. Because of probabilistic nature of fitted function, predictions are expressed in terms of predictive distribution that gives the probability distribution over $\mathbf{y}_i$ rather than simply a point estimate.

However, goal of practical supervised learning applications is a bit more general, indeed one should make a specific action based on understanding of values $\mathbf{y}_i$ is likely to take. This aspect is the subject of Decision Theory, that when combined with Probability Theory allows to make optimal decisions in situations involving uncertainty such as those encountered in supervised learning. In particular Decision Theory provides some useful methods that are able to decide either to do or do not an action given the appropriate probabilities considering that choice should be optimal in some appropriate sense. This is the decision step that is indeed generally very simple, even trivial, once inference problem is solved.

In order to present main concepts of Decision Theory in a more concrete way, let consider as starting point classification, with G classes C_g. Two distinct approaches to solving decision problems can be identified, both have been used in practical applications and consists of two phases with same goals. Indeed, both aim first to find class probabilities $P(C_g|\mathbf{X})$ and then to use Decision Theory to determine class membership for each new input $\mathbf{x}$. But former step is reached in different ways.
First kind of processes model explicitly or implicitly distribution of inputs as well as outputs and are known as generative models, because by sampling from them it is possible to generate synthetic data points in the input space. In practice, they solve inference problem of determining class conditional densities $P(\mathbf{X}|C_g)$ for each class C_g individually. Then, they infer separately class probabilities $P(C_g)$. Finally, they apply Bayes theorem to compute conditional class probabilities $P(C_g|\mathbf{X})$. Equivalently, they directly model joint distribution $P(\mathbf{X}, C_g)$ and then they normalize it to obtain class probabilities $P(C_g|\mathbf{X})$. On the contrary, models that belong to second method directly solve inference problem of determining class probabilities $P(C_g|\mathbf{X})$ and they are called discriminative models.
First techniques are the most demanding because they involve finding joint distribution over both $\mathbf{X}$ and C_g. For many applications $\mathbf{X}$ will have high dimensionality, and consequently a large training set is needed in order to be able to determine the class conditional densities to reasonable accuracy. Note that class probabilities $P(C_g)$ can often be estimated simply from fractions of training set data points in each of the classes. One advantage of this approach, however, is that it also allows marginal density of data $P(\mathbf{X})$ to be determined. This can be useful for detecting new data points that have low probability under the model and for which predictions may be of low accuracy, which is known as outlier detection or novelty detection. However, if the main goal is to make classification decisions, then to find the joint distribution $P(\mathbf{X}, C_g)$ can be wasteful of computational resources and excessively demanding of data. When in fact only probabilities $P(C_g|\mathbf{X})$ are really needed, then they can be obtained directly through the other approach. Indeed, class-conditional densities $P(\mathbf{X}|C_g)$ may contain a lot of structure that has little effect on probabilities $P(C_g|\mathbf{X})$.

Second step allows to assign the correct class between $\{C_g\}_{g=1}^{G}$ to each new observation $\mathbf{x}_i$ and involves to understand how probabilities play a role in making decisions. Informally, a correct idea is to follow a criterion that minimizes chance of assigning $\mathbf{x}_i$ to the wrong class. As a consequence, class having the higher probability $P(C_g|\mathbf{x}_i)$ should intuitively be chosen. More formally, a rule should be found that divides input space into regions $\{R_g\}_{g=1}^{G}$ called decision regions, one for each class, such that all points in R_g are assigned to class C_g. The boundaries between decision regions are called decision boundaries or decision surfaces. Each decision region need not be contiguous but could comprise some number of disjoint regions. In addition, this rule should assign each value $\{\mathbf{x}_i\}_{i=1}^{n}$ to one of available classes such that as few misclassifications as possible are make. A mistake occurs when an input vector belonging to one class is assigned to another different class. Probability of misclassification is equal to the sum of probabilities that every observation is assigned to uncorrected classes. Since goal of decision step is to minimize probability of making a mistake, the minimum is obtained if each input value $\mathbf{x}_i$ is assigned to class with higher joint probability $P(\mathbf{x}_i, C_g)$. Using the product rule $P(\mathbf{x}_i, C_g) = P(C_g|\mathbf{x}_i)P(\mathbf{x}_i)$, and noting that factor of $P(\mathbf{x}_i)$ is common to all terms, the class for which conditional probability

$$P(C_g|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|C_g)P(C_g)}{P(\mathbf{x}_i)}$$

is largest should be chosen, where any of the quantities appearing in Bayes theorem can be obtained from the joint distribution $P(\mathbf{x}_i, C_g)$ by either marginalizing or conditioning with respect to appropriate variables.

However, except for the case of two classes, it is usually slightly easier to maximize the probability of being correct, which is given by

$$\begin{aligned} P(correct) &= \sum_{g=1}^{c} P(\mathbf{x}_i \in R_g, C_g) \\ &= \sum_{g=1}^{c} \int_{R_g} P(\mathbf{x}_i, C_g) d\mathbf{x}_i \end{aligned}$$

which is maximized when the regions R_g are chosen such that each $\mathbf{x}_i$ is assigned to the class for which $P(\mathbf{x}_i, C_g)$ is largest that is equivalent to assign it to the class having the largest conditional probability $P(C_g|\mathbf{x}_i)$.

For many applications, goal will be more complex than simply minimizing the number of misclassifications. As a consequence, it is necessary to introduce a loss function, also called a cost function, which is a single, overall measure of loss incurred in taking any of the available decisions or actions. Goal is then to minimize the total loss incurred[13].

Considering classification, let suppose that, for a new input value of $\mathbf{x}_i$ true class is C_g and that $\mathbf{x}_i$ is assigned to class C_j (where j may or may not be equal to g). In so doing, some level of loss is reached that is denoted by L_{gj}, which we can view as the g, j element of a loss matrix. Optimal solution is the one which minimizes loss function. However, loss function depends on true class, which is unknown. For a given input vector $\mathbf{x}_i$, uncertainty in true class is expressed through joint probability distribution $P(\mathbf{x}_i, C_g)$ and so the average loss should instead be minimized, where

[13] Some authors consider instead a utility function, whose value they aim to maximize. These are equivalent concepts since utility is simply the negative of loss.

the average is computed with respect to this distribution, and is given by

$$E[L] = \sum_g \sum_j \int_{R_j} L_{gj} P(\mathbf{x}_i, C_g) d\mathbf{x}_i \tag{1.14}$$

Each input $\mathbf{x}_i$ can be assigned independently to one of decision regions R_j. Criterion is to choose the regions R_j in order to minimize the expected loss of Equation 1.14, which implies that for each $\mathbf{x}_i$, $\sum_g L_{gj} P(\mathbf{x}_i, C_g)$ should be minimized. As before, we can use the product rule to eliminate common factor of $P(\mathbf{x}_i)$. Thus, decision rule that minimizes expected loss is the one that assigns each new $\mathbf{x}_i$ to the class j for which quantity

$$\sum_g L_{gj} P(C_g|\mathbf{x}_i)$$

is a minimum. This is clearly trivial to do, once class probabilities are known (or fitted).

Classification errors arise from regions of input space where joint distributions $P(\mathbf{X}, C_g)$ have comparable values. These are the regions where there is relatively uncertainty about class membership. In some applications, it will be appropriate to avoid making decisions on the difficult cases in anticipation of a lower error rate on those examples for which a classification decision is made. This is known as the reject option that introduce a threshold and rejects those inputs $\mathbf{x}_i$ for which the largest of the probabilities $P(C_g|\mathbf{x}_i)$ is less than or equal to the threshold. Setting where threshold is equal to 1 will ensure that all examples are rejected, whereas if there are c classes then setting threshold less than $1/c$ will ensure that no examples are rejected. Thus fraction of examples that get rejected is controlled by the value of threshold. Reject criterion can be easily extended to minimize expected loss, when a loss matrix is given, taking account of loss incurred when a reject decision is made.

Until now, classification problem has been broken down into two separate steps, inference in which training data is used to learn a model for $P(C_g|\mathbf{X})$, and subsequent decision stage in which probability $P(C_g|\mathbf{X})$ is used to make optimal class assignments. An alternative would be to solve both problems together thereby combining inference and decision stages into a single learning problem. This implies simply to learn a function $f(\mathbf{x}_i)$, called discriminant function, that maps inputs $\{\mathbf{x}_i\}_{i=1}^n$ directly onto a class label, or onto decisions. For instance, in the case of two-class problems, discriminant function might be binary valued such that $f(\mathbf{x}_i) = 0$ represents class C_1 and $f(\mathbf{x}_i) = 1$ represents class C_2. This possibility is even simpler since probabilities $p(C_g|\mathbf{x})$ play no role even if there are many powerful reasons for wanting to compute these probabilities.

As regards regression, characterized by real-valued variables and a model expressed by Equation 1.3, Decision Theory allows to choose a specific prediction $\hat{y}_i$

of each value y_i for any new input $\mathbf{x}_i$ in such a way as to minimize the average, or expected, value of a suitably chosen loss function $L(y_i, \hat{y}_i)$, that is equal to

$$E[L] = \int\int L(y_i, \hat{y}_i)P(\mathbf{x}_i, y_i)d\mathbf{x}_i dy_i$$

A common choice of loss function in regression problems is the squared loss given by

$$L(y_i, \hat{y}_i) = \{\hat{y}_i - y_i\}^2$$

In this case, the expected loss can be written as

$$E[L] = \int\int \{\hat{y}_i - y_i\}^2 P(\mathbf{x}_i, y_i)d\mathbf{x}_i dy_i$$

Since criterion is to choose $\hat{y}_i$ so as to minimize $E[L]$, optimal solution is given by

$$\hat{y}_i = E[y_i|\mathbf{x}_i] = \phi(\mathbf{x}_i)^T\hat{b} \tag{1.15}$$

which is the conditional average of y_i on $\mathbf{x}_i$ shown in Equation 1.4 and it is known as the regression function.
In the light of this explanation a complete overview of all previous results can be done. First of all it is now clear the reason for which predictions are defined as conditional mean, and this result increases its plausibility since predictions are optimal with respect to some criteria. Then, criterion of minimizing expected squared loss function for computing optimal predictions confirms that of minimizing RSS for fitting parameters. Finally, it is now justified interest in conditional distribution and in fitting its parameters rather than in joint distribution.

As with classification, there are two strategies that can be followed: either to determine appropriate probabilities and then to use these to make optimal decisions, or to build models that make decisions directly. First approach can solve its inference problem determining joint density $P(\mathbf{x}_i, y_i)$, then normalizing to find conditional density $P(y_i|\mathbf{x}_i)$, and finally marginalizing to find conditional mean. Otherwise, it can first determine conditional density $P(y_i|\mathbf{x}_i)$, and then subsequently marginalizing to find conditional mean. Second approach finds a regression function $\hat{y}_i = \phi(\mathbf{x}_i)^T\hat{b}$ directly from training data.
The relative merits of these three approaches follow the same lines as for classification problems above.
The squared loss is not the only possible choice of loss function for regression. Indeed, there are situations in which squared loss can lead to very poor results and more sophisticated approaches should be developed. An important example concerns situations in which conditional distribution $P(y_i|\mathbf{x}_i)$ is multimodal, as often arises in solution of inverse problems.

2

Partial Least Squares (PLS)

This Chapter presents the supervised learning technique of Partial Least Squares (PLS), core argument of the thesis, dealing with all the main concepts related to its definition and use. After the introduction (Section 2.1), a formal description of PLS is provided, including its mathematical structure, algorithmic features and geometric representation (Section 2.2). Tasks of the PLS method are then illustrated (Section 2.3), so that PLS and its behaviour in several situations can be compared with alternative methods in the following Chapters. In addition, an overview of some PLS applications to different data analysis problems is given (Section 2.4). Finally, the main softwares that include PLS method are reviewed (Section 2.5).

2.1 Introduction

Machine Learning, as previously said, is a discipline based on data collection and analysis. Indeed, learners process data of the training set in order to reach their goals (representation and generalization). Thus data, that consist of measurements of individual properties done for each instance, are capital for solving Machine Learning problems.

Such prominence of data is confirmed by their key role in the more general process of understanding real phenomenons. Indeed, in order to know and study an observable event, researchers usually settle, work with and refer to a corresponding system, i.e., a group of interacting, interrelated, or interdependent variables[1]. Since the system is characterized by uncertainty it is usually assumed that it follows a probability distribution which is theoretically defined but at the same time unknown. Then, the aim of the survey is to identify significant variables excluding irrelevant ones and to understand relations between them, taking into account their random nature. Empirical evidence helps in this task, because measurements of quantities involved in the phenomenon produce data whose analysis allows to describe the system. As a consequence, data, that can be seen as observed values of variables[2], should be

[1] In other words, the defined system corresponds to an observable event whose specific quantities are represented by variables.

[2] Data are also called realizations, i.e. draws from the fixed and unknown probability distribution.

gathered together and analyzed with suitable techniques. More data are available, more information about the system can be extracted through data analysis. However, since the process of data collection is usually expensive and time consuming, researchers usually apply a specific sampling design. It consists in detecting the population, as the set of all individuals directly involved with the phenomenon, then, following specific sampling techniques, it defines the statistical sample as a smaller subset of the population on which measurements of variables are done. In this way, researchers are sure that data describe the system and that analyses on the training set explain also the real event.
With this premises, one can state that the system generates collected data and that data analysis allows to understand the real phenomenon. However, sometimes further conjectures should be done. The system could be indeed driven by a small number of latent, i.e. not directly measurable, variables. In this case some new quantities exist that straightforwardly control the phenomenon. So, it is necessary to build them with the original variables before fitting a model for the phenomenon of interest. The method presented in this Chapter makes this underlying assumption.

Partial Least Squares (PLS) is a wide class of techniques for modelling relations between sets of observed quantities by means of latent variables. In its general form, PLS defines new components by maximizing the covariance between two different blocks of original variables. In this way PLS builds a more compact latent space where measured data are firstly projected and then the model is fitted with this updated dataset. As regards assumptions that PLS requires to be assured by data for its applicability, such hypothesis are not so constraining. PLS emerges indeed as a multivariate method that can remove collinearity[3] from dataset. Furthermore, PLS solves the problem that arises when number of individuals is much lower than cardinality of variables. In addition, PLS has minimum demands in terms of residual distribution, since it does not need data to come from normal or known distributions.
PLS was developed by econometrician H. Wold since 1966 when he modified his Fixed Point technique, that uses an iterative Ordinary Least Squares (OLS) algorithm to estimate coefficients of a system of simultaneous equations (Wold, 1966 [74]). This method gave way to Nonlinear Iterative PArtial Least Squares (NIPALS) algorithm that consists of an iterative sequence of simple and multiple OLS regressions in order to calculate principal components and canonical correlations, respectively (Wold, 1973 [75]). General PLS algorithm is based on NIPALS and appeared at the end of 1977 as an iterative procedure for finding latent variables. There are different techniques to extract latent vectors, and each of them gives rise

[3] Collinearity exists when there is an extreme dependence between variables. Presence of collinearity involves difficulties in interpreting results because fitted coefficients can be insignificant to explained variable.

to a variant of PLS. For example the approach originally designed by Wold is called PLS Mode A, whereas PLS1 and PLS2 are the most frequently used forms. Moreover, some PLS variants model relations among a number of sets higher than two. Wold's son simplified PLS algorithm and added diagnostic interpretation with various co-workers since the eighties (Wold, 1984 [79]; Wold, 1989 [78] and Wold, 1992 [77]). These later works have turned the PLS method into the general scientific data analysis tool that it is today and that assures computational and implementation simplicity also with huge datasets (Wold, 2001 [80]).

PLS has many statistical tasks, indeed it comprises regression and classification ability, but at the same time it can be used also as descriptive tool. Furthermore, both dimension reduction and nonlinear PLS can be defined.

PLS has been proven to be nowadays a very powerful technique in many research areas and industrial contexts. This multidisciplinary applicability is a main feature of PLS that distinguishes it since its origins. Indeed PLS was initially developed from methods related to Econometric arguments. Then, interests in PLS shift from Social Sciences to Chemometrics, where PLS has firstly received a great amount of attention. This approval of PLS resulted in a lot of implementations in other disciplines including Food Research, Physiology, Medicine, Pharmacology, Economics (as business disciplines or strategic management area), Engineering and Computer Science (as bioinformatics, computer vision, image processing, human detection and face recognition) to name but a few.

The fruitful application of PLS in so many fields depends on the availability of a suitable software, since PLS needs sophisticated computations. Fortunately, all the free or commercial suites for multivariate analysis provides tools that implement also PLS methods. Such solutions can have a more general applicability, working in several industries, or be more subject oriented, as specific applications for well defined domains.

2.2 PLS theory

In this Section a complete description of the PLS procedure is provided. It includes firstly a mathematical part, that requires some basic knowledge of algebra, as the use of vectorial spaces, matrices and their properties. Then, NIPALS algorithm is presented in order to show how PLS is really computed. Some variants of PLS are listed here, which differ from PLS in the last step. An eigenvalue problem is also described, its solutions correspond to results obtained by means of NIPALS algorithm. Finally, a geometric representation of both data and PLS results is suggested for simple examples by means of basic plots, it is useful to better understand information carried by dataset.

In the following, matrices and vectors (columns for definition) are denoted by bold capital and bold lowercase letters, respectively. They represent data, variables, weights or unknown parameters. A superscript T denotes the transpose of a matrix or vector, so that transpose of a vector will be a row. Lowercase letters stay for real numbers, as indexes. In particular j is used as column index and i for row in presence of matrices. Other symbols respect standard assumptions or they are explained near corresponding equation.

2.2.1 Mathematical structure

Let n be the number of instances belonging to the collected training set on which variables of interest are measured.
Let the domain $\mathcal{X} \subseteq I\!R^k$ denote a subset of k-dimensional Euclidean space and $\mathbf{X} \in \mathcal{X}$ be a matrix of dimension $(n \times k)$. $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_k)$ includes k vectors $\mathbf{x}_j \in I\!R^n$. Every vector $\mathbf{x}_j$ records the values $\{x_{ij}\}_{i=1}^n$ of j-th variable that are measured for each instance of the training set.
Similarly, let $\mathcal{Y} \subseteq I\!R^d$ be a d-dimensional Euclidean space and $\mathbf{Y} \in \mathcal{Y}$ be a matrix of dimension $(n \times d)$. $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_j, \ldots, \mathbf{y}_d)$ collects d vectors $\mathbf{y}_j \in I\!R^n$ of observations $\{y_{ij}\}_{i=1}^n$.
$\mathbf{X}$ and $\mathbf{Y}$ pairs of row vectors are denoted by $\{\underline{\mathbf{x}}_i, \underline{\mathbf{y}}_i\}_1^n$, $\quad \mathbf{x}_i \in I\!R^k$, $\mathbf{y}_i \in I\!R^d$ and collect all the information about every measured instance.
Some statistics can be computed from these data. Let the sample mean of $\mathbf{X}$ and $\mathbf{Y}$ be respectively

$$\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_j, \ldots, \bar{x}_k)^T \qquad \bar{\mathbf{y}} = (\bar{y}_1, \ldots, \bar{y}_j, \ldots, \bar{y}_d)^T$$

i.e. the k- and d-dimensional vectors that collect sample means of every $\mathbf{x}_j$ and $\mathbf{y}_j$, computed as $\bar{x}_j = \frac{1}{n}\sum_{i=1}^n x_{ij}$ and $\bar{y}_j = \frac{1}{n}\sum_{i=1}^n y_{ij}$. Many statistical analyses settle the hypothesis of variables with a null mean. As a consequence, without loss of generality, it can be assumed that every vector of both $\mathbf{X}$ and $\mathbf{Y}$ matrices has zero mean; otherwise it can be easily centered. Moreover, let

$$\mathbf{S}_{\mathbf{X}} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} \qquad \mathbf{S}_{\mathbf{Y}} = \frac{1}{n-1}\mathbf{Y}^T\mathbf{Y} \qquad \mathbf{S}_{\mathbf{XY}} = \frac{1}{n-1}\mathbf{X}^T\mathbf{Y} \tag{2.1}$$

be respectively the $(k \times k)$ and $(d \times d)$ symmetric matrices of unbiased sample covariance and $(k \times d)$ sample cross product covariance matrix. Correct sample variance of vector $\mathbf{x}_j$ is denoted by $s_j^2 = \frac{1}{n-1}\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2$ and enters on main diagonal of matrix $\mathbf{S}_{\mathbf{X}}$ that contains outside $s_{ij} = \frac{1}{n-1}(\mathbf{x}_i - \bar{x}_i)^T(\mathbf{x}_j - \bar{x}_j)$, the correct sample covariance between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. The same concepts hold for $\mathbf{S}_{\mathbf{Y}}$.
Sometimes measured quantities can be standardized, or autoscaled, that means $x_{ij}^{new} = (x_{ij} - \bar{x}_j)/(\sqrt{s_j^2})$ obtaining new data characterized by mean and variance equal to 0 and 1 respectively. Analyses are then applied to the standardized variables and the resulting solutions transformed back to reference the original locations and scales of observed quantities.

All these sample statistics (and their values) can be used as estimators (and as estimates) of corresponding theoretical quantities. Indeed let $X = (x_1, \ldots, x_j, \ldots, x_k)$ and $Y = (y_1, \ldots, y_j, \ldots, y_d)$ denote the multivariate (k-variate and d-variate) random variables that represent properties of the observed phenomenon. Both are characterized by their own probability distribution. This means that they have a mean $\mu_X = (\mu_1, \ldots, \mu_j, \ldots, \mu_k)^T$ and $\mu_Y = (\mu_1, \ldots, \mu_j, \ldots, \mu_d)^T$ respectively, k- and d- vectors that collect mean of every unidimensional variable. Moreover, $(k \times k)$ and $(d \times d)$ covariance matrices $\mathbf{\Sigma_X}$ and $\mathbf{\Sigma_Y}$ can be defined as well as $(k \times d)$ cross product covariance matrix $\mathbf{\Sigma_{XY}}$. Mean is defined as the expectation, or equivalently the first moment, and it is usually denoted with $\mu_x = ave(x) = E[x]$ for a general unidimensional variable x. Variance is the second central moment defined as $\sigma_x^2 = var(x) = E[(x - E[x])^2] = E[x^2] - (E[x])^2$. Covariance is the joint central moment equal to $\sigma_{xy} = cov(xy) = E[(x - E[x])(y - E[y])]$. Obviously, probability distribution of variables is unknown, since it is impossible to observe and measure all their realizations. As a consequence, data are essential to compute the statistics that estimate corresponding theoretical quantities allowing to describe the unknown phenomenon.

In order to compute the best model for observed quantities by means of latent variables, PLS decomposes the zero mean matrices $\mathbf{X}$ and $\mathbf{Y}$ according to the following formulas

$$\begin{aligned} \mathbf{X} &= \mathbf{TP}^T + \mathbf{E} \\ \mathbf{Y} &= \mathbf{UQ}^T + \mathbf{F} \end{aligned} \tag{2.2}$$

with $\mathbf{T}$ and $\mathbf{U}$ $(n \times m)$ score matrices; $\mathbf{P}\,(k \times m)$ and $\mathbf{Q}\,(d \times m)$ loading matrices; $\mathbf{E}\,(n \times k)$ and $\mathbf{F}\,(n \times d)$ residual matrices. Quantity m can be considered a meta-parameter and it should be estimated from data through cross validation, a model selection criterion[4]. It has an upper bound equal to the rank of covariance matrix $\mathbf{S_X}$, that is $m \leq rk(\mathbf{X}^T\mathbf{X}) \leq k$.

In order to compute this decomposition, PLS looks for weight vectors (or projection vectors) $\mathbf{w}$ and $\mathbf{c}$, $(k \times 1)$ and $(d \times 1)$ respectively, such that

$$[cov(\mathbf{t}, \mathbf{u})]^2 = \underset{||\mathbf{w}||=||\mathbf{c}||=1}{\text{argmax}} \, [cov(\mathbf{Xw}, \mathbf{Yc})]^2 \tag{2.3}$$

where $\mathbf{t}$ and $\mathbf{u}$ are a column of $\mathbf{T}$ and $\mathbf{U}$ respectively[5] and cov indicates the sample covariance defined by $cov(\mathbf{t}, \mathbf{u}) = \frac{\mathbf{t}^T\mathbf{u}}{n}$. In the constraint, $||\mathbf{w}|| = \sqrt{^T\mathbf{w}} = \sqrt{\sum_{i=1}^k w_i^2}$ is the norm of vector $\mathbf{w}$ and states that the vector $\mathbf{w}$ has length 1. At the end of

[4] Subsection 1.2.2 gives a complete overview about model selection criteria and describes in details other alternative techniques. Straightforward application of many of the competing model selection criteria is not appropriate for PLS, since it is not a linear modelling procedure that is, $\mathbf{Y}$ values enter nonlinearly into the model estimates.

[5] $\{\mathbf{t}_j\}_{j=1}^m$ is a similar notation that can be used in the following.

PLS procedure m vectors $\mathbf{w}$ and $\mathbf{c}$ are computed and they are collected in matrices $\mathbf{W}$ and $\mathbf{C}$ of dimensions $(k \times m)$ and $(d \times m)$ respectively.
Informally, PLS extracts m latent variables as linear combinations of the original observed and measured quantities. These components define a new vector space that is a subspace of the original one which data belong to. PLS builds latent variables maximizing the covariance, i.e. the separation, in the subspace. In other words, PLS learns a subspace in which every latent variable is well distinguished from the others. Moreover, latent variables collect most of the variation of observable $\mathbf{X}$ in such a way that they may also be used to model $\mathbf{Y}$ quantities. Components explain indeed both $\mathbf{X}$ and $\mathbf{Y}$ variability. Then, PLS projects instances and variables computing matrices $\mathbf{T}$, $\mathbf{P}$, $\mathbf{U}$ and $\mathbf{Q}$. With these results PLS produces a sequence of m models and estimate which one is the best through cross validation.

2.2.2 NIPALS algorithm

In its classical form, PLS is based on the Nonlinear Iterative PArtial Least Squares (NIPALS) algorithm, reported in Table 2.1. NIPALS algorithm starts with a random initialization of the score vector $\mathbf{u}_i$. If $\mathbf{Y} = \mathbf{y}$, i.e. it is unidimensional, NIPALS assigns to $\mathbf{u}_0$ values of the original (standardized) data $\mathbf{y}$; otherwise $\mathbf{u}_0$ is equal to a random n-dimensional vector (step 1). Then, at each iteration $i = 1, \dots, k$ of NIPALS algorithm until convergence, residuals of $\mathbf{Y}$ variables are partially regressed on $\mathbf{X}$ residuals, both computed from previous iteration (step 2). Partial regression consists of computing covariance weight vector $\mathbf{w}_i$ (step a) and then using it as a projection vector to form a linear combination of $\mathbf{X}_{i-1}$ variables so that score vector $\mathbf{t}_i$ can be obtained (step b). $\mathbf{Y}_{i-1}$ variables are regressed on this linear combination, so that weight vector $\mathbf{c}_i$ is computed (step c) and used to project $\mathbf{Y}_{i-1}$ values in order to obtain a new score vector $\mathbf{u_i}$ (step d). All formulas of this phase are confirmed by Equation 2.3. Now vectors of loadings $\mathbf{p}_i$ and $\mathbf{q}_i$ can be computed as coefficients of regressing $\mathbf{X}_{i-1}$ on $\mathbf{t}_i$ and $\mathbf{Y}_{i-1}$ on $\mathbf{u}_i$ in agreement with relations explained in Equation 2.2 (step 3). Then, PLS model $\hat{\mathbf{Y}}_i$ can be fitted since all required elements are available (step 4). Furthermore, NIPALS algorithm performs a deflation of current matrices $\mathbf{X}_{i-1}$ and $\mathbf{Y}_{i-1}$, by subtracting their rank-one approximations based on their projections (step 5). As a consequence, any information captured by $\mathbf{w}_i$ is removed, residuals are formed and they can be used as variables for the next iteration if explanatory power of the regression is small and more projection vectors are required. Indeed, in order to achieve the desired latent space dimension, algorithm returns to the first step to evaluate a new $\mathbf{w}_i$ and the whole process is repeated until the wished number of latent vectors had been extracted. In this way, NIPALS algorithm produces a sequence of models $\{\hat{\mathbf{Y}}_i\}_1^m$, where $m \leq rk(\mathbf{S}_\mathbf{X})$, on successive passes through the "for" loop. The one $\hat{\mathbf{Y}}_i$ that minimizes the cross validation score is selected as the PLS solution. The test will cause the algorithm to terminate after as many steps as the rank of sample covariance matrix $\mathbf{S}_\mathbf{X}$ (step 6).

The asymptotic space and time complexity of NIPALS algorithm is $O(kn)$ (assuming $d \leq k$). Indeed, it is not possible to do away with $O(kn)$ space requirement because this is necessary to store the $\mathbf{X}$ matrix. As a consequence, in spite of its good performances, NIPALS algorithm for PLS can have computational bottlenecks in presence of very large sample size n and high number of variables k. Occurrence of these conditions implies extension of computational time. Although in some applications PLS modelling is an off line training step, accelerating it is paramount to enable large scale modelling. Time complexity can be addressed with efficient parallelization strategy using for example graphical processors that achieved $\sim 30X$ speedups against standard CPU-based implementations (Srinivasan et al., 2010 [62]).

Table 2.1: NIPALS algorithm

```
Nonlinear Iterative PArtial Least Squares (NIPALS)
Given: $\mathbf{X}_0 \leftarrow \mathbf{X}\,(n \times k)$
       $\mathbf{Y}_0 \leftarrow \mathbf{Y}\,(n \times d)$
       $\hat{\mathbf{Y}}_0 \leftarrow \mathbf{0}\,(n \times d)$
1)  Initialize score vector $\mathbf{u}_i$
    if $\mathbf{Y} = \mathbf{y}$ (1-dimensional)
       then $\mathbf{u}_0 \leftarrow \mathbf{y}$
       else $\mathbf{u}_0 \leftarrow$ random $n$-dimensional vector
2)  Iterate to convergence
    for $i = 1$ to $k$
    a) $\mathbf{w}_i = \mathbf{X}_{i-1}^T \mathbf{u}_{i-1} / (\mathbf{u}_{i-1}^T \mathbf{u}_{i-1})$
    b) $\mathbf{t}_i = \mathbf{X}_{i-1} \mathbf{w}_i$
    c) $\mathbf{c}_i = \mathbf{Y}_{i-1}^T \mathbf{t}_i / (\mathbf{t}_i^T \mathbf{t}_i)$
    d) $\mathbf{u}_i = \mathbf{Y}_{i-1} \mathbf{c}_i$
3)  Compute loading vectors
    $\mathbf{p}_i = \mathbf{X}_{i-1}^T \mathbf{t}_i / (\mathbf{t}_i^T \mathbf{t}_i)$
    $\mathbf{q}_i = \mathbf{Y}_{i-1}^T \mathbf{u}_i / (\mathbf{u}_i^T \mathbf{u}_i)$
4)  Compute model
    $\hat{\mathbf{Y}}_i = \hat{\mathbf{Y}}_{i-1} + \mathbf{u}_i \mathbf{q}_i^T$
5)  Deflate matrices $\mathbf{X}$ and $\mathbf{Y}$
    $\mathbf{X}_i \leftarrow \mathbf{X}_{i-1} - \mathbf{t}_i \mathbf{p}_i^T$
    $\mathbf{Y}_i \leftarrow \mathbf{Y}_{i-1} - \mathbf{u}_i \mathbf{q}_i^T$
6)  Ending condition
    if $\mathbf{X}_i^T \mathbf{X}_i = 0$
    then Exit
    end for
```

Variants

The name PLS applies to a very general statistical method. Indeed, PLS methodology can be seen as a family of procedures that includes a number of specific variants, that allow to achieve different tasks. Each PLS form is defined by the corresponding deflation scheme that is applied in the algorithm (Rosipal and Krämer, 2006 [56]). If the deflation is equal to

$$\mathbf{X} = \mathbf{X} - \mathbf{t}\mathbf{p}^T \text{ and } \mathbf{Y} = \mathbf{Y} - \mathbf{u}\mathbf{q}^T$$

as reported in the algorithm of Table 2.1 (step 5), the so defined PLS method is called PLS Mode A. This is the approach originally designed by H. Wold for modeling paths of causal relation between any number of variables blocks (Wold, 1975 [76]). In this case, each iteration of the algorithm applies a rank-one deflation of individual block matrices $\mathbf{X}$ and $\mathbf{Y}$ using corresponding score and loading vectors. This technique model connections between different blocks of variables whose relation is symmetric. Since PLS Mode A seems to be appropriate for modelling existing relations between sets of variables, it is similar to Canonical Correlation Analysis (CCA).
Otherwise, the following PLS variants can be defined applying the corresponding deflation schemes:

- PLS1 and PLS2

  $$\mathbf{X} = \mathbf{X} - \mathbf{t}\mathbf{p}^T \text{ and } \mathbf{Y} = \mathbf{Y} - \mathbf{t}\mathbf{t}^T\mathbf{Y}/(\mathbf{t}^T\mathbf{t}) = \mathbf{Y} - \mathbf{t}\mathbf{c}^T$$

  assuming that a linear inner relation between scores vectors $\mathbf{t}$ and $\mathbf{u}$ exists and is defined as

  $$\mathbf{U} = \mathbf{T}\mathbf{D} + \mathbf{H} \tag{2.4}$$

  where $\mathbf{D}$ is the $(m \times m)$ diagonal matrix and $\mathbf{H}$ denotes the matrix of residuals. In this approach score vectors $\mathbf{t}$ are good predictors of $\mathbf{Y}$. As a consequence, after extraction of score vectors $\mathbf{t}$, variables $\mathbf{Y}$ are regressed on $\mathbf{t}$. Moreover, deflation scheme removes a regression component from $\mathbf{Y}$ at each iteration of the algorithm. Here, weight vectors $\mathbf{c}$ are not scaled to unit norm.
  PLS1 and PLS2, that are the most frequently used PLS techniques, fit regression models (often denoted by acronym PLS-R), with one block of predictors and one block of responses. Indeed, relations between $\mathbf{X}$ and $\mathbf{Y}$ variables is asymmetric and computation of score vectors reflects that. In PLS1 one of the blocks of data consists of a single variable; in PLS2 both blocks are multidimensional. Furthermore, PLS1 and PLS2 deflation scheme of extracting one component at a time assures mutual orthogonality of score vectors $\mathbf{t}$. Finally, in PLS1 deflation of $\mathbf{y}$ is technically not needed during the iterations of PLS.

- PLS-SB
  In PLS-SB computation of all weight vectors $\mathbf{w}$ are done at once. This involves a sequence of implicit rank-one deflations. In contrast to PLS1 and PLS2, extracted score vectors $\mathbf{t}$ are in general not mutually orthogonal.

- SIMPLS
  SIMPLS, introduced by de Jong, avoids deflation steps at each iteration of PLS1 and PLS2 (de Jong, 1993 [16]). SIMPLS approach directly finds weight vectors $\widetilde{\mathbf{w}}$ which are applied to original not deflated matrix $\mathbf{X}$. Criterion of mutually orthogonal score vectors $\widetilde{\mathbf{t}}$ is kept. It has been shown that SIMPLS is equal to PLS1 but differs from PLS2 when applied to multidimensional matrix $\mathbf{Y}$.

These three variants are characterized by the asymmetric relation between different blocks of variables. For this reason, they all allow to make regression and they differ substantially, according to a theoretical perspective, from PLS Mode A (and the corresponding NIPALS algorithm of Table 2.1). Regarding regression, since PLS introduction, several different algorithms have been proposed that lead to the same sequence of models. Besides those listed above, the most elegant formulation is, perhaps, shown by Helland (Helland, 1988 [31]). Indeed, his work shows that for every iteration i the corresponding PLS model can be obtained by an OLS regression of the $\mathbf{Y}$ variable on the linear combination between $\mathbf{X}$ variables and covariance weights $\mathbf{w}_i$; that is score vectors $\mathbf{t}_i$, projection of $\mathbf{X}$ on latent space.

Eigenvalue problem

The NIPALS algorithm corresponds to traditional power iterations for finding dominant eigenvectors of a specific eigenvalue problem[6]. Indeed, the first weight $\mathbf{w}$ computed with NIPALS algorithm is equal to the first eigenvector of the following eigenvalue problem

$$\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w} = \lambda\mathbf{w} \tag{2.5}$$

where matrix involved in this equation $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$ is the squared covariance matrix between $\mathbf{X}$ and $\mathbf{Y}$. Because the rank of the above system is limited by number of instances n, $n < k$ yield a few dominant eigenvectors and hence PLS works best in this scenario.

Similarly, eigenvalue problems for extraction of $\mathbf{t}$, $\mathbf{u}$ or $\mathbf{c}$ estimates can be derived. User then solves for one of these eigenvalue problems and other score or weight vectors are readily computable using relations defined in NIPALS.

Considering PLS1 and PLS2 together with corresponding eigenvalue problem, some interesting properties can be highlighted. First of all singular values of cross product matrix $\mathbf{X}^T\mathbf{Y}$ equals to sample covariance values. Then, the first singular value of the deflated cross product matrix $\mathbf{X}^T\mathbf{Y}$ at iteration $i + 1$ is greater or equal than the second singular value of $\mathbf{X}^T\mathbf{Y}$ at iteration i. This result can be also applied to relation of eigenvalues of Equation 2.5 due to the fact that Equation 2.5 corresponds to singular value decomposition of transposed cross product matrix $\mathbf{X}^T\mathbf{Y}$. In particular, PLS1 and PLS2 algorithms differ from computation of all

[6] The Appendix A.1 provides a more technical description of a general eigenvalue problem.

eigenvectors of Equation 2.5 in one step.
In PLS-SB all eigenvectors of Equation 2.5 are computed at once. This method involves a sequence of implicit rank-one deflations of the overall cross product matrix.

2.2.3 Geometric representation

Algebraic definition of PLS by means of matrices and vectors allows to use geometric representation of these mathematical quantities in order to better understand information carried by data such as their structure, existing relations between variables and between instances but also to detect outliers. Indeed, an important aspect of PLS is the ability to easily visualize high-dimensional data through the set of extracted latent variables. For this reason some simple diagnostic PLS tools exist, as score and loading plots, that explain the meaning of all the results of PLS as components, score, loadings, residuals, etc.
Generally, in geometrical drawing of a dataset (matrix), number of variables (columns) defines dimensionality of the space, variables are represented by axes and every instance (row) corresponds to a point, where the former's values coincide with coordinates of the last. In the following the most important graphical tools are shown with the help of a simple example.
Let consider a very small dataset with one instance ($n = 1$). Three variables, denoted by $\mathbf{x1}, \mathbf{x2}, \mathbf{x3}$, are measured on it ($k = 3$). Their value is $1, 2, 3$ respectively. So, matrix $\mathbf{X}\,(1 \times 3)$ is equal to Table 2.2. With such data, a tridimensional space can be defined where each axis represents a variable and the instance becomes a point of the space. Its coordinates are equal to values of variables (Figure 2.1[7]).

Table 2.2: Dataset represented by **X** matrix.

X	**x1**	**x2**	**x3**
x	1	2	3

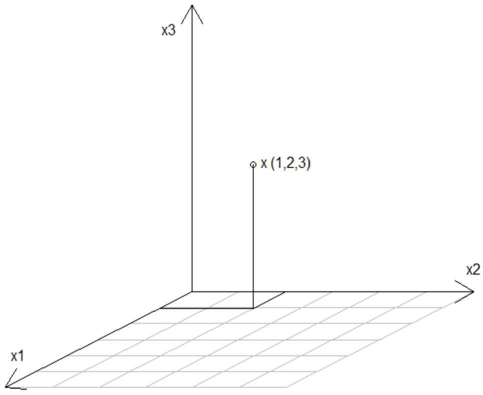


Figure 2.1: Tridimensional vectorial space for dataset **X**.

[7] All the figures of this Section are obtained with R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Table 2.3: PLS results.

(a) Score matrix.

$\mathbf{T}$	$\mathbf{t}_1$	$\mathbf{t}_2$
t	1	2

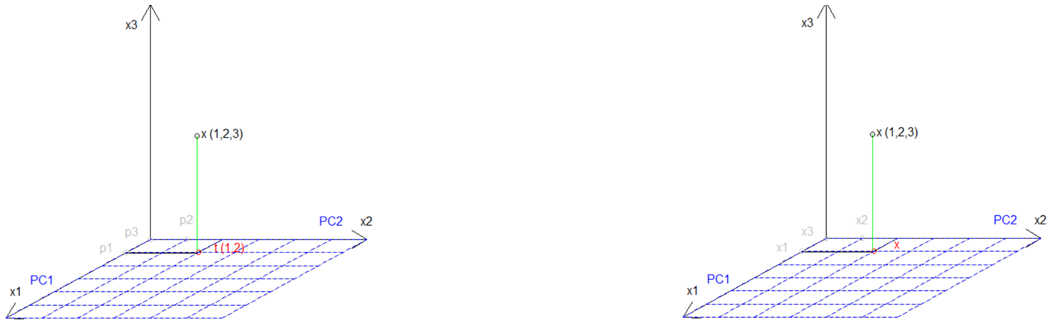
(b) Loading matrix.

$\mathbf{P}$	$\mathbf{p}_1$	$\mathbf{p}_2$
p1	1	0
p2	0	1
p3	0	0

(c) Residual matrix.

$\mathbf{E}$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
e1	0	0	3

Let suppose that PLS is applied to this dataset and that result is given by matrices $\mathbf{T}\,(1 \times 2)$, $\mathbf{P}\,(3 \times 2)$ and $\mathbf{E}\,(1 \times 3)$ as in Table 2.3. PLS defines a new vector space that is a subspace of the previous one in which original data are represented. Loading, score and residual matrices contain all the informations to draw this new space with its own observations and quantities. First of all, vectors $\mathbf{p}$ (columns of matrix $\mathbf{P}$) provide axes of the subspace and their position with respect to original space. Let denote with label PC these new axes that correspond to latent variables, sometimes called Principal Components. Moreover, rows of matrix $\mathbf{P}$ record the position of original variables in the new subspace. More formally, they hold projections of original variables (old axes) in the new subspace. These values are called loadings and their geometrical representation is named loading plot. Secondly, score matrix $\mathbf{T}$ collects coordinates of instance projected in the new subspace. These values are called scores and score plot is their graph. Finally, in matrix $\mathbf{E}$ there are residuals, defined as the difference between original values of instances and their projection. Information collected in matrices $\mathbf{T}$, $\mathbf{P}$ and $\mathbf{E}$ can be added to previous graph as Figure 2.2a shows for results of example. Blue, gray, red and green colours are chosen to highlight latent space, loading, score and residual values respectively. In this case, PLS defines a (blue) bidimensional horizontal plane with new axes position equals to $(1, 0, 0)$ and $(0, 1, 0)$ with respect to original tridimensional space. Projection of old



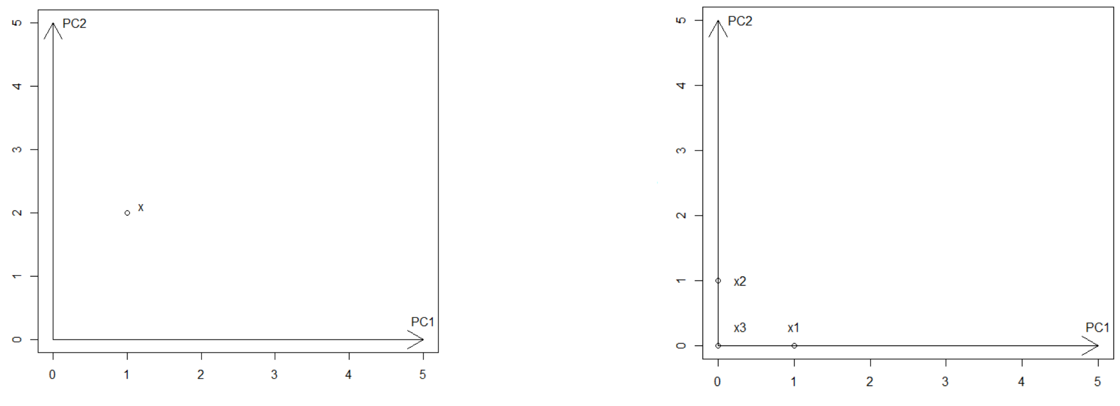
(a) Formal labels.

(b) Meaningful labels.

Figure 2.2: Tridimensional plots of PLS results.

variables are gray points $(1, 0)$; $(0, 1)$ and $(0, 0)$. The instance is projected into red point of the subspace with new coordinates $(1, 2)$ and residual is the green segment of length 3. Figure 2.2b shows same results but its labels extend information carried by plot. Indeed, these labels link points to corresponding elements of dataset, instances or variables, making easier plot's interpretation and description of dataset. This labelling is used in score and loading plots usually produced by most common softwares that assure PLS computation. Moreover, they generally provide several other graphical tools that can be managed in order to build every kind of useful plot.

Without loss of generality, these premises can be extended to every situation in which PLS is applied, hopefully characterized by more than one instance and usually with a very high number of variables so that PLS can compute several components. In these cases, in order to assure a clear and meaningful rendering of figures despite graphical difficulties of drawing more than tridimensional space, score and loading plot are normally represented as bidimensional graphs where the axis are two chosen latent variables. Then, different pairs of latent components can be considered as axis, creating a sequence of score and loading plots, surely more readable than multivariate graphs. Figure 2.3 shows an example with suggested simple data. Position of instances and variables in the latent space defined by PC can be easily recognized observing these plots. Considering points in score plot, one can generally infer if instances are similar or how much they differ. Indeed, groups of neighbouring points represent data with similar features; whereas distant points are quite different observations. Analysis of loading plot allows to understand which variables are important in considered and displayed PC. Indeed, points near origin of axes represent variables that enter in PC with a low weight, so that they are not meaningful for these PC. Otherwise, if a variable has a high value along a PC, this means that it is substantial and heavily considered in the PC computing. A further step allows



(a) Score plot.

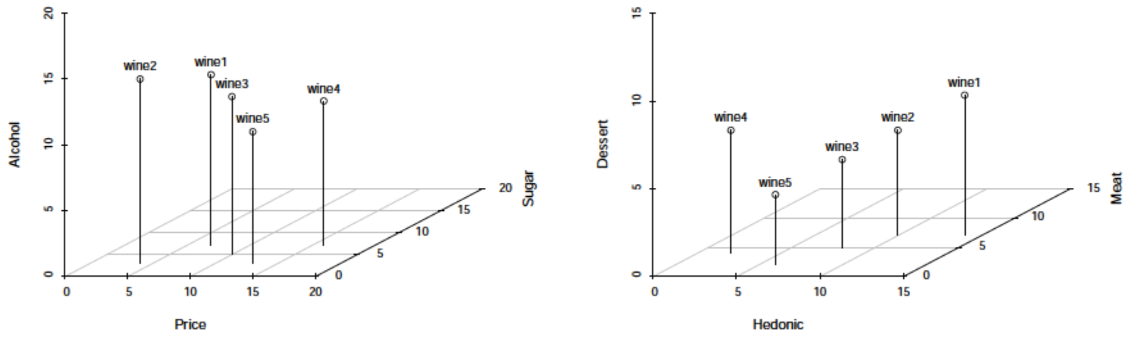
(b) Loading plot.

Figure 2.3: Bidimensional plot of PLS results.

Table 2.4: Dataset of a simple example of PLS regression.

	X				Y		
	Price	Sugar	Alcohol	Acidity	Hedonic	Meat	Dessert
wine1	7	7	13	7	14	7	8
wine2	4	3	14	7	10	7	6
wine3	10	5	12	5	8	5	5
wine4	16	7	11	3	2	4	7
wine5	13	3	10	3	6	2	4

to make a better interpretation of dataset comparing scores and loadings, overlapping ideally score and loading plot, since they are two different pictures of the same space. Indeed, if information extracted by both score and loading plot are mixed, one can say not only which observations are more or less similar but also why these differences exist, controlling which variables are more important for latent variables. After this preliminary and partial example, a simple case of PLS regression will be considered. Graphical outputs again help to understand more quickly and more easily information carried by the numerical results. As a consequence, with the helpful support of graphical tools analysis, evaluations and choice can be made. Let assume that a set of five wines are rated by a panel of experts on three aspects. The dependent variables record so the subjective evaluation about likeability of a wine, and how well it goes with meat, or dessert. In addition, there are four predictors: price, sugar, alcohol, and acidity content of each wine. The complete dataset[8] is shown in Table 2.4 and in Figure 2.4. In particular, the plot on the left illustrates the obser-



(a) Tridimensional **X** space (last variable excluded).

(b) Tridimensional **Y** space.

Figure 2.4: An example of graphical representation for the wine dataset.

[8] Data of this example are taken from H. Abdi, Partial Least Squares (PLS) Regression. In: Lewis-Beck M., Bryman, A., Futing T. (Eds.) (2003). *Encyclopedia of Social Sciences Research Methods.* Thousand Oaks (CA): Sage.

Table 2.5: Matrices of weights **W** and **C**.

(a) **W** matrix.

W	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$
Price	−0.5137	−0.3379	−0.3492
Sugar	0.2010	−0.9400	0.1612
Alcohol	0.5705	−0.0188	−0.8211
Acidity	0.6085	0.0429	0.4218

(b) **C** matrix.

C	$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$
Hedonic	0.6093	0.0518	0.9672
Meat	0.7024	−0.2684	−0.2181
Dessert	0.3680	−0.9619	−0.1301

Table 2.6: Score matrix **T** and loading matrix **P**.

(a) **T** matrix.

T	$\mathbf{t}_1$	$\mathbf{t}_2$	$\mathbf{t}_3$
wine1	0.4538	−0.4662	0.5716
wine2	0.5399	0.4940	−0.4631
wine3	0	0	0
wine4	−0.4304	−0.5327	−0.5301
wine5	−0.5633	0.5049	0.4217

(b) **P** matrix.

P	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$
Price	−1.8706	−0.6845	−0.1796
Sugar	0.0468	−1.9977	0.0829
Alcohol	1.9547	0.0283	−0.4224
Acidity	1.9874	0.0556	0.2170

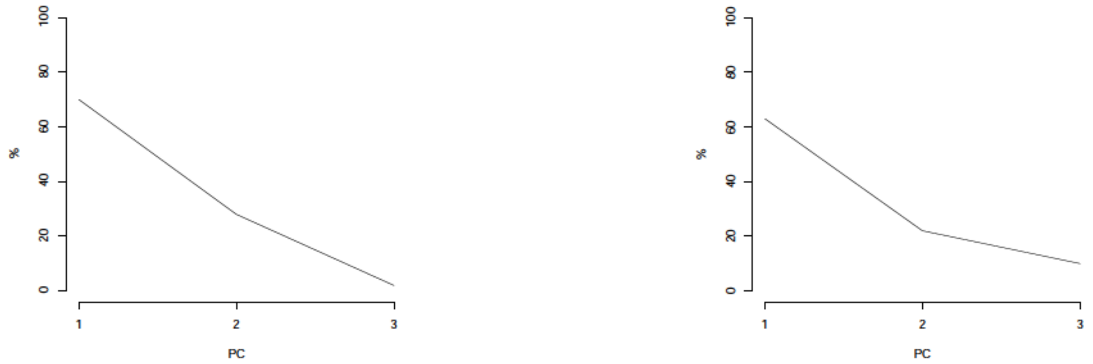
vations in the tridimensional space defined by the first three independent variables (Figure 2.4a) while the plot on the right represents the data on the tridimensional space of **Y** variables (Figure 2.4b). Let now suppose to apply a PLS regression on these data, setting the maximum number of latent variables equal to three and choosing a proper validation method. After the computations, all the output matrices can be viewed and numerical results displayed in plots that allow to evaluate them in a simpler and faster way. Matrices of weights **W** and **C** are reported in Table 2.5 With these matrices the latent variables can be computed and original data can be then projected on the new latent space. As a consequence matrices of scores and loadings can be computed for both **X** and **Y** variables and corresponding results are shown in Table 2.6 and Table 2.7. In addition, also explained variance is computed and it should be taken into account (Table 2.8 and Figure 2.5). Let note that the residual variance is strictly related to explained variance, since their sum is by definition equal to one. As a consequence, softwares usually compute both types of variance and switch between them can be done.

Table 2.7: Score matrix for **Y** variable.

U	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$
wine1	1.9451	−0.7611	0.6191
wine2	0.9347	0.5305	−0.5388
wine3	−0.2327	0.6084	0.0823
wine4	−0.9158	−1.1575	−0.6139
wine5	−1.7313	0.7797	0.4513

Table 2.8: Variance of **X** and **Y** explained by the latent vectors.

PC	**X** Percentage	**X** Cumulative	**Y** Percentage	**Y** Cumulative
1	70	70	63	63
2	28	98	22	85
3	2	100	10	95

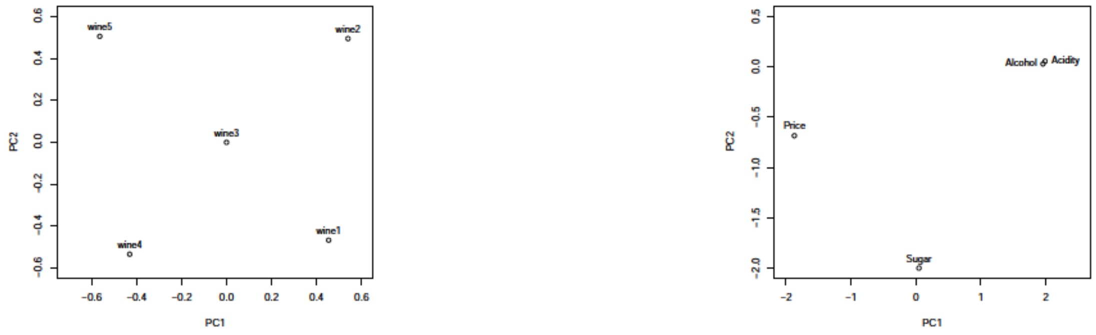


(a) Explained variance for **X**.

(b) Explained variance for **Y**.

Figure 2.5: Percentage of explained variance.

Observing these values and plots, one can find that two latent vectors explain 98% of the variance of **X** and 85% of **Y**. This suggests to keep these two dimensions for the final solution. For this reason scores and loadings can be analysed by means of these two PC, so that score and loading plots focus on the first two latent vectors (Figure 2.6).



(a) Score plot.

(b) Loading plot.

Figure 2.6: Score and loading plots for **X** variables.

2.3 PLS tasks

In this Section PLS tasks are considered and described. Indeed, PLS can be used as a regression or classification tool as well as for dimension reduction. Moreover, PLS deals also with nonlinearity[9].

Regression, as previously said, includes any technique for modelling an asymmetric relation between two blocks of variables, labels and inputs, where the former are constrained to be continuous values. Linear regression is the most common approach and assumes that the structural form of the relation is linear, i.e. the conditional mean (seldom median or some other quantile) of response variable given the value of explanatory variables is an affine function with respect to parameters. PLS, as regression tool, models a linear relation between few latent variables, instead of a large number of measured variables and in this way it simplifies the original system.

Classification differs from regression in the kind of the label taken into account. Indeed, it restricts to categorical response variables whose values correspond to classes, levels, modalities, or categories. PLS applies classification by encoding the class membership in an appropriate indicator matrix. Then, after extraction of relevant latent vectors, the best classifier is produced by the algorithm.

As regards dimensionality reduction, among many advantages of PLS approach there is the ability to analyze importance of individual observed variables potentially leading to deletion of irrelevant ones. This mainly occurs in the case of experimental design where many insignificant terms are measured. In such situations, PLS can guide the practitioner into more compact experimental settings with a significant cost reduction and without a high risk associated with the "blind" variables deletion.

Considering nonlinearity, PLS can be a competing alternative to traditional methods, even if it originally assumes linear relations between variables. Indeed, classical approach of PLS can be extended for modelling nonlinearity by two major methodologies. One is based on a nonlinear function between scores still computed as linear combinations of original variables; the other applies a nonlinear mapping to built a subspace where linear PLS can be applied.

2.3.1 Regression

In order to define PLS regression (PLS-R), PLS decomposition (Equation 2.2) should be combined with the assumption of a linear relation between scores vectors **T** and **U** (Equation 2.4). Considering the case of a unidimensional response variable

[9] Since the previous Chapter (Section 1.1) explained all these topics in their general form, only the main concepts are briefly recalled here. Then, the following Subsections will describe with more details how PLS implements the different tasks.

$\mathbf{y}$ the former result becomes $\mathbf{y} = \mathbf{Uq} + \mathbf{f}$, so that the following relation holds

$$\begin{aligned} \mathbf{y} &= \mathbf{TDq} + \mathbf{Hq} + \mathbf{f} \\ &= \mathbf{Tc} + \mathbf{f}^* \end{aligned} \tag{2.6}$$

This equation is simply the decomposition of $\mathbf{Y}$ using OLS regression[10] on orthogonal predictors $\mathbf{T}$. The m-dimensional regression coefficient vector[11] $\mathbf{c} = \mathbf{Dq}$ collects the not scaled to length one weights $\{c_i\}_1^m$ and $\mathbf{f}^* = \mathbf{Hq} + \mathbf{f}$ denotes the n-dimensional residual vector.

It is useful to redefine Equation 2.6 in terms of original predictors $\mathbf{X}$. To do this, orthonormalized score vectors, are now considered, that is, $\mathbf{T}^T\mathbf{T} = \mathbf{I} = \mathbf{U}^T\mathbf{U}$. Moreover, $\mathbf{W}$ denotes the $(k \times m)$ matrix that collects vectors of weights $\mathbf{w}$ computed as coefficients of regression between $\mathbf{X}$ and score vectors $\mathbf{u}$ in all m iterations of NIPALS algorithm. Assuming without loss of generality that $\mathbf{X} = \mathbf{TP}^T$ from Equation 2.2 and multiplying by $\mathbf{W}$, $\mathbf{T}$ can be obtained as in the following steps

$$\begin{aligned} \mathbf{X} =& \mathbf{TP}^T \\ \mathbf{XW} =& \mathbf{TP}^T\mathbf{W} \\ \mathbf{T} =& \mathbf{XW}(\mathbf{P}^T\mathbf{W})^{-1} \end{aligned}$$

Now, Equation 2.6 becomes

$$\mathbf{y} = \mathbf{XW}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{c} + \mathbf{f}^*$$

With assumption of orthonormalized score vectors the following three results can be simply obtained. From Equation 2.6 it holds that $\mathbf{c} = \mathbf{T}^T\mathbf{y}$. Similarly, $\mathbf{P}^T = \mathbf{T}^T\mathbf{X}$ is computed from relation $\mathbf{X} = \mathbf{TP}^T$. Whereas $\mathbf{W} = \mathbf{X}^T\mathbf{U}$ by definition. So, relation will be equivalent to

$$\mathbf{y} = \mathbf{XX}^T\mathbf{U}(\mathbf{T}^T\mathbf{XX}^T\mathbf{U})^{-1}\mathbf{T}^T\mathbf{y} + \mathbf{f}^*$$

Usually this relation is easily written as

$$\begin{aligned} \mathbf{y} &= \mathbf{Xb}_{PLS} + \mathbf{f}^* \\ \hat{\mathbf{b}}_{PLS} &= \mathbf{X}^T\mathbf{U}(\mathbf{T}^T\mathbf{XX}^T\mathbf{U})^{-1}\mathbf{T}^T\mathbf{y} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{c} \end{aligned} \tag{2.7}$$

where $\hat{\mathbf{b}}_{PLS}$ represents the k-dimensional vector of PLS regression coefficients. Different scalings of the individual score vectors $\mathbf{t}$ and $\mathbf{u}$ do not influence $\mathbf{b}_{PLS}$.

[10] Further details about OLS are given in Section 3.2 that presents a complete overview of this method.

[11] Attention should be payed noting that the m-dimensional vectors (so columns by definition) $\mathbf{q}$ and $\mathbf{c}$ are a row of the $(d \times m)$ matrices $\mathbf{Q}$ and $\mathbf{C}$, respectively.

2.3.2 Classification

PLS can be used also for classification analysis (Barker and Rayens, 2003 [2]). In the following, preliminary assumptions that define background traditionally used in classification analysis are presented.

The $(n \times k)$ matrix $\mathbf{X} = \{\underline{\mathbf{x}}_i \in \mathcal{X} \subset I\!\!R^k\}_{i=1}^n$ collects all data of the set of n instances. It is still assumed to be zero mean. Moreover, every observation belongs to a class: G denotes the number of classes and ${}_gn$ is the number of instances in the g-th class, so that $\sum_{g=1}^G {}_gn = n$. Statistical indexes can be computed both for every group and for the whole dataset. For example, if ${}_g\underline{\mathbf{x}}_i$ represents the k-dimensional vector for the i-th observation in the g-th class,

$$ {}_g\bar{\mathbf{x}} = \frac{1}{{}_gn}\sum_{i=1}^{{}_gn} {}_g\underline{\mathbf{x}}_i \qquad \bar{\mathbf{x}} = \frac{1}{n}\sum_{g=1}^{G}\sum_{i=1}^{{}_gn} {}_g\underline{\mathbf{x}}_i = \frac{1}{n}\sum_{g=1}^{c} {}_gn_g\bar{\mathbf{x}} $$

are the k-dimensional vectors that record for every variable its sample mean in the g-th group and its sample mean of the whole matrix $\mathbf{X}$, respectively. Furthermore, the $(k \times k)$ matrices

$$ \mathbf{SS_b} = \sum_{g=1}^{G} {}_gn({}_g\bar{\mathbf{x}} - \bar{\mathbf{x}})({}_g\bar{\mathbf{x}} - \bar{\mathbf{x}})^T \qquad \mathbf{SS_w} = \sum_{g=1}^{G}\sum_{i=1}^{{}_gn}({}_g\underline{\mathbf{x}}_i - {}_g\bar{\mathbf{x}})({}_g\underline{\mathbf{x}}_i - {}_g\bar{\mathbf{x}})^T \quad (2.8) $$

define the *between-classes* and *within-classes* sums of squares, respectively. Between, among or inter class are synonymous, whereas within equals intraclass. The $(n \times G - 1)$ matrix $\mathbf{Y}$ records for every observation its class membership. It is a collection of dummy variables that represent which group every instance belongs to. It is equal to

$$ \mathbf{Y} = \begin{pmatrix} \mathbf{1}_{{}_1n} & \mathbf{0}_{{}_1n} & \dots & \mathbf{0}_{{}_1n} \\ \mathbf{0}_{{}_2n} & \mathbf{1}_{{}_2n} & \dots & \mathbf{0}_{{}_2n} \\ \vdots & \vdots & \ddots & \mathbf{1}_{{}_{G-1}n} \\ \mathbf{0}_{{}_Gn} & \mathbf{0}_{{}_Gn} & \dots & \mathbf{0}_{{}_Gn} \end{pmatrix} $$

where $\mathbf{0}_{{}_gn}$ and $\mathbf{1}_{{}_gn}$ are $({}_gn \times 1)$ vectors of all zeros and ones, respectively. The matrix $\mathbf{Y}$ is zero mean, too. Two-class classification corresponds to one-dimensional $\mathcal{Y}$-space.

With these assumptions, previously proposed orthonormalized PLS method can be modified in order to define a special case of PLS that reveals useful for classification purposes. One-dimensional $\mathcal{Y}$-space is firstly considered. The optimization problem becomes

$$ \max_{||\mathbf{w}||=||c||=1} [cov(\mathbf{Xw}, \mathbf{y}c]^2 = \max_{||\mathbf{w}||=1} var(\mathbf{Xw})[corr(\mathbf{Xw}, \mathbf{y})] $$

Here $\mathcal{Y}$-space penalty $var(\mathbf{y})$ is not meaningful. As a consequence, it can be removed and only $\mathcal{X}$-space variance is involved. This modified PLS method is based on eigensolutions of *among-classes* sum of squares matrix $\mathbf{SS_b}$. Eigenvalue problem (Equation 2.5) with orthonormalized PLS is transformed into

$$\mathbf{X}^T\mathbf{y}(\mathbf{y}^T\mathbf{y})^{-1}\mathbf{y}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\widetilde{\mathbf{y}}\widetilde{\mathbf{y}}^T\mathbf{X}\mathbf{w} = \lambda w \tag{2.9}$$

where

$$\widetilde{\mathbf{y}} = \mathbf{y}(\mathbf{y}^T\mathbf{y})^{-1/2}$$

is the vector that represents uncorrelated and normalized $\mathbf{y}$ variable. Using the following relation

$$(n-1)\mathbf{S_{Xy}}\mathbf{S_y}^{-1}\mathbf{S_{Xy}}^T = \mathbf{SS_b}$$

eigenvectors of Equation 2.9 are equivalent to eigensolutions of

$$\mathbf{SS_b}\mathbf{w} = \lambda\mathbf{w} \tag{2.10}$$

which corresponds to maximizing the *between-class* variance. Interestingly, in case of two-class classification, direction of the first orthonormalized PLS score vector $\mathbf{t}$ is identical to the first score vector found by either the PLS1 or PLS-SB methods. This immediately follows from the fact that $\mathbf{y}^T\mathbf{y}$ is a number here. In this two-class scenario $\mathbf{X}^T\mathbf{y}$ is of a rank-one matrix and PLS-SB extract only one score vector $\mathbf{t}$. In contrast, orthonormalized PLS can extract additional score vectors, up to the rank of $\mathbf{X}$.

In case of multi-classes classification, rank of $\mathbf{Y}$ matrix is equal to $G-1$ which determines the maximum number of score vectors that may be extracted by orthonormalized PLS-SB method[12]. Again, similar to one-dimensional $\mathcal{Y}$-space deflation of $\mathbf{Y}$ matrix at each step can be done using score vectors $\mathbf{t}$ of PLS2. Consider this deflation scheme in $\mathcal{X}$ and $\mathcal{Y}$-spaces

$$\begin{aligned}&\mathbf{X}_i = \mathbf{X} - \mathbf{t}\mathbf{p}^T = (\mathbf{I} - \mathbf{t}\mathbf{t}^T/(\mathbf{t}^T\mathbf{t}))\mathbf{X} = \mathbf{P}_i\mathbf{X}\\ &\widetilde{\mathbf{Y}}_i = \mathbf{P}_i\widetilde{\mathbf{Y}}\end{aligned}$$

where $\mathbf{P}_i = \mathbf{P}_i^T\mathbf{P}_i$ represents a projection matrix. Using these deflated matrices $\mathbf{X}_i$ and $\widetilde{\mathbf{Y}}_i$ eigenproblem 2.9 can be written in the form

$$\mathbf{X}_i^T\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T\mathbf{X}_i\mathbf{w} = \lambda\mathbf{w}$$

Thus, similar to the previous two-class classification, solution of this eigenproblem can be interpreted as the solution of Equation 2.10 using the *among-classes* sum of squares matrix now computed on deflated matrix $\mathbf{X}_i$.

[12] It is considered here that $G \leq k$, otherwise number of score vectors is given by k.

2.3.3 Dimension reduction and nonlinear PLS

Nowadays current research leads to the use of new instruments that capture data properties better. Due to the wide application of such improved tools in several real situations, amount of available data has then grown fast in the last years. However, with these tall and fat datasets (large n and k, number of samples and variables respectively, usually with $n << k$), traditional regression algorithms scale often poorly. Computers must indeed handle huge amounts of data per second and presenting such inputs directly to a complex method may be computationally infeasible. Beside this, another problem can appear, since collinearity is usually present in so big datasets.

In order to apply successfully a regression or classification technique improving both algorithmic scaling and multicollinearity, the most direct solution consists of three steps: i) finding a new restricted set of variables where multicollinearity is removed; ii) projecting original observed data onto this low-dimensional subspace; iii) analyzing projections in this space, i.e. building a regression model or a classifier with this new representation. The first step is accomplished by dimension reduction techniques and one way to reach its goal is creating new variables, called latent vectors, as linear combination of the original ones in such a way that collinearity is deleted[13]. Dimension reduction can be defined also as a preprocessing tool since it is usually a preliminary step of a complete data analysis with the aim of data modelling. At the end, resulting projections onto just defined low dimensional space can be used with any standard regression or classification method.

Although originally proposed as a regression method, PLS is a dimensionality reduction technique, widely used in different problems because of its effective performance. PLS is indeed a powerful tool that provides dimensionality reduction for even hundreds of thousand of variables that are reduced to a much smaller subspace (with a dimension order of tens). In particular, PLS dimensionality reduction is performed by obtaining weight vectors $\mathbf{w}$ that allow to define directions of the new subspace. Then, observed data $\mathbf{X}$ can be projected by weight matrix $\mathbf{W}$. Moreover, since PLS method accounts for information of $\mathbf{Y}$ space, it can be defined supervised or class aware dimensionality reduction technique, as opposed to traditional ones that are unsupervised. In addition, PLS finds new variables that also preserve useful information. These features are then used as inputs to regression algorithm and they can prove very effective. Finally, PLS dimension reduction might also be performed in order to speed up computation. PLS finds indeed latent variables that are fast to compute.

In many areas of research and industrial situations data can exhibit nonlinear behaviours. Among other methods, also PLS can be used to model such nonlinearity. However, since PLS technique is originally based on linear assumptions, it should be

[13] This strategy belongs to the family of unsupervised methods called feature extraction as seen in the previous Chapter (Subsection 1.1.1).

slightly modified in order to be effective. As a consequence, some further hypothesis should be made in addition to its base statements[14]. In particular, two major methodologies exist for modelling nonlinearity. First group of approaches is based on reformulating considered linear relation 2.4 between score vectors $\mathbf{t}$ and $\mathbf{u}$ by a nonlinear model

$$\mathbf{u} = g(\mathbf{t}) + \mathbf{h} = g(\mathbf{X}, \mathbf{w}) + \mathbf{h} \tag{2.11}$$

where $g(\cdot)$ represents a continuous function modelling existing nonlinear relation. Again, $\mathbf{h}$ denotes a vector of residuals. Polynomial functions, smoothing splines, artificial neural networks or radial basis function networks have been used to model $g(.)$. Assumption that score vectors $\mathbf{t}$ and $\mathbf{u}$ are linear projections of original variables is kept. This leads to the necessity of a linearization of the nonlinear mapping $g(\cdot)$ by means of Taylor series expansions and to the successive iterative update of weight vectors $\mathbf{w}$.

Second approach to nonlinear PLS is based on a mapping of original data by means of a nonlinear function to a new representation (data space) where linear PLS is applied. Following this idea, nonlinear PLS can potentially be defined using for instance kernel trick. Recently developed theory of kernel-based learning can indeed be also applied to PLS. This approach allows to extend linear data analysis tools and to deal with nonlinear aspects of measured data. Moreover, powerful machinery of kernel PLS keeps computational and implementation simplicity of linear PLS and at the same time has been proven to be competitive with respect to other traditional kernel-based regression and classification methods.

2.4 PLS applications

PLS is considered nowadays a very powerful versatile data analytical tool in many research areas and industrial applications where it is used in order to analyze real world data and to understand existing relations between them. PLS is indeed characterized by some interesting properties that favour PLS method to be applied successfully in many actual situations. First of all, PLS solves problems due to collinearity of data and limited sample size, two common features of available datasets. Then, it does not require any assumption for data distribution. Thirdly, it assures computational and implementation simplicity with huge datasets, too. PLS history confirms such multidisciplinary applicability even if, despite its advantages, initially PLS did not develop easily. Its expansion was indeed slowed down because it was difficult to position PLS within a statistical context.

As previously said, PLS was introduced by Wold since sixties, when he worked

[14] Here basic ideas about nonlinear PLS are being presented, whereas Chapter 4, after a general overview, gives more details in particular in Section 4.3. In addition, Rosipal has provided a complete description of nonlinear PLS in his recent work (Rosipal, 2011 [55]).

on Econometric models related to estimation methods for systems of simultaneous equations[15]. Indeed, unlike his contemporary colleagues that preferred maximum likelihood, Wold studied different estimation techniques based on least squares and using iterative procedures. During seventies Wold continued to apply PLS technique to new issues, but without remarkable results. In the eighties, research interest in PLS shifted to questions of Chemistry into what is now known as Chemometrics (application of statistical methods to chemical data). The person responsible of this transition was the son of Wold. In 1983 S. Wold together with H. Martens adapted NIPALS to solve the problem of collinearity in linear regression models. This was the turning point. PLS had indeed firstly received a great amount of attention in Chemometrics, where the algorithm became a standard tool for processing a wide range of chemical, both organic and analytical, data problems. As a consequence, popularity of PLS resulted in a lot of implementations in other scientific fields. Now PLS is applied in the most different subjects, from Marketing to Engineering, from Medicine to Computer Science. Hereinafter some examples are presented.

2.4.1 Chemometrics

Chemometrics is the science of extracting information from chemical systems through data driven tools. It solves indeed problems in chemistry and related experimental life sciences as biochemistry, medicine, biology and chemical engineering. Recently it has emerged with a focus on analyzing data originating mostly from organic and analytical chemistry, food research, and environmental studies. Moreover, it is highly interdisciplinary and it uses methods frequently employed in core data analytic subjects such as multivariate Statistics, applied Mathematics, and Computer Science. In particular, statistical methodology has been successfully applied to many types of chemical problems for some time. Experimental design techniques have had for instance a strong impact on understanding and improving industrial chemical process. However, chemometricians suggest also their own techniques based on heuristic reasoning and intuitive ideas and there is a growing body of empirical evidence that they perform well in many situations. As a consequence, standard chemometric methodologies are widely used industrially. Moreover, analytical instrumentation and methodology are improved following developments of chemometric methods. At the same time academic groups continue to study chemometric theory and applications. For this reason, Chemometrics can be defined an application driven discipline.

Aim of Chemometrics is developing a model which can be used with descriptive or predictive purposes. In the first case, properties of chemical systems are modeled with the intent of describe, understand and identify the underlying relations and structure of the studied phenomenon. They include data exploration through prin-

[15] Before, Wold was already outstanding in Social Science arguments as statistical demand analysis or utility theory.

cipal components and cluster analysis, as well as modern computer graphics. On the other hand, the model is developed to predict properties and behaviour of interest. Predictive modelling (regression and classification) is indeed an important goal in most applications. For this reason, regression analysis on observational data forms a major part of chemometric studies.

Chemometric analyses are based on measured features of the chemical system as pressure, flow, temperature, infrared/visible/mass spectra. Such datasets can be small but they are often very large and highly complex, involving hundreds to thousands of both variables and observations, even if usually they are characterized by many measured variables on each of a few cases, so that number of variables k greatly exceeds observation count n. Furthermore, variables resulting from digitization of analog signals as infrared/UV/visible spectroscopy, mass spectrometry, nuclear magnetic resonance, atomic emission/absorption and chromatography experiments are all by nature highly collinear.

However, even if datasets may be highly multivariate, a strong and often linear low-rank structure is present. As a consequence, these data suggest the use of techniques such as PLS. PLS has firstly succeeded in Chemometrics as an alternative to OLS in the poorly or ill conditioned problems encountered there. Then, PLS has been shown over time very effective at empirically modelling the more chemically interesting latent variables, exploiting their interrelations in data, and providing alternative compact coordinate systems for further numerical analyses. So, PLS has been heavily promoted in corresponding literature and it became the most popular regression method used in Chemometrics. Finally, after that PLS was heavily used in chemometric applications for many years, it began to find regular use in other fields.

In nearly all chemometric analyses, variables are standardized (autoscaled). Then computations are applied to standardized quantities and resulting solutions transformed back to reference original locations and scales. As a consequence, regression methods are always assumed to include constant terms, thus making them invariant with respect to variable locations, so that translating them to all have zero means is simply a matter of convenience or numerics. Most of these techniques are not, however, invariant to the relative scaling of variables. So, choosing them to all have the same scale is a deliberate choice on the part of the user. A different choice would give rise to different estimated models.

Main advantages of multivariate tools as PLS are that fast, cheap, or nondestructive analytical measurements can be used to estimate sample properties which would otherwise require time consuming, expensive, or destructive testing. Equally important is that multivariate methods allow for accurate quantitative analysis in presence of heavy interference by other analytes. Selectivity of analytical techniques is provided as much by mathematical computations, as analytical measurement modalities. For example Near InfraRed (NIR) spectra, which are extremely broad and nonselective compared to other analytical measurements such as infrared spectra, can often be used successfully in conjunction with carefully developed multivariate methods to predict concentrations of analytes in very complex matrices.

Examples of chemometric applications include analysis of spectroscopic data that consist of multi-wavelength spectral measurements. In analytical chemistry, spectroscopy are applied on solutions with known concentrations of a given compound. Then, multivariate regression techniques, such as PLS, are used to relate concentration for the analyte of interest to infrared spectrum. Once a regression model is built, unknown concentration can be predicted for new samples, using the spectroscopic measurements as predictors. The advantage is obvious if the concentration is difficult or expensive to measure directly.

2.4.2 Computer vision

PLS has been nowadays successfully adopted also in several applications of Computer Science as for example bioinformatics and computer vision. Among others, core business of this last discipline is the development of devices that involve people's locations and movements. Such tools produce many pattern recognition problems as image processing, human detection and face recognition. As a consequence, over the last few years the issue of detecting humans in single images has received considerable interest and significant research has been devoted to locating and tracking people in images and videos. Then, variations in illumination, shadows, and pose, as well as frequent inter- and intra-person occlusion render the definition of effective techniques for human detection a challenging task.

Two main approaches to human detection have been recently explored. The first class of methods consists of a generative process where detected parts of the human body are mixed together according to a prior human model. The second group considers purely statistical analyses that combine a usually very big set of low level variables within a detection window in order to classify the window as containing a human or not. In particular, every detection window is firstly decomposed into overlapping blocks. Then, for each block of the window, many variables are extracted. Once the variables extraction process is performed for all blocks inside a detection window, variables are concatenated to construct an extremely high-dimensional vector.

As regards typical variables related to human detection problems, the main information is collected by grids of Histograms of Oriented Gradient (HOG). Humans in standing positions have distinguishing features, such as strong vertical edges that are present along the boundaries of the body. HOG descriptors look at the spatial distribution of the edge orientations and capture edge or gradient structures that are characteristic of local shape. Moreover, HOG has some nice properties. Since it is computed for regions of a given size within the detection window, it is robust to some location variability of body parts and it is also invariant to rotations smaller than the orientation bin size. In addition, HOG is a low level variable which outperforms features as wavelets and shape contexts. Other sources of information reveal very useful in order to avoid a number of false positive detections. Indeed, a strong set of variables can provide high discriminatory power, reducing the need for complex classification methods. As a consequence, a significant improvement in human detection can be achieved using different types or combinations of low level features. These complement classical variables and are, for example, homogeneity of textures and typical colour of both human clothing and background, or particularly skin colour. Both clothing and background textures are generally uniform but the former is very different from natural textures observed outside of the body due to constraints on the manufacturing of printed cloth. As a consequence, variables are extracted from co-occurrence matrices, a method widely used for texture analysis, since they provide information regarding homogeneity and directionality of patches. Co-occurrence matrices represent second order texture information, i.e. the joint probability distribution of gray-level pairs of neighboring pixels in a block. Descriptors used are for example angular second moment, contrast, correlation, variance, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, and directionality. The second type of information captured is colour. Although colours may not be consistent due to variability in clothing, certain dominant colours are more often observed in humans, mainly in the face/head regions. So discriminatory colour information is found both in cloths and in the face/head regions. In order to incorporate colour, the original HOG is used to extract a descriptor called colour frequency. In spite of its simplicity, colour frequency increases detection performance.

Typical low level variables of human detection are so the original HOG descriptors with additional colour information, called colour frequency, and texture variables computed from co-occurrence matrices. A consequence of such variable augmentation is an extremely high dimensional space, rendering many classical Machine Learning techniques, such as SVM, intractable. In contrast, the number of observations in the dataset is much smaller. Furthermore, variables are extracted from neighboring blocks within a detection window, which increases the multicollinearity of data. Since the number of detection windows within an image is very high (ten of thousands for a 640×480 image scanned at multiple scales), it is crucial to obtain good detection results at very small false alarm rates.

These properties of datasets related to pattern recognition problems make an ideal setting for some multivariate methods such as PLS. Indeed, in this case high dimensional vector should be firstly analyzed by a technique that reduces its dimensionality. Then, a simple and efficient classifier can be used to evaluate the window as containing a human or not. More formally, PLS projects high dimensional vector onto a set of weights resulting in a lower vector that can be handled by traditional classification methods. Recently, PLS methods have been used for many image processing, human detection and face recognition problems. Here, PLS is primarily used as a supervised dimensionality reduction tool to effectively combine several low level features. PLS assures indeed a better learning enhancing detection and recognition and adapted PLS was shown to greatly improve the performance, especially for two-class problems. Finally, PLS method outperforms other approaches that utilize many more sources of information such as depth maps, ground plane estimation, and occlusion reasoning (Schwartz et al., 2009 [61]).

Other examples of PLS application in pattern recognition problems are speaker recognition, experiments on finger movement detection, and cognitive fatigue prediction (Srinivasan et al., 2011 [63]).
Considering the last situation, a significant reduction of the recording electrodes have been achieved by the use of a PLS model without the loss of classification accuracy. As regards speaker recognition, it deals with the task of verifying a speaker's claimed identity from a sample utterance based on a number of training utterances for which the speaker is known. However, the speech data, apart from carrying the speaker specific characteristics, is often subject to noise and reverberation, since it also encapsulates phonemic content, channel variability, and session variability. Nevertheless, variability in the phonemic content can be removed by posing the problem of recognition over a collection of data spanning several utterances. As a consequence, speaker recognition systems should be able to learn the between class separability in a supervector setting where substantial progress in rejecting channel/session variability has been made via several techniques. So, the goal in supervector space is to discriminate between a speaker and impostors by accounting for the speaker variability while ignoring nuisance information. Commonly, only a few (often one) speeches from a very large database belong to the target speaker, which necessitates the use of a method capable of learning from not many observations in a very high dimensional space. PLS is able to do that since it learns a unique latent space for each speaker and dimension of latent space is much lower than original variable space. Then, PLS classifies speech in this latent space.

2.5 Software for PLS

Since PLS needs sophisticated computations, as previously seen with the description of its mathematical structure and algorithm (Section 2.2), its fruitful application depends on the availability of a suitable software.
Fortunately, nowadays PLS methodology is usually integrated within all the main computer programs for multivariate analysis. Such softwares can have a more general applicability, as solutions working in several industries, or be more subject oriented, as specific applications for well defined domains. For example SPM, which integrates a PLS regression module, is one one of the most widely used softwares for brain imaging. Similarly, there are some R packages that implement PLS methods with microarray data, as plsgenomics. However, only the former type of software is considered. Moreover, softwares can be free or proprietary.
Hereinafter the most common and worldwide used softwares for multivariate and

PLS analysis are listed, briefly presenting their key properties or adding some interesting historical informations.

Simca v. 13, MKS Umetrics AB, Sweden is a state-of-the-art suite of software products for multivariate analysis. Indeed, with the latest graphics functionality and the leading edge analytical methods, it offers all the tools to manage data from simple visualization to advanced batch modeling. In this way, it enables to effectively explore data, analyze process and interpret the results. As a consequence, it transforms data into information, allowing to make decisions quickly and with confidence.
Simca has some interesting properties. First of all, it is, and it has always been, a front runner in latest multivariate technology, continuously developed and updated. For this reason, PLS regression method was included from the beginning and remains still today a core part of the software. Secondly, it has a completely updated interface with amazing usability, interactivity and visualization. Thirdly, some tools for helping users are provided online and can be downloaded: a user guide and tutorial in order to get started by running through a series of examples step by step; a brief introduction to the methodology together with the knowledge base about multivariate data analysis; the technical support menu to find out how to get assistance. Finally, the best way to learn everything about the methods and how to use Simca to transfer data into information, is to attend a training courses. With these qualities, for many years Simca has been the standard tool for scientists, engineers, researchers, product developers and others coping with large amounts of data. Indeed, it can be applied in many industries and domains as for example Pharma R&D, Semiconductor, Plastics, Chemicals, Pulp and Paper, Metals, Minerals and Mining, Food and Beverage, Manufacturing. In particular, it is the standard for Process Analytical Technology and Quality by Design solutions used in pharmaceutical and biopharmaceutical development and production.
Regarding its origins, Umetrics AB[16] was founded in 1987 (with the name Umetri AB that was changed in 1999) by a group from Dept. of Organic Chemistry at Umeå University. Professor S. Wold was a member of this group that deals with multivariate data analysis and design of experiments under the name of Chemometrics. The first business idea was to make consulting and teaching these arguments to European chemical and pharmaceutical industry. Then, Umetrics market expansion program continued, until in 1991 the decision to develop an own multivariate software package resulted in a MS-DOS software called SIMCA-4R. In the same year a small package for design of experiments was also released and on-line software started to be designed. In 1992 was born SIMCA-P 3.0 for Windows. Since those years Umetrics AB continued to develop and improve a broad range of software products for multivariate data analysis and design of experiments.

[16] For more informations please visit the web site http://www.umetrics.com/

The Unscrambler® X v. 10.2 is a software for fast, smart, easy and accurate multivariate analysis of large and complex datasets. For this reason, it provides several new analytical methods, as advanced regression and classification modeling tools, that assure deeper insights and better predictions from data. It includes for example PCA, PCR, MLR, LDA, PLS for regression and discriminant analysis, and SVM. In addition, it is the only major software application combining the power of multivariate analysis with Design of Experiments functionality in one seamless package. It was originally developed in 1986 by H. Martens, and later by Camo Software AS, Oslo Norway[17] that has set the standard in multivariate analysis for over 25 years.
The last version is optimized for even greater usability, improved stability, increased security and reliability. Firstly, it is a program with exceptional ease of use, intuitive workflows, outstanding graphics and interactive data visualization. Secondly, it includes easy data import options that accept a wide range of data formats. Thirdly, it assures also powerful exploratory data analysis tools, as cluster analysis, that make easier data mining and allow to cut through large data sets to identify underlying patterns quickly and easily. Then, it has extensive data preprocessing options that ensures that data are suitable for multivariate analysis. As added value, Camo Software AS offers extensive knowledge base and help on hand. Indeed, softwares have detailed tutorials written by world-leading experts, which explain the theory and application of multivariate analysis methods, plus practical examples and tips on using the softwares. Moreover, compliance mode, digital signatures, password access, Windows domain authentication and audit trails provide the necessary security requirements for regulated industries. In addition, it has a flexible and adaptable architecture. Plug-in modules for specific methods or file formats give it the flexibility to meet the needs of any industry. Its models can also integrate into third party applications. Finally, it is suitable for almost any company and helps clients develop, manufacture and market products faster and more cost effectively. Indeed, it can be used for R&D, Engineering, Manufacturing, Quality Control, Data mining and analysis of customer, product or transaction data, and Predictive modeling in a wide range of industries from Aerospace and Defense to Food and Beverages, from Banking and Financial Services to Agriculture. But it is decisive also for scientists working in process analysis, Chemometrics, spectroscopy, metabolomics or sensometrics etc. Whereas it can be a useful support for lecturers wanting to introduce multivariate analysis to their syllabus with an easy-to-learn, user-friendly package.

R[18] is a language and environment for statistical computing and graphics. Indeed, it provides a lot of techniques as linear and nonlinear modelling, classical statistical

[17] The reference web site is http://www.camo.com/

[18] R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

tests, time-series analysis, classification, clustering, etc. and it is highly extensible adding new packages. R is a GNU project available since 1995 as Free Software in source code form under the terms of the Free Software Foundation's GNU General Public License (GPL) from the Comprehensive R Archive Network (CRAN)[19]. It compiles and runs on a wide variety of UNIX platforms and similar systems, Windows and MacOS.

Regarding PLS, the pls package implements PCR and PLS and is freely available under the GNU GPL. The package is written by R. Wehrens and B.-H. Mevik and it started as a merge of Wehrens's earlier package pls.pcr and an unpublished package by Mevik. A description of the package was published in R News (Mevik, 2006 [44]) whereas a slightly longer description can be found in an article by Mevik and Wehrens (Mevik and Wehrens, 2007 [45]). The pls package has many features. Firstly, it includes several algorithms as traditional NIPALS algorithm (orthogonal scores), kernel PLS, wide kernel PLS, Simpls and PCR through support vector decomposition as well as it manage multi-response models, i.e. it allows PLS2. Secondly, it assures flexible cross-validation and Jackknife variance estimates of regression coefficients. Thirdly, it provides extensive and custom plots of scores, loadings, predictions, coefficients, (R)MSEP, R^2 and correlation loadings. Then, it has a formula interface that follows the classical linear model function "lm()", with methods for tasks as predict, print, summary, plot, update, etc. In addition, there are extraction commands for coefficients, scores and loadings whereas MSEP, RMSEP and R^2 estimates can be computed. Moreover, multiplicative scatter correction is also considered. Finally, there are some interesting add-ons like VIP.R or specplot.R. The former implements a variable selection method called Variance Importance in Projection that should be used when multicollinearity is present, even if it currently only works with single-response orthogonal scores PLS regression models. The last one is a function to interactively plot spectra that allows zooming and panning of the spectra, even if so far only horizontally.

There are some other packages related to PLS. The package autopls is an extension of the pls package with automated backward selection of predictors. The package plsdepot contains different methods for PLS analysis of one or two data tables such as NIPALS, SIMPLS, PLS Regression, and PLS Canonical Analysis. The package plsdof provides Degrees of Freedom estimates and statistical inference for PLS Regression. Model selection for PLS is based on various information criteria (AIC, BIC) or on cross-validation. Estimates for the mean and covariance of the PLS regression coefficients are available. They allow the construction of approximate confidence intervals and the application of test procedures. Further, cross-validation procedures for RR and PCR are available. The package plsRglm makes Partial least squares Regression for (weighted) generalized linear models and k-fold cross validation of such models using various criteria. It allows for missing data in the explanatory

[19] All the informations, downloads and manuals can be found in the web site http://cran.r-project.org/

variables. Bootstrap confidence intervals constructions are also available. The package ppls allows Penalized PLS and contains linear and nonlinear regression methods based on PLS and penalization techniques. Model parameters are selected via cross-validation, and confidence intervals ans tests for the regression coefficients can be conducted via jackknifing.

Outside these domains, SAS/STAT® software is probably the most easily available program that provides a complete, comprehensive set of tools for data analysis. It works in both specialized and general enterprise application environments. For this reason, it enables to take advantage of all data in order to uncover new business opportunities and increase revenue.
SAS/STAT moves the scientific discovery process forward by applying the latest statistical techniques. Indeed, it does not only include tools for traditional statistical analysis of variance and predictive modeling, but also exact methods and statistical visualization techniques. Moreover, statistical procedures are constantly being updated to reflect the latest developments in statistical methodology, thus enabling to go beyond the basics for more advanced statistical analyses. In particular, among many methods there is regression that includes also PLS procedure implemented with the PROC PLS statement. Whereas OLS has the single aim of minimizing sample response prediction error, seeking linear functions of the predictors that explain as much variation in each response as possible, the techniques implemented in PROC PLS have the additional goal of accounting for variation in the predictors, under the assumption that directions in the predictor space that are well sampled should provide better prediction for new observations when the predictors are highly correlated. In PROC PLS data and methods can be specified as well as some other options. First of all, PROC PLS fits models using any one of a number of linear predictive methods as for example PCR, reduced rank regression[20] and PLS that balances the two objectives of explaining response variation and explaining predictor variation. Secondly, PROC PLS allows two different formulations for PLS: the original method of Wold and the SIMPLS method of de Jong. Thirdly, PROC PLS enables to choose the number of extracted factors by cross validation. Various methods of cross validation are available, including one-at-a-time validation, splitting the data into blocks, and test set validation. In addition, general linear modeling approach can be used to specify a model for the design, allowing for general polynomial effects as well as classification or ANOVA effects. Finally, the model fit by the PLS procedure in a data set can be saved and applied to new data by using the SCORE procedure.
SAS/STAT handles large data sets from disparate sources, so that analysts are freed to focus on analysis rather than data issues. Moreover, it provides technical sup-

[20] Reduced rank regression, also known as maximum redundancy analysis, extracts factors to explain as much response variation as possible and differs from multivariate linear regression only when there are multiple responses.

port by experienced master's and doctorate level statisticians who can help address almost any question quickly and provide a quality of service not often found with other software vendors. In addition, it achieves corporate and governmental compliance. Indeed, users can produce repeatable code that is easily documented and verified for legal compliance issues. Finally, it allows to gain higher model-scoring performance and faster time to results. When licensed with SAS Model Manager and SAS Scoring Accelerator, SAS/STAT linear models can indeed be published into database-specific functions and use in-database processing. This eliminates the need to move data between SAS and the database for scoring purposes, reducing cost, complexity and latency of the scoring process. Thus, performance of the entire modeling process is improved, enabling faster predictive results and competitive advantage. However, this aspect can become a drawback, because SAS software drives users toward dependency on only SAS-specific solutions (e.g., their proprietary data warehouses). Moreover, someone finds this software difficult to use, requiring specific SAS programming expertise. It is also expensive and carries high, unpredictable annual licensing costs. Furthermore, data visualization is integral for analytics, but SAS's graphics have major shortcomings.
Regarding its origins, once SAS stood for Statistical Analysis System and was born in 1966 at North Carolina State University as a software to analyze agricultural research. Then, as demand for such software grew, SAS company[21] was founded in 1976 to help all sorts of customers, from pharmaceutical companies and banks to academic and governmental entities. Nowadays, SAS is the leader in business analytics software and services, and the largest independent vendor in the business intelligence market.

The PLS methods can be implemented in practice also with the Statistics Toolbox™ included in MATLAB®. Indeed, MATLAB can be extended with a lot of add-on products for specialized application areas. Statistics Toolbox belongs to this extensive set providing key features for Math, Statistics, and Optimization. It provides algorithms and tools for organizing, analyzing, and modeling data. In particular, in presence of multidimensional data, with Statistics Toolbox users can apply regression or classification for predictive modeling with PLS techniques. Moreover, Statistics Toolbox has got algorithms that identify key variables with sequential feature selection, transform data with principal component analysis, or apply regularization and shrinkage. In addition, users can generate random numbers for Monte Carlo simulations, use statistical plots for exploratory data analysis, and perform hypothesis tests. Furthermore, Statistics Toolbox considers specialized data types for organizing and accessing heterogeneous data. Indeed, dataset arrays store numeric data, text, and metadata in a single data container. Then, built-in methods enable to merge datasets using a common key (join), calculate summary statistics

[21] The corresponding web site is http://www.sas.com/

on grouped data, and convert between tall and wide data representations. Finally, categorical arrays provide a memory-efficient data container for storing information drawn from a finite, discrete set of categories.
Statistics Toolbox, other add-on products, as well as MATLAB are designed and developed by MathWorks (Natick, Massachusetts, U.S.A.), the leading developer of mathematical computing software founded in 1984[22]. MATLAB is a high-performance interactive environment for numerical computation, visualization, and programming that combines comprehensive math and graphics functions with a powerful high-level language. It allows to analyze data, develop algorithms, create models and applications. The language, tools, and built-in math functions enable to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages. It can be used for a range of applications and more than a million engineers and scientists in industry and academia use this language of technical computing.

XLSTAT-PLS is the module of XLSTAT that includes several features for multivariate analysis. In particular, it provides both traditional modeling tools, as OLS, and advanced techniques as PCR, PLS regression and PLS discriminant analysis.
XLSTAT[23] version 2013.1 is the leading and most complete data analysis and statistical solution available for Microsoft Excel®. Originally, it was designed in 1993 by Addinsoft in order to help users gain time by eliminating the need for complicated and risky data transfers between different applications, that had been requisite for previous data analysis. For this reason, it uses Microsoft Excel as its interface providing a wide variety of functions that enhances Microsoft Excel analytical capabilities. As a consequence, it is integrated into Microsoft Excel making a familiar environment the ideal tool for everyday data analysis and Statistics requirements.
It is easy to install and to use, quick and highly reliable, intuitive and user-friendly, modular, didactic, affordable, accessible and available in many languages, automatable and customizable. It does not require learning a new software interface and it shares data and results seamlessly. It offers excellent customer services, as top level assistance and support team solutions to any customer inquiry within one business day.
For all these reasons, it ensures access to statistical methods and data analysis to everyone, so that people can focus on data analysis finding the information hidden in their data. Moreover, it encourages innovation and excellence in analytics in various fields and it promotes education making teachers and students life easier. Furthermore, it is also involved in several other programs to facilitate the access to the software to students. As a consequence, it has grown to be one of the most commonly used statistical software packages on the market.

[22] The company web site is http://www.mathworks.co.uk/

[23] For more information please visit the web site http://www.xlstat.com/en/

STATISTICA is a graphically oriented line of Statistics softwares that was designed since 1990 by StatSoft®, Inc. to offer in several areas entirely new levels of functionality not available at that time in any data analysis software. StatSoft®, Inc. was founded in 1984 in Tulsa, Oklahoma U.S.A. and is now one of the largest global providers of analytic software worldwide[24]. The latest version released in 2012 is STATISTICA 11.0 and this line includes many products, as for example STATISTICA Advanced. STATISTICA Advanced has the functionality of STATISTICA Base, STATISTICA Multivariate Exploratory Techniques, STATISTICA Advanced Linear/Nonlinear Models and STATISTICA Power Analysis and Interval Estimation. In particular, STATISTICA Advanced Linear/Nonlinear Models contains a wide array of the most advanced linear and nonlinear modeling tools on the market; supports continuous and categorical predictors, interactions, and hierarchical models; includes automatic model selection facilities as well as variance components, time series, and many other methods; and all analyses incorporate extensive, interactive graphical support and built-in complete Visual Basic scripting. Among all the modules that STATISTICA Advance Linear/Nonlinear Models features, there is also PLS.

Weka (Waikato Environment for Knowledge Analysis) is a Data Mining software developed at the University of Waikato, New Zealand[25]. It is an open source software issued under the GNU GPL and the latest version is Weka 3. It is fully implemented in the Java programming language and its algorithms can either be applied directly to a dataset or called from Java code. Moreover, it is also well-suited for developing new Machine Learning schemes. In addition, it runs on almost any modern computing platform and has a graphical user interface for an easy access to its functionalities.
Weka contains a comprehensive collection of Machine Learning algorithms and visualization tools that support several standard Data Mining tasks. Indeed, it implements techniques for data analysis and predictive modeling. More specifically, it allows data preprocessing, clustering, classification, regression, visualization, association rules, and feature selection. Whereas the original non-Java version of Weka was primarily designed since 1993 as a tool for analyzing data from agricultural domains, Weka 3, for which development started in 1997, is now used in many different application areas, in particular for educational purposes and research.
Regarding PLS, the Package weka.filters.supervised.attribute includes the Class PLSFilter, that runs PLS regression over the given instances and computes the resulting matrix for prediction. By default it replaces missing values and centers the data. Some options can be specified in order to turn on output debugging information; define the number of components to compute (default is 20); update the class at-

[24] The company web site is http://www.statsoft.com/

[25] For more information please visit the corresponding web site http://www.cs.waikato.ac.nz/ml/weka/

tribute (default is off); turn replacing of missing values on; choose the algorithm to use between SIMPLS or the default PLS1; decide the type of preprocessing that is applied to the data between none, the default center or standardize.

3

Related methods

This Chapter presents briefly some methods that have goals and tasks in common with PLS but that belong to a different scientific field. Indeed, even if PLS was initially developed and successfully applied in Chemometrics, it becomes early a topic also for statisticians (Section 3.1). First of all, Ordinary Least Squares is briefly described (Section 3.2), since it is the traditional estimation procedure used for linear regression and because it can be seen as the benchmark for all the following techniques. Then, PLS is related to Canonical Correlation Analysis where latent vectors with maximal correlation are extracted (Section 3.3). Moreover, Ridge Regression (Section 3.4), Principal Component Analysis and Principal Component Regression (Section 3.5) are presented. As regression procedures, all these techniques can be cast under a unifying approach called continuum regression (Section 3.6). Furthermore, as regards classification, there is a close connection between PLS and Linear Discriminant Analysis (Section 3.7). Aim of this Chapter is to bring all these methods together into a common framework attempting to identify their similarities and differences.

3.1 Introduction

Although PLS was heavily promoted and used in Chemometrics, it was at first overlooked by statisticians, since it was considered rather an algorithm than a rigorous statistical method. As a consequence, features of PLS eluded for some time theoretical understanding. This led to unsubstantiated claims concerning its performance relative to other regression procedures. In particular, statisticians assert that PLS makes fewer assumptions about nature of data. But simply not understanding the nature of hypothesis being made does not mean that they do not exist.

Nevertheless, successful applications of PLS on real world data make increase attention of statisticians to this techniques, so that interest in PLS statistical properties has risen and relevant advances in PLS are done within the last years. The effectiveness of PLS is hence been studied theoretically in terms of its variance and shrinkage properties. Moreover, the performance of PLS is investigated in several simulation studies. Some researches connected PLS with other, in statistical community better understood, methods and they shown very competitive behaviour of PLS. Further

surveys will surely reveal additional aspects of PLS and will help to better define structures of data and problems where the use of PLS will become strategic in comparison to other techniques.
Finally, connections between individual processes could help to design new algorithms by joining their good properties and thus resulting in more powerful tools. PLS and other approaches are indeed sometimes concurrent, but they could also be combined.

3.2 Ordinary Least Squares

Ordinary Least Squares (OLS) is the traditional estimation procedure used for linear regression (Equation 1.3). Considering the simplest form of linear regression, that involves a linear combination also of input variables, OLS estimator $\hat{\mathbf{b}}_{OLS}$ is formally defined as the solution of

$$\begin{aligned}
&\underset{\mathbf{b}}{\operatorname{argmin}} ||\mathbf{y} - \mathbf{X}\mathbf{b}||^2 = \\
&\underset{\mathbf{b}}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\mathbf{b})^T(\mathbf{y} - \mathbf{X}\mathbf{b}) = \\
&\underset{\mathbf{b}}{\operatorname{argmin}}\, var(\mathbf{y} - \mathbf{X}\mathbf{b})
\end{aligned} \tag{3.1}$$

Computing the first derivative and putting it equal to zero, this problem becomes equivalent to the equation

$$||\mathbf{y} - \mathbf{X}\mathbf{b}|| = 0 \tag{3.2}$$

then, applying the method of normal equations

$$\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y} \tag{3.3}$$

the solution[1] is given by

$$\hat{\mathbf{b}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.4}$$

Estimation process of $\hat{\mathbf{b}}_{OLS}$ requires computing of $(\mathbf{X}^T\mathbf{X})^{-1}$, that denotes the pseudoinverse of $\mathbf{X}$. As a consequence, matrix $\mathbf{X}^T\mathbf{X}$ should be invertible, i.e. it should not be a singular matrix or equivalently its determinant should not be null. This remark suggests which hypothesis data should comply if OLS would be correctly and successfully applied. Indeed, OLS regression requires a $(n \times k)$ input matrix where $n >> k$ with linearly independent columns and rows. This means that OLS regression should be especially used if instances and explanatory variables are uncorrelated. Observations are usually not related to each other; but collinearity of

[1] In Chapter 1 (Subsection 1.2.1) this result was introduced presenting the parameter fitting procedure that minimizes an error function, where the most common choice is Residual Sum of Squares (RSS). Then, the same equation is also obtained following a more general approach for estimating unknown parameters, called maximum likelihood (comparison with Equation 1.12 can be made).

variables should be accurately evaluated.
OLS estimator has two very interesting and important statistical properties related to its mean and variance, that are respectively

$$\begin{aligned} E[\hat{\mathbf{b}}_{OLS}] &= E[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= E[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\mathbf{b}+\mathbf{e})] \\ &= E[\mathbf{b}+(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e}] \\ &= \mathbf{b} \end{aligned} \tag{3.5}$$

$$\begin{aligned} var[\hat{\mathbf{b}}_{OLS}] &= var[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= var[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\mathbf{b}+\mathbf{e})] \\ &= var[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{e}] \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\Sigma_{\mathbf{e}} \end{aligned} \tag{3.6}$$

where $\Sigma_{\mathbf{e}}$ is the diagonal covariance matrix of independent and identically distributed (i.i.d.) random noise $\mathbf{e}$ that contains variances σ^2. Moreover, random noise is assumed to have null mean. Since the mean is equal to the vector $\mathbf{b}$, OLS estimator can be defined unbiased. Moreover, for the Gauss-Markov theorem its variance is lower than the variance of any other estimator that is a linear combination of observations $\mathbf{y}$. As a consequence, since it is unbiased and efficient, OLS estimator is called the Best Linear Unbiased Estimator (BLUE).
In order to briefly summarize other OLS characteristics some algebraic properties should be first reminded. In what follows, eigendecomposition and singular value decomposition (SVD) will be used[2].
SVD of matrix allows to write $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{R}\boldsymbol{\Gamma}\mathbf{Z}^T \tag{3.7}$$

with $\mathbf{R}$ and $\mathbf{Z}$ orthonormal matrices that consist of singular vectors of $\mathbf{X}$. Matrix $\boldsymbol{\Gamma}$ is diagonal and it collects singular values of $\mathbf{X}$. With some basic algebra it can be shown that

$$\mathbf{X}^T\mathbf{X} = \mathbf{Z}\boldsymbol{\Lambda}\mathbf{Z}^T = \sum_{i=1}^{k} \lambda_i \mathbf{z}_i \mathbf{z}_i^T \tag{3.8}$$

where $\{\mathbf{z}_i\}_1^k$ is a k-dimensional vector of matrix $\mathbf{Z}$. Eigendecomposition of $\mathbf{X}^T\mathbf{X}$ is so obtained starting from SVD of $\mathbf{X}$. As a consequence, singular vectors of the last coincide with eigenvectors of the former. Moreover, diagonal matrix $\boldsymbol{\Lambda} = \boldsymbol{\Gamma}^T\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^2$ has elements $\lambda_i = \gamma_i^2$ (squared singular values of $\mathbf{X}$ are equal to eigenvalues of $\mathbf{X}^T\mathbf{X}$). Here $\{\lambda_i\}_1^k$ are arranged in descending order ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$). Considering Equations 3.7 and 3.8, definition of Λ and properties of orthonormal matrices, it

[2] A more technical description of both is provided in Appendix A.1 and A.2, respectively.

follows that

$$\begin{aligned}\hat{\mathbf{b}}_{OLS} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\\ &=(\mathbf{Z}\mathbf{\Lambda}\mathbf{Z}^T)^{-1}(\mathbf{R}\mathbf{\Gamma}\mathbf{Z}^T)^T\mathbf{y}\\ &=\mathbf{Z}\mathbf{\Lambda}^{-1}\mathbf{Z}^{-1}\mathbf{Z}\mathbf{\Gamma}^T\mathbf{R}^T\mathbf{y}\\ &=\mathbf{Z}\mathbf{\Gamma}^*\mathbf{R}^T\mathbf{y}\end{aligned} \tag{3.9}$$

with $\mathbf{\Gamma}^* = \mathbf{\Lambda}^{-1}\mathbf{\Gamma}^T$ $(k \times n)$ diagonal matrix with elements equal to $\{\frac{1}{\gamma_i}\}_1^k = \{\frac{1}{\sqrt{\lambda_i}}\}_1^k$. This equation can be written as

$$\hat{\mathbf{b}}_{OLS} = \sum_{i=1}^{rk(\mathbf{X}^T\mathbf{X})} \frac{\mathbf{r}_i^T\mathbf{y}}{\sqrt{\lambda_i}}\mathbf{z}_i = \sum_{i=1}^{rk(\mathbf{X}^T\mathbf{X})} \hat{\mathbf{b}}_i \tag{3.10}$$

where $\hat{\mathbf{b}}_i = \frac{\mathbf{r}_i^T\mathbf{y}}{\sqrt{\lambda_i}}\mathbf{z}_i$ is the component of $\hat{\mathbf{b}}_{OLS}$ along $\mathbf{r}_i$ and $rk(\cdot)$ denotes the rank of a matrix.

Since, as seen in Equation 2.7, coefficient $\hat{\mathbf{b}}_{PLS}$ is related to an OLS estimator $\mathbf{c}$ computed from regression of $\mathbf{y}$ on orthogonal predictors $\mathbf{T}$, relation shown in Equation 3.10 holds also for $\hat{\mathbf{b}}_{PLS}$ with proper corrections. As a consequence,

$$\hat{\mathbf{b}}_{PLS} = \mathbf{W}(\mathbf{P}^{\mathbf{T}}\mathbf{W})^{-\mathbf{1}} \sum_{i=1}^{rk(\mathbf{T}^T\mathbf{T})} \frac{\mathbf{r}_i^T\mathbf{y}}{\sqrt{\lambda_i}}\mathbf{z}_i = \mathbf{W}(\mathbf{P}^{\mathbf{T}}\mathbf{W})^{-\mathbf{1}} \sum_{i=1}^{rk(\mathbf{T}^T\mathbf{T})} \hat{\mathbf{c}}_i$$

where $\hat{\mathbf{c}}_i = \frac{\mathbf{r}_i^T\mathbf{y}}{\sqrt{\lambda_i}}\mathbf{z}_i$ is the component of $\hat{\mathbf{c}}_{OLS}$ along $\mathbf{r}_i$ that, as $\mathbf{z}_i$ and λ_i, belongs to SVD of $\mathbf{T}$ matrix.

3.3 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA), introduced by Hotelling, extracts the latent vectors that describe directions of maximal correlation (Hotelling, 1933 [34]). Formally, CCA solves the following optimization problem

$$[corr(\mathbf{t},\mathbf{u})]^2 = \max_{||\mathbf{w}||=||\mathbf{c}||=1}[corr(\mathbf{X}\mathbf{w},\mathbf{Y}\mathbf{c})]^2 \tag{3.11}$$

where $[corr(\mathbf{t},\mathbf{u})]^2 = [cov(\mathbf{t},\mathbf{u})]^2/var(\mathbf{t})var(\mathbf{u})$ denotes the sample squared correlation.

The corresponding eigenvalue problem providing the solution to this optimization criterion is given by

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}\mathbf{w} = \lambda\mathbf{w} \tag{3.12}$$

where $\mathbf{w}$ represents a weight vector for projecting original $\mathcal{X}$-space data into the latent space.

Regression with PLS can be compared to CCA and connections between them become useful in order to understand further properties of both methods. CCA, as PLS, is a projection method to latent variables. It is easy to see that PLS criterion (Equation 2.3) represents a form of CCA where the principle of maximal correlation is balanced with the requirement of explaining as much variance as possible in both $\mathcal{X}$- and $\mathcal{Y}$-spaces.

$$\max_{||\mathbf{w}||=||\mathbf{c}||=1} [cov(\mathbf{Xw}, \mathbf{Yc})]^2 = \max_{||\mathbf{w}||=||\mathbf{c}||=1} var(\mathbf{Xw})[corr(\mathbf{Xw}, \mathbf{Yc})]^2 var(\mathbf{Yc}) \tag{3.13}$$

Note that in the case of a one-dimensional $\mathcal{Y}$-space only the $\mathcal{X}$-space variance is involved. In general CCA is solved in a way similar to PLS-SB, that is, eigenvectors and eigenvalues of Equation 3.12 are extracted at once by an implicit deflation of the cross product matrix $\mathbf{X}^T\mathbf{Y}$.

3.4 Ridge Regression

Ridge Regression (RR) was introduced by Hoerl and Kennard as a method for stabilizing regression estimates in presence of extreme collinearity, it means when sample covariance matrix $\mathbf{S}_\mathbf{X}$ is singular or nearly so (Hoerl and Kennard, 1970 [33]).
Considering the simplest linear model with unidimensional response variable and a linear combination also of the input (Equation 1.3), RR solves the following optimization problem

$$\begin{aligned}
&\max_{||\mathbf{w}||=||c||=1} \frac{[cov(\mathbf{Xw}, \mathbf{y}c)]^2}{(1-\eta_\mathbf{X})var(\mathbf{Xw}) + \eta_\mathbf{X}} = \\
&\max_{||\mathbf{w}||=1} \frac{[cov(\mathbf{Xw}, \mathbf{y})]^2}{var(\mathbf{Xw}) - \eta_\mathbf{X} var(\mathbf{Xw}) + \eta_\mathbf{X}} = \\
&\max_{||\mathbf{w}||=1} \frac{[cov(\mathbf{Xw}, \mathbf{y})]^2}{var(\mathbf{Xw}) + \eta} = \\
&\max_{||\mathbf{w}||=1} \frac{[corr(\mathbf{Xw}, \mathbf{y})]^2 var(\mathbf{Xw})}{var(\mathbf{Xw}) + \eta}
\end{aligned} \tag{3.14}$$

with $\eta_\mathbf{X}$ and $\eta = -\eta_\mathbf{X} var(\mathbf{Xw}) + \eta_\mathbf{X}$ representing regularization terms that are usually called "ridge" parameters. They range respectively in $0 \le \eta_\mathbf{X} \le 1$ and $\eta \le 0$. They are generally considered a metaparameter of the procedure and they are estimated through a model selection procedure. Since the response values $\{y_i\}_1^n$ enter linearly in the model estimates $\{\hat{y}_i\}_1^n$, any of the competing model selection criteria can be straightforwardly applied.
RR corresponding eigenvalue problem providing the solution to optimization criterion is given by

$$[(1-\eta_\mathbf{X})\mathbf{X}^T\mathbf{X} + \eta_\mathbf{X}\mathbf{I}]^{-1}\mathbf{X}^T\mathbf{y}\mathbf{y}^T\mathbf{Xw} = \lambda\mathbf{w} \tag{3.15}$$

where $\mathbf{w}$ represents a weight vector for projecting original $\mathcal{X}$-space data into the latent space.
RR optimization problem can be equivalently written as a penalized least squares criterion[3] with the penalty being proportional to squared norm of coefficient vector $\mathbf{b}$

$$\begin{aligned}&\underset{\mathbf{b}}{\operatorname{argmin}}\,||\mathbf{y}-\mathbf{Xb}||^2+\eta||\mathbf{b}||^2=\\&\underset{\mathbf{b}}{\operatorname{argmin}}\,var(\mathbf{y}-\mathbf{Xb})+\eta var(\mathbf{b})\end{aligned}$$

with $\eta \geq 0$. Then, regression coefficients $\mathbf{b}$ are taken to be the solutions

$$\hat{\mathbf{b}}_\eta=(\mathbf{X}^T\mathbf{X}+\eta\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.16}$$

with $\mathbf{I}$ being the $(k \times k)$ identity matrix. RR estimators are approximate solutions of Equation 3.4 where the inverse of the (possibly) ill-conditioned predictor covariance matrix is stabilized by adding to it a multiple of $\mathbf{I}$. Degree of stabilization is regulated by value of the ridge parameter.

3.4.1 Canonical Ridge Analysis

The relation between OLS, CCA, RR and PLS can be seen through the concept of Canonical Ridge Analysis. This method provides indeed a unique general approach that collects all the previously described estimation procedures as special cases. Canonical Ridge Analysis solves the following optimization problem

$$\max_{||\mathbf{w}||=||\mathbf{c}||=1}\frac{[cov(\mathbf{Xw},\mathbf{Yc})]^2}{[(1-\eta_\mathbf{X})var(\mathbf{Xw})+\eta_\mathbf{X}][(1-\eta_\mathbf{Y})var(\mathbf{Yc})+\eta_\mathbf{Y}]}$$

with $0 \leq \eta_\mathbf{X}, \eta_\mathbf{Y} \leq 1$ regularization terms usually called “ridge” parameters. Corresponding eigenvalue problem is

$$[(1-\eta_\mathbf{X})\mathbf{X}^T\mathbf{X}+\eta_\mathbf{X}\mathbf{I}]^{-1}\mathbf{X}^T\mathbf{y}[(1-\eta_\mathbf{Y})\mathbf{Y}^T\mathbf{Y}+\eta_\mathbf{Y}\mathbf{I}]^{-1}\mathbf{Y}^T\mathbf{Xw}=\lambda\mathbf{w}$$

There are two extreme situations of this optimization criterion and corresponding eigenvalue problem:

- for $\eta_\mathbf{X}=0$ and $\eta_\mathbf{Y}=0$ CCA is computed (Equation 3.11 and 3.12);
- for $\eta_\mathbf{X}=1$ and $\eta_\mathbf{Y}=1$ PLS results (Equation 2.3 and 2.5).

By continuous changing of $\eta_\mathbf{X}$ and $\eta_\mathbf{Y}$, solutions lying between these two cornerstones are obtained. In particular, interesting settings are

[3] As previously seen in Subsection 1.2.2, RR belongs to the family of shrinkage methods (or weight decay) that follow a regularization approach to control overfitting.

- $\eta_{\mathbf{X}} = 1$ and $\eta_{\mathbf{Y}} = 0$ which represents a form of orthonormalized PLS (OPLS) where the $\mathcal{Y}$-space data variance does not influence the final PLS solution;
- $\eta_{\mathbf{X}} = 0$ and $\eta_{\mathbf{Y}} = 1$ which consists of OPLS where the $\mathcal{X}$-space variance is similarly ignored.

The last important condition that should be highlighted is when $0 < \eta_{\mathbf{X}} < 1$ with unidimensional response variable $\mathbf{y}$. In this case RR is obtained (Equation 3.14 and 3.15). Reducing analysis to this setting, further considerations can be done. Indeed, a value of $\eta = \infty$ results in the model being the response mean $\hat{y} = 0$, whereas $\eta = 0$ give rise to the unregularized OLS estimates.

3.5 Principal Component Analysis and Regression

Principal Component Analysis (PCA) specifies directions of maximal variance, that are called principal components (PC). PC can be seen as latent variables that define a new space where original data are projected. Formally, the problem can be written as

$$\max_{||\mathbf{w}||=1} [var(\mathbf{Xw})] \tag{3.17}$$

that is equivalent to solving the following eigenproblem

$$[\mathbf{X}^T\mathbf{X}]\mathbf{w} = \lambda\mathbf{w}$$

Intuitively, PCA begins with computing of both sample covariance matrix $\mathbf{S_X}$ (Equation 2.1) and its eigendecomposition (Equation 3.8). Here weight vectors are equal to eigenvectors of $\mathbf{X}^T\mathbf{X}$, i.e. $\mathbf{w} = \mathbf{z}$. Then, $(n \times k)$ matrix $\mathbf{XZ}$ can be obtained. It records the projections of data on the latent space, i.e. the coordinates of every observation of the dataset on each PC.

PCA, as PLS, is a projection method to a new reduced space. As a consequence, they can be compared through the optimization criterion they use to define latent variables. Whereas PLS creates orthogonal weight vectors by maximizing covariance between observations in $\mathbf{X}$ and $\mathbf{Y}$ (Equation 2.3), PCA is based on the criterion of maximum data variation in the $\mathcal{X}$-space alone. PCA results indeed in a single projection irrespective of the task. Since PCA does not take into account for any information about the $\mathbf{Y}$ variables, it is defined as an unsupervised technique.
Furthermore, connection between PLS and PCA can be seen by means of CCA optimization criterion. Considering Equation 3.13, PLS is computed as a PCA where the criterion of maximal variance in $\mathcal{X}$-space is weighted with correlation between $\mathbf{X}$ and $\mathbf{Y}$ variables, and variance in $\mathcal{Y}$-space. PLS extends indeed PCA with a regression phase so that PC related only to $\mathbf{X}$ will explain covariance between $\mathbf{X}$ and

$\mathbf{Y}$ as far as possible.
Finally, both PCA and PLS are traditional dimensionality reduction techniques widely used in Machine Learning. But PLS provides a more principled dimensionality reduction in comparison to PCA.

When PCA is performed on $\mathbf{X}$ matrix and computed PC are used as predictors for $\mathbf{Y}$ response, Principal Component Regression (PCR) can be defined. Excluding PLS, PCR developed by Massy is the most popular and heavily promoted regression method in Chemometrics (Massy, 1965 [42]). On the contrary, it is known to but seldom recommended statisticians. Indeed, as for PLS, the original ideas motivating PCR where entirely heuristic, and it statistical properties remain largely undiscovered. So, PCR has been in the statistical literature for some time, even if it has seen relatively little use compared to OLS. Formally, PCR models the relation between $\mathbf{y}$ variable and PC linearly as

$$\mathbf{y} = \mathbf{XZa}$$

where k-dimensional vector $\mathbf{a}$ collects regression coefficients and can be computed as a traditional OLS estimator from $\mathbf{y}$ and $\mathbf{XZ}$

$$\begin{aligned}
\hat{\mathbf{a}} &= (\mathbf{Z}^T\mathbf{X}^T\mathbf{XZ})^{-1}\mathbf{Z}^T\mathbf{X}^T\mathbf{y} \\
&= \Lambda^{-1}\mathbf{Z}^T\mathbf{X}^T\mathbf{y} \\
\hat{a}_i &= \frac{\mathbf{z}_i^T\mathbf{X}^T\mathbf{y}}{\lambda_i} \quad i = 1, \ldots, k
\end{aligned}$$

since, from eigendecomposition of $\mathbf{X}^T\mathbf{X}$, it can be simply shown that $\mathbf{Z}^T\mathbf{X}^T\mathbf{XZ} = \Lambda$. Last line describes every element of regression coefficient vector $\hat{\mathbf{a}}$. Component of i-th position is the regression coefficient corresponding to i-th PC.
Moreover, $\mathbf{XZ} = \mathbf{R\Gamma Z}^T\mathbf{Z} = \mathbf{R\Gamma}$ and $\mathbf{\Gamma}^* = \Lambda^{-1}\Gamma^T$ as previously seen. As a consequence, estimator is equivalent to

$$\begin{aligned}
\hat{\mathbf{a}} &= \Lambda^{-1}\mathbf{\Gamma}^T\mathbf{R}^T\mathbf{y} \\
&= \Gamma^*\mathbf{R}^T\mathbf{y} \\
\hat{a}_i &= \frac{\mathbf{r}_i^T\mathbf{y}}{\sqrt{\lambda_i}} \quad i = 1, \ldots, k
\end{aligned}$$

Last line shows the i-th component of regression coefficients vector. With this estimator, PCR produces a sequence of regression models $\{\hat{\mathbf{y}}_m\}_{m=0}^{rk(\mathbf{X}^T\mathbf{X})}$ where the rank is equal to number of nonzero eigenvalues. It is assumed the convention that for $m = 0$ the model is just the response mean, $\hat{\mathbf{y}}_0 = 0$. The m-th model is the OLS regression of $\mathbf{y}$ on the first m PC.

$$\begin{aligned}\hat{\mathbf{y}}_m &= \sum_{i=1}^{m} \mathbf{X}\mathbf{z}_i a_i \\ &= \sum_{i=1}^{m} \frac{\mathbf{X}\mathbf{z}_i\mathbf{z}_i^T\mathbf{X}^T\mathbf{y}}{\lambda_i}\end{aligned} \tag{3.18}$$

The goal of PCR is to choose the particular model $\hat{\mathbf{y}}_m^*$ with the lowest prediction mean squared error

$$m^* = \underset{0 \leq m \leq rk(\mathbf{X}^T\mathbf{X})}{\text{argmin}} \; \tilde{E}(\mathbf{y} - \hat{\mathbf{y}}_m)^2$$

where $\tilde{E}$ is the average over future data, not part of the collected sample. The quantity m can thus be considered a metaparameter whose value is estimated from data through a model selection procedure.
Since PCR estimator that regresses $\mathbf{y}$ on the first m PC is an approximate solution of Equation 3.4, there is a strictly relation between $\hat{\mathbf{b}}_{PCR}$ and $\hat{\mathbf{b}}_{OLS}$. In particular considering the model with $m = rk(\mathbf{X}^T\mathbf{X})$ it holds

$$\begin{aligned}\hat{\mathbf{b}}_{PCR} &= \mathbf{Z}\hat{\mathbf{a}} \\ &= \mathbf{Z}(\mathbf{Z}^T\mathbf{X}^T\mathbf{X}\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{X}^T\mathbf{y} \\ &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \hat{\mathbf{b}}_{OLS}\end{aligned}$$

or equivalently

$$\hat{\mathbf{b}}_{PCR} = \mathbf{Z}\Gamma^*\mathbf{R}^T\mathbf{y} = \hat{\mathbf{b}}_{OLS}$$

PCR, as PLS, aims to reduce dimensionality and to thereby tackle the problems of regression parameters estimation that often occur in sets of explanatory variables which have high multicollinearity. In particular, PCR establishes maximum variability of explanatory variables and PLS does the same whilst also taking into account the relation between $\mathbf{X}$ and $\mathbf{Y}$. Indeed, PCR focuses on reducing the dimensionality of $\mathbf{X}$ without taking into account the relationship that exists between $\mathbf{X}$ and $\mathbf{Y}$. As a consequence, the optimum subset of PC should be chosen since PC have been computed with respect to $\mathbf{X}$ but there is no guarantee that these PC will be pertinent for explaining $\mathbf{Y}$. Finally, PCR is a less prediction oriented method than PLS.

3.6 Continuum regression

Comparison of PLS, RR and PCR allows to bring out some interesting properties of their corresponding estimators and models. These regression procedures can be evaluated by different perspectives. From description of their algorithmic procedure, for instance, it might appear that PCR, RR and PLS are very different methods leading to quite distinct model estimates. However, a heuristic comparison suggests that they are, in fact, quite similar, in that they are all attempting to achieve the same operational result in slightly different ways. As a consequence, a unifying approach called continuum regression can be defined that relates PLS to PCR and RR, considering OLS as a reference method (Frank and Friedman, 1993 [24]).

The unidimensional case is taken into account, where response n-dimensional vector $\mathbf{y}$ is regressed on $(n \times k)$ matrix of predictors $\mathbf{X}$ by means of regression coefficient $\mathbf{b}$, a k-dimensional vector. Then, each regression procedure can be regarded as a two step process: first a m-dimensional subspace of k-dimensional Euclidean space is defined, secondly regression is performed under the restriction that coefficient vector $\mathbf{b}$ lies in that subspace

$$\mathbf{b} = \sum_{i=1}^{m} b_i \mathbf{w}_i \tag{3.19}$$

where the k-dimensional unit vectors $\{\mathbf{w}_i\}_1^m$ span the prescribed subspace with $\mathbf{w}_i^T \mathbf{w}_i = 1$. Finally, properties of regression coefficient estimators can be highlighted. As a consequence, regression procedures can be compared by three tools: i) the way in which they define the subspace $\{\mathbf{w}_i\}_1^m$; ii) the manner in which the (constrained) regression is performed; iii) the behaviour of regression coefficient estimator and its form.

One possibility to assess the quality of an estimator $\hat{\mathbf{b}}$ is to compute its Mean Squared Error (MSE), which should be as lower as possible and it is defined as

$$\begin{aligned} MSE(\hat{\mathbf{b}}) &= E[(\hat{\mathbf{b}} - \mathbf{b})^T (\hat{\mathbf{b}} - \mathbf{b})] \\ &= (E[\hat{\mathbf{b}}] - \mathbf{b})^T (E[\hat{\mathbf{b}}] - \mathbf{b}) + E[(\hat{\mathbf{b}} - E[\hat{\mathbf{b}}])^T (\hat{\mathbf{b}} - E[\hat{\mathbf{b}}])] \\ &= (E[\hat{\mathbf{b}}] - \mathbf{b})^T (E[\hat{\mathbf{b}}] - \mathbf{b}) + var[\hat{\mathbf{b}}] \end{aligned}$$

This is the bias-variance decomposition of MSE where the first part is the squared bias and the second part is the variance term.

OLS estimator (Equations 3.4 and 3.9) has no bias if $\mathbf{b} \in range(\mathbf{X}^T\mathbf{X})$, as previously seen (Equation 3.5). As a consequence, one possibility to decrease MSE is to modify it by shrinking its directions that are responsible for a high variance. The variance term (Equation 3.6) depends on the nonzero eigenvalues of $\mathbf{X}^T\mathbf{X}$. Indeed, if some eigenvalues are very small, the variance of $\hat{\mathbf{b}}_{OLS}$ can be very high, which leads to a high MSE value. Moreover, small eigenvalues of $\mathbf{X}^T\mathbf{X}$ correspond to principal directions $\mathbf{z}_i$ of $\mathbf{X}$ that have a low sample spread. So, in order to improve OLS estimator by decreasing its MSE, new shrinkage estimators can be computed as

$$\hat{\mathbf{b}}_{shr} = \sum_{i=1}^{rk(\mathbf{X}^T\mathbf{X})} f(\lambda_i)\hat{\mathbf{b}}_i \tag{3.20}$$

where $\hat{\mathbf{b}}_i = \frac{\mathbf{r}_i^T\mathbf{y}\mathbf{z}_i}{\sqrt{\lambda_i}}$ as defined in Equation 3.10. In this way, the estimates of each method (for a given value of the metaparameter fitted through a model selection criterion) depend on data only through the vector of OLS estimates $\hat{\mathbf{b}}_i$ and the eigenvalues of the predictor covariance matrix $\{\lambda_j\}_1^k$. Moreover, $f(\cdot)$ is some real-valued function and $f(\lambda_i)$ are called shrinkage factors that scale the OLS solution along each eigendirection. If every element of Equation 3.20 does not depend on $\mathbf{y}$, that is, $\hat{\mathbf{b}}_{shr}$ is linear in $\mathbf{y}$, any factor $f(\lambda_i) \neq 1$ increases the bias of the i-th component. The variance of the i-th component decreases for $|f(\lambda_i)| < 1$ and increases for $|f(\lambda_i)| > 1$. The OLS estimator is shrunk in the hope that the increase in bias is small compared to the decrease in variance. Since PLS, PCR and RR estimators can be expanded in terms of OLS components $\hat{\mathbf{b}}_i$, they are characterized by a shrinkage structure, too. Indeed, PLS, PCR and RR, unlike OLS, produce biased estimates, since their criteria all involve the scale of predictor matrix through its sample variance. Usually, the degree of this bias is regulated by the value of the model selection parameter. The effect of this bias is to shrink the regression coefficient vector away from the OLS solution toward directions in the predictor variable space in which the projected data have larger spread. This is the main goal of continuum regression. Then, since shrinkage away from low spread directions controls estimate's variance, all these methods try also to improve properties of their own estimators.

Considering OLS, the subspace is defined by a single unit vector $\mathbf{w}_{OLS}$. Projecting original variables $\mathbf{X}$ by means of $\mathbf{w}_{OLS}$ in the new vector space, new predictor variables can be computed as linear combination $\mathbf{X}\mathbf{w}_{OLS}$. The single unit vector $\mathbf{w}_{OLS}$ maximizes the squared sample correlation between response and new predictor variables.

$$\mathbf{w}_{OLS} = \underset{\mathbf{w}^T\mathbf{w}=1}{\text{argmax}}\,[corr(\mathbf{y}, \mathbf{X}\mathbf{w})]^2 \tag{3.21}$$

The OLS model is then a simple least squares regression of $\mathbf{y}$ on new variables $\mathbf{X}\mathbf{w}_{OLS}$ belonging to vector subspace

$$\begin{aligned}\hat{\mathbf{y}}_{OLS} &= \mathbf{X}\mathbf{w}_{OLS}[\mathbf{w}_{OLS}^T\mathbf{X}^T\mathbf{X}\mathbf{w}_{OLS}]^{-1}\mathbf{w}_{OLS}^T\mathbf{X}^T\mathbf{y}\\ &= \mathbf{X}[\mathbf{X}^T\mathbf{X}]^{-1}\mathbf{X}^T\mathbf{y}\end{aligned}$$

As a consequence, as seen in Equation 3.4, the OLS estimator is given by

$$\begin{aligned}\hat{\mathbf{b}}_{OLS} &= [\mathbf{X}^T\mathbf{X}]^{-1}\mathbf{X}^T\mathbf{y}\\ &= \mathbf{w}_{OLS}[\mathbf{w}_{OLS}^T\mathbf{X}^T\mathbf{X}\mathbf{w}_{OLS}]^{-1}\mathbf{w}_{OLS}^T\mathbf{X}^T\mathbf{y}\\ &= \mathbf{w}_{OLS}b_{OLS}\end{aligned}$$

In this case there is a single factor b for Equation 3.19 that is

$$b_{OLS} = [\mathbf{w}_{OLS}^T \mathbf{X}^T \mathbf{X} \mathbf{w}_{OLS}]^{-1} \mathbf{w}_{OLS}^T \mathbf{X}^T \mathbf{y}$$

The OLS criterion is invariant to the scale of the linear combinations $\mathbf{Xw}$. Indeed, OLS estimation is equivariant under all nonsingular affine transformations of the variable axes as linear, rotation, scaling.

Analyzing RR, the subspace is defined by a single unit vector $\mathbf{w}_{RR}$, as in OLS, but the criterion is somewhat different

$$\mathbf{w}_{RR} = \underset{\mathbf{w}^T\mathbf{w}=1}{\operatorname{argmax}} \, [corr(\mathbf{y}, \mathbf{Xw})]^2 \frac{Var(\mathbf{Xw})}{Var(\mathbf{Xw}) + \eta}$$

where η is the ridge parameter and $\mathbf{w}_{RR}$ allows to project explanatory variables on the defined subspace. This result confirms optimization problem previously seen in Equation 3.14. The ridge model is then taken to be a (shrinking) ridge regression of $\mathbf{y}$ on $\mathbf{Xw}_{RR}$ with the same value for the ridge parameter

$$\begin{aligned}\hat{\mathbf{y}}_{RR} &= \mathbf{Xw}_{RR}[\mathbf{w}_{RR}^T \mathbf{X}^T \mathbf{X} \mathbf{w}_{RR} + \eta]^{-1} \mathbf{w}_{RR}^T \mathbf{X}^T \mathbf{y} \\ &= \mathbf{X}[\mathbf{X}^T\mathbf{X} + \eta \mathbf{I}_k]^{-1} \mathbf{X}^T \mathbf{y}\end{aligned}$$

So, the ridge estimator, as seen in Equation 3.16, is equal to

$$\begin{aligned}\hat{\mathbf{b}}_{RR} &= [\mathbf{X}^T\mathbf{X} + \eta \mathbf{I}_k]^{-1} \mathbf{X}^T \mathbf{y} \\ &= \mathbf{w}_{RR}[\mathbf{w}_{RR}^T \mathbf{X}^T \mathbf{X} \mathbf{w}_{RR} + \eta]^{-1} \mathbf{w}_{RR}^T \mathbf{X}^T \mathbf{y} \\ &= \mathbf{w}_{RR} b_{RR}\end{aligned}$$

The single factor b for Equation 3.19 now is

$$b_{RR} = [\mathbf{w}_{RR}^T \mathbf{X}^T \mathbf{X} \mathbf{w}_{RR} + \eta]^{-1} \mathbf{w}_{RR}^T \mathbf{X}^T \mathbf{y}$$

As regards RR estimator (Equation 3.16), it can also be written in the following form

$$\hat{\mathbf{b}}_{\eta} = \sum_{i=1}^{rk(\mathbf{X}^T\mathbf{X})} \frac{\lambda_i}{\lambda_i + \eta} \hat{\mathbf{b}}_i$$

since $\mathbf{X}^T\mathbf{X} = \Lambda$ and $\mathbf{X} = \mathbf{R\Gamma Z}^T$. As a consequence, RR estimator is an example of shrinkage estimator for regression coefficient $\mathbf{b}$ (Equation 3.20) with shrinkage factor

$$f(\lambda_i) = \frac{\lambda_i}{\lambda_i + \eta}$$

As previously seen, setting $\eta = 0$ in RR approach, all results yield the unbiased OLS solution, whereas $\eta > 0$ introduces increasing bias toward larger values of $var(\mathbf{Xw})$ and increased shrinkage of the length of the fitted coefficient vector. For small values of η, the former effect is the most pronounced; for example, for $\eta > 0$ the RR solution will have no projection in any subspace for which $var(\mathbf{Xw}) = 0$, and very little projection on subspaces for which it is small.
RR is not equivariant under scaling, even if it is under rotation.

PCR defines a sequence of m-dimensional subspaces each spanned by the first m unit vectors $\{\mathbf{w}_i\}_1^{rk(\mathbf{X}^T\mathbf{X})}$, that are the solution to

$$\mathbf{w}_i(PCR) = \underset{\substack{\{\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}_h=0\}_1^{i-1} \\ \mathbf{w}^T\mathbf{w}=1}}{\text{argmax}} \; var(\mathbf{X}\mathbf{w}) \tag{3.22}$$

This confirms previous result shown in Equation 3.17. The first constrain in this equation is the $\mathbf{X}^T\mathbf{X}$ orthogonality and it ensures that the linear combinations associated with the different solution vectors are uncorrelated over the data

$$corr(\mathbf{X}\mathbf{w}_i, \mathbf{X}\mathbf{w}_h) = 0 \quad i \neq h$$

As a consequence, they also turn out to be orthogonal $\mathbf{w}_i^T\mathbf{w}_h = 0$, $i \neq h$. Because of their definition, every unit vector $\mathbf{w}_i$ coincides with $\mathbf{z}_i$, the eigenvectors of $\mathbf{X}^T\mathbf{X}$. The m-th PCR model is then given by a least squares regression of the response on the m linear combinations $\{\mathbf{X}\mathbf{w}_i\}_1^m$, previously defined PC and collected in $(n \times m)$ matrix $\mathbf{XW}$. Since they are uncorrelated, this reduces to the sum of univariate regression on each one

$$\begin{aligned}\hat{\mathbf{y}}_m(PCR) &= \mathbf{XWb}_m \\ &= \sum_{i=1}^{m} \mathbf{X}\mathbf{w}_i(PCR)b_i\end{aligned}$$

where $(k \times m)$ matrix $\mathbf{W}$ collects k-dimensional unit vectors $\{\mathbf{w}_i\}_1^m$ and m-dimensional vector $\mathbf{b}_m(PCR)$ records factors b_i. Accordingly to Equation 3.19, for the m-th PCR model the regression coefficient is then the k-dimensional vector of form

$$\begin{aligned}\hat{\mathbf{b}}(PCR) &= \mathbf{Wb}_m \\ &= \sum_{i=1}^{m} \mathbf{w}_i(PCR)b_i\end{aligned}$$

Comparing this result with Equation 3.18, factors b_i are equal to

$$b_i = \frac{\mathbf{w}_i^T\mathbf{X}^T\mathbf{y}}{\lambda_i} \quad i = 1, \ldots, m$$

PCR is also a shrinkage regression method for estimating k-dimensional vector of regression coefficients $\mathbf{b}$. The shrinkage factors of Equation 3.20 are defined as

$$f_{PCR}(\lambda_i) = \begin{cases} 1 & \text{if } \lambda_i^2 \geq \lambda_m^2 \quad (i \leq m) \\ 0 & \text{if } \lambda_i^2 < \lambda_m^2 \quad (i > m) \end{cases}$$

both of which are linear in that they do not involve the sample response values.
In PCR, the degree of bias is controlled by the value of m, the dimension of the

constraining subspace spanned by $\{\mathbf{w}_i(PCR)\}_1^m$ that is, the number of components m used in the PCR model. If $m = rk(\mathbf{X}^T\mathbf{X})$ an unbiased OLS solution is obtained. For $m < rk(\mathbf{X}^T\mathbf{X})$, bias is introduced. The smaller the value of m, the larger the bias. As with RR, the effect of this bias is to draw the solution toward larger values of $var(\mathbf{Xw})$. This is because constraining $\mathbf{w}$ to lie in the subspace spanned by the first m eigenvectors of $\mathbf{X}^T\mathbf{X}$ places a lower bound on the sample variance of $\mathbf{Xw}$, that is $var(\mathbf{Xw}) \geq \lambda_m$. Since the eigenvectors, and hence the subspaces, are ordered on decreasing values of λ_m, increasing m has the effect of easing this restriction, thereby reducing the bias.
PCR, as RR, is not equivariant under scaling even if it is under rotation.

PLS regression also produces a sequence of m-dimensional subspaces spanned by successive unit vectors, and then the m-th PLS solution is obtained by a least squares fit of the response onto the corresponding m-linear combinations in a strategy similar to PCR. The only difference from PCR is in the criterion used to define the vectors that span the m-dimensional subspace and hence the corresponding linear combinations. The criterion that give rise to PLS is

$$\begin{aligned}\mathbf{w}_i(PLS) &= \underset{\substack{\{\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}_h=0\}_1^{i-1}\\ \mathbf{w}^T\mathbf{w}=1}}{\operatorname{argmax}} [corr(\mathbf{y},\mathbf{Xw})]^2 var(\mathbf{Xw})\\ &= \underset{\substack{\{\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}_i=0\}_1^{i-1}\\ \mathbf{w}^T\mathbf{w}=1}}{\operatorname{argmax}} [cov(\mathbf{y},\mathbf{Xw})]^2\end{aligned}$$

since squared correlation between two variables is equal to squared covariance between them divided by product of their variances and $var(\mathbf{y}) = 1$ because it is autoscaled (Stone and Brooks, 1990 [65]). This result corresponds to Equation 2.3. As with PCR, vectors $\{\mathbf{w}_i(PLS)\}_1^{rk(\mathbf{X}^T\mathbf{X})}$ are constrained to be mutually $\mathbf{X}^T\mathbf{X}$ orthogonal so that the corresponding linear combinations $\{\mathbf{Xw}_i\}_1^{rk(\mathbf{X}^T\mathbf{X})}$ are uncorrelated over the data. This causes the m-dimensional least squares fit to be equivalent to the sum of m univariate regressions on each linear combination separately, as with PCR. Unlike PCR, however, the $\{\mathbf{w}_i(PLS)\}_1^{rk(\mathbf{X}^T\mathbf{X})}$ are not orthogonal owing to the different criterion used to obtain them. Formally the model is

$$\begin{aligned}\hat{\mathbf{y}}_m(PLS) &= \mathbf{XWb}_m(PLS)\\ &= \sum_{i=1}^{m} \mathbf{Xw}_i(PLS) b_i\end{aligned}$$

where $(k \times m)$ matrix $\mathbf{W}$ collects k-dimensional unit vectors $\{\mathbf{w}_i\}_1^m$ and m-dimensional vector $\mathbf{b}_m(PLS)$ records factors b_i. Accordingly to Equation 3.19, for the m-th PLS model the regression coefficient is then the k-dimensional vector of form

$$\begin{aligned}\hat{\mathbf{b}}(PLS) &= \mathbf{W}\mathbf{b}_m(PLS)\\ &= \sum_{i=1}^{m}\mathbf{w}_i(PLS)b_i\end{aligned}$$

For PLS, the situation is similar to that of PCR. The degree of bias is regulated by m, the number of components used. For $m = rk(\mathbf{X}^T\mathbf{X})$, an unbiased OLS solution is produced. Decreasing m generally increases the degree of bias. An exception to this occurs when $\mathbf{X}^T\mathbf{X} = \mathbf{I}$ (totally uncorrelated predictor variables), in which case an unbiased OLS solution is reached for $m = 1$ and remains the same for all m (though for $m \geq 2$ the regressions are singular, all of the regressors being identical). This can be seen from the PLS criterion (Equation 2.3). In this case, $var(\mathbf{Xw}) = 1$ for all $\mathbf{w}$, and the PLS criterion reduces to that for OLS (Equation 3.21). With this exception, the effect of decreasing m is to attract the solution coefficient vector toward larger values of $var(\mathbf{Xw})$ as in PCR. For a given m, however, the degree of this attraction depends jointly on the covariance structure of the predictor variables and the OLS solution, which in turn depends on the sample response values. This fact is often presented as an argument in favour of PLS over PCR. Unlike PCR, there is no sharp lower bound on $var(\mathbf{Xw})$ for a given m.

PLS, unlike OLS, is not equivariant under scaling, even if it is under rotation, as PCR and RR.

Even if PLS estimator has shrinkage properties as well, its shrinkage factors can not be expressed by a simple formula, as those for RR and PCR. As a consequence, in order to show PLS shrinkage structure some characteristics of PLS method should be first described. In particular PLS can be seen as a regularized least squares fit by means of conjugate gradient method and Lanczos algorithms (see Phatak and de Hoog, 2002 [52]). Indeed, it can be shown that the PLS algorithm is equivalent to the conjugate gradient method. This is a procedure that iteratively computes approximate solutions of Equation 3.4 by minimizing the quadratic function

$$\frac{1}{2}\mathbf{b}^T\mathbf{X}^T\mathbf{X}\mathbf{b} - \mathbf{y}^T\mathbf{X}\mathbf{b}$$

along directions that are $\mathbf{X}^T\mathbf{X}$ orthogonal. The approximate solution achieved after m steps is equal to the PLS estimator obtained after m iterations. Moreover, the conjugate gradient method is in turn closely related to the Lanczos algorithm, a technique for approximating eigenvalues. Let define a matrix $\mathbf{K}$ as

$$\mathbf{K} = (\mathbf{X}^T\mathbf{y}, (\mathbf{X}^T\mathbf{X})\mathbf{X}^T\mathbf{y}, \ldots, (\mathbf{X}^T\mathbf{X})^{p-1}\mathbf{X}^T\mathbf{y})$$

The space spanned by the columns of $\mathbf{K}$ is denoted by $\mathcal{K}$ and it is called the m-dimensional Krylov space of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{y}$. In the Lanczos algorithm, an orthogonal basis

$$\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_p) \tag{3.23}$$

of $\mathcal{K}$ is computed. Given the linear map $\mathbf{X}^T\mathbf{X}$, its restriction to $\mathcal{K}$ for an element $\mathbf{k} \in \mathcal{K}$ is defined as the orthogonal projection of $\mathbf{X}^T\mathbf{X}\mathbf{k}$ onto the space $\mathcal{K}$. Moreover, the map is represented by the $m \times m$ matrix

$$\mathbf{L} = \mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}$$

This matrix is tridiagonal. Its m eigenvectors and eigenvalues pairs

$$(\mathbf{v}_i, \delta_i) \tag{3.24}$$

are called Ritz pairs. They are the best approximation of the eigenpairs of $\mathbf{X}^T\mathbf{X}$ given only the information that is encoded in $\mathcal{K}$. Finally, the weight vectors $\mathbf{w}$ in Equation 2.3 of PLS1 are identical to the basis vectors in Equation 3.23. In particular, the weight vectors are a basis of the Krylov space and the PLS estimator is the solution of the optimization problem

$$\begin{aligned} &arg\min_{\mathbf{b}} ||\mathbf{y} - \mathbf{X}\mathbf{b}||^2 \\ &subject\,to\,\mathbf{b} \in \mathcal{K} \end{aligned}$$

Now PLS shrinkage factors can be simply defined, since they are closely related to the Ritz pairs (Equation 3.24). PLS shrinkage factors $f(\lambda_i)$ that correspond to estimator $\hat{\mathbf{b}}_{PLS}$ after p iterations of PLS are indeed

$$f(\lambda_i) = 1 - \prod_{j=1}^{p}(1 - \frac{\lambda_i}{\delta_j})$$

These shrinkage factors have some remarkable properties. Most importantly, $f(\lambda_i) > 1$ can occur for certain combinations of i and m. Note however that the PLS estimator is not linear in the response values $\mathbf{y}$. Indeed, the factors $f(\lambda_i)$ depend on the eigenvalues of matrix $\mathbf{L}$ (Equation 3.24) and $\mathbf{L}$ in turn depends on $\mathbf{y}$. It is therefore not clear in which way this shrinkage behaviour influences MSE of PLS1. The amount of absolute shrinkage $|1 - f(\lambda_i)|$ is particularly prominent if m is small. Shrinkage structure of PLS estimator for a m-component model can be equivalently expressed as

$$f_{im}(PLS) = \sum_{h=1}^{m} b_h \lambda_i^h \tag{3.25}$$

where the vector $\mathbf{b} = \{b_h\}_1^m$ is given by $\mathbf{b} = \mathbf{W}^{-1}\mathbf{w}$ with the m components of the vector $\mathbf{w}$ being

$$w_i = \sum_{h=1}^{k} \hat{b}_h^2 \lambda_h^{i+1}$$

and the elements of the $(m \times m)$ matrix $\mathbf{W}$ are given by

$$\mathbf{W}_{ij} = \sum_{h=1}^{k} \hat{b}_h^2 \lambda_h^{i+j+1}$$

Both formulas contain element $\hat{b}_h = \mathbf{X}^T \mathbf{z}_h \mathbf{y} / \lambda_h$ that is the projection of the OLS solution on $\mathbf{z}_h$. This structure of shrinkage factors allows to confirm previous conclusions. Indeed they depend on both the number of components m used and the eigenstructure $\{\lambda_h\}_1^k$, as do the factors for RR and PCR, but not in a simple way. Moreover, they also depend on the OLS solution $\{\hat{b}_j\}_1^k$ which in turn depends on the response values. The PLS shrinkage factors are seen to be independent of the length of the OLS solution $||\hat{\mathbf{b}}||^2$, since they are related only with the relative values of $\{\hat{b_j}\}_1^k$.
Although the shrinkage factors for PLS can not be expressed by a simple formula, as those for RR and PCR, they can be computed for given values of m, $\{\lambda_j\}_1^k$ and $\{\hat{b}_j\}_1^k$, and compared to those of RR and PCR for corresponding situations, as done by Frank and Friedman in their work (Frank and Friedman, 1993 [24]).
An interesting aspect of the PLS solution is that, unlike RR and PCR, it not only shrinks the OLS solution in some eigendirections ($f(\lambda_i) \leq 1$) but expands it in others ($f(\lambda_i) > 1$). For a m-component PLS solution, the OLS solution is expanded in the subspace defined by the eigendirections associated with the eigenvalues closest to the m-th eigenvalue. Directions associated with somewhat larger eigenvalues tend to be slightly shrunk, and those with smaller eigenvalues are substantially shrunk. The expression for the mean squared prediction error suggests that, at least for linear estimators, using any $f(\lambda_i) > 1$ can be highly detrimental because it increases both the bias squared and the variance of the model estimate. This suggests that the performance of PLS might be improved by using modified scale factors $\{\widetilde{f_{im}}(PLS)\}_1^k$ where $\widetilde{f_{im}}(PLS) \leftarrow min(f_{im}(PLS), 1)$, although this is not certain since PLS is not linear and mean squared error was derived assuming linear estimates. It would, in any case, largely remove the preferences of PLS for (true) coefficient vectors that align with the eigendirections whose eigenvalues are close to the m-th eigenvalue.
In conclusion, PLS and PCR make the assumption that the truth distribution of theoretic random variables have particular preferential alignments with the high spread directions of the observed predictor variables distribution. As a consequence, PLS and PCR shrink more heavily away from the low spread directions than RR. In addition, PLS places increased probability mass on the true coefficient vector aligning with the m-th principal component direction, where m is the number of PLS components used, in fact expanding the OLS solution in that direction. However, the solutions and hence the performance of PLS, PCR and RR tend to be quite similar in most situations, largely because they are applied to problems involving high collinearity in which variance tends to dominate the bias, especially in the directions of small predictor spread, causing all three methods to shrink along those directions. In presence of more symmetric designs, larger differences between them might well emerge.

3.7 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is both a classification and a dimensionality reduction technique widely used in Machine Learning. LDA consists of a linear projection of the data on a new subspace whose directions should be computed solving a specific optimization problem. Projection weights $\mathbf{w}$ are indeed found maximizing *between-classes* variance and at the same time minimizing the *within-classes* variance in the projected space. Remembering $\mathbf{SS_b}$ and $\mathbf{SS_w}$, *among-classes* and *within-classes* sums of squares defined in Equation 2.8, LDA optimization criterion can be written as

$$\max_{||\mathbf{w}||=1} [\mathbf{SS_w}^{-1}\mathbf{SS_b}\mathbf{w}]$$

As a consequence, weight vectors coincide with the eigenvectors $\mathbf{w}$ of the following eigenvalue problem

$$[\mathbf{SS_w}^{-1}\mathbf{SS_b}]\mathbf{w} = \lambda\mathbf{w}$$

As usual, the connections between PLS and LDA can be seen through the optimization criteria they use to define projection directions. In particular in this case modified PLS method defined in Equation 2.10 should be considered. The among-classes sum of squares matrix $\mathbf{SS_b}$ appears in both criteria. Moreover, observing both criteria and tracing backwards the reasoning done for PLS classification (Subsection 2.3.2), it can be said that LDA, as PLS, has a supervised nature because it takes into account not only the variance between observations of the samples but it uses also the $\mathbf{Y}$ data to determine the projection space. As a consequence, LDA results in different subspaces for different $\mathbf{Y}$. For this reason, LDA is preferred over PCA for classification tasks.

However, if $\mathbf{Y}$ spans G classes, LDA produces at most a $(G-1)$-dimensional subspace, since it can compute at most $G-1$ meaningful latent variables. So, the size of the LDA projection space is bounded by $G-1$, which can limit the representation ability of the projected space. Additionally, when handling a few individuals with thousands of variables ($n << k$), the covariance estimates in LDA are not full of rank and the weight vectors cannot be extracted. On the contrary, PLS does not have the $G-1$ limit in the projection space dimension, since it can extract further projecting directions beyond $G-1$. Furthermore, unlike LDA, PLS is well suited for data with $n << k$.

Finally, in LDA classification directions are identical to the directions given by CCA using a dummy matrix $\mathbf{Y}$ for group membership (Bartlett, 1938 [3]). So, there is a close connection of classification with PLS to LDA also by means of CCA.

4

Support Vector Machines

This Chapter opens with a short introduction to Support Vector Machine (SVM) that presents intuitive ideas behind its algorithmic aspects, summarizes briefly its historical development, and lists some applications (Section 4.1). Then, general theory of SVM is described in order to define the most important concepts that are involved in every form of SVM algorithm (Section 4.2). Even if the chosen approach considers the simplest case of binary output, it can be easily generalized to more complex situations. This Section illustrates step by step three specific procedures. It distinguishes between linear and nonlinear classification. Moreover, it analyzes two different possibilities for linear classificators, i.e. when they are applied on linearly separable data and on nonlinearly separable data. Finally, examples of nonlinear kernel PLS are described (Section 4.3). They complete both approaches of nonlinear PLS regression previously described in the Subsection 2.3.3 with kernel-based technique of SVM.

4.1 Introduction

A Support Vector Machine (SVM) is a set of techniques that represents a major development in Machine Learning algorithms. It defines a group of supervised learning methods and has a twofold aim: firstly to analyze data and to recognize patterns; secondly to generalize informations learned before to new observations. Hence, SVM can be seen as a competitor of statistical methodologies that apply classification and regression in order to describe given known data and to predict future ones.
SVM covers a wide range of situations with respect to data features. First of all, with regard to categorical quantities SVM is part of generalized linear classificators family. In this case, SVM works both on linearly separable and on nonlinearly separable data. In particular, considering a set of training examples each marked as belonging to one of two categories, aim of SVM method is to specify a classification function that would hopefully classify new data. Hence, SVM defines a nonprobabilistic binary linear classifier. Indeed, a SVM training algorithm builds a model that assigns examples into one category or the other. Then, with a new dataset it is able to predict for each given input from which of two possible classes it comes.

Secondly, a kind of SVM consists of kernel-based methods that define a nonlinear classifier. Thirdly, SVM can provide an alternative approach for learning of polynomial classificators, opposed to classical techniques as neural networks. Indeed SVM overcomes problems and drawbacks of both one layer or multi layer neural networks. The former networks have a learning algorithm that is efficient[1], but they are useful only if data are linearly separable. On the other hand, the last can represent nonlinear functions but are very difficult to be trained since weights space has an high dimension[2]. On the contrary, SVM has an efficient algorithm and is able to represent complex nonlinear functions.

Graphical representation can help to understand a general SVM model. Given data become points in space, mapped so that examples of different categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. More formally, a SVM constructs a hyperplane (set of hyperplanes in a high- or infinite-dimensional space) which are used for classification, regression, or other tasks as prediction. Indeed, usually SVM learns firstly one or more separating hyperplanes between groups of training data, then SVM uses these hyperplanes to predict new observations. Intuitively, a good separation is achieved by the hyperplane that has the largest distance (so-called functional margin) to the nearest training data point of any class (so-called support vector), since in general the larger the margin the lower the generalization error of the classifier.

SVM has some interesting qualities. Usually SVM is able to handle high dimensional input space and at the same time it does not require dataset characterized by large cardinality. In addition, it can manage successfully presence of sparse instances and irrelevant variables. Then, SVM can assure a good generalization ability with an high prediction accuracy and without the risk of overfitting. Of course, its performance should improve as the amount of training data increases. At the end, it has also nice math properties since it involves a simple convex optimization problem which is guaranteed to converge to a single global solution.

However, there is some drawbacks, too. When data are modeled by means of a multidimensional space and solution refers to new features, interpretability of results can be not so easy and meaningful. A trade off should so be done between speed and advantages of computation or directness of results explanation. Then, SVM is sensitive to noise and a relatively small number of mislabeled examples can dramatically decrease the performance. Finally, users should learn how to tuning SVM because some phases, as kernel or parameters selection, are usually done with

[1] Characteristics parameters of network are obtained through the solution of a convex problem of quadratic programming with constrains of equality or box type (value of the parameter should belong to a range) that has a unique global minimum.

[2] Traditional techniques, as back propagation, allow to obtain weights of network solving a not convex and not constrained optimization problem. As a consequence, there is an indeterminate number of local minimum.

a sequence of trials where various values are tested in a length series of experiments. Domain experts can however give assistance as for example in formulating appropriate similarity measures. Otherwise, in the absence of reliable criteria, applications rely on the use of a validation procedure to set such parameters.
Traditional SVM approach usually assumes two classes defining binary classificators as a consequence, a tricky question is how to manage multi-class data. A possible answer is to consider as many SVM as groups data belong to. In this case, the first SVM learns the differences between the first group and everything else; the second SVM does the same with the second class, and so on. A problem can so be viewed as a series of binary classifications, one for each category. Then, in order to predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

SVM belong to statistical learning theory called Theory of Vapnik-Chervonensky introduced in Russia by Vapnik and colleagues during sixties - when Generalized Portrait algorithm are defined (Vapnik and Lerner, 1963 [71], Vapnik and Chervonenkis, 1964 [68] and [69]) - and seventies (Vapnik and Chervonensky, 1974 [70]). The original SVM algorithm was indeed proposed by V. N. Vapnik in 1982 as an extension of Generalized Portrait algorithm (Vapnik, 2006 [67]). Then, SVM were developed in AT& T Bell Laboratories by Vapnik and colleagues (Boser et al., 1992 [8], Guyon et al., 1993 [29]) until the current standard version (soft margin) was defined a few years later (Cortes and Vapnik, 1995 [15], Schölkopf et al., 1995 [58] and 1998 [60], [59]) and gained finally increasing popularity in late 1990s.

Since its origins in an industrial context, research in SVM has had many applications. First works interested optical and hand written character recognition, where shortly SVM became competitive with the best traditional methods. Other fields where SVM is successfully applied are object detection (Blanz et al., 1996 [7]); speaker recognition (Schmidt, 1996 [57]); content-based image retrieval and face recognition (Osuna et al., 1997 [49]); text and hypertext categorization (Joachims, 1998 [36]). In the last case aim of SVM is the classification of natural text documents into a fixed number of predefined categories based on their content. Applications are for example email filtering, web searching, sorting documents by topic, etc.
More recently SVM is largely used in bioinformatics applications as protein sequences, gene expression, phylogenetics informations, cancer classification, etc. For example a protein sequence can be mapped in a 20-dimensional space considering its amino acid composition, that is a vector of frequencies. Secondary and 3D structure of proteins can also be predicted, as remote homologies between proteins or gene can be identified. Finally, some other applications of SVM are functional classification of proteins, pattern recognition in biological sequences and data analysis of information carried on microarrays.
In conclusion, SVM is currently among the best performers for a number of classi-

fication tasks and it can be applied to complex data types beyond feature vectors (e.g. graphs, sequences, relational data) by designing kernel functions for such data. Moreover, SVM techniques have been extended also to unsupervised learning methods as PCA.

4.2 Binary classifier

In this Section a general SVM learning process is formulated. The simpler case of binary classification is considered. It can be viewed as the task of separating two classes in the input space. However its results can be easily generalized to situations with more than two classes. An opening part is used to describe preliminaries assumptions, notation and aim of SVM, highlighting its problems and solutions for overcoming them. Then, linear classifier are taken into account, considering both the case where it works on linearly separable data and the case of soft margin classification. Finally nonlinear classifier is presented. Every time learning process is defined step by step.

A set of training examples that are preclassified is defined by

$$\begin{array}{l} S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_n, y_n)\} \\ \{\mathbf{x}_i\}_{i=1}^{n} \in \mathcal{X} \subseteq I\!R^k \\ \{y\}_{i=1}^{n} \in \mathcal{Y} = \{-1, +1\} \end{array}$$

where $\mathcal{X}$ is called space of input and $\mathcal{Y}$ space of output. Space of input is a subsets of Euclidean space $I\!R^k$, so that every data can be drawn as a point of this space whose coordinates coincides with corresponding values of data vector. As a consequence, graphical representation allows to visualize vector algebra and calculus with data. Every $\{\mathbf{x}_i, y_i\}_{i=1}^{n}$ is taken by an unknown distribution $P(\mathbf{x}, y)$.
A SVM learns from these given and known data the border between observations belonging to different classes.

A set of threshold functions can be defined as

$$\begin{array}{l} H = \{f_\gamma(\mathbf{x}), \gamma \in \Gamma \subset I\!R\} \\ f_\gamma(\mathbf{x}) : I\!R^k \to \{-1, +1\} \end{array}$$

with $f(\cdot)$ real function and Γ set of real parameters γ that creates a machine able to solve a particular problem[3]. Each threshold function $f_\gamma(\mathbf{x})$ corresponds to an hyperplane in the Euclidean space that represents input space. Moreover, everyone is an hypothesis and it is also named classifier or separator, so that the set H is called space of hypothesis.
Some properties of H can be defined. The VC (Vapnik-Chervonenkis) dimension of

[3] For instance Γ can correspond to weights of synapses of a neural network.

hypothesis space H coincides with the VC dimension of classifier $f_\gamma(\mathbf{x})$. It is the natural number that corresponds to the highest number of points that can be split in all the ways from set of functions. VC dimension is a measure of complexity of set H and it is denoted h.
Given a set of n points, among every 2^n classifications $(-1, +1)$, if a function $f_\gamma(\mathbf{x})$ exists that correctly classifies observations, then set of points is split from set of functions.

Mean theoretic error is defined as

$$R(\gamma) = \int |f_\gamma(\mathbf{x}) - y| P(\mathbf{x}, y) d\mathbf{x} dy$$

and it is a measure of goodness of an hypothesis for predicting class y of a point $\mathbf{x}$. Aim of SVM learning process is to find a function $f^*_\gamma(\mathbf{x})$ that minimizes mean theoretic error

$$f^*_\gamma(\mathbf{x}) = \operatorname*{argmin}_\gamma R(\gamma)$$

However, mean theoretic error $R(\gamma)$ is impossible to compute since probability distribution $P(\mathbf{x}, y)$ is not known. This problem can be solved when a training set is available. Indeed a training set is a sample extract from a population characterized by probability distribution $P(\mathbf{x}, y)$. As a consequence, an approximation of mean theoretic error $R(\gamma)$ can be computed as

$$R_{emp}(\gamma) = \frac{1}{n} \sum_{i=1}^{n} |f_\gamma(\mathbf{x}_i - y_i)|$$

and it is called mean empiric error. Law of large numbers assures that mean empiric error converges in probability towards mean theoretic error. As a consequence, aim of learning process becomes to minimize mean empiric error.
Theory of uniform convergence in probability developed by Vapnik and Chervonenkis, gives an upper bound to deviation of mean empiric error from mean theoretic error

$$R(\gamma) \leq R_{emp}(\gamma) + \sqrt{\frac{h(log\frac{2n}{h}) - log(\frac{\eta}{4})}{n}}$$

with $0 \leq \eta \leq 1$ and h VC dimension of $f_\gamma(\mathbf{x})$.
In order to obtain minimum mean theoretic error, both mean empiric error and ratio between VC dimension and number of points (h/n) should be minimized. Usually mean empiric error is a decreasing function of h. As a consequence, for every given number of points, an optimal value of VC dimension exists, and it follows from a trade-off between R_{emp} and h/n. SVM algorithm solve efficiently this problem minimizing at the same time VC dimension and number of errors on training set.

Moreover, Vapnik has proved that the class of optimal linear separators has VC dimension h bounded above as

$$h \leq \min\left\{ \left\lceil \frac{D^2}{\rho^2} \right\rceil, k \right\} + 1$$

where ρ is the separation margin, D the diameter of the smallest sphere that enclose all of training examples, and k the dimensionality of learning data. Intuitively, this implies that regardless of dimensionality k VC dimension can be minimized by maximizing separation margin ρ. Thus, the complexity of the classifier is kept small regardless of data dimensionality. As a consequence, in order to minimize mean theoretic error through minimization of both mean empirical error and VC dimension, SVM should maximize separation margin; i.e. the best classification is achieved by the hyperplane that has the largest separation margin.

4.2.1 Linearly separable data

Before description of a SVM process for linearly separable data, characteristics of this kind of dataset are illustrated and some useful concepts introduced.

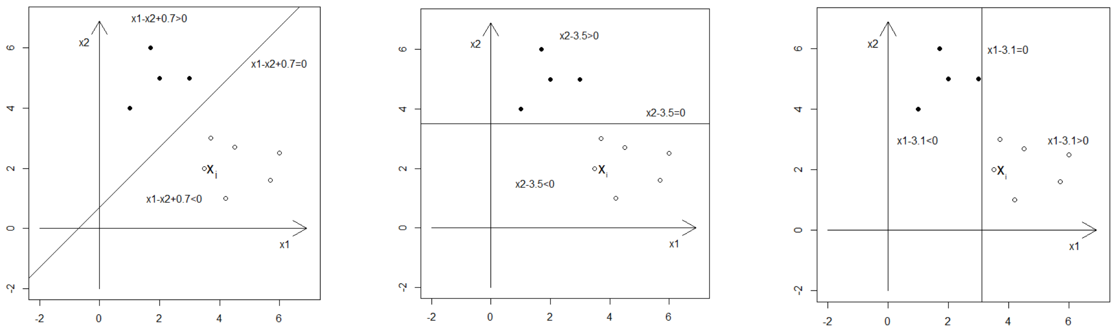
A set of data is linearly separable when it is possible to find at least a pair $(\mathbf{w}, b)$ such that

$$\begin{array}{ll} \mathbf{w}^T\mathbf{x}_i + b > 0 & \text{if } y_i = +1 \\ \mathbf{w}^T\mathbf{x}_i + b < 0 & \text{if } y_i = -1 \end{array} \tag{4.1}$$

where $\mathbf{w} \in I\!R^k$ is a k-dimensional vector and $b \in I\!R$ is a real number. Output value y_i corresponds to input vector $\mathbf{x}_i$ for every i-th observation of dataset. An hyperplane can be associated to every pair $(\mathbf{w}, b)$. It corresponds to points that satisfies the following relation

$$\mathbf{w}^T\mathbf{x}_i + b = 0$$

and it has distance equal to $\frac{|b|}{||\mathbf{w}||}$ from origin. As a consequence, a set of data is linearly separable when at least an hyperplane that correctly classifies data exists. Usually there are a number of hyperplanes with this required property.
Figure 4.1 shows an example where input data are bidimensional $\mathbf{x} = (x_1, x_2)^T$ so that input space is an euclidean space that can be represented as usual by a Cartesian plane. In this case points belonging to different classes are drawn with two different colours (white and black points). It easy to see that a pair $(\mathbf{w}, b)$ with properties required above exists, in this case it is for example $\mathbf{w} = [1, -1]$ and $b = 0.7$. Moreover, the hyperplane associated to this pair can be drawn, it is the line $x_1 - x_2 + 0.7 = 0$ because of bidimensional space (Figure 4.1a). Horizontal line $x_2 = 3.5$ can be also considered a good hyperplane, that means able to correctly classify data (Figure 4.1b). In this case $\mathbf{w} = [0, 1]$ and $b = -3.5$. Vertical line $x_1 = 3.1$ is another hyperplane that is able to separate different observations (Figure 4.1c). In this case $\mathbf{w} = [1, 0]$ and $b = -3.1$. Point $\mathbf{x}_i = (3.5, 2)$ has value $2.2 > 0$ in the firs



(a) Hyperplane $x_1 - x_2 + 0.7 = 0$ (b) Hyperplane $x_2 - 3.5 = 0$ (c) Hyperplane $x_1 - 3.1 = 0$

Figure 4.1: An example of binary linearly separable data with different hyperplanes.

case, $-1.5 < 0$ in the second and $0.4 > 0$ in the last. There is no problem if the same class has opposite output value considering different hyperplanes.

Space of hypothesis H is given by set of functions defined as

$$f_{\mathbf{w},b}(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x} + b)$$

where $sign(\cdot)$ is the binary discriminator $+/-$; $0, 1$; $true/false$, etc.

Distance between a point $\mathbf{x}_i$ and hyperplane associated to pair $(\mathbf{w}, b)$ is equivalent to

$$d(\mathbf{x}_i; \mathbf{w}, b) = \frac{|\mathbf{w}^T\mathbf{x}_i + b|}{||\mathbf{w}||}$$

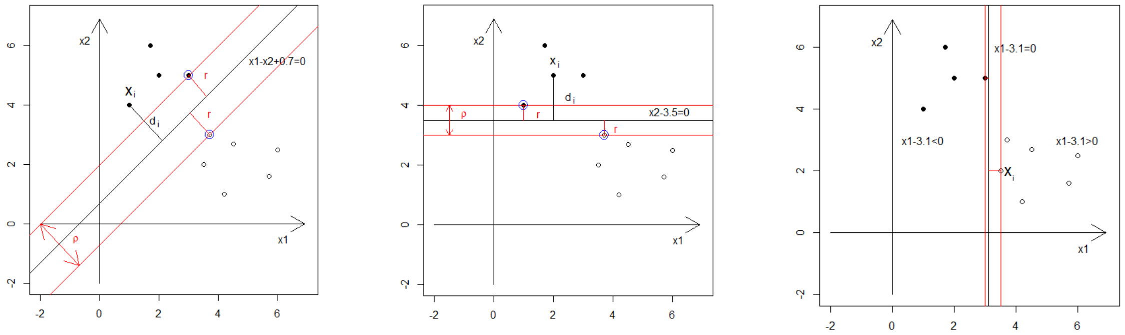
Points closest to the hyperplane are called support vectors. They are the most important training points because they define the separator hyperplane, whereas other training examples are ignorable. Their distance from hyperplane is called margin and it is denoted by r. Hence if $||\mathbf{w}|| \leq a$ with $a \in I\!R$, distance between hyperplane and the nearest point should be greater than $1/a$ and the number of possibly corrected hyperplane are so reduced.

Optimum hyperplane has the same distance from support vectors that belong to different classes. As a consequence, double of margin defines separation margin that is denoted ρ

$$\rho = 2r$$

Figure 4.2 shows point $\mathbf{x}_i$ and its distance d_i to hyperplane; support vectors (blue circled points) and their distance (margin r) to hyperplane and separator margin ρ, considering data represented in Figure 4.1. Since optimum hyperplane should have $\rho = 2r$, hyperplane of Figure 4.2c can not be a solution of optimization problem.

If data are linearly separable, as hypothesis state, aim of SVM is to find the optimum hyperplane among all those that correctly classify the training set. Ability



(a) Hyperplane $x_1 - x_2 + 0.7 = 0$ (b) Hyperplane $x_2 - 3.5 = 0$ (c) Hyperplane $x_1 - 3.1 = 0$

Figure 4.2: Examples of hyperplanes with corresponding support vectors, distance and margin for linearly separable data.

of correct classification is required in the following way.
As previously seen, data are linearly separable if there exist a hyperplane, defined by pair $\mathbf{w}, b$, such that Equation 4.1 holds. Since there are infinite solutions that differ for only a scale factor on $\mathbf{w}$, with other words hyperplane does not change if its normal vector is scaled, previous relation can be written as

$$\begin{array}{ll} \mathbf{w}^T\mathbf{x}_i + b \geq +1 & \text{if } y_i = +1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 & \text{if } y_i = -1 \end{array} \tag{4.2}$$

applying a scale transformation on both $\mathbf{w}$ and b. Here $\mathbf{x}_i$ represents every observation of training set. Equality holds if and only if $\mathbf{x}_i$ is a support vector. In this case

$$|\mathbf{w}^T\mathbf{x}_i + b| = 1 = r||\mathbf{w}||$$

from definition of r, distance between a support vector from hyperplane. As a consequence,

$$r = \frac{1}{||\mathbf{w}||}$$

and

$$\rho = 2r = \frac{2}{||\mathbf{w}||}$$

Condition of Equation 4.2 is equivalent to

$$y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 \tag{4.3}$$

for every point of dataset. This equation expresses ability of correct classification. Then, as previously defined optimum hyperplane should have minimum mean theoretical error; this means small mean empirical error and small VC dimension. The last condition is guaranteed if hyperplane has maximum margin $\rho = 2r = \frac{2}{||\mathbf{w}||}$ or equivalently minimum norm $||\mathbf{w}||^2$. In order to define optimum hyperplane, one should correctly classify points of training set among two classes $y_i \in \{-1, +1\}$ trough a hyperplane that uses a coefficient $\mathbf{w}$ with the smallest norm. Formally, the problem of Linear Support Vector Machine (LSVM) becomes

$$\begin{array}{l} \min \frac{1}{2}||\mathbf{w}||^2 \\ \text{sub } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, n \end{array} \tag{4.4}$$

Function that should be optimized is a quadratic form. Quadratic optimization problems are a well known class of mathematical programming problems for which several (nontrivial) algorithms exist[4]. In this case both cost function and constrains are strictly convex so that optimization problem has a dual form and solution of the last coincides with optimum of the former. Moreover, dual problem is usually simpler to solve than primal. In order to define dual problem, Kuhn-Tucker theorem is followed. A new unconstrained problem is firstly formulated by means of Lagrange multipliers λ_i that are introduced for every inequality constraint in the primal problem. The Lagrangian is

$$L(\mathbf{w}, b, \lambda) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \lambda_i [y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1] \tag{4.5}$$

where $\lambda = (\lambda_1, \dots, \lambda_n)$ is the vector of nonnegative Lagrange multipliers derived from constrains of Equation 4.4. Lagrangian should be minimized with respect to $\mathbf{w}$ and b and at the same time maximized with respect to $\lambda \geq 0$. Solutions are given computing derivatives and putting them equal to zero

$$\frac{\partial L(\mathbf{w}, b, \lambda)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i = 0$$

and

$$\frac{\partial L(\mathbf{w}, b, \lambda)}{\partial b} = \sum_{i=1}^{n} \lambda_i y_i = 0 \tag{4.6}$$

From the first equation it follows that

$$\mathbf{w} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i = \mathbf{X}^T \mathbf{\Lambda} \mathbf{y} \tag{4.7}$$

where $\mathbf{X}$ is the $(n \times k)$ matrix of input data; $\mathbf{y}$ is the n-dimensional vector of output data and $(n \times n)$ diagonal matrix $\mathbf{\Lambda}$ records n Lagrangian multipliers. This means that optimum hyperplane can be written as a linear combination of vectors belonging

[4] Most popular optimization algorithms for SVM use decomposition to hill-climb over a subset of Lagrangian multipliers at a time, e.g. Sequential Minimal Optimization (Platt, 1999 [53]).

to training set. Replacing Equation 4.7 and 4.6 in Equation 4.5 original problem can be rewritten in the following simpler dual form

$$\begin{aligned} &\max \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ &\text{sub } \lambda_i \geq 0 \qquad\qquad (4.8) \\ &\sum_{i=1}^{n} \lambda_i y_i = 0 \end{aligned}$$

with $\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_i = ||\mathbf{w}^*||^2$. In this way previous constrains are changed with conditions on multipliers λ_i that can be computed as solution of this equation and then defined λ_i^*. Optimum values λ_i^* allow to define optimum hyperplane except for b. Indeed from Equation 4.7

$$\mathbf{w}^* = \sum_{i^1}^{n} \lambda_i^* y_i \mathbf{x}_i = \mathbf{X}^T \mathbf{\Lambda}^* \mathbf{y}$$

However, for every support vector $\mathbf{x}_l$ it holds that $y_l(\mathbf{w}^T\mathbf{x}_l + b) = 1$ (from Equation 4.3) so that

$$b^* = \frac{1}{y_l} - \mathbf{x}_l^T \mathbf{w}^* = y_l - \mathbf{x}_l^T \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{x}_i \quad \forall \lambda_l > 0$$

Then optimum hyperplane is given by

$$\mathbf{x}^T\mathbf{w}^* + b^* = \mathbf{x}^T \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{x}_i + y_l - \mathbf{x}_l^T \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{x}_i$$

where data are only involved in products. Finally classificator is equal to

$$f(\mathbf{x}) = sign(\mathbf{x}^T \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{x}_i + y_l - \mathbf{x}_l^T \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{x}_i)$$

for every vector $\mathbf{x}$.

In the solution, every point $\mathbf{x}_i$ for which corresponding multiplier λ_i is strictly greater than zero gives a support vector that belong to one of the two hyperplanes defined by threshold function. All the remaining points of training set have corresponding λ_i equal to zero, so that they do not influence classifier. Support vectors are the critical points of training set and are the nearest to separation hyperplane. If all other points would be removed or their position changed without go beyond plane on H_1, H_2, and learning algorithm would be repeated, the result would be exactly the same.

Since vectors of dataset are only involved as products between test point $\mathbf{x}$ and support vectors $\mathbf{x}_i$, solving optimization problem implies computing the products $\mathbf{x}_i^T\mathbf{x}_j$ between all training points. The computational cost of SVM is so $O(k^2n)$ where n is the number of vectors and k is their dimension.

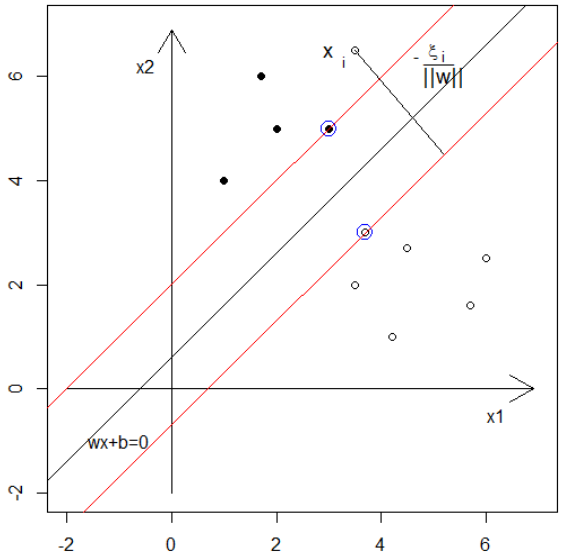


Figure 4.3: An example of binary nonlinearly separable data with hyperplane, support vectors and margin.

4.2.2 Soft margin classification

Sometimes training set holds points that are not linearly separable. This means that there exists some noisy data, that have an anomalous position with respect to other elements of the same class, as for instance point $\mathbf{x}_i$ in Figure 4.3. Aim of SVM with Soft Margin approach is to find a separator plane for this kind of set and in this case one should admit that some constraints could be broken. Hence, in order to allow misclassification of difficult examples, new quantities ξ_i, called slack variables, should be added to every constraint. Moreover, cost function should be modified so that not null slack variables will be penalized. Resulting margin is called soft margin.

Let suppose that if point $\mathbf{x}_i$ has an anomalous position, it has a distance equal to $-\xi_i/||\mathbf{w}||$ from its class as reported in Figure 4.3. As a consequence, ξ_i is large if anomalous points are distant from their class. If $\mathbf{x}_i$ belong to the correct class, $\xi_i = 0$. As in the previous case, the hyperplane has distance $b/||\mathbf{w}||$ from origin and is determined by support vectors. Formally, constrains of Equation 4.4 become

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0 \quad i = 1, \ldots, n$$

The new constrain allows a certain degree of tolerability, equal to ξ_i, to errors. A point of training set is misclassified if and only if corresponding ξ_i is greater than unit. As a consequence, $\sum_i \xi_i$ is an upper bound to maximum number of errors on dataset.

Then the problem represented by Equation 4.4 can be reformulated as

$$\begin{aligned} &\min \frac{1}{2}||\mathbf{w}||^2 + C(\sum_{i=1}^{n} \xi_i)^k \\ &\text{sub } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0 \end{aligned} \tag{4.9}$$

with constrained $\mathbf{w}, b, \xi$. Here C and k are parameters that should be defined a priori. Constant $C > 0$ can be viewed as a cost that allows to control overfitting since it trades off the relative importance of maximizing the margin and fitting the training data. A large value for C corresponds to high penalty assigned to errors. In practice, SVM algorithm tries to minimize $||\mathbf{w}||$ and at the same time to well separate given points making the minimum number of errors. Solution to optimization problem given by Equation 4.9 can be founded following the same procedure as for the case of linearly separable data. The Lagrangian is

$$L(\mathbf{w}, b, \lambda, \xi, \gamma) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \lambda_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{n} \gamma_i\xi_i + C(\sum_{i=1}^{n} \xi_i)^k$$

whose multiplier $\lambda = (\lambda_1, \ldots, \lambda_n)$ and $\gamma = (\gamma_1, \ldots, \gamma_n)$ are linked to constrains of Equation 4.9.

This Lagrangian should be minimized with respect to $\mathbf{w}, b, \xi$ and maximized with respect to $\lambda \geq 0$ and $\gamma \geq 0$.

Considering the simple case where $k = 1$, optimization problem can be reformulated in a manner similar to Equation 4.8

$$\begin{aligned} &\max \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\ &\text{sub } 0 \leq \lambda_i \leq C \\ &\qquad \sum_{i=1}^{n} y_i\lambda_i = 0 \end{aligned}$$

where the only difference with previous case is that dual variables λ_i are constrained by an upper bound equal to C. Solution of this problem is similar to solution of separable data. Again, $\mathbf{x}_i$ with nonzero λ_i will be support vectors. Solution to the dual problem is

$$\begin{aligned} \mathbf{w}^* &= \sum_{i=1}^{n} \lambda_i y_i \mathbf{x_i} \\ b^* &= y_l(1 - \xi_l) + \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i^T\mathbf{x}_l \quad \forall l \text{ s.t. } \lambda_l > 0 \end{aligned}$$

Then classifier is

$$\begin{aligned} f(\mathbf{x}) &= sign(\mathbf{x}^T\mathbf{w}^*) + b^*) \\ &= sign(\mathbf{x}^T\sum_{i=1}^{n} y_i\lambda_i^*\mathbf{x}_i) + y_l(1-\xi_l) + \sum_{i=1}^{n}\lambda_i y_i \mathbf{x}_i^T\mathbf{x}_l) \end{aligned}$$

4.2.3 Nonlinear classifier

In the previous considered situations data belong to a finite dimensional space where classification problem may be managed and solved. Indeed, both linear SVM and soft margin approach look for a separation hyperplane that maximizes its margin in input space. In particular, in the first case data are linearly separable directly in their input space. Similarly in the second case, classification can be correctly done working on input space introducing penalized slack variables. Here data are indeed linearly separable unless for few anomalous observations. However, the last solution does not always assure good performance since a hyperplane can only represent dichotomies. On the contrary, sometimes classes are not at all linearly separable in the input space. For example no line exists that splits correctly data represented in plot on left of Figure 4.4. For this reason, it was proposed that the original finite dimensional space be mapped into a much higher dimensional space, presumably making the separation easier in the new space. Cover's theorem on separability states indeed that a complex classification problem has a larger probability to be linearly separable if it is mapped through a nonlinear function of data into an high dimensional space than if it refers to a low dimensional space. Ideally, points are mapped through a nonlinear function ϕ in this new space $\mathcal{F}$ where they are linearly separable. Plot on the right side of Figure 4.4 represents this new space, called "feature" space. Then, optimum hyperplane is computed in the feature space following optimization problem with slack variables. Soft margin approach controls indeed overfitting. Moreover, to keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that products required in the solution may be computed easily from original variables. This requires definition of such products in terms of a kernel function selected to suit the problem. Hence, nonlinear SVM projects examples in a multidimensional space and looks for the best separation hyperplane in this space. Optimum separation hyperplane maximizes its margin with simple computations by means of kernel function.

Let suppose to map original data, that are not linearly separable in their input space $\mathcal{X} \subseteq I\!R^k$, in a new space of higher dimension $\mathcal{F} \subseteq I\!R^k$ with $p > k$ using a mapping function $\phi : \mathcal{X} \rightarrow \mathcal{F}$. Data are linearly separable in $\mathcal{F}$. As a consequence, coordinates in feature space of points that represents observations of dataset are equal to $\phi(\mathbf{x})$. Then optimum hyperplane that belongs to feature space is identified by equation

$$\phi(\mathbf{x}^T\mathbf{w}) + b = 0$$

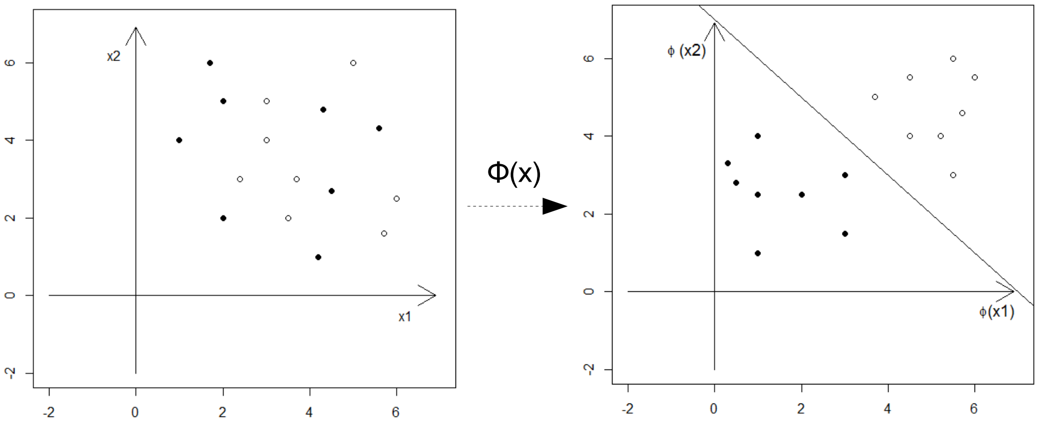


Figure 4.4: An example of data that requires nonlinear classifier represented in input space (on left) and feature space (on right).

with $\mathbf{w} \in I\!R^p$, that becomes

$$\phi(\mathbf{x})^T \sum_{i=1}^{n} y_i \lambda_i \phi(\mathbf{x}_i) + b$$

with $\mathbf{w} = \sum_{i=1}^{n} y_i \lambda_i \phi(\mathbf{x}_i)$. In this situation learning algorithm depends on data only by means of product between their images through ϕ in $I\!R^p$; i.e. $\phi(\mathbf{x}_i)^T \phi(\mathbf{x})$. As a consequence, high dimensional space causes computational problems, since learning algorithm should work with vectors of bigger dimension. In order to overcome this difficulty, a kernel function can be introduced. A kernel function gives the product between images of its two arguments, i.e.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \tag{4.10}$$

Using kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ inside the algorithm, it is so possible to ignore explicit form of ϕ, to avoid computing of data images (vectors of feature space) and of their product. Indeed hyperplane can be identified by equation

$$\sum_{i=1}^{n} y_i \lambda_i K(\mathbf{x}_i, \mathbf{x})$$

An example of a kernel function with bidimensional vectors $\mathbf{x} = [x_1, x_2]$ is $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2$. The proof need to show that $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ and is the following

$$\begin{aligned}
K(\mathbf{x}_i,\mathbf{x}_j) &= (1+\mathbf{x}_i^T\mathbf{x}_j)^2 \\
&= (1+x_{i1}x_{j1}+x_{i2}x_{j2})^2 \\
&= 1+x_{i1}^2x_{j1}^2+2x_{i1}x_{j1}x_{i2}x_{j2}+x_{i2}^2x_{j2}^2+2x_{i1}x_{j1}+2x_{i2}x_{j2} \\
&= [1 \quad x_{i1}^2 \quad \sqrt{2}x_{i1}x_{i2} \quad x_{i2}^2 \quad \sqrt{2}x_{i1} \quad \sqrt{2}x_{i2}] \begin{bmatrix} 1 \\ x_{j1}^2 \\ \sqrt{2}x_{j1}x_{j2} \\ x_{j2}^2 \\ \sqrt{2}x_{j1} \\ \sqrt{2}x_{j2} \end{bmatrix} \\
&= \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)
\end{aligned}$$

where $\phi(\mathbf{x}) = [1 \quad x_1^2 \quad \sqrt{2}x_1x_2 \quad x_2^2 \quad \sqrt{2}x_1 \quad \sqrt{2}x_2]$. Thus, a kernel function implicitly maps data to a high-dimensional space without the need to compute each $\phi(\mathbf{x})$ explicitly.

Kernel function can be represented by a kernel matrix $\mathbf{K}$ equal to

$$\begin{pmatrix} K(1,1) & K(1,2) & \dots & K(1,n) \\ K(2,1) & K(2,2) & \dots & K(2,n) \\ & & \dots & \\ K(n,1) & K(n,2) & \dots & K(n,n) \end{pmatrix} \tag{4.11}$$

It has the same number of rows and columns equal to number of data points n, and every cell records value of kernel function with arguments corresponding to elements shown by row and column indexes. It has all the informations needed by learning and summarizes information about data and kernel function. Since $K(i,j) = K(j,i)$ it is symmetric moreover, it is a semi-positive definite matrix. Mercers theorem state that every semi-positive definite symmetric function is a kernel, i.e. it can be seen as a product of two vectors[5]. In addition, semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix[6]. As a consequence, every semi-positive definite symmetric matrix can be seen as a kernel matrix.

Replacing $\mathbf{x}_i^T\mathbf{x}_j$ with $K(\mathbf{x}_i,\mathbf{x}_j)$ in every step of algorithm, a SVM that works on $I\!R^p$ is obtained. It gives the result in the same quantity of time as one that works with original nonmapped data. This is a linear classifier built in a different space, so all the previous considerations hold still now. Extension to complex decision spaces is rather simple: input variable $\mathbf{x}$ should be mapped in a new space with higher dimension, then a linear classifier work in this new space.

Formally a nonlinear classifier works as described in the following.

A point $\mathbf{x}$ is mapped in a new feature vector by means of function ϕ

$$\mathbf{x} \to \phi(\mathbf{x}) = (a_1\phi_1(\mathbf{x}), a_2\phi_2(\mathbf{x}), \dots, a_p\phi_p(\mathbf{x})) \tag{4.12}$$

[5] A more detailed description of Mercer's theorem is given on Appendix A.3

[6] A formal definition of Gram matrix is presented in Appendix A.4

where a_i are real numbers and ϕ_i are real functions, with $i = 1, \dots p$. Then the same algorithm of soft margin case can be applied where every value $\mathbf{x}_i$ is replaced by a new feature vector $\phi(\mathbf{x}_i)$. Decision function with mapping of Equation 4.12 becomes

$$f(\mathbf{x}) = sign(\phi(\mathbf{x})^T\mathbf{w} + b) = sign(\phi(\mathbf{x})^T \sum_{i=1}^{n} y_i\lambda_i\phi(\mathbf{x}_i) + b)$$

Replacing products $\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ with a kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) = \sum_{l=1}^{+\infty} a_l^2\phi_l(\mathbf{x}_i)^T\phi_l(\mathbf{x}_j)$$

the Lagrangian becomes

$$L(\lambda) = \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

and it should be maximized with constrains

$$\sum_{i=1}^{n}\lambda_i y_i = 0$$
$$0 \leq \lambda_i \leq C$$

Solving optimization problem the following result is obtained

$$f(\mathbf{x}) = sign(\sum_{i=1}^{n} y_i\lambda_i^* K(\mathbf{x}, \mathbf{x}_i) + b^*) = sign(\sum_{\mathbf{x}_i \in SV} y_i\lambda_i^* K(\mathbf{x}, \mathbf{x}_i) + b^*)$$

where SV is set of support vectors and λ^* are optimum values with respect to support vectors, because other are null so that they do not concur to sum.

Choice of kernel function is a tricky problem: it is always possible to map elements into a space with dimension higher than input space in order to produce a perfect classifier, however it will work poorly with new data, because of overfitting. Some kernel functions usually applied are for instance:

- Linear $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$

  mapping $\phi(\mathbf{x}) = \mathbf{x}$
- Polynomial of power m $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T\mathbf{x}_j)^m$

  mapping $\phi(\mathbf{x})$ that has $\binom{d+m}{m}$ dimensions
- Radial Basis function $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$

- Gaussian Radial Basis function $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2})$

  mapping $\phi(\mathbf{x})$ that is infinite-dimensional[7]

- Multi-layer Perceptron or Sigmoid $K(\mathbf{x}_i, \mathbf{x}_j) = tanh(b(\mathbf{x}_i^T \mathbf{x}_j) - c))$

As regards kernel, Gaussian or polynomial ones are used by default, but if they are ineffective more elaborate kernels should be defined.

A SVM is based on two concepts: a general learning module and a kernel function which is specific for the problem. Every SVM can work with every kind of kernel. Every SVM algorithm is then composed by four main steps: i) selection of kernel function; ii) choice of value for C; iii) solution of quadratic programming problem for which many software packages are available; iv) construction of discriminant function from found support vectors.

Nonlinear SVM has the ability to handle large feature spaces and its complexity does not depend on the dimensionality of the feature space.

4.3 Kernel PLS

Two major strategies of constructing nonlinear PLS have been previously mentioned (Subsection 2.3.3). One defines a nonlinear model for score vectors $\mathbf{t}$ and $\mathbf{u}$, the other maps original input data into a new feature space where linear PLS is applied as usual. Both can be mixed with Machine Learning techniques defined in this Chapter.

As regards former approach (Equation 2.11), the concept of kernel-based learning (Subsection 4.2.3) can also be used for modelling nonlinear relation between score vectors $\mathbf{t}$ and $\mathbf{u}$. An example would be a support vector regression model for nonlinear continuous function $g(\cdot)$. However, among all tools of Machine Learning, powerful machinery of kernel-based algorithm can be also usefully combined with PLS following the last approach. Indeed it is an elegant way of extending linear data analysis procedures to nonlinear problems. As a consequence, nonlinear kernel PLS methodology is briefly described in this Section.

Idea of kernel PLS method is based on mapping of original data from their k-dimensional space ($\mathcal{X} \subseteq I\!R^k$) into a high dimensional feature space $\mathcal{F}$ corresponding to a reproducing kernel Euclidean space. Then, estimation of PLS into feature space $\mathcal{F}$ reduces to application of linear algebra as simple as linear PLS. Formally, a function

$$\phi : \mathcal{X} \rightarrow \mathcal{F}$$

should be defined in order to compute new coordinates $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ since products $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ are required for classification. However, by applying the kernel trick all

[7] Every point is mapped to a Gaussian function and the separator is the combination of functions for support vectors.

Table 4.1: Kernel form of NIPALS algorithm.

```
Kernel form of NIPALS algorithm
Given: $\mathbf{X}$ $(n \times k)$
       $\mathbf{Y}$ $n \times d)$

1)  Initialize vector $\mathbf{u}$,
    if $\mathbf{Y}$ is 1-dimensional, assign $\mathbf{u} := \mathbf{Y}$,
    else $\mathbf{u} :=$ random $(n \times 1)$ vector
2)  Iterate to convergence:
    a) $\mathbf{t} = \mathbf{\Phi\Phi}^T\mathbf{u} = \mathbf{Ku}$
    b) $||\mathbf{t}|| \to 1$
    c) $\mathbf{c} = \mathbf{Y}^T\mathbf{t}$
    d) $\mathbf{u} = \mathbf{Yc}$
    e) $||\mathbf{u}|| \to 1$
3)  $\mathbf{p} = \mathbf{X}^T\mathbf{t}/(\mathbf{t}^T\mathbf{t})$ and $\mathbf{q} = \mathbf{Y}^T\mathbf{u}/(\mathbf{u}^T\mathbf{u})$
4)  Deflate $\mathbf{X}$ and $\mathbf{Y}$:
    $\mathbf{X} \leftarrow \mathbf{X} - \mathbf{tp}^T$ $\mathbf{Y} \leftarrow \mathbf{Y} - \mathbf{uq}^T$
If more projection vectors are required, go to step 2
```

these steps are simplified. Indeed kernel trick uses the fact that value of product between two vectors in $\mathcal{F}$ can be easily evaluated by a kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j), \quad \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$$

as previously seen in Equation 4.10. As a consequence, Gram matrix $\mathbf{K}$ of cross products between all mapped input elements can be defined, as shown in Equation 4.11, equal to

$$\mathbf{K} = \mathbf{\Phi\Phi}^T$$

where $\mathbf{\Phi}$ denotes the matrix of mapped data $\{\mathbf{\Phi}(\underline{\mathbf{x}}_i) \in \mathcal{F}\}_{i=1}^n$. Kernel trick implies that elements i, j of $\mathbf{X}$ correspond to value of kernel function $K(\underline{\mathbf{x}}_i, \underline{\mathbf{x}}_j)$.

Now, a modified version of the NIPALS algorithm should be considered where steps $2a$ and $2d$ of classical form described in Table 2.1 are merged and vectors $\mathbf{t}$ and $\mathbf{u}$ are scaled to unit norm instead of the vectors $\mathbf{w}$ and $\mathbf{c}$. The kernel form of the NIPALS algorithm is so obtained (Table 4.1)[8]. Similarly, in applications where an analogous kernel mapping of the $\mathcal{Y}$-space is considered, steps $2c$) and $2d$) can be further merged.

When kernel and traditional PLS methodology are compared, it is difficult to define the favourable one. While the kernel PLS approach is easily implementable, computationally less demanding and capable to model difficult nonlinear relations,

[8] In the case of the one-dimensional $\mathbf{y}$-space computationally more efficient kernel PLS algorithms have been proposed.

a loss of the interpretability of the results with respect to the original data limits its use in some applications. On the other hand, it is common to find situations where the classical approach of keeping latent variables as linear projections of the original data may not be adequate. In practice, a researcher needs to evaluate the adequacy of a particular approach based on the problem and he should made a trade off between requirements like simplicity of the solution and of implementation or interpretation of the results.

II

Second Part

5

Fumonisins detection in maize

In this Chapter a preliminary description of concepts involved by a PLS application on real data is reported. First of all, Section 5.1 gives a gradual but complete overview of applicability area introducing all topics that are deepen in the following and showing reasons that are behind this kind of problems and that support further analyses. Then, Section 5.2 presents issues of interest, fungi and mycotoxins, illustrating in a simple way their specific features together with the most troublesome factors related to their development and their management strategies. Section 5.3 goes into details explaining core arguments, Fusarium and fumonisins, and their critical aspects. Finally, Section 5.4 describes applied measurement processes that include both traditional reference analytical methods and some alternatives, based on spectroscopic techniques and chemometric analyses.

5.1 Introduction

A real data PLS application, whose goal is to quickly detect fumonisins contamination in maize meal from NIR spectra, is taken into account as reference situation. Maize plants indeed host Fusarium fungi that produce fumonisins. Since fumonisins are toxic and both cattle and people eat maize-based feed or food, contaminated crops are a very serious problem. In the following the most important elements that characterize this question are introduced and in the next Sections of this Chapter more details are given in order to know step by step considered application field.

Fumonisins are a kind of mycotoxins, poisonous substances produced by metabolism of mushrooms that commonly infect agricultural commodities. In every environment many species of fungi exist as well as many types of mycotoxins are synthesized. Even if fungi and mycotoxins are widely studied in past, there are a lot of aspects about both of them that are not yet well understood. Such variability and uncertainty cause a lot of difficulties in mycotoxin control. As a consequence, after definition of mycotoxins and their features, the most troublesome factors related to fungi and mycotoxins development and their management strategies should be highlighted.

As natural contaminant of maize, Fusarium lives within the plant taking some nutrients from it and changing maize conditions of life as well as plant aspect. Fusarium damages indeed the whole host organism and infects also grains. Then, if it produces fumonisins, Fusarium contaminated maize becomes toxic. When poisonous maize is used as cattle feed or food, it causes diseases both to animals and human beings. There are a lot of animal diseases directly linked with regular and continuous consumption of contaminated maize-based feed and some human illness are probably due to ingestion of fumonisins. However, as regards fumonisins as well as many other mycotoxins, there is a Tolerable Daily Intake (TDI) that, if it is not overcome, assures healthy life conditions. In order to avoid dangerous situations both to cattle breedings and to human beings, international authorities fix such values as law limits for many mycotoxins with respect to different maize-based feed and foodstuffs. Other regulations suggest methods to keep mycotoxins at a practical level.

In order to abide laws limits and at the same time to have no economical loss, maize producers should monitor fumonisins content in maize batches so that only uncontaminated grains can be selected. For this reason, procedures for fungi and mycotoxin detection are very important. Actually, reference measurement processes consist of analytical methods that include a wide array of in-laboratory testing. They are nowadays chosen as quality and safety control tools methodologies for food and feed industries. However, even if accurate, they are expensive and time consuming. As a consequence, some cheaper and faster methods are highly needed. Recently, alternatives based on spectroscopic techniques are for instance been suggested. Indeed, Near Infrared (NIR) spectra bring on useful information about composition of analyzed substance and they can be recorded quickly. Relationship between NIR spectra and fungi or mycotoxin content should then be investigated. Since this is not a so simple and direct link, and it is based on many variables, chemometric analysis and statistical tools are required. With a multidisciplinary approach a successful application of NIR methodology can be obtained in which prediction of mycotoxin can be done for future observations.

5.2 Micotoxins

Food and health are primary requirements for every human being as Maslow's hierarchy of needs (Maslow, 1954 [41]) states but as daily own experience shows, too. As a direct consequence, everyone is interested in consuming safety and healthy dishes that do not cause him disease or illness. At the other hand, from producers' point of view, safety assessment and quality assurance of foodstuffs are the most important goals of industries which take care of consumers' interests to win in competitive market. But health and quality are not only properties of final product, that can be reached in the last phase of productive process, actually they are still required at the beginning of supplier chain, from fields where agricultural commodi-

ties grow and are harvested. Here, real situations are usually characterized by many critical factors that producers should control in order to assure optimal features of products that consumers want to be satisfied.
Presence of mycotoxins in food or feed components is for example one of the most common problem that causes several alarming situations throughout the world.

Agricultural commodities called high organism usually host micro organisms like mushrooms and moulds already in field or after harvest in a symbiotic relationship. Some fungi are indeed defined endophyte, namely they live within another living being without harming apparently plant and ears.
Metabolism is the set of chemical reactions that occur in living organisms in order to maintain life: breaking down organic matter or using energy this process allows organism to grow, developing and maintaining its structures, to reproduce and to respond to their environment. Metabolites are the relatively small molecules of organic compounds that play the role of intermediates or products in chemical reactions of metabolism. There is a distinction between two groups of metabolites based on their functions: primary metabolites are directly involved in normal growth, development and reproduction of organism; secondary metabolites are neither necessary for growth nor for development of fungi but usually have other functions.

Mycotoxins are toxic secondary metabolites produced by organisms of the fungus kingdom including mushrooms, moulds and yeasts.
Their role in fungi life is not ever completely known. Sometimes importance of these compounds to organism is of an ecological nature since they are used as defences against predators, parasites and diseases, for interspecies competition and to facilitate the reproductive processes (colouring agents, attractive smells, etc.). Sometimes mycotoxins are harmful to other micro organisms such as other fungi or even bacteria (penicillin is one example). But in many cases the reason for mycotoxins' production is not yet discovered.
Moreover, unlike primary metabolites, absence of secondary metabolites results not in immediate death, but in long-term impairment of the organism's survivability and fecundity or aesthetics, or perhaps in no significant change at all.
In addition, relationship between fungi infection and mycotoxins occurrence is also not yet very well understood. Once the fungus enters the plant mycotoxins can eventually reach food or feed chain. If plant has visible symptoms of fungi infection but mycotoxins levels are null, there is no problem even if fungi will probably produce mycotoxins as soon as suitable conditions occur. Usually it get worse, because fungi often occurs in symptomless kernels beyond the seedling stage and it could contribute, without visual signs, to the total mycotoxins contamination. So even if there are no visible symptoms of fungal infection, both fungi and toxin can exist. Dowell et al. observed for instance that all analyzed groups of asymptomatic kernels had measurable fumonisins (Dowell et al., 2002 [19]). Furthermore, mycotoxins can

be found in cereals where there are no fungi, because they can persist beyond fungi death and presence. Finally, it should be stressed that both fungi and mycotoxins may occur at each post-harvesting stage, for example inadequate drying of the fruit before storage promotes fungi and mycotoxin presence in cereal batches delivered to elevators or mills. But here fungi development and mycotoxin synthesis can be potentially well controlled, respecting the maximum limits of relative humidity and other environmental conditions.

Usually one species of fungi may synthesize many different mycotoxins and a mycotoxin can be produced by several fungi. Due to minute size of spores, fungi and mycotoxins are found almost everywhere in extremely small quantities; then under suitable conditions fungi proliferate into colonies and mycotoxin levels become high. Mycotoxins can be found in many types of food components like spices, peanuts and nuts throughout the world. But cereals are often the most suitable environment where some species of fungi grow up, live together, develop and produce mycotoxins. Cereals, especially maize (Zea mays L.), are indeed the most commonly contaminated crops in Europe. Fungi infection of field grown maize is a chronic troublesome problem because moulds and mushrooms generally attack the whole plant impairing health and quality, reducing yield and causing great economical losses.

Mycotoxins are poisonous substances that greatly resist decomposition or being broken down in digestion, and neither temperature nor treatments such as cooking and freezing destroy them. So, whereas fungi damage plants, from roots to ears, mycotoxins ingestion can cause several acute or chronic disease (sometimes even death) both to cattle breedings and human beings, considering that cereal grains are components of cattle feed and they contribute to over 60% of the total world food production. In particular mycotoxins enter food chain either directly, with contamination of cereal-based food, or indirectly because mycotoxins can be found still in meat and dairy products.

Table 5.1 reports five mycotoxins considered to be extremely important throughout the world both from a toxicological and economical point of view; fungi which they are produced by and registered disease that they cause to maize-fed livestock and to humans. Aflatoxin (there are 4 types) is the most abundantly found and most commonly known mycotoxin. AFB1 typically occurs in small concentrations living little indication of its presence on the kernel surface; it infects the kernel germ and its effects on cereals are alteration in membrane integrity and inhibition of protein synthesis. Fungi that produce Deoxynivalenol (DON) are the most important and spread agents of Fusarium head blight (FHB), fungal disease of small grains, mainly winter cereal and maize, which reduces yield and seed quality in several areas worldwide.

Fusarium is a species that commonly live in maize and between others mycotoxins it also produces fumonisins (further details are given in Section 5.3).

Table 5.1: Mycotoxins, fungi that produced them and disease they cause.

Mycotoxin	**Fungi Species**	**Disease** (IARC*)
Aflatoxin	Aspergillus (type A. flavus, A. parasiticus, A. nominus); Penicillium	AFB1, AFM are carcinogenic (liver cancer), mutagenic, hepatoxic, immunosuppressive; they belong to group 1 of carcinogen elements classification for animals and humans
Ochratoxin	Aspergillus ochraceu; Penicillium verrucosum	Ochratoxin A is a potent renal toxin in animals, but there are no evidence for renal carcinogenicity
Deoxynivalenol	Fusarium graminearum; Fusarium culmorum	Protein synthesis inhibition, effects on DNA and RNA synthesis, mitochondrial function inhibition, effects on cellular membranes and on cellular correct division, apoptosis; immunosuppressor effect
Zearalenon	Fusarium graminearum	
Fumonisins	Fusarium verticillioides; Fusarium proliferatum; Fusarium graminearum	B1 causes leukoencephalomalacia, pulmonary edema in pigs and cancer in rats. FB1 belongs to “Group 2B carcinogens”, i.e. possibly carcinogenic to humans.
*IARC: International Agency for Research on Cancer, 2002 [35]		

5.2.1 Fungi and mycotoxin development

Even if food-related fungi and mycotoxins were studied extensively worldwide throughout the 20th century, at present there is still a lack of accurate knowledge on their development conditions. Indeed, fungi inoculum and following mycotoxin production can be defined as very complex systems that depend on many variables, someone not yet clearly identified, and whose relationships are still poorly understood.

In general, amount of fungi inoculum in a field varies greatly in their severity and it is caused by many different factors.
First of all use of infected seeds are surely the most obvious source of fungi occurrence in field. Then field location has a primary role because it defines extrinsic environment conditions, as weather variables (like temperature, humidity or drought) and soil properties, that clearly support fungi occurrence in kernels.

Rain and wind are directly involved in process of spores dispersal, because they can scatter fungi even from a distance of 300 – 400 km. In particular rainfall duration and intensity as well as droplet dimension are important: the heavier the rain, the more spores are released from source and lost to ground, but rain intensity is not linearly related to number of dispersed spores. As regards wind, a minimum speed is required for spore liberation and dispersal. However even if some conceptual models were proposed, the specifics of rain and wind role in spore liberation and dispersal have yet to be quantified in detail.

Other sources of inoculum in field are soil itself and cereals residues incorporated into or covering the soil. The litter layer formed by previous crop residues breaks down into organic matter modifying soil's chemical and biological characteristics: humus is less well degraded and soil quickly becomes more acidic, favouring the development of fungi. Moreover, residues keep water on surface of soil increasing release of Fusarium spp spores. In particular previous crop residues like stalks, grains or straw of maize, barley, wheat and other cereals are considered the main inoculum sources for F. graminearum and F. culmorum. Some studies state indeed that about 90% of the Fusarium spp. population is located in the first 10 cm of soil. In addition, it was observed that mean of total amount of previous crop residues (especially maize) on soil surface shows a significant positive correlation with wheat Fusarium spp. infection causing plant disease called FHB (Fusarium Head Blight) and DON contamination; so that analogous results can be expected for other cereals and mycotoxins (Maiorano et al., 2008 [39]).

Also, fungi inoculum is promoted by stress or damage of plant due to insect activity at and after harvest. In temperate areas the second generation larvae of Ostrinia nubilalis, commonly named European Corn Borer (ECB), are an important vector principally of F. verticillioides infection in maize kernels. Its feeding activity is crucial because damaged ears can suffer fumonisins contamination at rates forty times higher than healthy ones. The ECB makes easy the infection of F. verticillioides in two ways: larvae directly damage kernels by breaking the pericarp and giving the fungus a point of direct entry and the same larvae can act as vectors of the inoculum and carry it inside the kernels.

Fungi inoculum may occur in field at any moment of preharvesting but the more sensible stages of maize growth are the late milk and early dough stage of kernel maturity (anthesis and grain filling) when the process of grain dry-down takes place as grain moisture loss. During flowering spores are deposited on tassel, then spores through silk infection enter ears and fungi invade kernels. This process is switched on/off by the appearance and duration of silking which is determined by the maize phenological stage. This is another critical factor of fungi occurrence in maize.

Then, hybrid genotype of host organism promote or prevent with more or less strength fungi development. Indeed it defines intrinsic environment conditions as susceptibility, metabolism, defence mechanisms and other micro climatic factors inside the crop that change fungi levels.

Finally, once inoculum enters maize, most fungi, due to their aerobic nature (use of

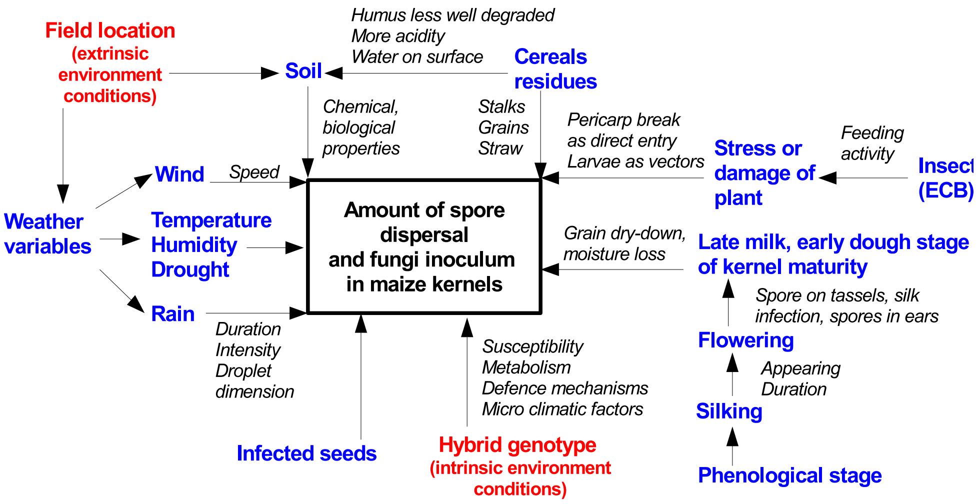


Figure 5.1: Diagram of critical factors that influence fungi occurrence. Arrows explain relationships of cause-effect and more details are given near them.

oxygen), consume organic matter wherever humidity and temperature are sufficient, so they live and grow up producing also mycotoxins.
In conclusion, a lot of variables causes fungal infection in cereal crops and Figure 5.1 summarizes all those listed above. However, since many aspects are not yet under control, further investigation to develop more accurate quantitative relationships between critical factors are hopefully needed. Nevertheless, applying suitable agronomic techniques, effects of these critical factors and of interaction between them on fungal proliferation and mycotoxins synthesis can be reduced. Some examples of best practices are given in the following Subsection.

5.2.2 Risk management strategies

Until now general features of mycotoxins have been described as toxicity, unknown role and uncertain relationships with their producers, fungi frequent occurrence in cereals and cereals wide geographical distribution, high resistance and association with known animal and human diseases. Moreover, previous Subsection presents many variables that influence fungi and mycotoxins development but that can be not so simply controlled. Considering all these critical aspects, fungi and mycotoxins can potentially cause nowadays rather dangerous situations, since they can not be completely removed. However, risk related to consumption of fumonisins contaminated maize-based products both by maize-fed livestock and by human beings can be minimized and yield improved. Indeed, authorities fix some guidelines as regards mycotoxins management strategies and promote agricultural technique modernization in order to obtain higher hygienic and sanitary standards for grain

lots, above all for those destined for human consumption, so that health can be assured. In particular they recommend some agricultural and manufacturing best practices that allow to manage carefully critical factors and interactions between them. As a consequence, fungi and mycotoxin suitable development conditions can be prevented, contamination's probability will decrease, mycotoxins occurrence can be reduced and critical situations for both producers and consumers can be avoided.
EC 583/2006 [11] concerns prevention and reduction of Fusarium toxins in cereals and cereal products and it lists risk factors to be taken into account in good agricultural practices to improve maize grain quality from sowing to harvesting, from storage to transport, from field to dry kiln, mills or farm in every phase and in every place where fungi and mycotoxins can develop. Some examples of risk management strategies are hereinafter described.

Accurate choice of sowing time is for many reasons the main simple strategy that can be adopted to reduce probability of fungi infection. In temperate regions indeed sowing time sets pedoclimatic conditions of the whole next growth period.
First of all a wrong sowing time can change dramatically conditions as relative humidity, temperature and silk wetness that control silking stage.
Secondly, delay of sowing leads also to a later harvest with a more prolonged kernel dry down. As a consequence, extending crop permanence in field increases probability of having ripening final part with both the worst weather conditions and high kernel moisture that favour fungi development.
Thirdly, later sowing times shifts ear development phase to a period with greater insect activity producing more kernel injuries. Considering that mycotoxins are clearly related to ear damages caused by insect activity and that ECB attack is consistent in European temperate areas, early sowing time is preferred to later one, particularly in cooler and wetter sites, in order to obtain a lower amount of stresses and damaged plants. It is indeed an effective technique to reduce ECB impact on maize production and fumonisins accumulation, because it delays appearing of ECB injury and reduces ECB feeding activity. It should hopefully be added to use of chemical treatments, main strategy adopted in Italy to manage ECB damage.

As regards cereal residues on soil, maize debris are the principal substrate where process of infection starts its cycle and represent an ideal source of inoculum up to second year.
Direct drilling and a minimum tillage, with only a partial and very superficial burying, leave a high density of crop residues on surface of soil. As a consequence, even if their cost is almost null they cause a high fungi infection and mycotoxin contamination. Then, expected sanitary quality level of grain is very low.
Stalk bailing, removing residues from surface of soil can be an effective agronomic management strategy to assure a medium-high quality level of grain and it can represent a good compromise between grain safety and management costs. Effectiveness

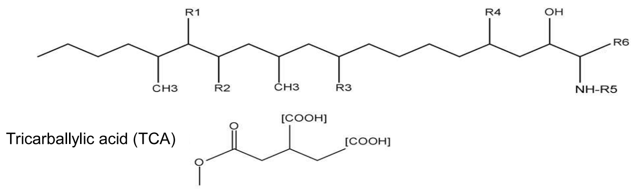
of this strategy can not be completely compared with that of a manual harvesting of residues from soil's surface.
Ploughing is the most useful and safe crop residues management strategy because it removes residues from first layers of soil to a depth of almost $15 - 30$ cm, and thus it highly decreases amount of inoculum source available for disease development. Negative aspect of ploughing is represented by its higher costs. In Italy and in all of Europe during the last years, a trend to reducing tillage was observed for environmental (avoiding erosion and increasing organic matter in soil) and economical (costs of tillage) reasons. As a consequence, minimum tillage and no tillage (direct drilling) systems are used without taking into account the consequences on products health and quality.

In addition, selection of hybrids with higher early vigour and tolerance to biotic and abiotic stresses improves maize productivity and genetic resistance to visible kernel mould. In the last years this is one of the most common practice chosen. On the contrary, use of a late-maturity hybrid could lead to a major risk of mycotoxin contamination as it can prolong length of ripeness and maintain better fungal development conditions.
Finally, plant density and nitrogen fertilization are other important crop techniques that, as sowing and choice of hybrids maturity group, could influence vegetative plant growth and length of maturation period.

5.3 Fumonisins

Fumonisins, first isolated in 1988 by Gelderblom et al., are mycotoxins synthesized as secondary metabolites by sixteen fungi species primarily belonging to Fusarium genus (Gelderblom et al., 1988 [27]). The main producers of fumonisins are Fusarium verticillioides (Saccardo) Nirenberg 1976 (synonym: Fusarium moniliforme J. Sheldon 1904; teleomorph Gibberella moniliformis Wineland) and Fusarium Proliferatum (Matsushima) Nirenberg 1976, both belonging to the Liseola section and Fusarium graminearum from Discolor section (Nelson et al., 1983 [47]).
Twenty eight different fumonisins have been identified to date and they are clustered into four groups (A, B, C and P series) based on structural similarities. Occurrence of fumonisin B_1 (FB1) and fumonisin B_2 (FB2) are the most frequent and have toxicological significance, they supports indeed cancer promoting activity to maize-fed livestock and to humans. On the contrary others types of fumonisins as [B.sub.3], [B.sub.4], [A.sub.1] and [A.sub.2]) occur in very low concentrations and are less toxic.

Fusarium species are among the most common natural contaminant of maize plants (Zea mays) worldwide and compared with several cereals as sorghum, wheat and barley, maize has the highest fumonisins production (Munkvold and Desjardins,



Toxin	R1	R2	R3	R4	R5	R6
FB1	TCA	TCA	OH	OH	H	CH_3
FB2	TCA	TCA	H	OH	H	CH_3

Figure 5.2: Chemical structure of fumonisins FB1 and FB2.

1997 [46]). In particular FB1 usually accounts for about 70% of the total fumonisins content found in naturally infected maize (Rheeder et al., 2002 [54]). Every year indeed Fusarium infections can be isolated from most field-grown maize and fumonisins contaminations are reported in the most important maize areas, especially where maize is a monoculture. Visible symptoms of Fusarium infection in maize ear can be reddish-purple streaks restricted to the pericarp, a small portion of the endosperm showing a white discolouration, shriveling over the kernel surface. In addition, Fusarium infects also cereals' seedling, roots, stalks and grains robbing nutrients, decreasing fat, protein, vitamin content, often changing their colour and consistency (texture). However both Fusarium and fumonisins can be found in symptomless kernels and plants, too.

Fusarium have been historically recognized as the cause of different potentially serious disease of maize-fed livestock. Fumonisins, indeed, bring on acute or chronic primary mycotoxicosis since they inhibit the biosynthesis of complex sphingolipids, endogen antitumor agents. Moreover, fumonisins mycotoxicosis can involve secondary maize-fed livestock disease, so that fumonisins can be considered carcinogens and immunosuppressive causing sometimes animals' death. For instance, FB1 is directly related to equine leukoencephalomalacia[1] (Marasas et al., 1988 [40]) and pig pulmonary edema (Harrison et al., 1990 [30]). In several animal species, as rats and mice, it is hepatoxic and nefrotossic (Gelderblom et al., 1991 [28]), then it is linked to hypercholesterolemia, immunological alteration, renal and liver toxicity.
Fumonisins are characterized by an high thermostability, since their molecular structure (reported in Figure 5.2) can be destroyed only after thermic exposure greater than 220 ℃. Moreover, fumonisins are not metabolized by animals so that, due to carry over, their occurrences could be found in cattle products like milk, cheese, meat or eggs. As a consequence, human contamination has two kind of sources: consumption of directly contaminated maize-based dietary staples or of indirectly

[1] Leukoencephalomalacia (ELEM) is a fatal brain disease of horses, donkeys, mules, and rabbits.

toxic food produced by infected animals. A possible association between human esophageal cancer and a diet based on maize with high levels of fumonisins in South Africa and China has been suggested, although at present no definitive conclusion can be made. International Agency for Research on Cancer states that there is inadequate evidence in humans for carcinogenicity of fumonisins so that it defines FB1 belonging to "Group 2B carcinogens", i.e. possibly carcinogenic to humans (IARC, 2002 [35]).

Beyond previously cited best practice, international authorities set also some legal limit for mycotoxins with respect to several maize-based feed and foodstuffs in order to avoid unsafe situations both to animals and humans.
EU Scientific Committee established a Tolerable Daily Intake[2] (TDI) for fumonisins with respect to several maize products intended both to animals or humans. Commission Regulation (EC) No 1126/2007 [13] amending (EC) No 1881/2006 [12] sets maximum levels for certain contaminants in several foodstuffs as regards Fusarium toxins in maize and maize products. In particular law limits for the sum of FB1 and FB2 with respect to different kind of food are listed in Table 5.2, whereas Table 5.3 shows law limits for cattle feed as reported by Commission Recommendation 2006/576/*EC* [10].

[2] TDI is an estimate of mycotoxin amount in food and feed that can be taken daily over a lifetime without appreciable health risk.

Table 5.2: Maximum levels for fumonisins sum [FB1+FB2] in maize and maize-based foodstuffs as fixed by (EC) No 1126/2007.

2.6	**Fumonisins**	**[FB1+FB2]** ($\mu g\, kg^{-1}$)
2.6.1	Unprocessed maize, with the exception of unprocessed maize intended to be processed by wet milling*	4000
2.6.2	Maize intended for direct human consumption, maize-based foods for direct human consumption, with the exception of foodstuffs listed in 2.6.3 and 2.6.4	1000
2.6.3	Maize-based breakfast cereals and maize-based snacks	800
2.6.4	Processed maize-based foods and baby foods for infants and young children	200
2.6.5	Milling fractions of maize with particle size $>$ 500 micron falling within CN code 1103 13 or 1103 20 40 and other maize milling products with particle size $>$ 500 micron not used for direct human consumption falling within CN code 1904 10 10	1400
2.6.6	Milling fractions of maize with particle size $\leq$ 500 micron falling within CN code 1102 20 and other maize milling products with particle size $\leq$ 500 micron not used for direct human consumption falling within CN code 1904 10 10	2000
*The exemption applies only for maize for which it is evident e.g. through labelling, destination, that it is intended for use in a wet milling process only (starch production).		

Table 5.3: Maximum levels for fumonisins sum [FB1+FB2] in maize and maize-based feed as fixed by Commission Recommendation 2006/576/EC.

Fumonisins	**[FB1+FB2]** ($\mu g\, kg^{-1}$)
Feed materials: maize and maize products	60
Complementary and complete feedingstuffs for:	
pigs, horses (Equidae), rabbits and pet animals	5
fish	10
poultry, calves ($<$ 4 months), lambs and kids	20
adult ruminants ($>$ 4 months) and mink	50

5.4 Measurement methods

Considering that large amounts of cereals are processed in food and feed industry every day and that really serious consequences are caused by mycotoxin ingestion, farmers and breeders, food producers in general, have to pay great care and extreme attention to abide law limits in order to assure healthy standards that prevent risk both to humans and animals, and to avoid economic loss due to both legal sanctions and dead stock. Since, as previously seen, mycotoxin contamination starts in field and even following the best agricultural and manufacturing strategies their complete removal is impossible, the last solution to reduce critical situations is extraction of contaminated lots from productive chain and their eventual use in some different business. Producers are so primarily interested in assessing mycotoxins presence in their own commodities because, in the logic of free market, this knowledge assures competitive gain.

As a consequence, it is necessary to monitor and accurately measure mycotoxins content in cereal batches so that grains can be screened and only those with low levels of mycotoxins can be selected as feed or food. Methodologies for quality and safety control of raw materials able to detect as soon as possible undesirable or toxic substances are so required. Ideal method should comply some basic properties as accurate measurements, then if it assures also cheap equipment, fast and simple procedure and it is nondestructive it would be better. Moreover, measurements are needed along the whole productive chain but of course, the earlier they are done, the lower is economical loss. So, it will be a smart idea thinking to procedures that could be used directly at harvesting on farm yard or that can screen for fumonisins occurrence in cereal batches delivered to elevators or mills.

Unfortunately, due to their intrinsic nature better described hereinafter, existing techniques provide accurate results but they are expensive, time consuming and only experts can manage their elaborate phases in laboratory, so that traditional methods are unsuitable for real-time control measures. As a consequence, alternatives for fungal and mycotoxins detection and quantification in food and feed components should be looked for. They should be of course accurate, but hopefully they should work also quickly and with minimum effort and cost.

NIR methodology, illustrated below, is a successful solution for this kind of problems. It consists of a multidisciplinary approach that involves general Spectroscopy, reference Chemistry, Statistics and Computing. NIR spectroscopy provides a technology for recording information of biological materials. In particular measured spectrum of a sample can be considered as its fingerprint, so that it brings on all the relevant needed data. Statistical analyses, as specific inference methods based on multivariate regression (like PLS calibration and full cross validation), change this information in knowledge, finding out relationship between spectra and mycotoxins level. Core business of this kind of statistical analyses is indeed to define a model based on NIR spectroscopy data for detecting mycotoxins in cereals. Then, with obtained statistical model prediction of mycotoxins value can be done for future observations. Both collecting NIR spectra and application of statistical model are few time consuming activities. Moreover, they do not require expensive equipment nor trained staff, and they are nondestructive. For this reasons, they can be considered as measurement method for safety and quality evaluation of agricultural commodities.

In conclusion, NIR methodology could provide a first screening method to be used along the production chain and it would be a great potential for on line monitoring quality control in food and feed industries or inspection administration services at the maximum mycotoxins level allowed by legislation. Indeed, it can detect mycotoxins in cereals samples. Then, a following selection of cereals harvested batches with low or negligible fumonisins levels can be done. Finally, cereal quality and health standards can be achieved.

5.4.1 Reference analytical method

To guarantee that mycotoxins are kept at a practical level, various compliance programs are actually chosen as tools to control food and feed industries. In particular, the process of mycotoxin regulation includes a wide array of in-laboratory testing. Usually, every general analytical method consists of several phases as extensive extraction, clean up and separation techniques (derivatization procedures). After sampling, mycotoxin of interest should indeed be extracted from matrix. The extraction is usually followed by a clean-up step where unwanted substances, which may interfere with the detection of the analyte, are removed from the extract. A final separation step will complete the procedure (Krska et al., 2007 [38]).

At present, the most official regulations and control methods are based on high-performance liquid techniques (HPLC) whose standards in Europe are set by the European Committee for Standardization (CEN). Other quantitative methods of analysis for most Fusarium toxins are gas chromatography (GC) with or without mass spectroscopy, molecular immunoaffinity fluorescence identification techniques as enzyme-linked immunosorbent assay (ELISA), or thin layer chromatography (TLC). In past HPLC with an electro spray mass spectrometer and evaporative light scattering detectors, capillary zone electrophoresis or ion pairing chromatographic separation with post column derivatization and subsequent fluorometric detection were sometimes used.

Fusarium mycotoxins constitute a very heterogeneous group of compounds and separate procedures are usually applied for quantification of individual toxins; however, multiple analytical protocols for determination of various toxins belonging to different groups are also available. Simultaneous screening of mycotoxins was for instance developed using an image analysis system in combination with a line immuno blot assay.

All these available chemical procedures are the most widely accepted reference methods for determining mycotoxin levels in cereals. They assure indeed reliable and accurate measurements. Moreover, they are the best techniques when compared with alternatives studies that pay attention to relationship between fungal mass (ergosterol) and mycotoxin content, considering that they are not strictly related. Nevertheless, existing methods are considered time consuming, because of extensive procedures which every phase consists of. The more recent techniques require indeed at least 30 min to process each mixture. Moreover, expensive solvents and reagents are needed and only trained staff can do such complex chemical analyses. In addition, these methods are frequently destructive and involve large quantities of grain. Finally, reliable analytical methods does not still exist for some mycotoxins causing lack of sufficient surveillance data.

5.4.2 NIR measurements

Organic molecules have specific absorption patterns in NIR region of electromagnetic spectrum that report chemical composition of biological material being analyzed (Williams and Norris, 2001 [73]). For this reason, both reflectance and transmittance[3] NIR spectroscopy was used routinely to detect organic compounds and nutritive parameters as protein, oil, starch, fatty acids and moisture content in matter like meat, milk, diary products, soft and durum wheat, maize, barley, oats, soy and oleaginous seeds or flours. For example single kernel NIR data have been used to develop a predictive model for wheat protein content (Delwiche and Hruschka, 2000 [18]). Simultaneous determination of multiple constituents is also possible.

But recent studies expand fields of NIR applications, and use NIR measurements also to detect a wide range of critical factors in a lot of different elements. Pearson built for instance a near infrared inspection system to automatically detect whole almond kernels with concealed damage (Pearson, 1999 [50]). Moreover, Dowell et al. showed that NIR spectroscopy is able to differentiate uninfested wheat kernels from seeds infested with larvae of three primary insect pests (Dowell et al., 1998 [21]). Furthermore, NIR measurements are applied to identify different mycotoxins in many commodities, as aflatoxins in spices (Hernández-Hierro et al., 2008 [32]) or DON in single wheat kernels (Dowell et al., 1999 [20]).

Regarding maize, nowadays visible and near infrared spectroscopy are commonly used to predict composition of maize kernels as content of gross energy, oil, protein, starch, and fatty acid. However, comparisons between transmittance or reflectance, between maize kernels or meal, between absolute or relative values were studied in order to define which NIR measurement is better. In the first years of nighties, Orman and Schumann evaluated for instance accuracy of protein, oil, and starch content prediction in maize grain, using Near InfraRed calibrations developed from three types of spectral data: diffuse reflectance on both ground grain and whole

[3] NIR spectra can be collected either from reflectance of element or transmittance through substance.

grain, and diffuse transmission on whole grain (Orman and Schumann, 1991 [48]). They found that the best constituent predictions were by equations developed from ground grain diffuse reflectance data.
Moreover, Cogdill et al. confirmed that attempts to develop single kernel predictive models for maize oil content using NIR transmittance would be difficult to develop (Cogdill et al., 2004 [9]). Indeed, potentially, transmittance spectra are more sensitive to density or total mass of kernel, because maize kernels are relatively large and longer wavelength light does not penetrate individual kernels.
In addition, a more recent research of Baye et al. tried to determine if NIR reflectance or transmittance spectroscopy could be used to predict absolute or relative composition (protein, oil, fatty acids, calorie, starch and weight) in both maize kernels and meal (Baye et al., 2006 [4]). They concluded that transmittance spectra have high levels of noise and are not suitable for estimating internal kernel composition, as well as percent values of constituents in meal give poor predictions, whereas single kernel NIR reflectance spectra are reporting an absolute amount of each component.

In the last years, interest on NIR spectroscopy widens up to include fungal infections and mycotoxins presence in maize. Some examples are listed hereinafter. Fernández-Ibañez et al. explored and evaluated aflatoxin AFB1 detection in intact, nonmilled grains of maize and barley (Fernández-Ibañez et al., 2009 [22]). Pearson et al. analyzed transmittance and reflectance spectra to show that both could be used to distinguish aflatoxin contamination in single whole corn kernels (Pearson et al., 2001 [51]). Kramer et al. investigated whether NIR spectroscopy of maize single kernels could be used to identify transgenic kernels containing high levels of avidin, which is toxic to and prevents development of insects that damage grains during storage (Kramer et al., 2000 [37]).

Finally, considering fumonisins in maize, Dowell et al. used reflectance and transmittance visible and NIR spectroscopy to detect fumonisins in single corn kernels infected with Fusarium verticillioides (Dowell et al., 2002 [19]). With their research, they conclude that NIR analysis of many kernels will not be as sensitive to fumonisins contamination as single kernel analysis. Moreover, classification results were generally better for oriented kernels than for kernels that were randomly placed in the spectrometer viewing area. In addition, as previously said, models based on reflectance spectra have generally higher correct classification than models based on transmittance spectra. In addition, Berardo et al. checked for instance that NIR could accurately predict percentage of F. verticillioides infection in maize kernels, as well as the quantity of ergosterol and fumonisin B1 in maize meal (Berardo et al., 2005 [6]).

NIR spectra measurements are in practice very simple and only previous phase of material preparation demands little attention. Indeed, as in every measurement process, a standard procedure should be followed in order to assure reproducible sampling and to obtain spectra that can be compared and analyzed together avoiding noise in the following statistical analysis caused by some variables that are not taken into account. For this reason, researchers should decide for example how place single kernel into spectrometer viewing area and staff should successively maintain the same starting conditions for every analysis. Similarly, if maize meal is used, same granularity should be guaranteed during milling process and NIR should analyze every time the same amount, characterized by a homogeneous and compact surface in contact with glass of dish.
Considering this, measurements of NIR spectra are faster than traditional techniques because materials can be screened rapidly (less than one minute) and, as described above, minimum preparation is required with respect to elaborated and time consuming procedures of chemical methods. Moreover, NIR spectroscopy needs no reagents during analyses, so it is cheaper than analytical methods, too. Due to its simplicity, no expert staff is required. Finally, NIR spectroscopy is a nondestructive technology since it preserves maize after the measurement for further analyses.
As a consequence, NIR spectroscopy may have practical applications toward on line samples screening, and automated systems for detecting properties of single seeds in real time have been described. For example, in contrast to many other procedures used to detect internal insect infestations in wheat grain, previously cited system developed by Dowell et al. could be incorporated into the current grain inspection process and provide the grain industry with quantitative data on internal insect infestations in wheat. Similarly, technology of Pearson's study should provide a valuable tool for rapidly detecting aflatoxin in corn. Finally, technology developed by Dowell et al. can be used to rapidly and nondestructively screen single corn kernels for the presence of fumonisins, and may be adaptable to on line detection and sorting. However, the first result that give importance to NIR spectroscopy is Canadian wheat grading and classification methods; its precision is comparable to official methods, but it has a much lower cost.
From a technical point of view, a comparison between NIR spectroscopy tools has confirmed quality of spectral data when using Fourier transform instruments (FT-NIR) as opposed to NIR dispersive (grating instruments) because their use can improve accuracy and quality of spectral informations. Moreover, regression models developed on FT-NIR spectral data allow to obtain better statistical results without maths pretreatment.
Other methods able to detect fungal infection and toxic metabolites in infected corn samples are infrared photo acoustic spectroscopy (IR-PAS); diffusive reflectance spectroscopy (DSR); mid infrared spectroscopy with attenuated total reflection (ATR); single kernel nuclear magnetic resonance (NMR) technology for liquid constituents.

6

Model fitting with PLS

This Chapter presents an application of PLS regression to real data. In this case, PLS regression is iteratively computed in order to define the relationship between fumonisins content in maize and its NIR spectra. Goal of the research is indeed the discovery of an accurate, fast, cheap, easy and nondestructive method able to detect fumonisins contamination in maize. First of all, data are described in Section 6.1. Here, procedures of data collection are explained, that involve sampling and applied measurements methods. Variable of interest, that consists of fumonisins contamination level of maize, is also analyzed. Then, Section 6.2 shows the final result that is the best model obtained during regression analyses. Both numerical values and graphical outputs are given and discussed. Finally, last Section 6.3 makes some conclusions in order to highlight the most important aspects that should be more accurately taken into account in future.
The outcome depicted in this Chapter resulted from a number of different analyses in which several hypothesis are tested and specific work procedures are followed. In particular, among all the evaluated models, an interesting previous one was presented in Gaspardo et al. (Gaspardo et al., 2012 [26]). Since those results arose from a partial analysis and they were promising, further researches were done in order to improve them and to extract as much information as possible from available observations. The following model belongs to these later analyses and it is also described in a corresponding article submitted to Food Chemistry (Della Riccia and Del Zotto [17]).

6.1 Dataset

Dataset (134 rows × 927 columns) contains information on 134 instances, i.e. analyzed maize amounts, about their NIR spectra (926 wavelengths) and their fumonisin contamination level, measured by HPLC technique. Hereinafter, followed data collection procedures are described: firstly sampling method, then HPLC, finally NIR measurements.
Sampling method allows to define where, when and how maize amounts should be picked up in order to build a consistent and significant dataset. All the observations should indeed be chosen following decided rules, so that they comply same

properties and they can be compared together. By doing so, instances represent population they belong to. As a consequence, results extracted from specific and limited available data have a more general meaning and they can be extended to the whole population. Moreover, in this way, unknown sources of variability and uncertainty are hopefully removed and noise brought on data is minimized. Finally, helpful informations provided by sampling method are usually summarized in the label that identifies every instance. These further data can become explicit if they fill new variables. Then, more useful analyses can be done.

As regards variable of interest, fumonisin contamination level in maize, presence of fumonisins FB1 and FB2 are measured and their sum considered for the following survey. Chosen reference analytical method is HPLC. Procedure and equipment are described in detail. Then, a preliminary descriptive analysis is developed and some critical aspects are highlighted.

Procedure to record NIR measurements is given, as well as NIR instrument used and its specific features. Chosen NIR variables are declared and some examples of NIR spectra are shown.

6.1.1 Sampling

Maize amounts are collected in Friuli Venezia Giulia Region (Italy) from 4 drying and storage stations and 18 dairy farms located in 5 different areas previously detected on the basis of latitude, pedoclimatic conditions, agronomic features and farms concentration, such that representative data would be available. Maize just harvested are collected during September and October 2008 from drying and storage stations.

Maize harvested and stored within farm silos for at least 8 months are taken three times (May, July and October) for two consecutive years (2008 and 2009) from dairy farms.

Sampling is carried out according to EC Regulation No 401/2006 [14] and, after drying at 60 ℃ over night, maize grains are ground in a laboratory mill for 2 min at high speed, mixed accurately to ensure homogeneity and stored in a cool, dry place until analysis.

6.1.2 HPLC

FB1 and FB2 concentration in maize is determined according to the AOAC Official Method 2001.04 (Visconti et al., 2001 [72]).

FumoniTest Wide Bore immunoaffinity columns and ortho-phthaldialdehyde solution are purchased from Vicam (Watertown, MA); HPLC-grade solvents and sodium dihydrogen phosphate are from Sigma-Aldrich (Chemie GbH, Steinhiem, Germany); fumonisins from Alexis Biochemicals, Axxora (Deutschland GmbH) is used; a Nova-Pack C18 (3.9 × 150 mm, $4\mu m$ particle size, Waters, Ireland) is employed.

HPLC system (Varian Analytical Instruments, USA) is equipped with a Prostar 363

Table 6.1: Values of some statistics for response variable fumonisins sum [FB1+FB2].

[FB1+FB2] mg kg^{-1}	
N. of missing	0
Min	0.357
Max	11.845
Mean	3.353
Standard deviation	2.188
Skewness	1.057
Kurtosis	1.512

fluorescence detector set at 335 nm excitation and 440 nm emission.
Twenty grams of maize were analyzed each time.
Fumonisins sum is used in the following statistical analysis and it is denoted by [FB1+FB2]. Values are in “ppm” scale, that means “parts per million” and is equivalent to mg kg^{-1}.

A preliminary descriptive analysis on variable of interest is done both with graphic tools, as histogram, and numerical results like main statistics (mean, min, max, standard deviation, number of missing data, etc.).
Sum of fumonisins FB1 and FB2 varies from 0.357 mg kg^{-1} to 11.845 mg kg^{-1} with a mean value equals to 3.353 mg kg^{-1}, there are no missing values and [FB1+FB2] main statistics are reported in Table 6.1. Positive skewness and histogram (Figure 6.1) confirm that statistically there are less highly contaminated observations than instances with low value of fumonisin contamination. In particular there are only 59 observations with [FB1+FB2] greater than 4 mg kg^{-1}, whereas the remaining 66% observations do not exceed this threshold, that is the law limit fixed by European Commission with respect to human consumption of unprocessed maize.

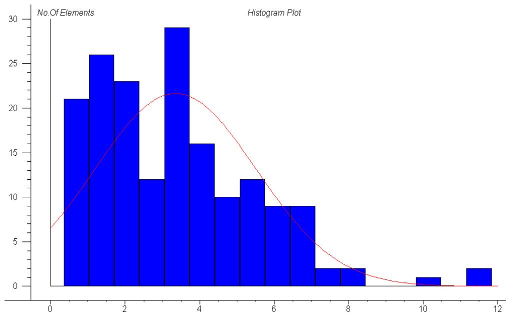


Figure 6.1: Histogram of [FB1+FB2] dependent variable that records fumonisins contamination level measured by HPLC.

6.1.3 FT-NIR

One hundred grams of maize meal are scanned with a Perkin Elmer FT-NIR spectrometer (Perkin Elmer Italia S.p.A.) using the integrating sphere with an adapter Sample Cup Spinner for rotation. Scanning time is approximately 50 sec. Software Spectrum v.5.3 (Perkin Elmer Italia S.p.A.) installed in a personal computer interfaced to spectrometer records FT-NIR spectrum as result of the following procedure. An interferogram, a component of the detector signal modulated as a function of optical path difference, is firstly stored. Successively the interferogram is converted to a frequency domain single beam spectrum. A reference reflectance spectrum, that had been taken before scanning, is finally subtracted from the single-beam spectrum to give a spectrum collected in diffuse reflectance mode. This procedure is repeated until fifty spectra are done and automatically averaged into one absorbance spectrum. Absorbance is defined as the logarithm of the inverse of reflectance. Values are collected at 2 nm of interval for wavelengths between 650 nm and 2500 nm (926 variables). The final spectral data are exported into standard JCAMP data format for the following statistical analysis.

Every NIR spectrum consists of absorbance value for each wavelength from 650 nm to 2500 nm at intervals of 2 nm recorded in 926 predictors (Figure 6.2 shows some spectra as examples).

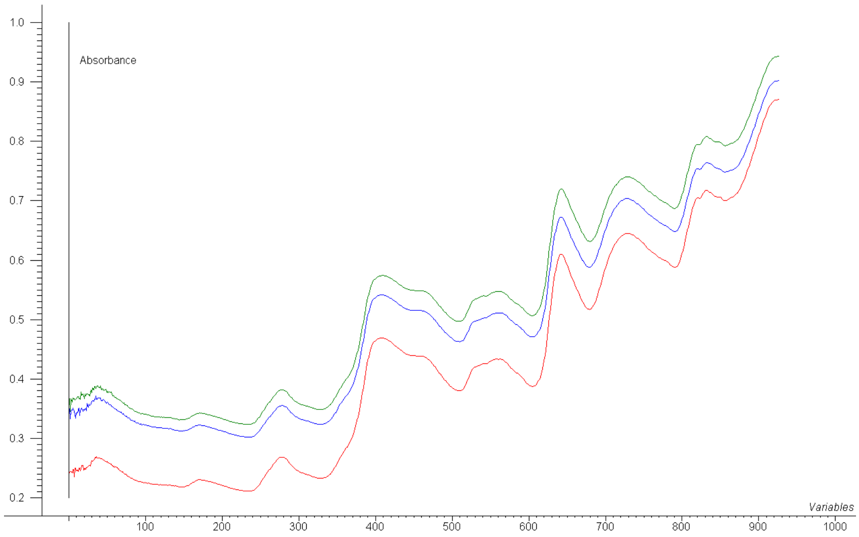


Figure 6.2: Examples of maize NIR spectra.

6.2 Results

Statistical inference is applied in order to explain and quantify the relationship between fumonisins contamination level (response or dependent variable) and maize NIR spectra (926 explanatory variables or predictors).
Specific techniques are required with spectral data, because of predictors high cardinality and collinearity. In this case, multivariate linear regression is applied choosing Partial Least Squares (PLS) with full cross validation and every computation is applied to centered data. After running PLS algorithm with full cross validation, graphical and numerical results are provided. They allow to control outcome model, verifying whether it satisfies required criteria and parameters. Summarizing statistics are indeed calculated from both calibration and validation residuals. Moreover, several graphical tools show model's properties.
All the chemometric analyses are developed with the software The Unscrambler v.9.6 (CAMO Software AS, Oslo, Norway) that applies the algorithm PLS1 to compute the regression model for a single response variable[1].

Model defined through PLS with full cross validation should have a twofold purpose: i) to assure screening of maize amounts with respect to threshold of 4 mg kg^{-1} in order to comply European Regulation for human consumption of maize-based products; ii) to provide a tool for classification and rather accurate predictions into the whole scale of fumonisin contamination level.
Both these aims, together with properties that every statistical model should comply, suggest some criteria and some guide lines for model assessment.
First of all final model should separate, correctly and without false positives, observations of dataset with respect to EU defined threshold of 4 mg kg^{-1}.
Secondly, indexes of models goodness and their predictive ability should be satisfactory. In particular correlation between measured and predicted values is taken into account. Correlation shows rate of similarity between measured values and predicted ones, telling how much near they are, so it is a pointer of model's accuracy. Ideal model has correlation between measured and estimated values that is as high as possible and predicted vs measured plot should show points along the diagonal.
Thirdly, residual variance should be decreasing and final model should not overcome a reasonable number of PC. Calibration or validation residual variance is a measure of information that model can not taken into account. It is defined as the mean of the squared deviation of residuals from their distribution's average. Aim of regression analysis is to find a model that minimizes residual variances. So, plot of both calibration and validation residual variance versus number of PC included in

[1] More details about the algorithms used by the software The Unscrambler v.9.6 can be found in the document "The Unscrambler Appendices: Method References" on the web page http://www.camo.com/downloads/

the model should show decreasing lines and it suggests, where variance is minimum, how many PC enter into the model.
Finally, with respect to the previous partial analysis (Gaspardo et al., 2012 [26]), the following researches were done in order to i) improve full cross validation outcomes; ii) do a more precise survey of wavelengths that enter into the model; iii) extract as much information as possible from observations.
Analyses are developed in order to meet all these requirements, even if sometimes a trade off between opposite goals should be made. Some aspects are so discussed in order to show which ones are chosen as priorities.

Regarding regression analysis, several PLS models with full cross validation have been tried considering as starting point the whole dataset and all the wavelengths. Then, in order to refine obtained results, some rather little changes on observations and subsets of variables taken into account are tested, making different hypothesis for instances and variables that enter into computations. Indeed, some maize amounts and some ranges of wavelengths show specific properties when different models are fitted. As a consequence, their behaviour should be controlled and effect of their presence or absence into model should be afterwards evaluated.
Finally, first 52 variables are excluded from every following computation and one observation is extracted from dataset. Removed wavelengths belong to visible spectrum (650 – 752 nm), whose signal brings too much noise making more difficult extraction of significant information. This is probably due to NIR instrument that works also with visible spectrum even if it should be considered an extreme range where spectra reproducibility, accuracy and precision can not be assured. Similarly, chosen observation behaves differently from others, standing alone in plots and showing unexpected values. This is likely caused by errors occurred during measurement process such as following a procedure that is not the standard one or being in presence of polluting substances in the measured amount. A model is so fitted with this new reduced dataset and its results are hereinafter described.
Calibration set has 133 observations and Table 6.2 summarizes defined model reporting its statistics and the values of its more important indexes. In particular some results quantify shape of regression line, as slope and offset, whereas the remaining characterize probability distribution of model's errors.
The curve of residual variance is a monotonically decreasing function with respect to PC's number considered in the model (Figure 6.3) and they suggest a number of PC equal to 21 that is rather low compared with an amount of almost thousand original variables. In particular increments become irrelevant after that 21 PC enter into the model (in the plot, line tends to be horizontal after the 21th PC). Relation between fumonisin content in maize and NIR spectra can so be expressed by a linear combination of 21 PC. With less PC models would be incomplete and unable to describe correctly and exhaustively phenomenon under study; while with more PC models would over fit and consider many informations, even those that could be

Table 6.2: Summary statistics and values of most important indexes of described models.

Final model		
number of observations in CS		133
number of PC		21
number of removed variables		294
Calibration	Correlation	0.998
	Slope	0.995
	Offset	0.015
	Bias	5.724e-05
	SEC	0.144
	RMSEC	0.143
Validation	Correlation	0.909
	Slope	0.799
	Offset	0.669
	Bias	-0.012
	SEP	0.893
	RMSEP	0.890

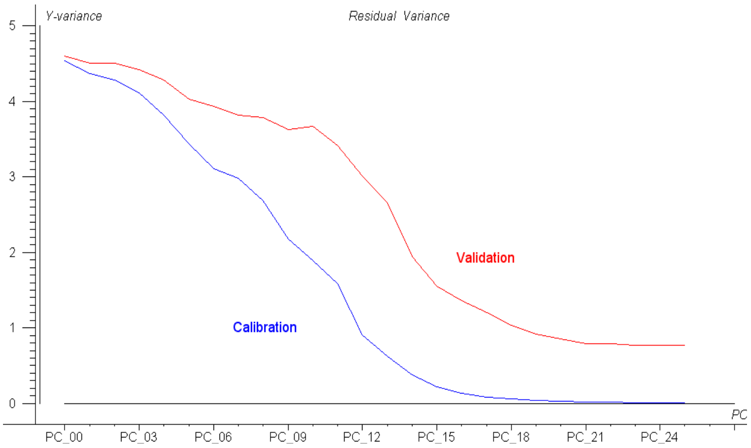


Figure 6.3: Plot of calibration and validation residual variance versus number of PC.

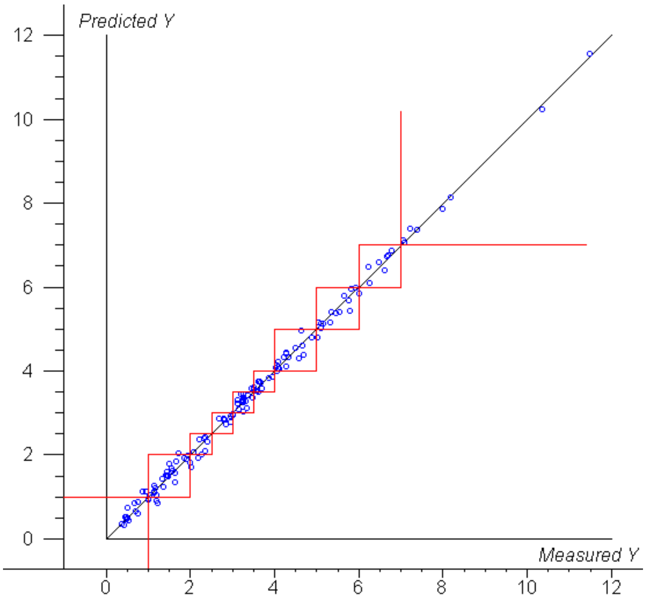


Figure 6.4: Calibration plot: measured [FB1+FB2] versus calibrated values. Threshold equal to EU limit of 4 mg kg^{-1} is highlighted as some other classes.

noise.

Calibration and validation value of every observation that enter into model can be compared with its corresponding measured value. So, predicted and measured values can be plotted into a graph in which points display themselves hopefully along diagonal and correlation between them can be computed. Correlations of calibration and validation are respectively equal to 0.998 and 0.909. Since both correlation values are very high, we can state that model i) describes accurately the relation between fumonisin content and NIR spectra avoiding overfitting; ii) would have a good prediction ability for future unknown observations that could be defined as members of the population to which dataset belongs (i.e. they have similar properties with respect to observations of the dataset, such that model can be applied).

Further remarks can be done. Indeed, in order to evaluate screening and classification ability of the model, lines corresponding to target thresholds can be added in calibration plot (Figure 6.4). They allow to visualize easily if there are false positives and/or false negatives (by positive we mean contaminated above 4 mg kg $^{-1}$), if estimated values agree with measured one or if some observation is not fitted in the right group. First of all, analyzing calibration plot around value of 4 mg kg^{-1}, we can observe that no observation is misclassified and that legal limit is satisfied without false positives and/or false negatives. Since no contaminated maize amount is predicted as safe one and vice versa, the developed model assures a good screening ability. Moreover, some other interesting intervals are drawn in this figure. Every calibrated

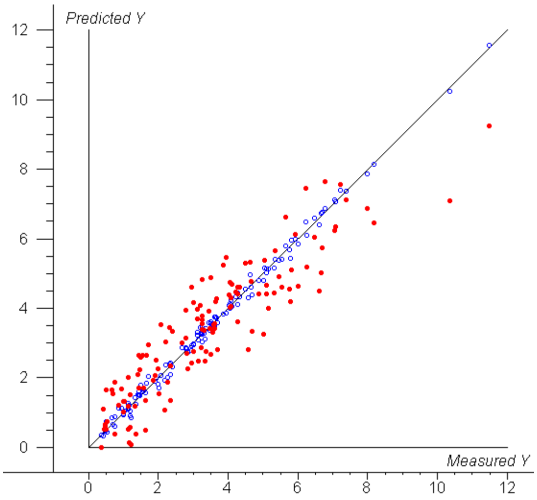


Figure 6.5: Calibration and Validation plot: measured [FB1+FB2] versus calibrated values (blue circled points); measured [FB1+FB2] versus validated values (red filled points).

contamination level is indeed sufficiently accurate so that no point is outside highlighted areas. As a consequence, model can be used also as a classifier. Identified classes are $[0;1)$, $[1;2)$ $[2;2.5)$ $[2.5;3)$, $[3;3.5)$, $[3.5;4)$, $[4;5)$, $[5;6)$, $[6;7)$ and values higher than 7 mg kg^{-1}. In particular, law limits equal to 1 and 2 mg kg^{-1} are also in evidence, they are two other maximum levels established in Commission Regulation (EC) No 1126/2007 as reported in Table 5.2.

Finally, behaviour of two observations whose fumonisin level overcomes 10 mg kg^{-1} should be noticed. Even if both are well calibrated, their validation value is rather distant from diagonal (Figure 6.5). This is not troublesome, because actually there are few high contaminated maize amounts, but if similar extreme observations are collected in the future and enter into the model, these kind of values will support each other and their validation will improve. Further studies, new data (in particular more observations with high level of fumonisin content) and other analyses are required so that model will be more robust and its behaviour as accurate as possible for highly infected maize amounts, too. In this research, collected values about fumonisins belong to a wide range but descriptive statistics highlight an unbalanced data distribution with few extreme samples. Such levels of fumonisin contamination correctly reflect historical trend of the last years in Friuli Venezia Giulia Region. From a strictly statistical perspective, this situation shows two alternatives. Described result presents an unique complete model that covers the whole range of fumonisin contamination level. In this case, model definition forces software suggestions about observations with high fumonisin content considered as outliers due to their low and weak frequency. On the other hand two distinct models can be estimated one for low levels of fumonisin contamination, the other for extreme observations if original dataset could be split into two subsets (threshold should be accurately chosen). Future researches could consider this hypothesis to develop other models.

Described model is characterized by interesting and promising properties. First of all, its correlation of calibration is very large as a consequence, model assures an accurate fitting of data. Secondly, it has high correlation of validation and rather low number of PC. Both work benefits to the model because they involve a good prediction ability for a simple model. However, a very important feature is given by set of variables that enter into the model. From this point of view, some considerations should be discussed and a trade off between opposite goals should be made before taking some decisions or reaching a conclusion.

As to the number of variables to be used in the definition of the model, all frequencies are considered as starting point. Then, in order to improve the results it is checked whether some variables should be removed. At the end the model reduces its number of independent variables dropping three intervals of wavelengths and combining 632 variables to build PC. On the contrary several models that work with all the wavelengths, except those belonging to visible spectrum, were developed during analysis. Comparisons between these two opposite approaches can be done. Last strategy is surely the most universal, complete, and with a general validity whereas the former focusing on particular intervals risks to ignore important features of data. Moreover, we lose almost 300 variables to gain few PC, while we would expect to simplify strongly the model in front of a so high number of removed wavelengths. Nevertheless, since outcomes of model are very fulfilling we can state that excluded wavelengths are redundant and bring on noisy information, maybe they record only noise, so that they can be disregarded without troubles. In addition, first strategy can be supported by further analyses and researchers that follow a chemical approach according to which spectra can be used in a selective way. Indeed someone can suggest that a more direct relation exists between fumonisin content and a specific, limited and well defined range of NIR wavelengths due to fumonisin chemical structure so that other unnecessary ranges can be removed, corresponding presumably to humidity or other chemical components unrelated to fumonisins. As a consequence, in future chemists can confirm strategy adopted and will help to choose the most appropriate range of NIR wavelengths.

In conclusion, such aspects should be investigated more deeply. For example a choice between two approaches should be done by a work and research group with members coming preferably from different scientific fields, in order to evaluate their advantages and drawbacks, to take into account every interest and to decide the best shared solution.

6.3 Discussion

Maize is a dietary staple both for cattle breedings and human beings in many countries. Fusarium development in maize is a very complex phenomenon that depends on environmental patterns (relations and interaction between plant, soil and weather properties). Fusarium species damage maize plant then, when suitable conditions appear, produce fumonisins as secondary metabolites. Factors that cause fumonisins infection are not so clear. Fumonisins role in fungi life is also not always known. Moreover, Fusarium presence does not always involve fumonisins occurrence and at the same time Fusarium absence does not imply lack of fumonisin contamination, since fumonisins can be observed without visible symptoms of Fusarium instances. Both Fusarium and fumonisins can occur in every phase of the maize's life from sowing to storage, from field to mills. Actually Fusarium and fumonisin contamination can be controlled and reduced, applying some reasonably achieved agricultural and manufacturing practices, but not completely removed.

Taking into account that i) Fusarium and fumonisins infection of maize causes both serious health consequences and economic loss due to decreased quality and quantity of batches; ii) legal limits must be abode; fumonisin presence in maize should be periodically monitored in order to know as soon as possible if lots are contaminated. With this provided information maize with significant fumonisins level can be identified, so that infected maize amounts can be extracted before they are commingled with maize characterized by low or negligible fumonisins content. Then, fumonisins contaminated maize can be safely removed from batches intended for food or feed use and assigned for other purpose, as energy production.

Since traditional fumonisins detection procedure, like HPLC, are accurate but expensive and time consuming, a method based on NIR spectroscopy and multivariate linear regression is checked. Results show that it allows screening of maize to comply with European Regulation, a good classification ability and a rather accurate fumonisins prediction, since validation outcomes are promising. These properties together with cheaper, simpler and faster almost real time measurements would suggest NIR methodology as a suitable alternative to traditional analyses of fumonisin detection in maize. Further studies are however needed in order to confirm its good properties, verify its applicability or develop other models based on slightly different assumptions suggested by these preliminaries remarks.

Conclusions

C.1 Theoretical background

The first Chapter opens with an introduction to Machine Learning that defines it before listing the most common techniques developed in this discipline. Machine Learning, that is strictly related with both Statistics and Data Mining, consists of five subfields that collect a wide variety of algorithms. In this thesis focus is made on unsupervised and supervised learning for which more details are given. Indeed, they present the general framework where PLS takes place.

Successively, a general working procedure for dealing with supervised problems is described according to an incremental explanation and an intuitive approach. Thus, model fitting, model selection and decision stage are illustrated in order to understand gradually how to address the purpose of supervised learning. After considering the twofold nature of data, that includes both an underlying regularity and a random noise, the adaptive model can be defined as the sum of a structural component and a random variable. Then, curve fitting is explained as the problem of computing the model with its form and values of its parameters. There are many criteria to solve this task, but everyone should take into account the problem of model selection that controls overfitting and assures a good prediction ability, the two opposite aims that a model should reach at the same time. Suitable strategies deals with this trade off and their application assures that fitted model describes accurately data having also a good prediction ability. As a consequence, after obtaining predictions, supervised learning algorithms can decide either to do or do not an action based on understanding of values that target is likely to take and considering that choice should be optimal in some appropriate sense.

The second Chapter provides a simple but complete overview of PLS. Since the aim is to be formally exhaustive, the methodology is described gradually, including theoretical formulas together with algorithmic features, graphical tools and informal descriptions.

PLS is a wide class of supervised multivariate techniques for modelling relations between sets of observed collinear quantities through latent variables defined by maximizing the covariance in the original space. In this way, PLS projects measured observations and with these new data it fits a model. PLS is a powerful analytical tool with rather soft assumptions and minimum demands in terms of measurements scales, sample size and residual distribution. Firstly, PLS not only manages collinearity but can also remove it from dataset. Secondly, PLS is effective

even if instances are less than number of variables. Thirdly, PLS does not need data to come from normal or known distributions. With these relaxed constrains, PLS results very flexible and it brings on many benefits.
Moreover, PLS assures also computational and implementation simplicity and good performances are always confirmed. In particular, PLS algorithm consists of an iterative process that can handle high amounts of data and whose computational time scales well with dataset dimensions. The asymptotic space requirement and time complexity of PLS algorithm are those necessary to store the dataset matrix.
Furthermore, PLS proves to be a useful tool for problems of regression, classification and dimensionality reduction as well as in presence of nonlinear relations between variables. Regarding regression and classification, PLS was originally developed as a technique that avoids the effect of multicollinearity in the parameters estimation. But in turn, PLS may be more appropriate for predictive purposes. Indeed, Wold himself asserted that PLS is mainly suited to predictive causal analyses in highly complex situations with poorly developed theoretical understanding. With respect to dimensionality reduction, PLS analyzes the importance of individual observed variables potentially leading to deletion of irrelevant ones. PLS defines indeed more compact settings with an aware variables deletion. Even if its classical approach assumes linear relations between variables, PLS can also be extended to model nonlinearity trough recently developed theory of kernel-based learning. Kernel PLS keeps computational and implementation simplicity of linear PLS and at the same time is competitive with respect to other traditional nonlinear methods.
All these interesting properties make PLS a very powerful versatile data analytical tool and favour PLS to be applied successfully in many research areas and industrial contexts that provide tall and fat datasets, where often collinearity occurs and data exhibit nonlinear behaviour. PLS has been so proven to be very effective and with a wide multidisciplinary applicability, that distinguishes it since its origins.
This Chapter ends with a review of the most common packages that include the PLS methods. Indeed, PLS needs sophisticated computations, so that its fruitful application depends on the availability of a suitable computer program. Fortunately, nowadays PLS methodology is usually integrated within all the main suites for multivariate analysis.
PLS is the core argument of the thesis and after its introduction in this Chapter, it will be successively compared with competitive techniques.

The third Chapter presents methods that share goals and tasks of PLS but that achieve them in a different way. Comparisons reveal both additional aspects and the competitive behaviour of PLS, showing that it performs better than alternatives. Connections between individual techniques help to design new algorithms resulting in more powerful tools.
First, Ordinary Least Squares (OLS) is the traditional estimation procedure used for linear regression and it can be seen as the benchmark for the other options. OLS

usually defines the best linear unbiased estimator, but if and only if data comply restrictions, as the absence of collinearity. There is a relation between OLS and PLS, because the last is computed as an OLS regression of observed response on orthogonal latent predictors. This link requires weight, loading and score vectors and allows to transform an OLS estimator computed in the latent space to a PLS estimator that refers to original variable space.

CCA is, as PLS, a projection method to latent variables but it extracts latent vectors with maximal correlation. In particular PLS criterion represents a form of CCA where the principle of maximal correlation is balanced with the requirement of explaining as much variance as possible in both $\mathcal{X}$ and $\mathcal{Y}$-spaces. So, PLS is more complete since it takes into account a larger amount of information brought by data.

RR was born as a method for stabilizing regression estimates in presence of extreme collinearity. RR optimization criterion is equivalent to that of PLS divided by the sum between $\mathbf{X}$ covariance matrix and a metaparameter that should be estimated through a model selection procedure. Furthermore, RR estimator of regression coefficient can be computed from a penalized OLS criterion with the penalty proportional to its squared norm.

Canonical Ridge Analysis provides a unique approach that collects all the previously defined estimation procedures as special cases. Its optimization problem consists of a weighted PLS criterion where the weight is in the denominator and includes a metaparameter. Choosing specific values of this metaparameter, or letting it change into its allowed range, optimization criterion of every presented technique is obtained.

Considering other chemometric techniques, PCA, as PLS, is a projection method, that specifies directions of maximal variance, called PC. However, PCA is defined as an unsupervised technique, since it does not take into account for any information about the $\mathbf{Y}$ variables. Indeed, PLS extends PCA with a regression phase so that PC related only to $\mathbf{X}$ will explain covariance between $\mathbf{X}$ and $\mathbf{Y}$ as far as possible. Similarly, both PCA and PLS are traditional dimensionality reduction techniques widely used in Machine Learning. But PLS provides a more principled dimensionality reduction in comparison to PCA. If PC computed by a PCA are used as predictors in a linear regression model, PCR is defined. However, there is the problem of choosing the optimum subset of PC, since PC have been computed with respect to $\mathbf{X}$ but there is no guarantee that these PC will be pertinent for explaining $\mathbf{Y}$. Finally, PCR, as PCA and PLS, deals with collinearity but it is a less prediction oriented method than PLS.

All these procedures can be cast under a unifying approach called continuum regression that relates PLS to PCR and RR, taking OLS as a reference method. The comparison is done through the mean standard error, that should be as lower as possible for assuring an high estimator quality, and which can be written in the form of its bias-variance decomposition. OLS estimator has usually no bias. PLS, PCR and RR, unlike OLS, produce biased estimates. The effect of this bias is to shrink the regression coefficient directions that are responsible for a high variance.

Thus, PLS, PCR and RR estimators can be expanded in terms of OLS components, and they can be characterized by a shrinkage structure that involves good features of the corresponding estimators.

As regards classification, there is a close connection between PLS and LDA that is also a dimensionality reduction technique widely used in Machine Learning. LDA, as PLS, has a supervised nature and is a projection method, however it is limited by some constrains that PLS does not have.

After this critical comparison, that confirms PLS predominance over concurrent approaches, PLS can be extended to nonlinear problems in the following in order to complete the list of its benefits.

The fourth Chapter introduces SVM, a set of supervised learning techniques that can also be seen as competitors of PLS methods. A general overview about SVM is firstly presented that includes its benefits and drawbacks, its historical development and its applications. Secondly, SVM classical theory is described considering binary classification. Following an approach with a gradually increasing complexity, both linear and nonlinear cases are explained and, with regard to linear classificator, both linearly and nonlinearly separable data are taken into account. Thirdly, nonlinear PLS procedures previously illustrated are completed with SVM approach.

In general terms, considering categorical quantities, SVM can be defined as a non-probabilistic binary linear classifier, so that it belongs to generalized linear classificators family. SVM has some interesting properties. Indeed, it is usually able to handle high dimensional input space and at the same time it does not require dataset characterized by large cardinality. In addition, it can successfully manage presence of sparse instances and irrelevant variables. Then, SVM can assure a good generalization ability with an high prediction accuracy and without the risk of overfitting. SVM has also nice math qualities, since it consists of a simplex convex optimization problem which is guaranteed to converge to a single global solution. At the end, SVM has been also extended to unsupervised learning methods, as PCA. However, there is also some drawbacks. For example interpretability of results can be not so easy and meaningful. As a consequence, a trade off should be done between speed and advantages of computation or directness of results explanation. Then, SVM is sensitive to noise. In addition, users should learn how to tuning SVM, sometimes with either a length series of experiments or the assistance of domain experts or a validation procedure. Moreover a tricky question is how to manage multi-class data, since traditional SVM usually defines binary classificators.

Research in SVM has had many applications in various fields where it is become shortly competitive with traditional methods. Since PLS can be used in the same real situations to reach the same goals, comparison between SVM and PLS can be also done through a critical analysis of such concrete cases.

If SVM are combined with nonlinear PLS, linear data analysis procedures can be ex-

tended to nonlinear problems. The simpler solution assumes that kernel-based SVM is used for modelling nonlinear relation between PLS score vectors. Another methodology, called nonlinear kernel PLS, maps original input data into a new feature space where linear PLS is applied as usual. The kernel trick should be also considered since it allows to easily evaluate product between two vectors in the feature space. So, the kernel form of the NIPALS algorithm is obtained as a modified version of the traditional NIPALS algorithm. Comparison of kernel and traditional PLS methodology is not so simple and it is difficult to choose the favourable one. While the kernel PLS approach is easily implementable, computationally less demanding and capable to model difficult nonlinear relations, a loss of the interpretability of the results with respect to the original data limits its use in some applications. On the other hand, it is common to find situations where the classical approach of keeping latent variables as linear projections of the original data may not be adequate.
This Chapter allows to complete the PLS theoretical background including also nonlinear PLS that assumes a kernel based approach developed on SVM theory. Even if in this thesis nonlinear PLS were not applied to any case study, first of all because the available data were not characterized by nonlinear relations, one interesting perspective for future research will be the application of nonlinear PLS to real data. In this way theoretical properties and qualities of this technique can be verified and compared with concurrent solutions. A concrete linear case study is instead described in the following.

C.2 A case study

The second part of this thesis presents a case study where PLS and full cross validation are applied to define a model that detects fumonisin content in contaminated maize from NIR spectra. Firstly, a preliminary gradual but complete description of the applicability area is given in order to show the main questions that are behind this kind of problems and that support further analyses. Indeed, fungi and mycotoxins are briefly introduced together with their features and the most troublesome factors related to their development and management. Then, more details are given about Fusarium and fumonisins and their critical aspects. Measurement processes are also explained, they include both traditional reference analytical methods (HPLC) and some alternatives, based on spectroscopic techniques (NIR). Secondly, results of a research on real data are shown. Its aim is to evaluate an accurate, fast, cheap, easy and nondestructive method able to detect fumonisin contamination in maize. Dataset is described explaining sampling procedure, applied measurements methods and variable of interest, that consists of fumonisins sum. Then, the best multivariate regression model is illustrated. Both numerical values and graphical outputs are given and discussed. Final comments highlight the most important conclusions and the aspects that should be more accurately taken into account in future.

The fifth Chapter introduces the problem of cereal crops contaminated by fungi that produce mycotoxins in order to focus successively on maize contaminated by Fusarium species that synthesize fumonisins.
Agricultural commodities usually host micro organisms in a symbiotic relationship both directly in the field or after harvest. Mycotoxins are toxic secondary metabolites produced by organisms of the fungus kingdom including mushrooms, moulds and yeasts. Their role in fungi life is not ever completely known. In addition, there is a lack of knowledge about relationship between fungi infection and mycotoxins occurrence. Moreover, fungi and mycotoxins can not be completely removed, even if fungal proliferation and mycotoxins synthesis can be reduced, applying suitable agronomic techniques. As a consequence, authorities fix some guidelines as regards mycotoxins management strategies and recommend some best practices in order to obtain higher hygienic and sanitary standards for grain lots. Furthermore, mycotoxins greatly resist decomposition or being broken down in digestion, and neither temperature treatments destroy them. Thus, mycotoxins enter food chain with contamination of cereal-based food, or can be found still in meat and dairy products. So, mycotoxins ingestion can cause several acute or chronic disease both to cattle breedings and human beings.
Mycotoxins can be found in many types of food components throughout the world, but cereals are often the most suitable environment for some species of fungi. In particular, Fusarium species are among the most common natural contaminant of maize plants worldwide and maize has the highest fumonisins production with respect to other cereals. Occurrence of fumonisin FB1 and FB2 is the most frequent and they have toxicological significance. As a consequence, EU Scientific Committee established values of Tolerable Daily Intake (TDI) for fumonisins with respect to several maize-based feed and foodstuffs.
Procedures for fungi and mycotoxin detection are important because if fumonisins content in maize batches is monitored, only uncontaminated grains enter in food chain allowing to assure safety and quality of maize, to abide laws limits and to have no economical loss. Actually, reference measurement processes for most of the mycotoxins are analytical methods that include a wide array of in laboratory testing and are based on HPLC. These tools are largely used for food and feed industries, since they are accurate and reliable, but they are also expensive, time consuming, destructive and only experts can do such complex phases. As a consequence, some cheaper and faster alternatives for fungal and mycotoxins detection in food and feed components are highly required. Recently, alternatives based on NIR spectroscopic techniques are been suggested. NIR spectroscopy involves a minimum preparation of raw materials, that can be screened quickly and following a very simple procedure where neither expert staff nor expensive equipment are needed, and it preserves maize after the measurement for further analyses. So, NIR spectroscopy is nondestructive, very simple, faster and cheaper than traditional analytical methods. After recording NIR spectra, relationship between these and fungi or mycotoxins contamination should be investigated through statistical analysis. Multivariate regression

techniques allow indeed to define a model that should be able to predict future values for the target variable. For these reasons, NIR spectroscopy can be considered as a measurement method for safety and quality evaluation of agricultural commodities. As a consequence, it may have practical applications toward on line screening and real time automated systems.

The sixth Chapter opens with the description of the available dataset that records all the information of analyzed maize amounts about their NIR spectra and their fumonisin contamination level, measured by HPLC technique. Firstly, sampling was carried out according to the corresponding EC Regulation. Secondly, presence of fumonisins FB1 and FB2 are measured and their sum considered for the following survey. Chosen reference analytical method is HPLC applied according to the AOAC Official Method. Moreover, a preliminary descriptive analysis of fumonisins content is done both with graphic tools and main statistics, this allows to highlight some critical aspects. Thirdly, procedure to record NIR measurements is given, as well as description of used FT-NIR spectrometer. In addition, chosen NIR variables are declared, they record absorbance values, and some examples of NIR spectra are shown.
In order to explain and quantify the relationship between fumonisins contamination level and NIR spectra, PLS with full cross validation is applied to centered data. The model should have a twofold purpose: i) to assure screening of maize amounts with respect to law limit of 4 $\mathrm{mg\,kg^{-1}}$; ii) to provide a tool for classification and rather accurate predictions into the whole scale of fumonisin contamination level.
The dataset reduces to a calibration set with 133 observations and the model has 21 PC suggested by a decreasing residual variance curve. Correlations of calibration and validation are very high. This means that the model describes accurately the relation between fumonisin content and NIR spectra avoiding overfitting; and it would have a good prediction ability for future unknown observations. Moreover, observing calibration plot, legal limit is satisfied and other interesting intervals can be drawn, so that no observation is misclassified. As a consequence, developed model assures a good screening ability; and it can be used also as a classifier. Considering the set of variables that enter into the model, two opposite approaches should be compared. Such critical aspects should however be investigated more deeply in order to decide the best solution.

In conclusion, maize is one of the most common dietary staple for both cattle breedings and human beings in many countries. Fusarium development in maize is a very complex phenomenon that depends on environmental patterns. Factors that cause fumonisins infection are not so clear, as well as fumonisins role in fungi life is not always known. Fusarium and fumonisins occurrence in maize causes serious health consequences to both animal and humans. Moreover, they involve economic loss due to decreased quality and quantity of batches. Legal limits are fixed by international organizations and they must be abode. Actually Fusarium and fumonisin contamination can be controlled and reduced, applying some reasonably achieved agricultural and manufacturing practices, but not completely deleted. As a consequence, fumonisin presence in maize should be periodically monitored in order to know as soon as possible if lots are contaminated. In this way, maize with significant fumonisins level can be identified and extracted before that they are commingled with safe maize characterized by low or negligible fumonisins content. Then, fumonisins contaminated maize can be safely removed from batches intended for food or feed use and assigned for other purpose. Since traditional fumonisins detection procedure, like HPLC, are accurate but expensive and time consuming, a method based on NIR spectroscopy and multivariate linear regression is checked. Results show that it allows screening of maize to comply with European Regulation, a good classification ability and a rather accurate fumonisins prediction, since validation outcomes are promising. These properties together with cheaper, simpler and faster almost real time measurements would suggest NIR methodology as a suitable alternative to traditional analysis of fumonisin detection in maize. Further studies are however needed in order to confirm its good properties, verify its applicability or develop other models based on slightly different assumptions suggested by these preliminaries remarks.

A
Technical developments

A.1 Eigenvalues and eigenvectors decomposition

Let $\mathbf{X} \quad (k \times k)$ denote a square matrix, λ be a scalar and $\mathbf{v}$ represent a nonzero k-dimensional vector. Let consider the following linear equation

$$\mathbf{X}\mathbf{v} = \lambda\mathbf{v} \tag{A.1}$$

It defines the eigenvalue problem and its solutions, i.e. the values λ and $\mathbf{v}$ that satisfy it, are called respectively eigenvalues of $\mathbf{X}$ and eigenvectors of $\mathbf{X}$ corresponding to λ. The prefix "eigen" is adopted from the German word "eigen" for own in the sense of a characteristic description. As a consequence, eigenvectors and eigenvalues are sometimes also known as characteristic vectors and characteristic values. Moreover, the set of eigenvalues is also sometimes named the spectrum of the matrix.

Eigenvalue equation can be interpreted geometrically. Usually, the multiplication of a vector by a square matrix changes both the length and the direction of the vector it acts on. But in the special case of eigenvectors and eigenvalues, it changes only the scale of the vector. Indeed it stretches, shrinks, leaves unchanged or flips (switches in the opposite direction) the vector. In other words, the eigenvectors of a square matrix are the nonzero vectors that, after being multiplied by the matrix, remain parallel to the original vector. For each eigenvector, the corresponding eigenvalue is the factor by which the eigenvector is scaled when multiplied by the matrix. If the eigenvalue $\lambda > 1$ the vector is stretched by this factor. If $\lambda = 1$ the vector is not affected at all by multiplication by the matrix. If $0 < \lambda < 1$ the vector is shrunk or compressed. The case $\lambda = 0$ means that the vector shrinks to a point represented by the origin. If $\lambda < 0$ then the vector flips and points in the opposite direction as well as being scaled by a factor equal to the absolute value of λ.

In order to solve the problem, the eigenvalue equation A.1 can be written as a linear omogeneus system

$$\begin{aligned} \mathbf{X}\mathbf{v} - \lambda\mathbf{v} &= \mathbf{0}_k \\ (\mathbf{X} - \lambda\mathbf{I}_k)\mathbf{v} &= \mathbf{0}_k \end{aligned} \tag{A.2}$$

where $\mathbf{I}_k$ is the $(k \times k)$ identity matrix, that is, for definition, a squared diagonal matrix with all elements equal to 1. $\mathbf{0}_k$ is a k-dimensional null vector, whose elements are all equal to zero. The eigenvalue problem consists on a k-equation system. However, since there are $k+1$ unknown quantities, the k elements of the vector $\mathbf{v}$ plus the scalar λ, it is undetermined. As a consequence, two cases should be distinguished. Firstly, if the inverse $(\mathbf{X} - \lambda\mathbf{I}_k)^{-1}$ exists, then both sides can be left-multiplied by it, to obtain the result $\mathbf{v} = 0$. In this case, the problem has a trivial solution that cannot be accepted, because it goes against the hypothesis of nonnull vector $\mathbf{v}$. Moreover, if λ is such that $(\mathbf{X} - \lambda\mathbf{I}_k)$ is invertible, λ cannot be an eigenvalue. This is the reason why $\mathbf{v}$ is assumed to be not null. However, it can be shown that the converse holds, too. Indeed, if $\mathbf{X} - \lambda\mathbf{I}_k$ is not invertible, λ is an eigenvalue. A matrix is not invertible if and only if its determinant is zero. Thus, adding the constrain that $\mathbf{v}$ should be normalized, i.e. $\mathbf{v}^T\mathbf{v} = 1$, for a given value of λ this system of equations admit a nontrivial solution if and only if

$$|\mathbf{X} - \lambda\mathbf{I}_k| = 0 \tag{A.3}$$

This is called characteristic equation. The left-hand side of this condition provides (using Leibniz' rule for the determinant) a polynomial function in λ, whose coefficients depend on the entries of matrix. Its degree is k, that is the highest power of λ occurring in this polynomial is λ^k.

After computing the k values of λ, every value of λ can be substituted in Equation A.2. For each λ, solving the omogeneus system of k equations in k variables, the corresponding eigenvector $\mathbf{v}$ can be found. They can be collected in the matrix $V = [v_1, \dots, v_k]$.

If the matrix $\mathbf{X}$ is symmetric, eigenvalues and eigenvectors have some intresting properties. Indeed, in this case $\mathbf{X}$ admits k distinct eigenvalues and the corresponding eigenvectors are orthogonal $\mathbf{v}_i^T\mathbf{v}_j = 0$. As a consequence, matrix $\mathbf{V}$ is also orthogonal, so that $\mathbf{V}^{-1} = \mathbf{V}^T$ since it holds that $\mathbf{V}^T\mathbf{V} = I = \mathbf{V}\mathbf{V}^T$. Matrix $\mathbf{V}$ has also full rank.

Furthermore, the k functions of Equation A.3 can be collected in the matrix $\mathbf{XV} = \mathbf{V\Lambda}$ where $\Lambda = diag(\lambda_1, \dots, \lambda_k)$. As a consequence,

$$\mathbf{V}^T\mathbf{X}\mathbf{V} = \mathbf{\Lambda}$$

with other words matrix of eigenvectors $\mathbf{V}$ diagonalizes matrix $\mathbf{X}$ and matrix $\mathbf{X}$ is defined diagonalizable. Moreover,

$$\mathbf{X} = \mathbf{V\Lambda V}^T = \sum_{i=1}^{k} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$$

This equation, called eigendecomposition or sometimes spectral decomposition, is the factorization of a matrix into a canonical form. Let note that only diagonalizable matrices can be factorized in this way. Since $\mathbf{V}$ has full rank, $rank(\mathbf{X}) = rank(\mathbf{\Lambda})$;

that means rank of a matrix is equal to number of not-null eigenvalues. If a matrix has one or more null eigenvalues, then $\mathbf{Xv} = 0$ and matrix $\mathbf{X}$ is singular (for definition its determinant is null) and has reduced rank. Other important properties of eigendecomposition are that determinant of a matrix is equal to the product of eigenvalues

$$|\mathbf{X}| = |\mathbf{V\Lambda V}^T| = |\mathbf{V}^T||\mathbf{\Lambda}||\mathbf{V}|' = |\mathbf{V}^T\mathbf{V}||\mathbf{\Lambda}| = |\mathbf{\Lambda}| = \prod_{i=1}^{k} \lambda_i$$

whereas trace of a matrix is equal to sum of eigenvalues.

$$tr(\mathbf{X}) = \sum_{i=1}^{k} \lambda_i$$

A.2 Singular values decomposition

Let $\mathbf{X}$ be a general $(n \times k)$ matrix. The Singular Values Decomposition (SVD) allows to write every matrix $\mathbf{X}$ in the following way

$$\mathbf{X} = \mathbf{R\Gamma Z}^T \tag{A.4}$$

where matrices $\mathbf{R}$ and $\mathbf{Z}$ are orthogonal of dimension $(n \times n)$ and $(k \times k)$ respectively. $\mathbf{\Gamma}$ is a $(n \times k)$ "diagonal" matrix. Indeed, with $h = \min\{n, k\}$, elements of $\mathbf{\Gamma}$ are equal to $\{\gamma_i\}_1^h$ if $i = j$, with other words if they stay on the "diagonal"; otherwise they are equal to zero. Moreover, elements $\{\gamma_i\}_1^h$ are ordered, $\gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_h \geq 0$. Elements $\{\gamma_i\}_1^h$ are called singular values, whereas matrices $\mathbf{R}$ and $\mathbf{Z}$ collect left singular vectors and right singular vectors respectively.

Generally, the SVD can be reduced in the following way

$$\begin{aligned}
\mathbf{X} &= \begin{pmatrix}\mathbf{R}_1 & \mathbf{R}_2\end{pmatrix}\begin{pmatrix}\Gamma_1 & 0 \\ 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{Z}_1^T \\ \mathbf{Z}_2^T\end{pmatrix} \\
&= \mathbf{R}_1\mathbf{\Gamma}_1\mathbf{Z}_1^T \\
&= \sum_{i=1}^{l} \gamma_i \mathbf{r}_i \mathbf{z}_i^T
\end{aligned}$$

with l equal to the number of singular values that are not null.

From the SDV of a general matrix $\mathbf{X}$ some interesting properties can be found. Computing the quantity $\mathbf{X}^T\mathbf{X}$

$$\begin{aligned}\mathbf{X}^T\mathbf{X} &= (\mathbf{R}\boldsymbol{\Gamma}\mathbf{Z}^T)^T(\mathbf{R}\boldsymbol{\Gamma}\mathbf{Z}^T)\\ &= \mathbf{Z}(\mathbf{R}\boldsymbol{\Gamma})^T\mathbf{R}\boldsymbol{\Gamma}\mathbf{Z}^T\\ &= \mathbf{Z}\boldsymbol{\Gamma}^T\mathbf{R}^T\mathbf{R}\boldsymbol{\Gamma}\mathbf{Z}^T\\ &= \mathbf{Z}\boldsymbol{\Gamma}^T\mathbf{I}_k\boldsymbol{\Gamma}\mathbf{Z}^T\\ &= \mathbf{Z}\boldsymbol{\Gamma}^2\mathbf{Z}^T\end{aligned}$$

its eigendecomposition is obtained, since $\mathbf{R}$ is orthogonal, i.e. $\mathbf{R}^T\mathbf{R} = \mathbf{I}_k$. As a consequence, $\mathbf{Z}$ collects the eigenvectors of $\mathbf{X}^T\mathbf{X}$. Similarly,

$$\mathbf{X}\mathbf{X}^T = (\mathbf{R}\boldsymbol{\Gamma}\mathbf{Z}^T)(\mathbf{R}\boldsymbol{\Gamma}\mathbf{Z}^T)^T = \mathbf{R}\boldsymbol{\Gamma}^2\mathbf{R}^T$$

so that $\mathbf{R}$ collects the eigenvectors of $\mathbf{X}\mathbf{X}^T$, since $\mathbf{Z}$ is also orthogonal. Furthermore, $\boldsymbol{\Gamma}^2$ records eigenvalues λ_i of both $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}\mathbf{X}^T$. In particular the following relationship holds $\gamma_i = \sqrt{\lambda_i}$.
Let note that singular vectors, so matrices $\mathbf{R}$ and $\mathbf{Z}$ are not univocally determined. On the contrary singular values and matrix $\boldsymbol{\Gamma}$ are unique.

A.3 Mercer's Theorem

Let $K(\mathbf{x}_i, \mathbf{x}_j)$ be a continous symmetric kernel that is defined into close interval $\mathbf{a} \leq \mathbf{x}_i, \mathbf{x}_j \leq \mathbf{b}$. Kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ can be expanded into series

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{+\infty} \gamma_l \phi_l(\mathbf{x}_i)\phi_l(\mathbf{x}_j) \tag{A.5}$$

with $\gamma_l > 0$. This expansion is true with an absolute and uniform convergence if and only if

$$\int_{\mathbf{b}}^{\mathbf{a}}\int_{\mathbf{b}}^{\mathbf{a}} K(\mathbf{x}_i, \mathbf{x}_j)\psi(\mathbf{x}_i)\psi(\mathbf{x}_j)d\mathbf{x}_i d\mathbf{x}_j \geq 0 \tag{A.6}$$

for all $\psi(\cdot)$ that satisfy

$$\int_{\mathbf{b}}^{\mathbf{a}} \psi^2(\mathbf{x})d\mathbf{x} < +\infty \tag{A.7}$$

A.4 Gram matrix

Given a $(n \times k)$ matrix $\mathbf{X}$ that collects n k-dimensional vectors $\mathbf{x}_i \in I\!R^k$, the Gram matrix $\mathbf{G}$ is the $(n \times n)$ matrix of all possible inner products between elements of $\mathbf{X}$.

Its element, whose position is i-th row and j-th column, is denoted by $g_{i,j}$ and is equal to the inner product between i-th and j-th element of $\mathbf{X}$, i.e.

$$g_{i,j} = \mathbf{x}_i^T \mathbf{x}_j \tag{A.8}$$

For example if $\mathbf{X}$ is equal to

$$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \tag{A.9}$$

with $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^k$, its Gram matrix is

$$G = \begin{bmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \mathbf{x}_1^T \mathbf{x}_2 \\ \mathbf{x}_2^T \mathbf{x}_1 & \mathbf{x}_2^T \mathbf{x}_2 \end{bmatrix} \tag{A.10}$$

Since property of symmetry holds for inner product, with other words $\mathbf{x}_i^T \mathbf{x}_j = \mathbf{x}_j^T \mathbf{x}_i$, Gram matrix is symmetric.
Let note that Gram matrix as here defined is equal to $\mathbf{X}\mathbf{X}^T$ and is the Gram matrix of $\mathbf{X}$ rows; whereas $\mathbf{X}^T\mathbf{X}$ is the Gram matrix of $\mathbf{X}$ columns.

Bibliography

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] M. Barker and W. Rayens. Partial least squares for discrimination. *Journal of Chemometrics*, 17:166–173, 2003.

[3] M. S. Bartlett. Further aspects of the theory of multiple regression. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 34(1), pages 33–40. Cambridge Univ. Press, 1938.

[4] T. M. Baye, T. C. Pearson, and A. M. Settles. Development of a calibration to predict maize seed composition using single kernel near infrared spectroscopy. *Journal of Cereal Science*, 43(2):236–243, 2006.

[5] R. E. Bellman. *Adaptive control processes: a guided tour*. Princeton University Press, 1961.

[6] N. Berardo, V. Pisacane, P. Battilani, A. Scandolara, A. Pietri, and A. Marocco. Rapid detection of kernel rots and mycotoxins in maize by near-infrared reflectance spectroscopy. *Journal of Agricultural and Food Chemistry*, 53(21):8128–8134, 2005.

[7] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. N. Vapnik, and T. Vetter. Comparison of viewbased object recognition algorithms using realistic 3D models. In *Proceedings ICANN96 - International Conference on Artificial Neural Networks - , Lecture Notes in Computer Science*, volume 1112, pages 251–256. Springer, 1996. Berlin.

[8] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, Pittsburgh, 1992.

[9] R. P. Cogdill, C. R. Hurburgh, and G. R. Rippke. Single-kernel maize analysis by near-infrared hyperspectral imaging. *Transactions of the ASAE*, 47(1):311–320, 2004.

[10] COMMISSION RECOMMENDATION. of 17 August 2006 on the presence of deoxynivalenol, zearalenone, ochratoxin A, T-2 and HT-2 and fumonisins in products intended for animal feeding (Text with EEA relevance) (2006/576/EC), 23.8.2006.

[11] COMMISSION RECOMMENDATION. of 17 August 2006 on the prevention and reduction of *Fusarium* toxins in cereals and cereal products (Text with EEA relevance) (2006/583/EC), 29.8.2006.

[12] COMMISSION REGULATION. (EC) of 19 December 2006 No 1881/2006 setting maximum levels for certain contaminants in foodstuffs (Text with EEA relevance). *Official Journal of the European Union*, L364/5-24, 20.12.2006.

[13] COMMISSION REGULATION. (EC) of 28 September 2007 No 1126/2007 amending Regulation (EC) No 1881/2006 setting maximum levels for certain contaminants in foodstuffs as regards *Fusarium* toxins in maize and maize products. *Official Journal of the European Union*, L255/14-17, 29.9.2007.

[14] COMMISSION REGULATION. (EC) of 23 February 2006 No 401/2006 laying down the methods of sampling and analysis for the official control of the levels of mycotoxins in foodstuffs (Text with EEA relevance). *Official Journal of the European Union*, L70/12-34, 9.3.2006.

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

[16] S. de Jong. SIMPLS: an alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18(3):251–263, 1993.

[17] G. Della Riccia and S. Del Zotto. A multivariate regression model for detection of fumonisins content in maize from Near Infrared spectra. *Food Chemistry*, November 2012. (Submitted).

[18] S. R. Delwiche and W. R. Hruschka. Protein content of bulk wheat from near-infrared reflectance of individual kernels. *Cereal Chemistry*, 77(1):86–88, 2000.

[19] F. E. Dowell, T. C. Pearson, E. B. Maghirang, F. Xie, and D. T. Wicklow. Reflectance and transmittance spectroscopy applied to detecting fumonisin in single corn kernels infected with fusarium verticillioides. *Cereal Chemistry*, 79(2):222–226, 2002.

[20] F. E. Dowell, M. S. Ram, and L. M. Seitz. Predicting scab, vomitoxin, and ergosterol in single wheat kernels using near-infrared spectroscopy. *Cereal Chemistry*, 76(4):573–576, 1999.

[21] F. E. Dowell, J. E. Throne, and J. E. Baker. Automated nondestructive detection of internal insect infestation of wheat kernels by using near-infrared reflectance spectroscopy. *Journal of Economic Entomology*, 91(4):899–904, 1998.

[22] V. Fernández-Ibañez, A. Soldado, A. Martínez-Fernández, and B. de la Roza-Delgado. Application of near infrared spectroscopy for rapid detection of aflatoxin b1 in maize and barley as analytical quality assessment. *Food Chemistry*, 113(2):629–634, 2009.

[23] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7:179–188, 1936.

[24] I. E. Frank and J. H. Friedman. A statistical view of some Chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

[25] B. Gaspardo and S. Del Zotto. Progetto sperimentale MICOSAFE: metodologie di analisi e risultati dell'indagine regionale. In *Sandri M. e Gaspardo B. Controllo delle micotossine negli allevamenti: salute animale e sicurezza alimentare. Volume redatto nell'ambito del progetto di ricerca MICOSAFE. Regione Autonoma Friuli Venezia Giulia*, pages 50–59, 2011.

[26] B. Gaspardo, S. Del Zotto, E. Torelli, S. R. Cividino, G. Firrao, G. Della Riccia, and B. Stefanon. A rapid method for detection of fumonisins B_1 and B_2 in corn meal using Fourier transform near infrared (FT-NIR) spectroscopy implemented with integrating sphere. *Food Chemistry*, 135:1608–1612, 2012.

[27] W. C. A. Gelderblom, K. Jaskiewicz, W. F. O. Marasas, P. G. Thiel, R. M. Horak, R. Vleggaar, and N. P. J. Kriek. Fumonisins-novel mycotoxins with cancer-promoting activity produced by *Fusarium moniliforme*. *Applied and Environmental Microbiology*, 54(7):1806–1811, 1988.

[28] W. C. A. Gelderblom, N. P. J. Kriek, W. F. O. Marasas, and P. G. Thiel. Toxicity and carcinogenicity of the *Fusarium moniliforme* metabolite, fumonisn B_1, in rats. *Carcinogenesis*, 12(7):1247–1251, 1991.

[29] I. Guyon, B. Boser, and V. N. Vapnik. Automatic capacity tuning of very large VC-dimension classifiers. In S. Hanson et al., editor, *Advances in Neural Information Processing Systems 5 (NIPS 92)*, pages 147–155. Morgan Kaufmann, San Mateo CA, 1993.

[30] L. R. Harrison, B. M. Colvin, J. T. Greene, L. E. Newman, and J. R. Cole. Pulmonary edema and hydrothorax in swine produced by fumonisin B1, a toxic metabolite of *Fusarium moniliforme*. *Journal of Veterinary Diagnostic Investigation*, 2(3):217–221, 1990.

[31] I. S. Helland. On the structure of partial least squares regression. *Communications in Statistics - Simulation and Computation*, 17(2):581–607, 1988.

[32] J. M. Hernández-Hierro, R. J. García-Villanova, and I. González-Martín. Potential of near infrared spectroscopy for the analysis of mycotoxins applied to

naturally contaminated red paprika found in the spanish market. *Analytica Chimica Acta*, 622(1):189–194, 2008.

[33] A. E. Hoerl and R. W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[34] H. Hotelling. Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[35] International Agency for Research on Cancer (IARC). *Some Traditional Herbal Medicines, Some Mycotoxins, Naphthalene and Styrene*, volume 82 of *IARC Monographs on the Evaluation of Carcinogenic Risk to Humans*. World Health Organization, Lyon, France, 2002.

[36] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML)*, pages 137–142, Berlin, 1998. Springer.

[37] K. J. Kramer, T. D. Morgan, J. E. Throne, F. E. Dowell, M. Bailey, and J. A. Howard. Transgenic avidin maize is resistant to storage insect pests. *Nature Biotechnology*, 18(6):670–674, 2000.

[38] R. Krska, E. Welzig, and H. Boudra. Analysis of *Fusarium* toxins in feed. *Animal Feed Science and Technology*, 137(3-4):241–264, 2007.

[39] A. Maiorano, M. Blandino, A. Reyneri, and F. Vanara. Effects of maize residues on the fusarium spp. infection and deoxynivalenol (DON) contamination of wheat grain. *Crop Protection*, 27(2):182–188, 2008.

[40] W. F. O. Marasas, T. S. Kellerman, W. C. A. Gelderblom, J. A. W. Coetzer, P. G. Thiel, and J. J. van der Lugt. Leukoencephalomalacia in a horse induced by fumonisin B1 isolated from *Fusarium moniliforme*. *The Onderstepoort Journal of Veterinary Research*, 55(4):197–203, 1988.

[41] A. H. Maslow. *Motivation and personality.* New York: Harper and Row, 1954.

[42] W. F. Massy. Principal components regression in exploratory statistical research. *Journal of the American Statistical Association*, 60:234–256, 1965.

[43] J. McCarthy. The Dartmouth Summer Research Project on Artificial Intelligence (conference). Dartmouth College, New Hampshire, 1956.

[44] B.-H. Mevik. The pls package. *R News, The Newsletter of the R Project*, 6(3), August 2006.

[45] B.-H. Mevik and R. Wehrens. The pls package: Principal component and partial least squares regression in r. *Journal of Statistical Software*, 18(2):124, 2007.

[46] G. P. Munkvold and A. E. Desjardins. Fumonisins in maize: Can we reduce their occurrence? *Plant Disease*, 81(6):556–565, 1997.

[47] P.E. Nelson, T. A. Toussoun, and W. F. O. Marasas. *Fusarium species: an illustrated manual for identification.* Pennsylvania State University Press, University Park, USA, 1983.

[48] B. A. Orman and R. A. Schumann. Comparison of near-infrared spectroscopy calibration methods for the prediction of protein, oil, and starch in maize grain. *Journal of Agricultural and Food Chemistry*, 39(5):883–886, 1991.

[49] E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 130–136. June 17-19, San Juan , Puerto Rico, 1997.

[50] T. C. Pearson. Use of near infrared transmittance to automatically detect almonds with concealed damage. *Lebensmittel Wissenschaft und Technologie*, 32(2):73–78, 1999.

[51] T. C. Pearson, D. T. Wicklow, E. B. Maghirang, F. Xie, and F. E. Dowell. Detecting aflatoxin in single corn kernels by transmittance and reflectance spectroscopy. *Transactions of the American Society of Agricultural Engineers*, 44(5):1247–1254, 2001.

[52] A. Phatak and F. de Hoog. Exploiting the connection between PLS, Lanczos methods and conjugate gradients: alternative proofs of some properties of PLS. *Journal of Chemometrics*, 16(7):361–367, 2002.

[53] J. C. Platt. Fast training of support vector machines using sequential minimal optimization., 1999.

[54] J. P. Rheeder, W. F. O. Marasas, and H. F. Vismer. Production of fumonisin analogs by *Fusarium* species. *Applied and Environmental Microbiology*, **68**(5):2101–2105, 2002.

[55] R. Rosipal. Nonlinear Partial Least Squares: an overview. In IGI Global ACCM, editor, *Chemoinformatics and Advanced Machine Learning Perspectives: Complex Computational Methods and Collaborative Techniques*, pages 169–189. Lodhi H., Yamanishi Y. (eds.), 2011.

[56] R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection*, volume 3940 of *Lecture Notes in Computer Science*, pages 34–51. Springer, 2006.

[57] M. Schmidt. Speaker identification via support vector classifiers. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing, ICASSP-96*, volume 1, pages 105–108. IEEE, 1996.

[58] B. Schölkopf, C. J. C. Burges, and V. N. Vapnik. Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy (eds), editors, *Proceedings, First International Conference and Knowledge Discovery and Data Mining AAAI Press, Menlo Park, CA*, pages 252–257, 1995.

[59] B. Schölkopf, A. Smola, K. R. Müller, C. Burges, and V. N. Vapnik. Learning and feature extraction with support vector methods. In T. Downs, M. Frean, and M. Gallagher, editors, *Proc. of the Ninth Australian Conf. on Neural Networks.* Brisbane, Australia, University of Queensland, 1998.

[60] B. Schölkopf, A. Smola, K. R. Müller, C. J. C. Burges, and V. N. Vapnik. Support vector methods in learning and feature extraction. *Australian Journal of Intelligent Information Processing Systems*, 1:3–9, 1998.

[61] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human detection using partial least squares analysis. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 24–31. IEEE, 2009.

[62] B. V. Srinivasan, W. R. Schwartz, R. Duraiswami, and L. S. Davis. Partial least squares on graphical processor for efficient pattern recognition. Technical Report. UM Computer Science Department, 18/10/2010.

[63] B. V. Srinivasan, D. N. Zotkin, and R. Duraiswami. A partial least squares framework for speaker recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5276–5279. IEEE, 2011.

[64] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B (Methodological)*, 36(1):111–147, 1974.

[65] M. Stone and R. J. Brooks. Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal component regression. *Journal of the Royal Statistical Society, Series B (Methodological)*, 52(2):237–269, 1990.

[66] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[67] V. N. Vapnik. *Estimation of dependences based on empirical data. Second Edition.* Springer-Verlag New York Inc, 2006.

[68] V. N. Vapnik and A. Y. Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25:103–109, 1964. USSR.

[69] V. N. Vapnik and A. Y. Chervonenkis. On a class of pattern recognition learning algorithms. *Automation and Remote Control*, 25(1):838–845, 1964. USSR.

[70] V. N. Vapnik and A. Y. Chervonenkis. *Teoriya raspoznavaniya obrazov: Statisticheskie problemy obucheniya. (Russian) [Theory of pattern recognition: Statistical problems of learning].* Moscow: Nauka, 1974.

[71] V. N. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780, 1963.

[72] A. Visconti, M. Solfrizzo, and A. De Girolamo. Determination of fumonisins B_1 and B_2 in corn and corn flakes by liquid chromatography with immunoaffinity column cleanup: Collaborative study. *Journal of AOAC International*, 84(6):1828–1837, 2001.

[73] P. Williams and K. Norris. *Near-Infrared Technology in the Agricultural and Food Industries Second ed.* American Association of Cereal Chemists Inc., 2001.

[74] H. Wold. Estimation of principal components and related models by iterative least squares. In P. R. Krishnaiah, editor, *Multivariate Analysis*, Academic Press, New York, pages 391–420. 1966.

[75] H. Wold. Nonlinear iterative partial least squares (NIPALS) modeling: some current developments. In P. R. Krishnaiah, editor, *Multivariate Analysis II, Proceedings of an international symposium on multivariate analysis held at Wright State University, Dayton, Ohio, June 19-24, 1972*, Academic Press, New York, pages 383–407, 1973.

[76] H. Wold. Path models with latent variables: The non-linear iterative partial least squares (NIPALS) approach. In *H. M. Blalock et al. eds.* Quantitative Sociology: Intentional Perspective on Mathematical and Statistical Modeling, pages 307–357. Academic Press, 1975.

[77] S. Wold. Nonlinear partial least squares modelling II. Spline inner relation. *Chemometrics and Intelligent Laboratory Systems*, 14:71–84, 1992.

[78] S. Wold, N. Kettaneh-Wold, and B. Skagerberg. Nonlinear PLS modeling. *Chemometrics and Intelligent Laboratory Systems*, 7(1)(1):5365, 1989.

[79] S. Wold, A. Ruhe, H. Wold, and W. J. Dunn III. The collinearity problem in linear regression. the partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5:735–743, 1984.

[80] S. Wold, M. Sjöström, and L. Eriksson. PLS-regression: a basic tool of Chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58:109–130, 2001.