UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

LAUREA MAGISTRALE IN BIOINGEGNERIA

DEVELOPMENT AND ASSESSMENT OF LINEAR REGRESSION TECHNIQUES FOR MODELING MULTISENSOR DATA FOR NON-INVASIVE CONTINUOUS GLUCOSE MONITORING

MICHELA RIZ

Relatore:
Prof. Giovanni Sparacino
Correlatore:
Ing. Mattia Zanon

ANNO ACCADEMICO 2010/2011

to my family...

“ La mente è come un paracadute.
Funziona solo se si apre. ”

Albert Einstein

Contents

Summary

Diabetes is a worldwide problem and the number of people with diabetes is constantly increasing due to several reasons including population growth, age, and increasing prevalence of obesity and physical inactivity. In particular, the long-term complications make diabetes a social and economical problem, since they have great impact on subject daily life and its management is financially expensive. As a consequence, considerable efforts have been made to control this disease also by using engineering technologies.

During the last decade, it has been proven that diabetes therapy can be improved by monitoring blood glucose levels by means of the so-called Continuous Glucose Monitoring (CGM) sensors. Different types of sensors, with different degrees of invasiveness, have been already developed in the literature and, at present time, new technologies are also under investigation. Among them, new completely Non-Invasive CGM sensors (NICGM) are very appealing for obvious practical reasons (Chapter 1). In particular, Solianis Monitoring AG (Zurich, Switzerland) has recently proposed a NICGM sensor based on the multi-sensor concept, i.e. a system that includes several sensors (for impedance, optics, temperature, acceleration, ..) on one single substrate, which can be attached to the human body, in order to allow a broad characterisation of the skin and the underlying tissues (Caduff et all, Biosensors and Bioelectronics, pp. 2778-2784, 2009). Such Multisensor signals allow the indirect measurement of glucose level in the blood through a mathematical model (Chapters 2 and 3).

The present work is performed under the aegis of a research agreement between the Department of Information Engineering of the University of Padova and Solianis Monitoring AG. The scope of the project is the development and the assessment of a model for estimating glucose level from the Solianis Multisensor data. Specifically, in the present thesis three different methods for building a linear regression model to describe glucose data from Multisensor signals will

be investigated, assessed and compared: Ordinary Least Squares (OLS), Partial Least Squares (PLS) and Least Absolute Shrinkage and Selection Operator (LASSO). In particular, we will first review the methodological and algorithmical issues connected with the three methods and offer a survey of their implementation through the exploitation of tutorial examples (Chapters 4, 5, and 6). Then, we will apply them to a data base consisting of 32 sets of Solianis data (Chapter 7). Results show that LASSO has the best performance in predicting glucose level from unseen Multisensor data, while OLS suffers from overfitting and PLS is too sensitive to the noise in data (Chapter 8). Some new developments are also proposed to further improve the accuracy of glucose estimates through the exploitation of a few finger-prick glucose samples per day (Chapter 9).

Sommario

Il diabete è un problema diffuso a livello mondiale e il numero di persone affette da diabete è in costante aumento a causa di diversi fattori come la crescita della popolazione, l'invecchiamento, l'obesità e l'inattività fisica. In particolare, le complicazioni altamente invalidanti fanno del diabete un problema socio-economico, poiché esse rappresentano un aggravio della qualità di vita del paziente ed impongono, per la loro diagnosi e cura, un notevole impegno finanziario. Per questo motivo, notevoli risorse sono state investite nelle tecnologie ingegneristiche volte al controllo di questa patologia.

Nel corso degli ultimi decenni, è stato dimostrato che la terapia dei pazienti diabetici può essere migliorata grazie a sistemi di monitoraggio continuo del glucosio (Continuous Glucose Monitoring, CGM). In letteratura si trovano differenti tipologie di sensori con diversi gradi di invasività e tuttora nuove tecnologie sono in via di sviluppo. Tra queste, i nuovi sensori completamente non invasivi sono molto interessanti per ovvie ragioni pratiche (Capitolo 1). In particolare, Solianis Monitoring AG (Zurigo, Svizzera) ha recentemente proposto un sensore non invasivo basato sul concetto di multi-sensore. Tale concetto corrisponde ad un sistema composto da differenti sensori (per misure di impedenza, ottiche, di temperatura, di accelerazione, ...) su uno stesso substrato, che può esser posizionato sul paziente permettendo una più ampia caratterizzazione della pelle e dei tessuti sottostanti (Caduff et all, Biosensors and Bioelectronics, pp. 2778-2784, 2009). I segnali del Multisensore Solianis consentono, attraverso l'utilizzo di un modello matematico, una misura indiretta della glicemia (Capitoli 2 e 3).

Questo lavoro di tesi si inserisce all'interno di un progetto di collaborazione tra il Dipartimento di Ingegneria dell'Informazione dell'Università di Padova e Solianis Monitoring AG. Lo scopo del progetto di ricerca consiste nello sviluppo e nella valutazione di un modello per la stima della glicemia a partire da dati del Multisensore Solianis. Nello specifico, in questo lavoro di tesi tre differenti metodi

per la stima di un modello di regressione lineare saranno valutati e comparati: Ordinary Least Squares (OLS), Partial Least Squares (PLS) and Least Absolute Shrinkage and Selection Operator (LASSO). In particolare, prima verranno descritti i tre metodi dal punto di vista metodologico e algoritmico, presentando le loro caratteristiche attraverso l'utilizzo di esempi dimostrativi (Capitoli 4, 5 e 6). Successivamente, i tre metodi saranno applicati ad un database comprendente 32 serie di dati Solianis (Capitolo 7). I risultati ottenuti dimostrano che LASSO presenta la migliore performance nel predire la glicemia a partire da dati indipendenti del Multisensore, mentre OLS risulta affetto da overfitting e PLS è troppo sensibile al rumore contenuto nei dati (Capitolo 8). Infine saranno proposti alcuni metodi, basati sull'utilizzo di pochi campioni di glicemia da sangue capillare al giorno, per un ulteriore miglioramento delle stime(Capitolo 9).

Chapter 1

Diabetes and Continuous Glucose Monitoring Sensors for its Therapy

This introductory Chapter has the aim to outline the context within the present thesis is embedded. Therefore, a brief description about metabolic processes connected with regulation of glucose concentration will be first presented in order to define diabetes. Then, starting from the characterisation of diabetes therapies, different kind of glucose monitoring systems will be illustrated. In particular, we will describe glucose sensors with different degrees of invasiveness. Among the so-called non invasive sensors, the Solianis Multisensor[1] will be later described in detail in Chapter 2, in order to better understand the characteristics and the origin of the data that will be used in this thesis.

1.1 General Overview of the Diabetes Disease

The human body is a complex system that is controlled in an extremely accurate way; its regulatory system produces many substances, which have the aim to keep parameters, like body temperature, blood pressure and substrate concentration, in a physiological range. A defect in this organization causes different kinds of disease. Glucose is a substrate that is essential for the correct working of many organs and tissues, since it is their principal energy provider.

1.1.1 The Glucose Regulatory System

Thanks to a complex regulatory mechanism, glucose concentration in blood, the glycaemia, is tightly kept in a limited rage, i.e. 70-120 mg/dl. Different hormones are involved in this regulation. The most important is insulin, which is produced by the beta-cells of pancreas. Insulin is physiologically secreted at every meal, in order to bring glycaemia within the euglycaemic range. It is also the principal control signal for conversion of glucose to glycogen for internal storage in liver[2].

Glucose is used by many organs, tissues and cells in different ways. Some, like brain or red blood cells, consume glucose continuously and independently of insulin and the interruption of this supplying may cause severe damages. For muscles, fatty tissue and liver the absorption of glucose is proportional to insulin concentration. In fact, in order to transport glucose from bloodstream into cells, where it is used for growth and energy, insulin must be present.

Glucose in blood derives both from intestinal absorption of carbohydrates and from internal production. In particular, the latter consists in the conversion to glucose of glycogen stored in the liver or in gluconeogenesis (the "re-construction" of glucose using substrate derived from glucose degradation).

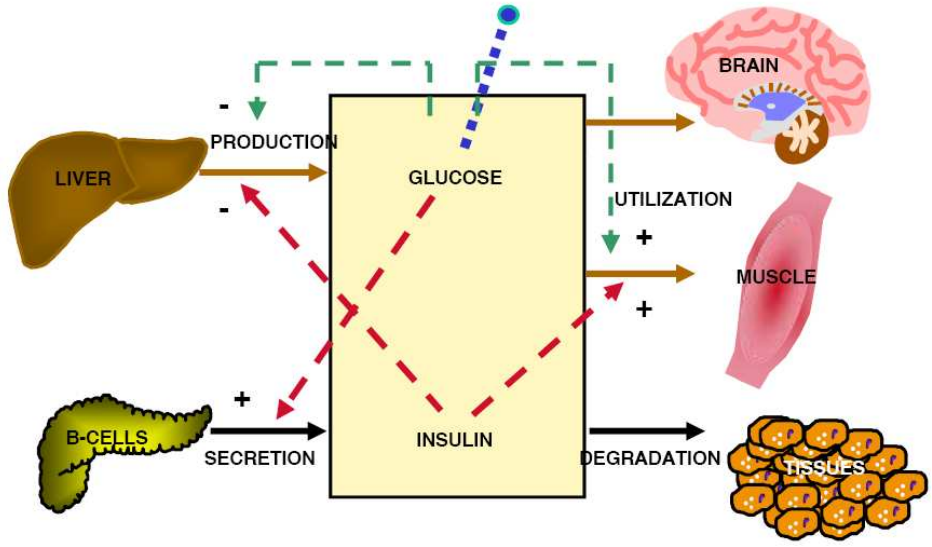

Figure 1.1: Scheme of the glucose regulatory system

The concentration of glucose in blood is subject to fluctuations due to its utilization, by organs and tissues, and its production. Nevertheless, as said before, it is tightly kept in a limited rage, thanks to a complex regulatory system, in which glucose itself is involved (see Figure 1.1).

An increase in blood glucose concentration causes an increase in insulin secretion. Glucose and insulin concentration have the same effect on the glucose production and utilization: an increase in insulin (or glucose) concentration causes a decrease of glucose production and an increase of glucose utilization by muscle, while there is no influence on glucose utilization by brain.

1.1.2 Types of Diabetes

However, in people with diabetes, either the pancreas produces little or no insulin (type 1 diabetes), or the cells do not respond appropriately to the insulin that is produced (type 2 diabetes).

"*Type 1 diabetes*" or Insulin Dependent Diabetes Mellitus (IDDM) is characterized by loss of the insulin-producing beta cells or the islets of Langherhans in the pancreas leading to insulin deficiency. In most cases type 1 diabetes has an autoimmune origin and affects children or young adults, and in fact it is also called "juvenile diabetes".

"*Type 2 diabetes*" or Non-Insulin Dependent Diabetes Mellitus (NIDDM) is characterized by insulin resistance which may be combined with relatively reduced insulin secretion. Insulin resistance corresponds to a loss of efficacy of insulin action, in particular, its effect on the glucose transport from the bloodstream into the cells is reduced. It is frequently associated with obesity and a sedentary lifestyle. Type 2 is the most common diabetes type (90 % of cases) and mostly affects adult people.

1.1.3 Diabetes-Related Complications

Fail of the glucose counter regulatory system in diabetic people causes Blood Glucose Levels (BGL) to exceed the correct range. This situation might lead to short (hypoglycaemia) and long (hyperglycaemia) term complications.

Hyperglycaemia has not an immediate damaging consequence on organism, but, if this state is frequent and persist for long time, it can lead to several

invalidating complications. These long term complications include *micro-vascular complications* (involving small blood vessels) and *macro-vascular complications* (involving large blood vessels)[3]. The former, like neuropathy, nephropathy and retinopathy can lead to nerves damage, renal failure and blindness respectively, the latter to coronary heart disease, strokes and peripheral vascular disease.

In order to prevent the onset of these complications, diabetes therapies attempt to keep BGL as close to euglycemia as possible undue patient danger. This can usually be done with close dietary management, physical activity and use of appropriate medications, like insulin injections before meals. The association of the faulty glucose regulatory system and the scrupulous therapy that may not always be applied tightly could cause, principally during sleep hours or physical activity, an even more dangerous unfavorable effect, i.e. hypoglycemia (i.e. too low blood glucose level).

The main problem caused by hypoglycemia affects the brain, given its continuous glucose demand. Therefore, when glucose levels fall, brain functions diminish and people may lose cognitive abilities and even enter a coma condition. Hypoglycemia, at the opposite of hyperglycemia, has short-term effects[4] and could be classified according to how much low the glucose level is:

- *mild hypoglycemia* (blood glucose levels between 55 and 70 mg/dl) is characterized by palpitations, extreme hunger, trembling, cold or excessive sweating and visual paleness, due to blood redirection to the vital organs and minimization of the peripheral blood circulation. In this case a small amount of carbohydrates eaten or drunk could restore normal levels;

- *moderate hypoglycemia* (between 55 and 40 mg/dl), whose symptoms include mood changes, irritability, confusion, blurred vision, weakness and drowsiness since it affects the central nervous system;

- *severe hypoglycemia* less than 40 mg/dl) is characterized by convulsions, loss of consciousness, coma, and hypothermia. If this condition is prolonged in time could cause irreversible brain damages and heart problems, or even death. In this case, intravenous dextrose or an injection of glucagon is required.

1.1.4 Diabetes Therapies and Monitoring

Diabetes is a worldwide problem and the number of people with diabetes is increasing due to population growth, aging, and increasing prevalence of obesity and physical inactivity. Besides its large diffusion, the long-term complications make diabetes a social and economical problem, since they have great impact on subject daily life and its diagnosis and management are financially expensive. As a consequence, considerable efforts have been made to control this disease also by using engineering technologies[5].

For type 1 diabetes, conventional therapies consist in insulin injection, that compensates the lack of insulin secretion and have the goal to restore euglycaemic levels. A suitable dosage is determined using information on food intakes and current BGL. In the early stage of type 2 diabetes, a diet modification and physical exercise, associated with medications improving insulin sensitivity, may be sufficient to control glycaemic levels. If diabetes proceeds, exogenous insulin injections may be needed.

In both cases, monitoring of BGL is important. In this regard, the traditional system is Self-Monitoring Blood Glucose (SMBG), i.e. patients have to take a finger-prick blood sample on specific strips and measure BGL with the dedicated device 3-4 times a day.

Self Monitoring Blood Glucose

The most common test for measuring BGL involves pricking a finger with a lancet device to obtain a small blood sample, applying a drop of blood onto a reagent test-strip, and determining the glucose concentration by inserting the strip into a measurement device for an automated reading. Different manufacturers use different technology, but most systems measure an electrical characteristic and use this to determine the glucose level in the blood[6].

There is also a painless alternative to blood sample, using other fluids in substitution like saliva, urine, sweat or tears. However, in these cases, delay in the appearance of glucose in these fluids must be taken into account.

These systems make also a direct measure, i.e. they measure a specific property of glucose. This means that if the same property is investigated for another kind of substance, a significantly different output is produced than the one obtained from glucose. Signals generated measuring a specific property of glucose comprise

its spectral, chemical and competitive binding profiles.

Direct measurements tend to be more stable than indirect ones because the signal being measured is usually unique and interferences more predictable. In fact, indirect measurements are affected by the presence of other chemicals and substances within the body that may produce the same signal, since they measure glucose effect on some secondary process[7].

Efforts have been made in order to reduce the level of SMBG invasiveness by decreasing the blood sample volume and measuring areas of body less sensitive to pain than fingertips.

Self Monitoring Blood Glucose

The main drawback of SMBG is the lack of glucose measures during sleeping or daily activity, or the excessive insulin dosage that can lead to hypoglycaemia episodes. The only way to prevent these episodes is to monitor glucose fluctuations during all day, and this is the reason why in the last decade many devices for Continuous Glucose Monitoring (CGM) have been developed. CGM systems obviously require to limit device invasiveness and to improve device portability.

The main advantage of CGM is the possibility to monitor BGL in a nearly continuous way, providing data about the direction, magnitude, duration, frequency and potential causes of fluctuations in blood glucose levels. In the last years, several algorithms have been developed for glucose prediction and alarm generation, with the aim of rendering glucose sensors "smart"[8].CGM is also required to develop the so-called artificial pancreas, which implements a closed-loop control that has the aim to normalize BGL. This device infuses the necessary amount of insulin subcutaneously using a micro-infusor driven by a control algorithm, which, in turn, exploits the measurements provided by a CGM sensor[9].

CGM provides much more information than SMBG (i.e. 3/4 times per day). Nevertheless, at the time of writing, the finger-prick systems, that measure capillary blood glucose, are still predominantly used to adjust diabetes treatment given their accuracy.

1.2 A Classification of Continuous Glucose Monitoring Sensors

In the previous Sections, the diabetes disease and its management were described with particular attention to the reasons calling for a tight and accurate monitoring of blood glucose levels by CGM sensors.

There are many ways to classify CGM sensors. For instance CGM sensors could be divided according to the kind of measure (direct or indirect), to the level of invasiveness or to the physical principle the sensor is based on. A classification of CGM sensors divided according to their level of invasiveness and their physical principle is shown in Figure 1.2.

1.2.1 Invasive CGM Sensors

As shown in Figure 1.2, a direct measurement of BGL could be obtained, in a rather invasive manner, by using sensors implanted into the body[10]. Most of these sensors are based on the glucose-oxidase principle. Other sensors are based on competitive binding of glucose with other molecules or glucose spectral properties[7].

Intravenous Implantable Sensors

Glucose oxidase-based sensors technology depends on the reaction of glucose with oxygen in presence of glucose oxidase to create gluconic acid. The limitation of using this method is that the reaction requires one oxygen molecule for each glucose molecule. Since glucose is more present in the body than oxygen, the limiting reagent results to be the oxygen. For this reason, the sensor would measure oxygen levels instead of glucose levels. To avoid this problem, sensors must give oxygen an advantage over glucose, using alternative electron donors, called mediators.

The competitive binding-based sensors measures fluorescence of a binding molecule: the more glucose is bound to this molecule, the less intense is the fluorescent signal so that if glucose levels increase the measure decreases.

This technique has still problems related to biocompatibility and to the risk inherent to surgical placement of these devices in blood vessels, hence it is not wide applied.

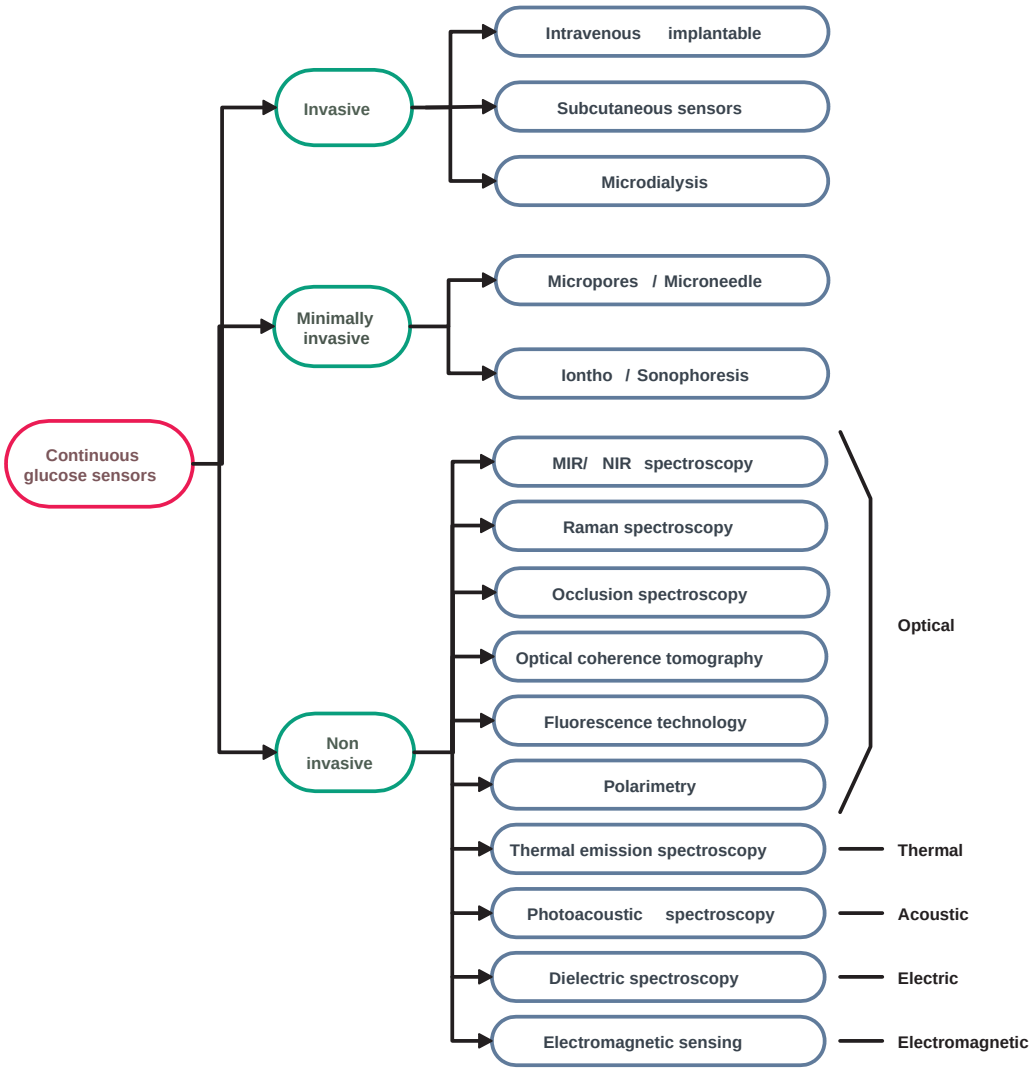

Figure 1.2: Classification of sensors for continuous glucose monitoring

A new intravascular continuous glucose monitoring system is under development, using a glucose-sensitive hydrogel. When this hydrogel is bound with glucose, it changes in volume. The result is a measurable change in the hydrogel impedance that is correlated to glucose concentration. Preliminary studies have been made on a prototype of the sensor, integrated with stents as antennas for wireless data transfer from within the body[11].

Subcutaneous Sensors

Instead of implanting the sensor into the body, a subcutaneous needle may be used to sense glucose. Usually these systems are based on enzyme electro-

des, which use enzymes like glucose-oxidase. The measure is not exactly continuous, because these sensors typically provide data every 3-5 minutes. However subcutaneous needles provide much more information if they are compared to a finger-prick system. These sensors require frequent calibration to compensate drifts attributable to protein and cell coating of the sensor, variable tissue oxygen tension and wound response to the sensor, which alters local blood flow. To perform the periodic recalibration of the sensor a traditional measurement using SMBG systems is needed.

Example of available subcutaneous sensor are: *FreeStyle Navigator*, *MiniMed Guardian* and *Dexcom*.

Figure 1.3: *FreeStyle Navigator* CGM System[12].

The *FreeStyle Navigator* CGM System (Abbott) consists of four components: a miniature electrochemical sensor placed in the subcutaneous adipose tissue, a disposable sensor delivery unit, a radiofrequency transmitter connected to the sensor, and a hand-held receiver to display continuous glucose values. The sensor can be used for 5 days, the glucose data on the receiver are updated once a minute and include a trend arrow to indicate the direction and rate of change averaged over the preceding 15 min. The user interface of the receiver allows the threshold alarms to be set at different glucose levels. The receiver contains a built-in FreeStyle blood glucose meter for calibration of the sensor as well as for confirmatory blood glucose measurements. The sensor requires four calibrations over the 5-day wearing period at 10, 12, 24, and 72 h after sensor insertion. It was approved by FDA in 2008[13][14].

Figure 1.4: *DexCom* SEVEN. *Left*: the receiver. *Right*: subcutaneous sensor and transmitter[15].

DexCom SEVEN (DexCom) consists of 3 part: a small sensor placed in the subcutaneous adipose tissue, a wireless transmitter and a receiver. It performs a new measure every 5 minutes for 7 days. The receiver displays the sensor glucose value along with a graph showing glucose trend of the last 1, 3 or 9 hours. The receiver contains memory up to 30 days of continuous glucose information and has programmable high and low glucose alerts and a non-changeable low glucose alarm set at 55 mg/dL. It must be calibrated every 12 hours. It was approved by FDA in 2007[16].

Figure 1.5: The *Guardian REAL-Time*[17].

The *Guardian REAL-Time* (MiniMed) device is similar to *FreeStyle Navigator*. It performs a new measure every 5 minutes for 3 days. The receiver contains memory up to 21 days of continuous glucose information and has alerts if a glucose level falls below or rises above preset values. It must be calibrated every 12

hours either manually or automatically via telemetry. It was approved by FDA in 2005. This sensor, integrated with an insulin delivery device composes the *MiniMed Paradigm REAL-Time* system, that was launched in 2006[18].

Microdialysis

Another type of subcutaneous sensor is based on a microdialysis systems, which use a fine, hollow microdialysis fibre placed subcutaneously. This probe is perfused with isotonic fluid from an external pool, while glucose, present in the interstitial fluid, freely diffuses into the fibre, where it is pumped out of body to a glucose-oxidase based sensor[19]. The main problem related to this kind of sensor consists in modifications of chemical and physical properties of the membrane, caused by modifications in tissues characteristics such as pressure, volume, temperature and hydration. These modifications affect flow rate and composition of perfusate, which may influence glucose concentration.

GlucoDay-S (Menarini) is a microdialysis-based glucose monitoring system. It is based on enzymatic-amperometric measurement analyzing the fluid coming from the subcutis of the abdominal region. The system comprises s Walkman-size apparatus, and a sensor fibre as well as two plastic bags (one for the buffer solution, one for the waste products) as disposables. The apparatus contains also a measurement cell and a peristaltic pump. The buffer solution is pumped from a bag into the subcutaneous tissue through the microfibre and rinses the interstitial fluid, from which the measures are obtained every 3 min and stored in memory. Data are downloaded after monitoring (maximum monitoring time, 48 h) via a serial or infrared connection to a standard PC for further analysis. It incorporates safety alarms form hypo or hyperglycaemia events and requires one daily calibration only[20].

1.2.2 Minimally Invasive Sensors for CGM

There is no agreement in literature about which kind of sensors should be considered as minimally invasive. In our review, this category will include all the sensors that need the creation of even microscopic holes in the skin to perform the measurement.

Micropores and Microneedle Techniques

For example micropores techniques perforate the stratum corneum without penetrating the full thickness of the skin. A pulsed laser or the local application of heat are considered to form micropores allowing the collection of interstitial fluid applying vacuum. A measure of glucose concentration is then derived from this sample.

SpectRx is made mainly of two units. The first unit is a handheld laser, which creates micropores (size of a hair) in the stratum corneum of the skin. The interstitial fluid, containing glucose, flows through the micropores and is collected by a patch. Then, it reaches a traditional glucose sensor, which is the second unit. The meter also includes a transmitter that sends wirelessly the glucose measurements to a handheld display device[21].

Similarly capillary blood could be sampled using a hollow microneedle, which is almost sensation-less and analyses blood using an enzyme-based system.

Iontophoresis and Sonophoresis

Among minimally invasive sensors, we also include transdermal methods, which stimulate the skin from outside in different manners in order to extract glucose from the skin for its direct measures. This group comprises different techniques like reverse iontophoresis and sonophoresis[21].

The first method is based on the flow of a low electrical current applied across the skin between an anode and a cathode positioned on the skin surface. The application of an electrical potential causes the migration of sodium and chloride ions from beneath the skin towards the cathode and anode respectively, at rates significantly grater than passive permeability. The convective flow induced by this technique carries out neutral molecules, including glucose, along with sodium. Thus, interstitial glucose is transported across the skin towards the cathode, where it is collected and measured by a glucose oxidase-based electrode. The concentration of glucose is low so oxygen is not a limiting factor to glucose oxidase. This technique tends to generate skin irritation and cannot be used if the subject is sweating significantly; in addition it needs a long warm-up and calibration. Skin irritation may be limited by shortening the time interval of the electrical potential application. However, a minimum duration is required to get sufficient amount of glucose for measurement.

GlucoWatch (no more available) device is based on reverse iontophoresis technology. It has a wrist-watch format and measures glucose through the skin using a disposable pad, which clips into the back of the meter. The pad uses an adhesive to stick to the skin allowing it to come in contact with a small electrical current, which causes the reverse iontophoresis, and then the glucose levels in the interstitial fluid can be estimated.

Compared with finger-stick readings, the meter measurements have a 15-min lag time. The meter is intended for use to supplement, but not to replace, information obtained from a standard blood glucose meter. The meter has 2-3 h warm-up period, to remove the glucose on the superficial epidermis and to onset a continuous convective flow. A single-point calibration, performed using a fingerstick blood glucose measurement, accounts for variability in both biosensor sensitivity and skin permeability and is used to convert subsequent biosensor measurements into glucose readings. Afterwards, the meter provides readings every 10 min: 3 min of electrical stimulation (glucose extraction), then 7 min of glucose measurement. The meter has a memory that can store up to 8500 records and the data can be download to a PC for a subsequent analysis.

An alarm also occurs in the case of a rapid change is seen in the blood sugar, in the case of sweating, and for any measurements above or below the patient's target levels. A trend indicator appears to show the direction of the blood sugar when the current measurement is more than 18 mg/dl higher or lower than the previous measurements. Event markers can be recorded for activities like meals, insulin intake and exercise.

However, the meter had several limitations. In fact, the measurements could fail or be inaccurate, if the patient was sweating, or in cases of rapid temperature changes, excessive movement of the meter, or strenuous exercise. Most users reported that the electrical discharge is quite noticeable during the first use of the meter, although it becomes less noticeable on subsequent use. Moreover the disposable pad must be replaced every 12-13 h of monitoring time to ensure continued accuracy; the meter must then go through the warm-up period and calibration again. In addition, it may take more than one try to calibrate the meter, thus requiring additional finger-stick tests. Finally, the meter causes skin irritation to some extent, which limits reuse of the same site to a week or two[22][23].

It has received FDA approval for adults in 1999, and for children aged 7-17

years in 2002. Animas Corporation has withdrawn it from the market since 2007, in preparation for the development of future diabetes management products.

A new Reverse Iontophoresis based Glucose Monitoring Device (RIGMD) has been developed in Korea[24]. It measures a weak electric current that is dependent on glucose concentration in the interstitial fluid, by using an electrochemical enzymatic sensor located on the forearm skin. The sensor is made up of electrodes and a gelatinous material which contains glucose oxidase. A current is produced between the electrodes causing reverse iontophoresis[23].

In [25], it is described the results of preliminary experiments for the development of a mediated glucose biosensor incorporated with reverse iontophoresis function for noninvasive glucose monitoring, using an optimum combination of glucose oxidase and ferrocene.

The sonophoresis uses low-frequency ultrasounds to create an array of microscopic holes on human skin which increase its permeability and allow the migration of glucose contained in interstitial fluid through the skin to a glucose sensor placed in contact with the skin. Thus a direct measure is feasible.

Echo Therapeutics produces a device based on sonophoresis technique. The meter is made essentially of two units: an ultrasonic device (*SonoPrep*), coupled with the skin through an aqueous medium, which increases skin permeation, and a glucose biosensor (*Symphony*), which measure glucose in the interstitial fluid reaching the sensor through the micropathways generated on the skin.

SonoPrep is an ultrasonic skin preparation generator, controlled by a microprocessor. This device delivers low-frequency ultrasound (53-56 kHz), which creates a cavitating force at the point of contact with the skin surface. This force reduces transiently the normally robust lipid barrier of normal intact skin, causing the outermost layer of skin to become increasingly conductive and permeable. Since the relationship between skin conductance and skin permeability, the active ultrasound is terminated when the skin reaches the predetermined level of permeability by continuously measuring skin conductance. This ensures that the site is properly prepared without pain, trauma (such as burn), or irritation. It is claimed that the application of the ultrasonic device for 15 s is enough to make the skin permeable for several hours (between 12 and 24 hours)[26].

Prelude SkinPrep System is a new skin preparation device under development, that can be used in alternative to *SonoPrep*. The system consists of a disposable

Figure 1.6: *Left:Prelude SkinPrep System. Right: Symphony*[27].

abrasive end driven by an electrical motor in a standalone hand piece. Instead of ultrasound,*Prelude* utilizes a mechanical means to remove stratum corneum, with the process controlled by the same conductance-based feedback mechanism used in *SonoPrep*[28].

The *Symphony* is a fully functional prototype biosensor instrument designed to measure glucose through permeated skin. The biosensor is able to maintain reliable fluid contact with the skin through a proprietary biocompatible hydrogel, which utilizes glucose oxidase to measure glucose concentration. The biosensor is housed in a wireless transmitter, which acquires, stores, and transmits coded data to the receiver/monitor to display a reading every minute in addition to trends and alarms for excessively high and low BGL[28].

A common limitation of all these methods is the delay between plasma and interstitial glucose concentration. This phenomenon is due to the glucose transport from plasma to interstitium that act as a low pass filter.

1.2.3 Non Invasive CGM Sensors

Non invasive CGM sensors measure glucose concentration through the skin without extracting blood or interstitial fluid or without a needle penetrating the skin for reaching these fluids. Hence, these sensors are very comfortable for the patient and do not cause displeasing physiological reactions. However, the

measure is affected by different confounding factors, making more difficult to perform an accurate measurement.

These sensors are based on different physical properties of the skin and underlying tissues (optical, thermal, acoustic and electrical) which are influenced by glucose concentration. Given the special importance of these sensors in the present thesis, in the next Chapter, the physical principles of these sensors will be described in detail. For each technique, an example of its application for CGM will be presented, with particular attention to the Solianis Multisensor.

Chapter 2

Non Invasive Continuous Glucose Monitoring: Principles, Open Problems and Aim of the Thesis

A review of Non Invasive Continuous Glucose Monitoring (NICGM) systems will be presented in this Chapter. We will start with the optical ones, based on the interaction of light with skin and then we will move toward the thermal-, acoustic, electric-based ones. For each method, principle and an example of application will be described. Finally, in Section 2.2, particular attention will be used to report the features of the Solianis Multisensor and an example of Multisensor signals will be shown, along with the problems connected with its modeling.

2.1 Physical Principles

NICGM sensors measure glucose concentration without extracting blood or interstitial fluid or without a needle penetrating the skin for reaching these fluids. Thus, the measure is performed through the skin that is a particular multi-layer biological tissue. Consequently, to understand the characteristics of these sensors is opportune to describe the skin morphology and the non-uniform blood distribution within the layers.

2.1.1 Skin Properties

The skin is composed by several distinctive layers as illustrated in Figure 2.1.

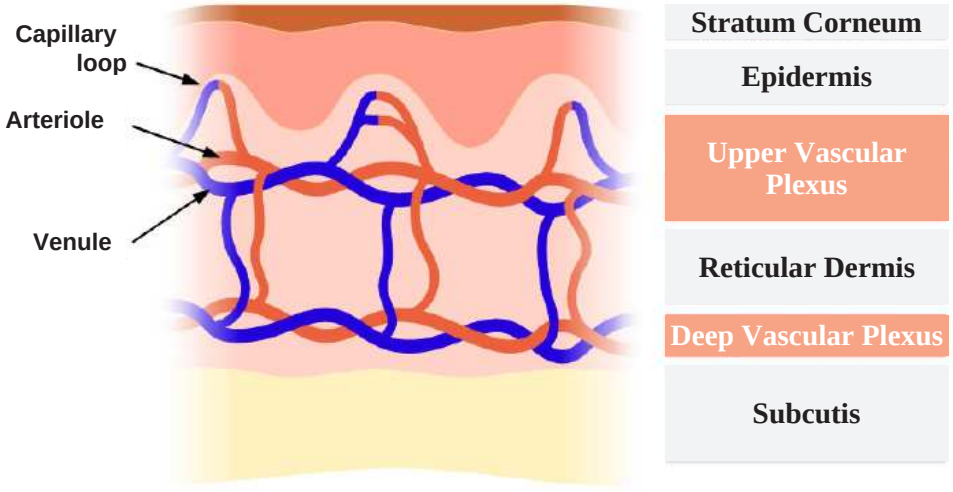

Figure 2.1: Representation of the skin layered structure highlighting the distribution of blood vasculature [29].

The uppermost skin layer is the *stratum corneum* of epidermis, composed of dead keratinized cells, followed by the living epidermis and the connective tissue of the dermis. The subcutaneous tissue is composed by an underlying fat layer and muscle. The dermis can be subdivided into three different layers: the upper vascular plexus, the reticular dermis and the deep vascular plexus. The epidermis does not have its own vasculature. The volume fraction occupied by blood vessels in the dermis is in the range of 1-20% and is concentrated in the upper and deep vascular plexus.

Most of NICGM sensors, e.g. Diasensor[23], TANGTEST[30], OrSense[31], Sentris-100[32] and other prototypes in development, are optical transducers that use light in variable frequencies to track glucose. Indeed, they exploit different properties of light to interact with glucose molecules in a concentration-dependent manner. These optical sensors monitor glucose variations in the dermal blood; hence the radiation needs to penetrate at least through the epidermis to reach the vascularised compartments of the dermis. Along with these optical sensors, other non-invasive approaches exploit on thermal, acoustic and electrical properties. This classification is reflected into the scheme of Figure 1.2.

2.1.2 Optical Techniques for NICGM

A beam of light interacts in different manners when it passes through a tissue like skin. A part of the beam is reflected by the stratum corneum, another part is absorbed from the tissue and the remaining part is scattered (i.e. it is deviated from the straight trajectory) and diffused into a number of different directions. Figure 2.2 shows a general scheme that summarizes the different kinds of interaction of light with skin.

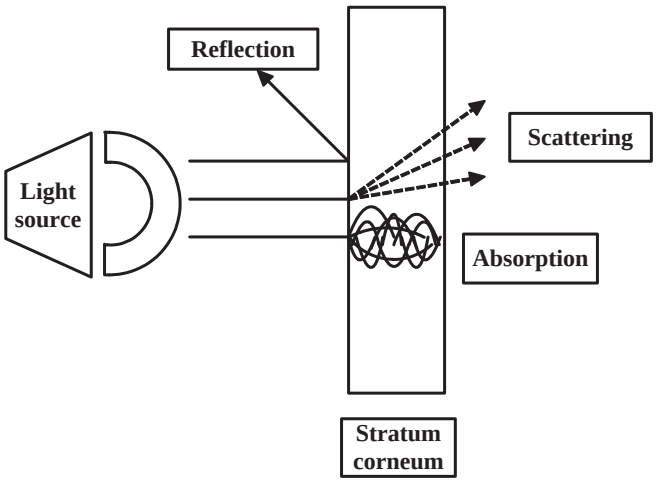

Figure 2.2: Optical properties of light utilized in glucose detection[33].

Spectroscopy analyses the optical properties of light in relation to the wavelength of the radiation. Spectroscopy also provides a precise analytical method for finding the constituents (and their concentration) in materials having unknown chemical composition, since each substance exhibit characteristic spectra, which may be interpreted as the "fingerprint" of that substance. The different types of spectroscopy may be classified according to which optical properties of the light is employed.

Absorption (MIR/NIR spectroscopy)

Infrared absorption spectroscopy is based on absorption phenomena: changes in glucose concentration can influence the absorption coefficient of tissues and thus the absorption bands[21].

In particular, the so called Near InfraRed spectroscopy (**NIR**) uses light in the near infrared range (750-2000 nm). Specific spectra are chosen in order to

minimize background absorption, in particular by water. The light in these wavelengths passes through the *stratum corneum* and epidermis into the subcutaneous space, allowing to measure in the deep tissues (in the range of 1 to 100 mm of depth). Perturbing factors that may interfere with glucose measurement include all the variables that influence absorption coefficient, like blood pressure, body temperature and skin hydration. Errors can also occur due to environmental variations such as changes in temperature, humidity, carbon dioxide, and atmospheric pressure. The absorption coefficient of glucose in the NIR band is low and is much smaller than that of water by virtue of the large disparity in their respective concentrations. Thus, in the NIR, the weak glucose spectral bands not only overlap with the stronger bands of water, but also those of hemoglobin, proteins, and fats.

Changes in glucose may affect the measurement process also in other indirect ways: for example, hyperglycaemia causes increased perfusion, which influences the spectrum and can be considered as a confounding factor. Furthermore, diabetic subjects can exhibit "thick skin" and "yellow skin"[34]. Thus light reflected from skin of a diabetic patient may differ from that of a healthy subject at equal level of glycaemia.

In contrast to NIR, Mid InfraRed spectroscopy (**MIR**) utilizes light at a wavelength between 2500-10000 nm. With respect to NIR, MIR exhibits less scattering phenomena and greater absorption. Hence the tissue penetration of light in MIR can reach only the *stratum corneum*.

The *TANGTEST Blood Glucose Meter* seems to be based on NIR technology. It is a prototype, which measures glycemia by analyzing intensity variations in the spectrum of a weak light (about 0.1 W) transmitted through the tested finger (middle or index finger). The authors claim that the signal noise due to other tissues is avoided by using the optical signal of pulsatile microcirculation: the signal obtained by the meter is in fact divided into a pulsatile and a direct component. The pulsatile component, which is synchronized with heart rate, is used to monitor blood glucose[23][30].

The *Diasensor* device is based on the near infrared (NIR) spectroscopy technology. It operates by placing the patient's forearm on the arm tray of the meter. The dimensions of the meter are relevant compared with other meters, but it is still sufficiently compact to be used in a domiciliary environment. The blood

glucose test is obtained in less than 2 minutes. However, it is not intended as a replacement for the traditional invasive blood glucose meter. It seems that the distributor was EuroSurgical Ltd., UK. However, the web site of the company does not currently mention *Diasensor*, and hence it can be concluded that it is not on sale anymore[23].

InLight Solutions is developing a device based on NIR spectroscopy and multivariate analysis to make quantitative and qualitative measurements. Appropriate optic and software have been develop to clearly distinguish glucose molecules from water molecules. The devices are made up of three components: a light source, an optical detector, and a spectrometer. The measures are performed using the differences between the light that was sent into the skin and the light that the detector collects[35].

Scattering(Raman Spectroscopy)

Raman spectroscopy measures scattering of single wavelength light and is based on the fact that a small fraction of scattered light shows wavelengths different from the one of the exciting beam. This fraction is dependent on rotational or vibrational energy states within a molecule. Raman spectroscopy shows highly specific absorption bands and, compared with MIR and NIR spectroscopy, it has the benefit of suffering of less interferences from water. However, the Raman signal is weaker than its counterpart in other technologies due to the fact that measured photons normally have lower intensity than the original light and thus requires powerful detectors[36].

Recently, an improvement in traditional Raman spectroscopy has been proposed (surface-enhanced Raman spectroscopy), which may increase the sensitivity of the acquisition. It has only been tested in rats[37].

A prototype of sensor based on Raman spectroscopy has been described and tested. Raman spectra were collected by means of a specially designed instrument, optimized to collect Raman light emitted from a scattering medium (tissue) with high efficiency and a diode laser as the Raman excitation source[38].

Scattering(Occlusion Spectroscopy)

Another technique that measures scattered light is occlusion spectroscopy[33], which is based on the property of glucose to decrease the diffusion coefficient and

on the enhanced transmission of light due to erythrocyte aggregation that can be reproduced in vivo by applying a pressure to the fingertip for 2-3 seconds, greater than the systolic one. One signal is collected when no pressure is applied and it is combined with the occlusion signal in order to calculate glucose concentration thanks to a specific algorithm. The advantage of this method is that it measures arterial glucose level. However, intrinsic erythrocyte aggregation and free fatty acid concentration may interfere with the measure. Calibration is needed for glucose predicting parameters using four blood glucose reference points in the first three hours, and an additional reading after 8 hours.

Figure 2.3: *OrSense NBM-200G*[39].

OrSense NBM-200G is based on Occlusion Spectroscopy. European CE mark was granted in 2007. The measurement is performed using an annular probe, which is positioned on the finger's root and contains light sources, detectors and pneumatic cuffs producing oversystolic pressure to occlude blood flow. The optimization of sensitivity and specificity is achieved by the following:

- *Transmission mode.* In the transmission mode the light traverses the whole organ (finger), and the photons typically encounter many more glucose molecules along their paths than in the reflection mode. This strategy enhances the sensitivity to glucose and reduces the influence of local factors such as skin morphology and pigmentation.

- *Dynamic signal.* Occlusion spectroscopy is based on generation of an optical signal that changes with time. The signal is induced by oversystolic occlusion at the finger's root, which causes cessation of blood flow throughout

the finger. This strategy allows us to collect not only one data point per wavelength, but rather a whole function. It results in a better signal-to-noise ratio.

- *Multispectral data.* Multiple wavelengths of light sources are used. This is beneficial for specificity/selectivity, as the different behaviour of the optical signal among wavelengths allows cleaning the influence of unwanted interferences, such as the absorption of hemoglobin and changes of oxygen saturation.

- *Sophisticated algorithms.* The data are processed with sophisticated algorithms, which use only a small number of parameters, hence avoiding overfitting and false correlations.[31]

Optical Coherence Tomography

Other types of glucose sensors are based on Optical Coherence Tomography (OCT) that was originally developed to perform the tomographic imaging of the eye. An OCT system uses a low-power laser source, an interferometer with two arms (reference and sample) and a photodetector to measure the interferometric signal[33].

The skin is irradiated with a low coherence light (light in which the emitted photons are synchronized in time and space). Backscattered radiations from tissues are combined with light returned from reference arm and the resulting interferometric signal is detected by the photodetector. Basically, it measures the delay correlation between the two original signals. Using this technique, glucose concentration in the dermis can be determined, since an increase of glucose concentration in the interstitial fluid causes an increase in the refractive index, thus determining a decrease in the refractive index mismatch.

This technique is affected by motion artefacts. In addition, while small changes in skin temperature have negligible effects, changes of several degrees have a significant influence on the signal.

The *Sentris-100* device is based on optical coherence tomography technology. It uses an infrared light to scan a cylindrical volume of skin in several steps from the skin surface down to the subcutaneous tissue. Acute changes in protein (collagen and myosin) conformation occur in response to glucose concentration

Figure 2.4: The *Sentris-100*.

changes and creates a high sensitivity in the optical coherence tomography signal; localization of signal detection to blood vessel walls minimizes any observed signal lag[23][32].

Fluorescence Technology

Fluorescence technology has also been proposed for glucose monitoring and is based on the generation of fluorescence by human tissue when excited by lights at specific frequencies. These sensors are able to measure glucose levels exploiting the dependence between fluorescence intensity and glucose concentration in the solution. Other fluorescence-based glucose sensors are based upon the affinity sensor principle, where glucose and a fluorescein-labelled analogue bind competitively with a receptor site specific for both ligands. Thus, an increase in glucose concentration causes a decrease in the binding of receptor with fluorescein-labelled analogue resulting in a decreased light emission[33].

A glucose-sensing contact lens has been developed using boronic acid to measure lachrymal glucose concentration[40]. The main drawback of this system is that it requires a hand-held external light source/detector. Thus, even if theoretically the lens is able to monitor continuously the glucose concentration, the

information is carried out only with the detector usage.

Recently an injectable hydrogel microbeads has been developed for fluorescence-based in vivo continuous glucose monitoring. A fluorescent monomer based on diboronic acid has been developed. It enables reversible responsiveness to glucose without any reagents and enzymes. The fluorescent monomer has long, hydrophilic spacers and polymerization sites to bind flexible supports. The fluorescent monomer has sufficient intensity for in vivo transdermal monitoring; even when it is immobilized in a solid support (microbeads). Due to the virtue of their small size, the fluorescent microbeads are injectable, minimally invasive, and rapidly respond to glucose change. The microbeads have been tested with success in rats[41].

Polarimetry

Polarimetry is based on the optical properties of glucose, due to its chemical structure that makes glucose a chiral molecule. When polarized light (light with all waves oscillating in the same plane) passes through a solution containing optically active solutes, such as chiral molecules, its polarization plane is rotated by a certain angle, which depends on solutes concentration. Measuring the rotation angle with a polarimeter allows calculating glucose concentration. This technique is sensitive to scattering properties of tissues that depolarizes light. However, skin cannot be investigated by polarimetry since it shows high scattering due in particular to the *stratum corneum*. For this reason the preferential measurement site is the eye, in particular the *aqueous humor* beneath the *cornea*, which has low scattering properties. However, this particular measurement site raise a second problem: a time delay between glucose concentration in *aqueous humor* and blood. Other general sources of errors are variations in temperature and pH of the solution[36].

A new real-time optical polarimetric approach for glucose sensing utilizing two wavelengths is presented in [42]. Only in vitro experiments have been reported. In fact an efficient eye coupling mechanism has not been developed yet, allowing in vivo experiments on rabbits eyes.

2.1.3 Thermal Emission Spectroscopy

Thermal emission spectroscopy measures IR signals generated in the human body as a result of glucose concentrations changes. The tympanic membrane is used as measuring site, since this membrane shares the blood supply with the centre of temperature regulation in the hypothalamus[43]. Body movements and ambient temperature are the most significant sources of noise in this method[36].

2.1.4 Photoacoustic Spectroscopy

Photoacoustic spectroscopy uses the principle that absorption of a laser light causes consequent acoustic response. Tissue is illuminated by a short laser pulse, at a specific wavelength, and the absorbed radiation causes localised heating. The small temperature increase is dependent on the specific heat capacity of the tissue irradiated. Volumetric expansion due to heating generates an ultrasound pulse, which can be detected by a microphone. Increasing tissue glucose concentrations reduce the specific heat capacity of tissue and thus increase the velocity of the generated pulse making photoacoustic spectroscopy an indirect technique for glucose estimation[21].

Besides this, the photoacoustic spectrum considered as a function of laser light wavelength mimics the absorption spectrum in clear media (i.e. optically thin) and has the advantage to present higher sensitivity in the determination of glucose, thanks to the poor photoacoustic response of water.

The main limitation of this technique is its sensitiveness to chemical interferences from some biological compounds and to physical interferences from temperature and pressure changes.

The *Aprise* device is based on photoacoustic technology. It exploits in fact the photoacoustic properties of the blood and tissues to estimate the prevailing glucose levels. The sensor is attached to the skin above a blood vessel, and it generates ultrasound waves by illuminating the tissue with laser pulses. Analysis of the acoustic signals provides information on the absorbance of light in the tissue at different depths, which is influenced by glucose concentration. An ultrasonic image of the optical properties of tissue directly beneath the sensor is obtained. The ultrasonic image resolves the blood vessel from the tissue layers around it, enabling separated analysis of changes in optical properties of blood

and surrounding tissues[23][44].

2.1.5 Electromagnetic Sensing

Another technique for investigating dielectric parameters of blood utilizes the electromagnetic coupling between two inductors turned around the medium under study. Indeed, the coupling is influenced by variations in the dielectric parameters of the solution, which are modified by glucose. This method is based on the application of a voltage signal with proper frequency to the primary inductor and for electromagnetic coupling a signal will be produced on the secondary inductor. There exists an optimal frequency, where the sensitivity to glucose change is maximal, but it is significantly influenced by temperature. The main problem of this technique is that several other components may have an influence upon the blood dielectric parameters and not only upon glucose[21].

A new electromagnetic sensor is described in [45]. Its in vitro ability to estimate variations in glucose concentration of different solutions with similarities to blood (sodium chloride and Ringer-lactate solutions) has been tested, differing though in the lack of any cellular components. The sensor was able to detect the effect of glucose variations over a wide range of concentrations.

The *Glucoband* is a non-invasive glucose monitor that uses bio-electromagnetic resonance to measure the blood glucose levels of the human body. This device is worn like a wrist watch and displays results of the test on an LCD screen. The initial measurement process takes only a few minutes. However, in the monitoring mode, measurements can be continuous. Since each concentration of glucose has its unique electromagnetic molecular self-oscillation signature-wave, the *Glucoband* perform the measure matching the self-oscillation frequencies of glucose molecules with those of hundreds of reference solutions with different levels of glucose stored in an internal database of “signatures”.

2.1.6 Impedance Spectroscopy

Another kind of spectroscopy investigates the dielectric properties of a tissue using a current flow instead of a light beam. It is called Dielectric Spectroscopy (DS) or Impedance Spectroscopy (IS).

The impedance of one tissue can be estimated by applying current of known

intensity. The experiment is repeated with low alternating currents at different frequencies in order to measure the dielectric spectrum (the impedance as a function of frequency). Glucose is indirectly measured by its interaction with red blood cells. In particular, variations in plasma glucose concentration induce in red blood cells a decrease in sodium and an increase in potassium ion concentration. These variations cause changes in the red blood cells membrane potential, which can be measured by determining the permittivity and conductivity of the cell membrane through the dielectric spectrum[21]. The different sources of error include temperature variations, sweating and motions. Furthermore, this technique requires calibration.

The IS technology has been used for many years to investigate the dielectric properties of cells and organelles. Before going into the details of this technology it is worth having a review of dielectric properties of biological materials.

DISPERSION	**FREQUENCIES**	**ORIGIN**
α	low(10-100Hz)	electrical double layers and electrolytes at membrane boundaries
β	radio(100kHz-10MHz)	cell suspension(blood)
δ	radio(10MHz-1GHz)	water bound to proteina and internal protein motion
γ	microwave(1-100GHz)	reorientation of free water molecules

Table 2.1: Dielectric properties of biological materials.

The dielectric properties of biological materials are characterized by four major dispersions, which are termed α, β, δ, and γ. Different mechanisms account for low frequency (α), radiofrequency (β), and microwave frequency (γ) dispersions[46]. The α-dispersion is generally considered to be associated with interfacial polarization linked with electrical double layers and surface ionic conduction effects of electrolyte at membrane boundaries. The β-dispersion has essentially two components arising from two different mechanisms: the capacitive shorting out of membrane resistances and rotational relaxations of biomolecules. Cell suspensions such as blood will typically exhibit a significant β-dispersion in the radiofrequency range between 100 kHz and 10 MHz. In addition, reorientation of free water molecules causes γ-dispersion. Water bound to protein and internal protein motion will also cause a subsidiary process, called the δ-dispersion, that

is observed in the frequency region between the β- and γ-dispersion. All these processes are summarized in Table 2.1.

Thus, IS can investigate the relaxation processes of complex systems in an extremely wide range of time constants, ranging from 10^{-12} to 10^{4} seconds. In particular, it is sensitive to intramolecular interactions and it is able to monitor cooperative processes.

IS-based techniques cannot measure glucose concentrations directly, since changes in glucose levels do not directly affect the dielectric properties of skin and the underlying tissue in the kHz and MHz frequency band. However, variations in plasma glucose lead to changes in the electrolyte balance in blood, cells and interstitial fluid. An increased concentration of glucose in blood involves a cellular biochemical response, which leads to changes of membrane components, nucleotide and ionic rearrangement. In particular, as a consequence of water movement, there is a decrease of sodium and an increase of potassium inside the erythrocyte. This variation of the electrolyte balance has an influence over the erythrocyte membrane potential and capacitance, which causes changes in the ac and dc conductivity and tissue permittivity that can be measured using IS[47]. A sensor based on IS uses electromagnetic waves in the selected frequency band that interact with the skin and the underlying tissue for monitoring these electrical properties.

In [48] and [49], clinical experiments in controlled conditions using IS showed promising results in monitoring changes in blood glucose levels. But, as soon as environmental conditions become less favourable, going towards a routine use for daily life, this technique exhibits its limitations, resulting in a significant reduction in signal quality. This is a direct consequence of the deleterious effects of many perturbing factors, such as temperature fluctuations, variations of skin moisture and sweat, changes in cutaneous blood perfusion and body movements affecting the sensor-skin contact surface[50]. Consequently, all these perturbations affecting the main glucose related signals have to be identified, characterised and compensated for. As better discussed in the following, this suggested in [1] the development of a Multisensor Glucose Monitoring System, where the multi-sensor concept means a system that includes several sensors on one substrate attached to the human body to allow broader bio-physical characterization of the skin and underlying tissues.

2.2 The Solianis Multisensor Approach to NI-CGM

This section will focus on the Solianis Multisensor approach to NICGM. After the description of the different sensors, an example of Multisensor data will be presented, along with its analysis procedure.

2.2.1 Description of the Sensor

Solianis Monitoring AG (Zurich, Switzerland) has proposed a system for NI-CGM based on the multi-sensor concept. The Solianis Multisensor is mainly based on a combination of dielectric and optical sensors, as well as temperatures and others, for the characterisation of the skin with the aim of monitoring blood glucose changes.

The Multisensor performs a continuous glucose monitoring collecting one set of signals, containing the information from each sensor every 20 seconds. As shown in Figure 2.5 the Multisensor is attached to the upper arm of the patient with a flexible band and it is powered with a battery pack.

IS electrodes

As described in Section 2.1.6, changes in blood glucose levels cause dielectric changes of skin and underlying tissues within the frequency range of 0.1-100 MHz, which is measured utilizing particular capacitive fringing-field electrodes[1]. In order to achieve different penetration depths of the electromagnetic field into the various tissue layers, three electrodes with different characteristic geometries are used in the Solianis Multisensor. In fact, the interaction between an applied electromagnetic field and the skin depends not only on the frequency band, but also on the geometric properties of the electrode. The differences between the three IS electrodes consist in the distance between the active electrode and the ground potential. In particular, a distance of 0.3, 1.5 and 4 mm is associated with shallow, mid and deep penetration respectively and the sensors are referred as short, middle and long, respectively (see Figure 2.5) .

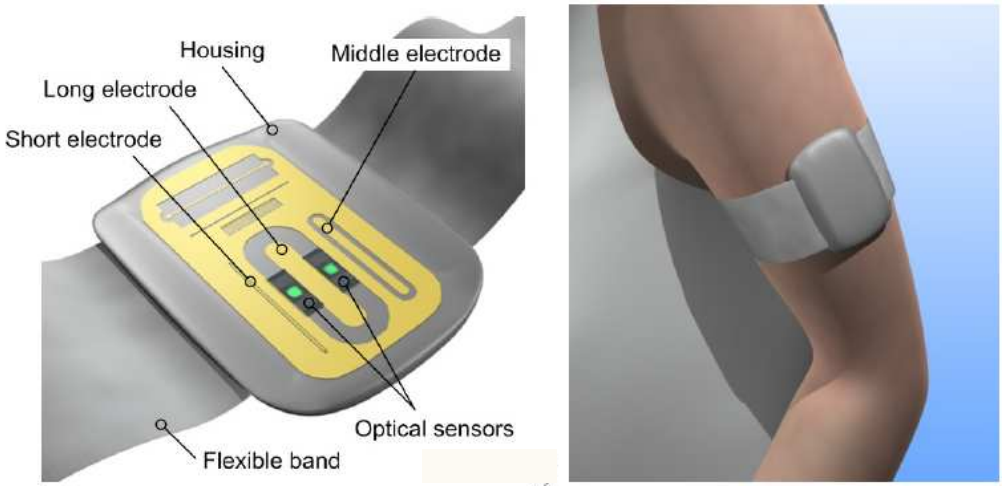

Figure 2.5: *Left*: Optical and dielectric sensors composing the Solianis Multisensor. *Right*: Solianis Multisensor attached to the upper arm with a flexible band.

The short electrode penetrates only the upper skin layers, thus it cannot yield information about glucose levels, but it may still contain information about perturbing effects related to the uppermost layers. Data from long and middle electrodes are regarded as primary signals, since they penetrate also the lower skin layers that are well micro-vascularised (see Figure 2.1) and hence particularly affected by glucose variations.

Optical sensors

As mentioned before, other sensors are used with the aim of obtaining useful information to compensate the perturbing factors: two optical sensors are embedded within the Multisensor substrate for the measurement of skin blood perfusion, which is a perturbing factor for dielectric signals[29]. Each optical sensor features 3 LEDs, located closely to each other, with the following wavelength: green (568 nm), red (660 nm) and infrared (798 nm). Light reflected back from the skin is detected by two photo-detectors (signal diodes), while the variation of emitted LEDs intensity are monitoring by two reference diodes (monitoring diodes) located near the LEDs.

Sweat sensors

An interdigitated electrode is used to measure the dielectric response at lower frequencies in the range of 1-200 kHz for obtaining information about sweating events and superficial hydration levels. An other sensor exploits the frequencies in the range of GHz to estimate hydration levels of the underlying skin layers, since GHz excite free water molecules (see Table 2.1).

Acceleration sensors

An integrated accelerometer has the aim to monitor continuously the acceleration and the position relative to the centre of gravity of the device.

Other sensors

Finally, others sensors monitor skin and housing temperature, and ambient humidity close to the device.

2.2.2 Example of Solianis Multisensor Data

As described in the previous Section, the Solianis device includes several sensors on one substrate in contact with the human body. Each sensor provides its specific set of signals. However, in order to obtain glucose readings, this set of measures needs to combined in a proper way through a relationship, namely a model linking what is measured with glucose levels. Learning this relationship involves also the use of some glucose references (one approximately every 15 minutes) that are collected using finger-stick methods, while the Multisensor is attached to the patient's arm.

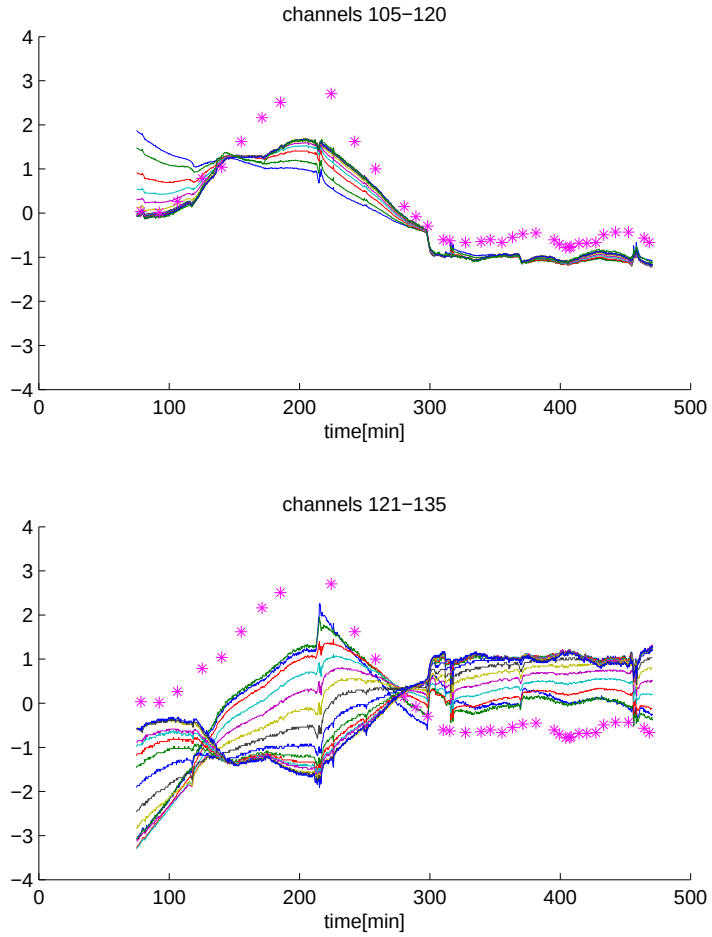

Figure 2.6: Example of normalised impedance signals (continuous lines) vs. normalised reference BGL (star).

In Figure 2.6, representative time series collected from the same electrode at different frequencies are shown. In particular, the impedance at different frequencies is represented using its magnitude (Figure 2.6, top) and its phase (Figure 2.6, bottom). As shown in the top panel the magnitude signals at different frequencies are similar but not identical, thus they show a strong correlation. The same is for the phase signals, which are also correlated with the magnitude signals. Since the impedance channels, as mentioned in the previous section, contain glucose information, they are referred as the primary "glucose signals"[1].

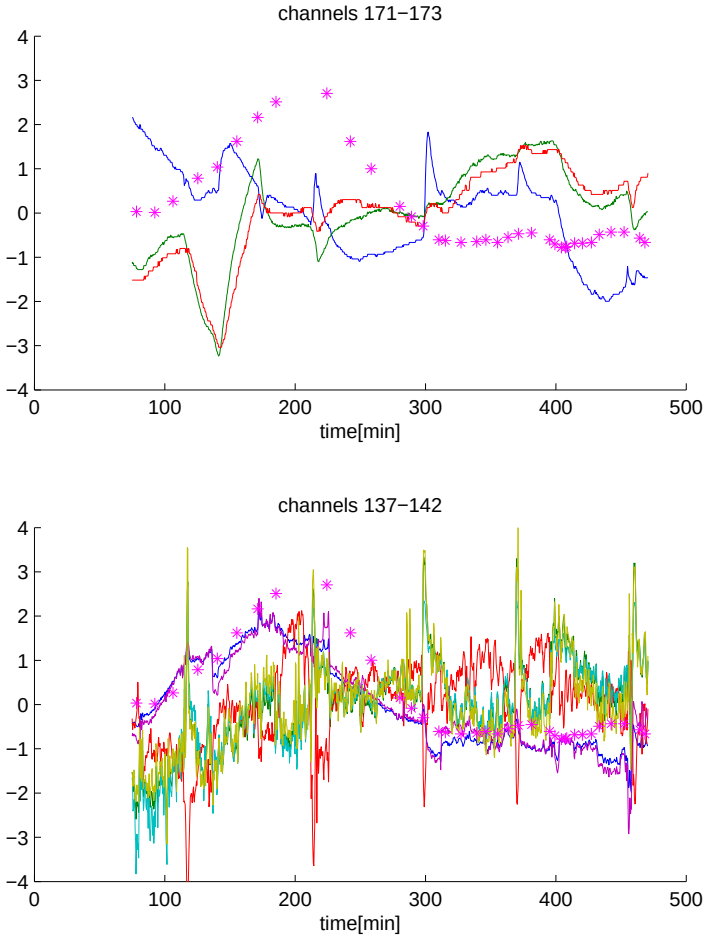

Figure 2.7: Example of normalised impedance signals (continuous lines) vs. normalised reference BGL (star).

In Figure 2.7 we plot the time series from channels associated with other sensors embedded within the Solianis device. In the top panel the skin and the housing temperature are plotted, along with ambient humidity. In the bottom panel an example of optical channels is shown. Some of them are correlated with the glucose reference. However, they are noisier then impedance channels.

2.2.3 From Multisensor Data to Glucose: Necessity of a Model

In the previous Section the Multisensor data have been described. This section has the aim to illustrate how to "link" these measurements with BGL.

As explained in Section 2.2 some signals contain the glucose information, while the other are used to characterise the perturbations affecting the primary signals, allowing their compensation. Hence, the Multisensor signals have to be combined to estimate BGL correctly through a relationship between the measured variables and the concentration of glucose in blood.

Representing with $\boldsymbol{S}(t)$ the Multisensor signals at the t-th time instant, the unknown relationship can mathematically be expressed as:

$$\mathrm{BGL}(t) = f(\boldsymbol{S}(t), \boldsymbol{\theta})$$

where f represents a generic function used to convert the signals into BGL values and $\boldsymbol{\theta}$ is an unknown vector containing the parameters that characterise this conversion.

To fully express this relationship, first of all one has to select what kind of function f suits best for the specific sensor (e.g. linear or not). Then, once the mathematical model for the connection between the signals is chosen, there are different methods to estimate its parameters. To perform this parameters estimation some BGL samples, collected using finger-stick method, along with the corresponding sensor signals are needed.

At the end of this estimation process, the sensor is calibrated, i.e. the measured signals $\boldsymbol{S}$ can be automatically converted to glycaemic values, which is the output to be displayed.

2.2.4 Open Problems

The principal Multisensor signals are those that mostly contain the information about glucose fluctuations. However, in the everyday life, these signals are affected by different perturbing factors (temperature fluctuations, skin moisture, sweat, blood perfusion ...). The multi-sensor concept derives from the necessity of compensate these perturbations. Hence, the other signals have the aim to quantify the perturbation allowing a better estimation of BGL.

As said before, there are different ways to combine the signals with the view of estimating BGL accurately. It is difficult to evaluate which is the exact effect of the perturbing factors on the principal signals and how the auxiliary signals are connected to the perturbing factors. Hence, different mathematical models and different estimation techniques have to be evaluated in order to determine which is the best way of combining the signals.

The context, in which this thesis is part, focus on the definition of a multivariate linear regression model for combining the Multisensor measures in order to estimate glucose profiles. In particular, the intrinsic Multisensor signals's features may cause the learning algorithm to fail in estimating good model parameters. This is because each learning technique has its own advantage and drawbacks that will be fully and deeply described in the next Chapters.

2.3 Aim of the Thesis and Outline

The thesis concerns the development of a model for estimating glucose level from the Solianis Multisensor data. In particular, different methods for estimating a linear regression model will be applied to the data in order to perform glucose levels estimation.

Three different methods have been compared: Ordinary Least Squares (OLS), Partial Least Squares (PLS) and LASSO. OLS uses all the signals and make a linear combination of them in order to obtain the best fit. PLS constructs new orthogonal predictors, starting from the original signals and makes a linear combination of a selected number them to calculate the regression. Finally, LASSO makes a linear combination of the signals, but penalizes the sum of absolute coefficients to prevent the multiplication coefficients from assuming too large values. In addition, LASSO has the characteristic to yield to sparse solutions, which means that some coefficients will be exactly zero.

The aim of the thesis is to evaluate the performance of these methods and highlight their advantages and drawbacks in modelling Solianis Multisensor data.

After a theoretical presentation of the regression problem (Chapter 3) and of

the parameter estimation methods (Chapters 4, 5 and 6), they will be applied to real data and results will be compared in order to select the best technique (Chapters 7 and 8). Finally, further topics and margins of improvement for modeling Solianis Multisensor data will be described (Chapter 9).

Chapter 3

Fundamental Concepts of Linear Regression for High-Dimensional Data

This chapter reports an introduction to the regression problem, with particular regard to high-dimensional datasets. The notation used throughout this thesis will be also introduced.

3.1 Problem Formulation and Used Notation

In general, the regression problem can be stated as the estimation of a variable (which is not easily measurable in practice, e. g. glucose in our case), from a set of measured variables (the Solianis Multisensor data). The unknown variable to be predicted is called *output* or *target*, while the measured variables are called *inputs*, or *regressors* (because they contain the information for the regression model) or *predictors* (because they are used to predict the output).

Thus the aim of regression is to build and identify a prediction model. This model can be later used for estimating the outcome for new unseen data. In our case study, our aim is to model the relationship between BGL and Solianis Multisensor data. The relationship is mathematically expressed using a model, which can be used to combine new Solianis Multisensor data in order to predict glucose concentration. A good prediction model accurately predicts such an outcome.

In the general case, the output variable (i.e. the target that should be described

by the model through a combination of the inputs) consists of a multi-dimensional vector. However, since our purpose is to estimate glucose, throughout this thesis, only the case of a single output variable will be considered. Hence, the output will be represented by a column vector $\boldsymbol{y}$ of dimension N x 1, where N is the number of available samples. In symbols:

$$\boldsymbol{y} = [y_1 \quad y_2 \quad \dots \quad y_N]^T \tag{3.1}$$

where y_i denotes the i-th sample of the reference.

The input variables are contained in the matrix $\boldsymbol{X}$ of dimension N x p, where the element x_{ij} represents the i-th sample of the j-th variable.

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \\ x_{N1} & x_{N2} & & x_{Np} \end{bmatrix} \tag{3.2}$$

While each row of the matrix $\boldsymbol{X}$ is formed by a set of p variables relative to the same i-th time instant (represented by the row vector $\boldsymbol{X}_{ip}$ (1 x p)), each column contains the N samples of the j-th variables (symbolized using the column vector $\boldsymbol{X}_{jN}$ (N x 1)). Hence, while subscript $i \in [1, 2 \dots N]$ indicates the sample, subscript $j \in [1, 2 \dots p]$ identifies the variable. To distinguish for example, $\boldsymbol{X}_1$ the set of p variables at the first time instant from $\boldsymbol{X}_1$ the N samples of the first variable, a second subscript is added, indicating the dimension of the vector.

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & \dots & \overline{x_{1j}} & \dots & x_{1p} \\ \vdots & & \vdots & & \vdots \\ \boxed{x_{i1}} & \dots & \boxed{x_{ij}} & \dots & \boxed{x_{ip}} \\ \vdots & & \vdots & & \vdots \\ x_{N1} & \dots & \underline{x_{Nj}} & \dots & x_{Np} \end{bmatrix} \Rightarrow \boldsymbol{X}_{ip} \quad \Downarrow \quad \boldsymbol{X}_{jN} \tag{3.3}$$

After having introduced some notation and basic concepts, the general regression problem can be mathematically expressed as finding the function f that defines the relationship between the input $\boldsymbol{X}$ and the target $\boldsymbol{y}$:

$$\boldsymbol{y} = f(\boldsymbol{X}) \tag{3.4}$$

The relationship, described by f, is not deterministic, since the measured input variables are affected by random noise (zero mean and uncorrelated).

Throughout this thesis only linear model are considered. Therefore, eq. (3.4) turns into:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}_0 \tag{3.5}$$

where the output is given by a linear combination of the inputs weighted by the column vector $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots \beta_p]^T$. The vector $\boldsymbol{\beta}_0$ (N x 1) describes the data offset and can be dropped if the data have zero mean. Hence, the aim of regression is to find an estimate $\hat{\boldsymbol{\beta}}$ of the unknown coefficients $\boldsymbol{\beta}$, given the knowledge of the reference vector $\boldsymbol{y}$ and of the causally related inputs $\boldsymbol{X}$. The set of data used to estimate the coefficients is called *training set*. After $\hat{\boldsymbol{\beta}}$ is determined, it can be used to calculate the correspondent model prediction of the target $\hat{\boldsymbol{y}}$ and, more important, to predict the unseen output from inputs.

It is worth saying that the real data, which will be described in Chapter 7 and analysed in Chapter 8, are composed by a glucose reference, obtained using finger-stick method, that corresponds to the vector $\boldsymbol{y}$ and by all Solianis Multisensor signals, which are included in the input matrix $\boldsymbol{X}$. As a consequence, the aim of the analysis is to predict blood glucose levels from the signals measured by the Solianis Multisensor.

3.2 Issues of High-Dimensional Regression

It could seem reasonable that, if the training set is large enough, it would be easy to generalize data behaviour and identify a good prediction model. However this it not true dealing with high-dimensional data. This is exactly our case since we have to deal with more than 150 input variables (Multisensor signals). The algorithms for solving regression problems suffer from the so called curse of dimensionality [51] when applied to high-dimensional datasets.

3.2.1 Curse of Dimensionality

Consider a p-dimensional unit hypercube and suppose the N regressors samples to be uniformly distributed in it. The fraction of samples included in a hypercube with side $r(< 1)$ is:

$$frac = r^p$$

Extracting the side of the hypercube as a function of the desired fraction and the dimension p, one gets:

$$r = frac^{1/p}$$

Hence, for example, to include 10% of the samples, we need a hypercube with side 0.1 for p=1, and a hypercube with side 0.8 for p=10. The different curves plotted in Figure 3.1 show the side of the hypercube as a function of the fraction of included samples for different values of dimension p.

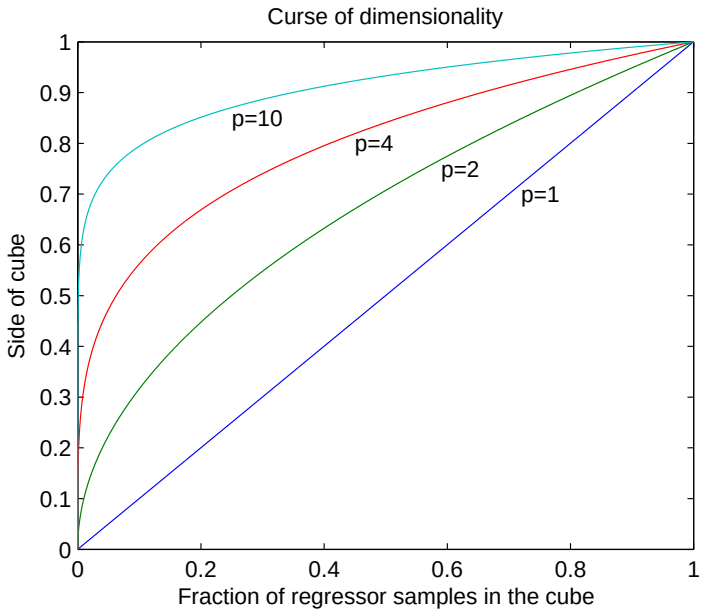

Figure 3.1: side of the cube as a function of the fraction of included samples for different values of dimension p.

As shown in the Figure 3.1 the hypercube side needed for a given fraction increases even more as the dimension p increases. Hence, as the number of the regressors increases, it becomes more difficult to generalized data behaviour. In fact, the samples are more distant to each other and, in particular, they tend to be close to an edge of the sampling area, because the prediction is much more difficult near the edges of the training sample.

One can also formulate the problem considering that the sampling density is proportional to $N^{1/p}$. In high dimensions all feasible training samples sparsely populate the input space. In fact, the density rapidly decreases to zero as p increases.

3.2.2 Overfitting

In addition, high-dimensional regression algorithms have to deal with overfitting, namely the risk of fitting a predictive model not only to the information yield by training set but also to the noise contained in it. To overcome this problem, regression techniques implements different tricks like: a)*dimensionality reduction*, which uses $M(\leq p)$ new regressors calculated from a linear combination of the original ones, or b) *regularisation*, putting a price on the values of the unknown coefficients $\boldsymbol{\beta}$ of model 3.5.

An example of regression technique using dimensionality reduction is Partial Least Squares (PLS), which will be described in detail in Chapter 5, while an example of regression technique using regularisation is Least Absolute Shrinkage and Selection Operator (LASSO), which will be widely discussed in Chapter 6. Both these methods require the setting of one parameter related to the model complexity (i.e. describing the new dimensionality M in PLS and the amount of regularisation in LASSO). The next Section will illustrate the parameter selection procedure of this parameter in the general case.

3.3 Criteria for Selection of Model Complexity

Model complexity should be selected such that the best performance of the chosen identification method is achieved. Now the problem is: which is the right way to evaluate the performance of one method? Through this section it will be described the logic and the steps for selection of the model complexity.

3.3.1 The Bias-Variance *Dilemma*

Considering the training set, it seems reasonable to assume that if model complexity increases the model will better describe the target. Hence, Residual Sum of Squares (RSS) on training data (describing the distance between the reference y and its model prediction) will decrease as model complexity increases.

$$\text{RSS} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\| \tag{3.6}$$

This is a key aspect of the so called internal validation. RSS of eq. (3.6) is expected to have a monotonic decreasing behaviour (see Figure 3.2).

This means that we cannot use it to determine model complexity, since we can always obtain, for sufficiently complex models, zero residuals. The so-determined model usually fails in predicting new data, different from those of the training set. In fact, a too complex model normally fits the reference data but also the noise (overfitting) and is thus not able to generalize the data behaviour properly.

As a consequence, the performance of the learning method has to be determined using independent test data. Suppose the measurement model to be a combination of a deterministic part and a random part due to the noise (zero mean, uncorrelated and homoscedastic noise $\boldsymbol{\epsilon}$ affecting each measure $\boldsymbol{y}$):

$$\boldsymbol{y} = f(\boldsymbol{X}) + \boldsymbol{\epsilon} \qquad\qquad \boldsymbol{\epsilon} \sim N(0, \sigma^2) \tag{3.7}$$

The Mean Square Error (MSE) can be defined for measuring errors between the true value $\boldsymbol{y}_{true}$ and the model prediction $\hat{\boldsymbol{y}}$:

$$\mathrm{MSE}(\hat{\boldsymbol{y}}) = \mathrm{E}\left[(\hat{\boldsymbol{y}} - \boldsymbol{y}_{true})^2\right] \tag{3.8}$$

The equation (3.8) can be divided in two terms, one representing the estimation variance and the other the bias (difference between the expected value of the estimation and the true value $\boldsymbol{y}_{true}$):

$$\mathrm{MSE}(\hat{\boldsymbol{y}}) = \mathrm{E}\left[(\hat{\boldsymbol{y}} - \mathrm{E}\left[\hat{\boldsymbol{y}}\right])^2\right] + \left(\mathrm{E}\left[\hat{\boldsymbol{y}}\right] - \boldsymbol{y}_{true}\right)^2 = \mathrm{Var}(\hat{\boldsymbol{y}}) + \mathrm{Bias}(\hat{\boldsymbol{y}})^2 \tag{3.9}$$

Generally, the variance term increases as model complexity gets higher. This can be explained observing that the more complex the model is, the more is the adherence to the data and thus the sensitivity of the model parameter estimation to the particular realization used to identify them (learning). On the other hand, the bias term decreases as model complexity increases. As a consequence, even if estimates are influenced by noise, its effects tend to be eliminated by averaging different estimates.

Summarizing, the training error tends to decrease when model complexity is increased. If the model overfits the data (too high complexity), it will not generalize well and the estimates will have too high variance. On the other side, if the model is not complex enough, it may underfit the data and have large bias. This brief discussion highlights the *dilemma* of fixing the bias-variance tradeoff and suggests that model complexity should be chosen in such a way to minimize the error on independent test data.

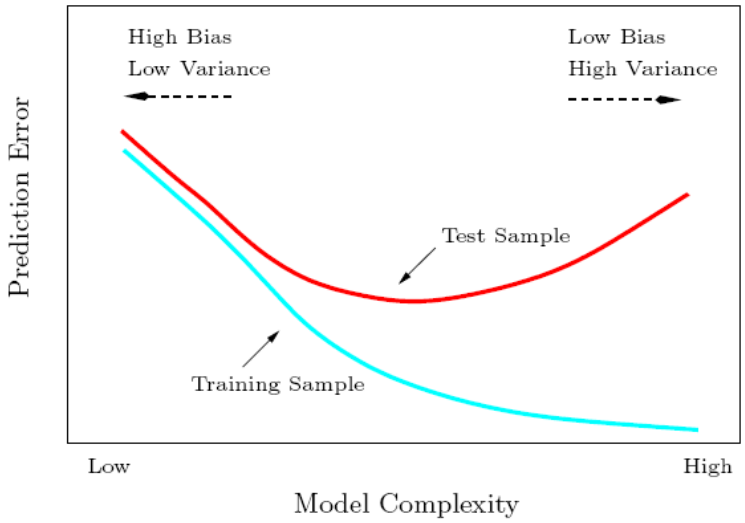

Figure 3.2: Test and training error as a function of model complexity [52].

As shown in Figure 3.2, the prediction error has a monotonic decreasing behaviour as model complexity increases, when calculated on the training set. Hence, it can not be used to select the correct amount of model complexity. In Figure 3.2 the prediction error behaviour when calculated on the test set is also plotted. Usually, it has concave behaviour, due to the bias-variance tradeoff. In this case, having the curve a minimum, it can be used to fix the model complexity. In the next Section a method to construct the test error curve is described.

3.3.2 The Cross-Validation Principle

As far as we observed that the training set is not useful to select the model complexity, another set of data has to be considered (test set). As a consequence, before describing how to calculate the prediction error curve on the test set, we have to discuss how to handle the available data.

In a data-rich situation, the best way to divide the available dataset is in three parts: a training set, a validation set and a test set. The training set is used to fit the model, the validation set is used to select the complexity parameter and the test set is used for assessing the generalization error of the final chosen model (Section 3.3). However, the data are often scarce (as in our case), and thus the previous approach is not applicable.

K-fold cross-validation is a method to estimate test error, using the training set. In particular, K-fold cross-validation splits the data into K parts of approximately equal size. Iteratively, one part is left aside to calculate the test error (using MSE), while the other $K-1$ parts are used to "learn" the coefficients of the model. In this way a test error upon each K-th part is calculated and, averaging these values, an estimation of the test error is obtained.

For example, suppose that a training set of 100 samples is available and that we want to perform 5-fold cross-validation. The 100 samples are randomly and equally divided in 5 parts, each of about 25 samples as shown in Figure 3.3.

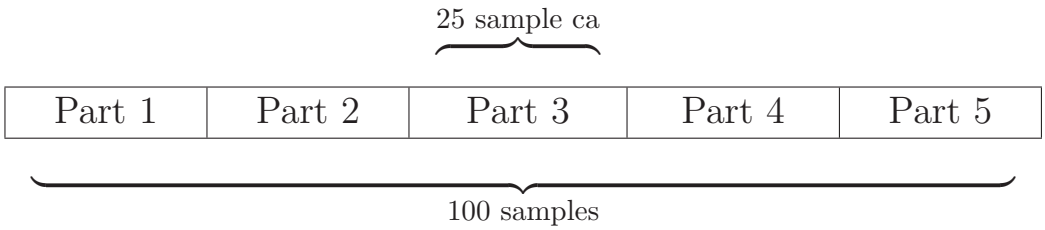

Figure 3.3: example of dataset division for 5-fold cross-validation.

At the first iteration part 2-3-4-5 of the training set are used to estimate the coefficients of the model, obtaining $\hat{\boldsymbol{\beta}}^{-1}$, where the superscript indicates the part that was not used in the learning procedure. The estimated coefficients $\hat{\boldsymbol{\beta}}^{-1}$ are used to predict the reference of part 1 ($\boldsymbol{y}_1$) from the inputs variable of part 1 ($\boldsymbol{X}_1$):

$$\hat{\boldsymbol{y}}_1 = \boldsymbol{X}_1 \hat{\boldsymbol{\beta}}^{-1} \tag{3.10}$$

The RSS is then used to calculated the test error on part 1, where the residuals denote the distance between the model predictions $\hat{\boldsymbol{y}}_1$ and the available reference points $\boldsymbol{y}_1$:

$$\mathrm{RSS}_1 = \sum_{i=1}^{N_1} (\hat{\boldsymbol{y}}_1 - \boldsymbol{y}_1)^2 = \|\hat{\boldsymbol{y}}_1 - \boldsymbol{y}_1\|^2 \tag{3.11}$$

where N_1 is the number of samples included in part 1.

At the second iteration, part 2 is left aside to calculate the RSS_2, using the coefficients estimated from part 1-3-4-5. Similarly, the procedure is iterated for

other three times in order to calculate RSS_3, RSS_4 and RSS_5. These five values of RSS are then averaged in order to estimate the test error.

$$\mathrm{E}_{test} = \overline{\mathrm{RSS}} = \frac{\sum_{i=1}^{5} \mathrm{RSS}_i}{5} \tag{3.12}$$

The whole procedure is repeated for different values of the complexity parameter in order to estimate the test error as a function of the model complexity (see Figure 3.2). Usually, this function has a minimum corresponding to the bias-variance tradeoff.

Cross-validation, averaging the RSS calculated on different datasets, allows also to estimate the confidence interval for the estimated test error. Using the previous example the confidence interval for a given model complexity can be calculated as follows:

$$\mathrm{SD} = \sqrt{\sum_{i=1}^{5} (\mathrm{RSS}_i - \overline{\mathrm{RSS}})^2 \Big/ 5} \tag{3.13}$$

As a consequence, instead of choosing the complexity parameter at the minimum of the test error function, usually "one-standard error" rule is used to choose the model.

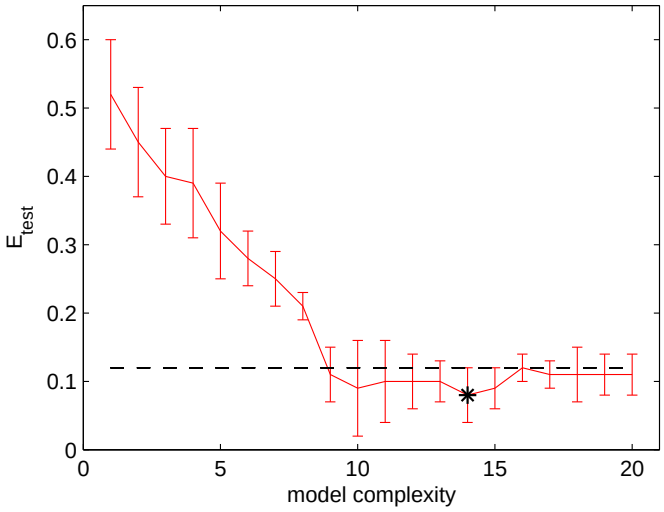

Figure 3.4: Example of test error curve in K-fold cross-validation. Black star is the minimum of the test error. Black dashed line is the upper limit.

The upper limit of the test error is placed at the level correspondent to the minimum of the test error plus its standard deviation. It is represented by the black dashed line in Figure 3.4. The selected model is the most parsimonious one, whose error is less then the limit fixed as explained before. In Figure 3.4 the chosen model complexity is 9.

3.4 Models Assessment (External Validation)

In this Section how to describe the performance of the selected model will be presented.

3.4.1 Principles for External Validation

In the previous Section we described how the training set is used in cross-validation to choose model complexity. Once model complexity is determined the coefficients of the model can be estimated from the whole training set using different techniques. For instance, Ordinary Least Squares (Chapter 4), Partial Least Squares (Chapter 5) or Least Absolute Shrinkage and Selection Operator (Chapter 6). The further step is to determine which learning method best suits for our particular problem. As a consequence, some indicators have to be defined to evaluate model performance on a test set of data. Since the error estimated from data used to learn the coefficients of the model tend to underestimate the real error, the test set must be composed by unseen data, i.e. data that are not used in cross-validation procedure nor in the learning procedure. Hence, this procedure is called external validation.

In formal terms, in external validation the coefficients of the linear model estimated from the training set $\hat{\boldsymbol{\beta}}_{train}$ are used to predict the target of the test data $\boldsymbol{y}_{test}$:

$$\hat{\boldsymbol{y}}_{test} = \boldsymbol{X}_{test}\hat{\boldsymbol{\beta}}_{train} \tag{3.14}$$

the subscript "train" denotes what is calculated from the training set, while the subscript "test" is appended to test set quantities through the equations.

To quantify the prediction quality, different indicators can be taken into account. These key indicators measure in different ways how well the prediction given by the linear model $\hat{\boldsymbol{y}}_{test}$ approximates the reference target $\boldsymbol{y}_{test}$. The most commonly used indicators for evaluating the performance of the models are: Root

Mean Square Error (RMSE), Mean Absolute Difference (MAD), Mean Absolute Relative Difference (MARD), Pearson coefficient of determination (R^2) and FIT. They are formally defined in the following Section.

3.4.2 Key-Indicators Definition

The indicators defined before can be used to evaluate the performance of the identified model on unseen data (i.e. when the test data set is considered). Hence, they allow the comparison between different models.

The Mean Square Error (MSE) was defined as a stochastic quantity in (3.8). However, a realization can be observed as normalized distance between prediction $\hat{\boldsymbol{y}}_{test}$ and reference data $\boldsymbol{y}_{test}$:

$$\text{MSE} = \sum_{i=1}^{N} (y_{itest} - \hat{y}_{itest})^2 \Big/ N \tag{3.15}$$

Root Mean Square Error (RMSE) is the square root of (3.15) and thus has the same units as the quantity being estimated.

The Mean Absolute Difference (MAD) is defined as follows:

$$\text{MAD} = \sum_{i=1}^{N} |y_{itest} - \hat{y}_{itest}| \Big/ N \tag{3.16}$$

which differs from the (3.15) since, instead of summing the square of the difference, its absolute value is summed up.

The Mean Absolute Relative Difference (MARD) is the same as (3.16), but it is an absolute indicator, since every difference $(y_{itest} - \hat{y}_{itest})$ is divided for the reference value y_{itest}:

$$\text{MARD} = \sum_{i=1}^{N} \left| \frac{y_{itest} - \hat{y}_{itest}}{\hat{y}_{itest}} \right| \Big/ N \tag{3.17}$$

While these three key indicators are based only upon the distance between the test reference data $\boldsymbol{y}_{test}$ and its prediction $\hat{\boldsymbol{y}}_{test}$, others like R^2 and FIT measure how much the prediction is a good approximation of the reference variation.

The Pearson correlation coefficient R measures the linear dependence between two variables $\boldsymbol{y}$ and $\boldsymbol{x}$ (test reference $\boldsymbol{y}_{test}$ and prediction $\hat{\boldsymbol{y}}_{test}$ in our case). The

general formula for its calculation is:

$$\mathrm{R} = \frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{N\sum_{i=1}^{N} {x_i}^2 - \left(\sum_{i=1}^{N} x_i\right)^2}\sqrt{N\sum_{i=1}^{N} {y_i}^2 - \left(\sum_{i=1}^{N} y_i\right)^2}} \tag{3.18}$$

The correlation coefficient R ranges from -1 to +1 included. A value of +1 or -1 implies a linear relationship between the two variables. In the case R equals +1 it means that if $\boldsymbol{y}$ increases, $\boldsymbol{x}$ increases too (correlation); in the case R equals -1 a decrease in $\boldsymbol{x}$ will correspond to an increase of $\boldsymbol{y}$ (anticorrelation). A value of 0 implies that there is no correlation between the variables.

The square of correlation coefficient, R^2, ranges from 0 to +1. Hence, it does not distinguish negative from positive correlation. This indicator turns out to be useful when interested to the connection between the variables and not to the sign of the relation.

A key mathematical property of the correlation coefficient is that it is invariant to changes in location and scale, i.e. if one of the variables is transformed linearly as $a + bx$ (with a and b constants) the correlation coefficient does not change its value. This can be useful to determine if the prediction $\hat{\boldsymbol{y}}_{test}$ has the same fluctuations of the reference $\boldsymbol{y}_{test}$, without having the same scale. In this case R^2 would assume a high value (good correlation), even if the distance between the reference and test sample is high, causing bad values for RMSE, MAD or MARD.

Finally, FIT quantifies the percentage of the output variation that is explained by the model. The total variance of the target ($\boldsymbol{SS_{tot}}$) can be seen as the sum of two terms: the variance explained by the prediction $\boldsymbol{SS_{reg}}$(variance of the prediction referred to the reference mean) and the variance not explained by the prediction $\boldsymbol{SS_{err}}$ (residual sum of squares).

$$\boldsymbol{SS_{tot}} = \boldsymbol{SS_{reg}} + \boldsymbol{SS_{err}} \tag{3.19}$$

where:

$$\begin{aligned} \boldsymbol{SS_{tot}} &= \| \boldsymbol{y}_{test} - mean(\boldsymbol{y}_{test})\| \\ \boldsymbol{SS_{reg}} &= \| \hat{\boldsymbol{y}}_{test} - mean(\boldsymbol{y}_{test})\| \\ \boldsymbol{SS_{err}} &= \| \hat{\boldsymbol{y}}_{test} - \boldsymbol{y}_{test}\| \end{aligned} \tag{3.20}$$

Hence the ratio between the explained variance and the total one, which correspond to the definition of FIT, is:

$$\text{FIT} = \frac{\boldsymbol{SS_{reg}}}{\boldsymbol{SS_{tot}}} = 1 - \frac{\boldsymbol{SS_{err}}}{\boldsymbol{SS_{tot}}} \tag{3.21}$$

then, substituting (3.20) into (3.21) and expressing in percent:

$$\text{FIT} = 100 * \left(1 - \frac{\| \hat{\boldsymbol{y}}_{test} - \boldsymbol{y}_{test} \|}{\| \boldsymbol{y}_{test} - mean(\boldsymbol{y}_{test}) \|}\right) \tag{3.22}$$

If the estimated model has a good performance, then RMSE, MAD and MARD will assume low values, while R^2 and FIT will take high values.

While, by visual inspection of the estimated profiles versus reference data, one can guess which model has the best performance, these indicators will allow a quantitative assessment of how much one method works better than the others in estimating linear model for regression.

3.5 Conclusions

This Chapter presented an introduction to the regression problem. First of all the notation used throughout this thesis was introduced. Then, starting with the consideration that algorithms dealing with high-dimensional data suffer from the curse of dimensionality and overfitting, a general introduction to the methods trying to solve these problems was also presented. As these algorithms usually require the setting of a parameter to adjust the model complexity, the general estimation procedure of this parameter was illustrated. In particular it was described how to construct a test error curve as a function of model complexity, using K-fold cross-validation. This curve is used to fix the model complexity. Hence, at the end of this Chapter, some indicators for the performance comparison of different models were presented.

As a consequence this Chapter has described the different steps to solve the regression problem: using the internal validation, we may check that as the model complexity increases, it better describes the training set; then, the cross-validation can be used to set the correct level of model complexity; finally, the external validation allows to compare different models. This whole procedure is used in this thesis to evalute the performance of the regression methods presented in the next Chapters, when applied to the Solianis Multisensor data. In particular,

different estimation methods for regression will be described in the next Chapters 4, 5 and 6, while the results of their application to the data of Chapter 7 will be presented in Chapter 8.

Chapter 4

Ordinary Least Squares

While in the previous Chapter we defined and stated the theoretical aspects of the notations that will be used throughout this thesis, Chapters 4, 5 and 6 will present some techniques for estimating linear regression models. Then, Chapter 8 will discuss the application of these methods to Multisensor data.

The goal of regression is to describe a target variable at the i-th time instant y_i given a set of p available input variables $\boldsymbol{X}_{ip}$ (1 x p) at the same i-th time instant. The previous sentence can be expressed in mathematical terms as:

$$y_i = f(\boldsymbol{X}_{ip}) \tag{4.1}$$

where f is any function modeling the relationship between the input variables $\boldsymbol{X}_{ip}$ and the target y_i. If the model is linear, eq. (4.1) turns into:

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j = \beta_0 + \boldsymbol{X}_{ip}\boldsymbol{\beta} \tag{4.2}$$

In model (4.2), p input variables $\boldsymbol{X}_{ip}$ are weighted using the p time-constant coefficients $\beta_1 \dots \beta_p$, while the coefficient β_0 plays the role of describing the data offset. These coefficients are unknown and represent the parameters that completely define and characterise the linear model. The p inputs in $\boldsymbol{X}_{ip}$ are also called predictors, since the target is fitted by a linear combination of them.

Consider now a training set consisting of a reference vector $\boldsymbol{y}$ (N x 1), containing N samples of the target at different time instants, and the corresponding input matrix $\boldsymbol{X}$ (N x p), whose rows represent the input variables $\boldsymbol{X}_{ip}$ at the same time instants, while each column $\boldsymbol{X}_{jN}$ contains all the samples referred to the j-th variable (see Section 3.1).

The most easy and well-known method for finding an estimate of the column vector $\boldsymbol{\beta} = [\beta_0, \beta_1 \ldots, \beta_p]$, $\hat{\boldsymbol{\beta}}$, given the reference vector $\boldsymbol{y}$ and the corresponding inputs $\boldsymbol{X}$, is Ordinary Least Squares (OLS). OLS makes no assumption about the validity of the model, but simply finds the best set of parameters $\boldsymbol{\beta}$ by adjusting them in order to minimize the error in the description of the training data. The errors are quantified through the Residual Sum of Squares (RSS) between the target and the model estimates. This function has a quadratic form, allowing a closed form solution for the model parameters.

This Chapter will present the characterization of the OLS estimation procedure. Then, with the support of two simple examples, advantages and drawbacks of OLS will be shown.

4.1 Definition of OLS

The model (4.2) for a training set can be written in matrix form, adding a column of ones at the beginning of the input matrix $\boldsymbol{X}$, obtaining matrix $\boldsymbol{X}^{(p+1)}$; this allow a more compact notation including the bias parameter β_0:

$$\boldsymbol{y} = \begin{bmatrix} 1 & \boldsymbol{X}_{1p} \\ \vdots & \vdots \\ 1 & \boldsymbol{X}_{Np} \end{bmatrix} \boldsymbol{\beta} = \boldsymbol{X}^{(p+1)}\boldsymbol{\beta} \tag{4.3}$$

alternatively, the bias parameter β_0 could be dropped if a preliminary centring of the data is performed. In order to simplify the notation, for the time being any superscript to the matrix $\boldsymbol{X}$ will be omitted and the matrix itself will represents both cases (i.e. including or not the bias parameter β_0), and its form will be clear from the context.

OLS determines the estimate $\hat{\boldsymbol{\beta}}$ by minimizing the Residual Sum of Squares (RSS), where the residuals denote the distance between the model predictions (4.2) and the available reference points y_i:

$$\mathrm{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \tag{4.4}$$

that can be written in matrix form as:

$$\mathrm{RSS}(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{4.5}$$

where $\boldsymbol{X}$ is the same as in eq. (4.3). It is easy to see that RSS is a quadratic function of the unknown parameter vector $\boldsymbol{\beta}$. Minimizing RSS in (4.5) can thus be done by setting to zero the first derivative of (4.5) with respect to $\boldsymbol{\beta}$:

$$\frac{\partial \text{RSS}}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{4.6}$$

$$\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = 0 \tag{4.7}$$

The matrix equation (4.7) collects the so called *normal equations*. If the matrix $\boldsymbol{X}^T\boldsymbol{X}$ is not singular, a close formula for the solution $\hat{\boldsymbol{\beta}}$ can be computed. In this case, the solution is unique and takes the form:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{4.8}$$

The estimated parameter vector $\hat{\boldsymbol{\beta}}$ could then be placed into (4.3) to obtain an estimate of the target $\hat{\boldsymbol{y}}$, also termed "model prediction":

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{4.9}$$

Once the model parameters $\hat{\boldsymbol{\beta}}$ are estimated from the training set, the linear model of eq. (4.3) can thus be used to predict unseen data through a linear combination of the inputs.

4.2 Properties of OLS

This Section will present some insight into the OLS technique for estimating linear regression models.

4.2.1 Geometrical Properties

OLS has also a geometrical interpretation as seen in Figure 4.1, that represents the case of two different input variables $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$, each having three time samples.

Each input $\boldsymbol{X}_{j3}$ could be considered as a vector in the three-dimensional space, where the vector $\boldsymbol{y}$ is also defined.

Supposing that $\boldsymbol{X}$ has full column rank and since the number of different inputs (two) is smaller than the number of data points of the training dataset

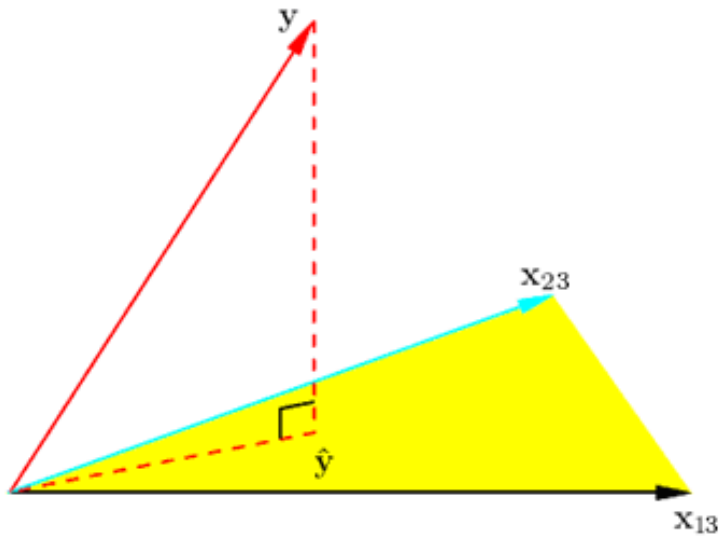

Figure 4.1: geometrical interpretation of OLS. Target vector $\boldsymbol{y}$, estimation of target vector $\hat{\boldsymbol{y}}$, input vectors $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$ and in yellow the linear subspace S [1].

(three), the inputs span a linear subspace S (a plane in this case), called columns space of $\boldsymbol{X}$.

Using the linear model, the estimation $\hat{\boldsymbol{y}}$ could be any linear combination of the inputs $\boldsymbol{X}_{j3}$. For this reason the estimate could lie anywhere in the bi-dimensional subspace S and the RSS represents the squared euclidean distance between the reference $\boldsymbol{y}$ and its estimation $\hat{\boldsymbol{y}}$.

Since OLS adjusts the parameters $\boldsymbol{\beta}$ of the linear model to minimize the RSS, the OLS model prediction $\hat{\boldsymbol{y}}$ is the particular vector lying in the subspace S, which is the closest as possible to the reference $\boldsymbol{y}$. For this reason, $\hat{\boldsymbol{y}}$ corresponds to the orthogonal projection of $\boldsymbol{y}$ onto the subspace S, which is described mathematically by:

$$\boldsymbol{X}^T(\boldsymbol{y} - \hat{\boldsymbol{y}}) = 0 \tag{4.10}$$

(4.10) represents the orthogonality condition of the vector $(\boldsymbol{y} - \hat{\boldsymbol{y}})$ with respect to the subspace S defined by $\boldsymbol{X}$.

Substituting (4.9) into (4.10) one gets:

$$\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) = 0 \tag{4.11}$$

This condition corresponds to the normal equations in (4.7) and is satisfied by the OLS solution.

4.2.2 Singularity Condition

If the regressors $\boldsymbol{X}_{jN}$ are not linearly independent, $\boldsymbol{X}^T\boldsymbol{X}$ is singular and can not be inverted to calculate the parameters in (4.8) yielding to a not uniquely defined $\hat{\boldsymbol{\beta}}$. However, the multiple solutions are still the projection of $\boldsymbol{y}$ onto the column space of $\boldsymbol{X}$, though there are more ways to express this projection, as there are more ways to define the subspace S.

The linear dependency of the columns of $\boldsymbol{X}$ is a consequence that one or more qualitative inputs $\boldsymbol{X}_{jN}$ are coded in a redundant fashion. If a couple of columns are nearly to be linearly dependent, the correlation between the two variables is high and the matrix $\boldsymbol{X}$ is not full rank. In this case the problem of inverting $\boldsymbol{X}^T\boldsymbol{X}$ is ill-conditioned leading to low accuracy of the estimated vector $\hat{\boldsymbol{\beta}}$. A typical solution for this problem is recoding and/or dropping redundant columns in $\boldsymbol{X}$. Other methods, as explained in the next Chapters, provide a regularization term to cope with this low rank issue.

The most common method to recode redundant columns is the QR decomposition of $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R} \tag{4.12}$$

where $\boldsymbol{Q}$ is an orthogonal matrix ($\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$) of dimension ($N$ x p), while $\boldsymbol{R}$ is an upper triangular matrix of dimension (p x p). Without going into the details, these matrices are obtained by recursive orthogonalisation of the inputs, leading to an orthonormal basis for the column space of $\boldsymbol{X}$.

The QR decomposition is used to transform model (4.2) in a simpler, more stable triangular system. From (4.7) we have:

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}^T\boldsymbol{y} \tag{4.13}$$

then, substituting (4.12) in (4.13) we get:

$$\begin{aligned} \boldsymbol{R}^T\underbrace{\boldsymbol{Q}^T\boldsymbol{Q}}_{\boldsymbol{I}}\boldsymbol{R}\boldsymbol{\beta} &= \boldsymbol{R}^T\boldsymbol{Q}^T\boldsymbol{y} \\ \boldsymbol{R}\boldsymbol{\beta} &= \boldsymbol{Q}^T\boldsymbol{y} \end{aligned} \tag{4.14}$$

Using QR decomposition the OLS solution is given by:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \boldsymbol{R}^{-1}\boldsymbol{Q}^T\boldsymbol{y} \\ \hat{\boldsymbol{y}} &= \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{y} \end{aligned} \tag{4.15}$$

The number of the estimated coefficients that are not zero is equal to the rank of matrix $\boldsymbol{X}$ and the solution coincide to (4.8) and (4.9) if $\boldsymbol{X}$ has full column rank.

4.2.3 Statistical Properties

Now suppose the measurement model to be a combination of a deterministic part (linear combination of regressors) and a random part (zero mean, uncorrelated and homoscedastic noise ϵ_i affecting each measure y_i):

$$\begin{aligned} y_i &= \boldsymbol{X}_i\boldsymbol{\beta} + \epsilon_i \qquad\qquad \boldsymbol{\epsilon} \sim N(0, \sigma^2) \\ y_i &= y_{true} + \epsilon_i \end{aligned} \tag{4.16}$$

The Mean Square Error (MSE) of an estimation $\hat{\boldsymbol{y}}$ in estimating the true value $\boldsymbol{y}_{true}$ is:

$$\mathrm{MSE}(\hat{\boldsymbol{y}}) = \mathrm{E}\left[(\hat{\boldsymbol{y}} - \boldsymbol{y}_{true})^2\right] \tag{4.17}$$

The equation (4.17) can be divided in two terms, one representing the estimation variance and the other the bias (difference between the expected value of the estimation and the true value $\boldsymbol{y}_{true}$):

$$\mathrm{MSE}(\hat{\boldsymbol{y}}) = \mathrm{E}\left[(\hat{\boldsymbol{y}} - \mathrm{E}\left[\hat{\boldsymbol{y}}\right])^2\right] + (\mathrm{E}\left[\hat{\boldsymbol{y}}\right] - \boldsymbol{y}_{true})^2 = \mathrm{Var}(\hat{\boldsymbol{y}}) + \mathrm{Bias}(\hat{\boldsymbol{y}})^2 \tag{4.18}$$

The Gauss-Markov theorem tells us that the OLS estimator of the parameter vector $\boldsymbol{\beta}$ has the smallest variance among all linear unbiased estimators, namely it presents the lowest possible MSE of the estimate. For this reason OLS is also known as Best Linear Unbiased Estimator (BLUE).

However, it may well exist a biased estimator with smaller MSE. Since this estimator is biased it must have a very small variance in order to have smaller MSE than OLS (that is unbiased). Any method that shrinks or sets to zero some of the coefficients of the linear combination may result in a biased estimate but in a lower variance.

4.3 Implementation of OLS

The OLS estimator can be implemented by the following (Matlab-like notation) pseudo-code:

load X, y

$\left.\begin{array}{l} \hat{\beta} \leftarrow \text{inv}(X^T X) X^T y; \\ \text{(or using QR decomposition)} \\ \hat{\beta} \leftarrow X \backslash y; \end{array}\right\}$ parameter estimation

$\hat{y} \leftarrow X\hat{\beta}$ } target estimation

The Matlab code used in this thesis to implement OLS is reported in Appendix A.1.

4.4 Tutorial Examples

In this Section we will present a couple of examples that will help us to highlight the features of OLS in estimating linear regression models.

4.4.1 Example 1 (Diabetes data)

The data for this first example are taken from [53]. They examine the correlation between a number of clinical measures in diabetes patients and a measure of "diabetes progression" (**dp**). To model this connection, 442 samples referred to different subjects are available.

The data have been randomly split in two sets: one form the training set, which is used to learn the parameters of the model and the other is used to estimate the prediction error. In particular, the training set contains two thirds (294 sample) of the total available samples.

Considering the linear model (4.2):

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}$$

the measure of "diabetes progression" represents our target vector $\boldsymbol{y}$:

$$\boldsymbol{y} = \left[\begin{array}{c} dp(1) \\ dp(2) \\ \vdots \\ dp(294) \end{array}\right] \Bigg\} \textbf{dp}$$

while the clinical measures, including **age**, **sex**, body mass index (**bmi**), average blood pressure (**bp**), and six serum measurements (**sm1**,...,**sm6**), represent our input variables $\boldsymbol{X}_{jN}$, which compose the matrix $\boldsymbol{X}$:

$$\boldsymbol{X} = \left[\begin{array}{ccc|c|ccc} age(1) & sex(1) & | & \overline{bmi(1)} & | & bp(1) & \ldots & sm6(1) \\ age(2) & sex(2) & | & bmi(2) & | & bp(2) & \ldots & sm6(2) \\ \vdots & \vdots & | & \vdots & | & \vdots & & \vdots \\ age(294) & sex(294) & | & \underline{bmi(294)} & | & bp(294) & \ldots & sm6(294) \end{array}\right] \begin{array}{l} \Rightarrow subject\ \ 1 \\ \Rightarrow subject\ \ 2 \\ \\ \Rightarrow subject\ \ 294 \end{array}$$

$$\Downarrow$$

bmi

Thus each column of the matrix $\boldsymbol{X}$ contains one of the ten input variables, while the rows correspond to 294 samples referred to different subjects, forming the training set.

Our aim is to determine the coefficients vector $\boldsymbol{\beta}$ that describes the influence of the clinical variables upon the "diabetes progression". Since the training data have been standardised (mean=0 and standard deviation=1), the offset parameter β_0 can be dropped and the unknown vector $\boldsymbol{\beta}$ has dimension ten:

$$\boldsymbol{\beta} = \left[\begin{array}{c} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{10} \end{array}\right]$$

First of all it can be useful to calculate the correlation between the different input variables.

	sex	bmi	bp	sm1	sm2	sm3	sm4	sm5	sm6
age	0.174	0.185	0.335	0.260	0.219	-0.075	0.204	0.271	0.302
sex	-	0.088	0.241	0.035	0.143	-0.379	0.332	0.150	0.208
bmi	-	-	0.395	0.250	0.261	-0.367	0.414	0.446	0.389
bp	-	-	-	0.242	0.186	-0.179	0.258	0.393	0.390
sm1	-	-	-	-	0.897	0.052	0.542	0.516	0.326
sm2	-	-	-	-	-	-0.196	0.660	0.318	0.291
sm3	-	-	-	-	-	-	-0.738	-0.399	-0.274
sm4	-	-	-	-	-	-	-	0.618	0.417
sm5	-	-	-	-	-	-	-	-	0.465

Table 4.1: correlation between the different input variables. Highlighter the most elevated correlations.

Sm1 and **sm2** are the most correlated variables, while the variables **sm3** and **sm4** show a high negative correlation. This can be checked visually plotting the variables together (see Figure 4.2)

The OLS coefficients have been estimated using the closed form solution (4.8), obtaining the following coefficients:

	age	sex	bmi	bp	sm1	sm2	sm3	sm4	sm5	sm6
coeff	0.009	-0.167	0.318	0.190	-0.765	0.498	0.157	0.184	0.485	0.063

Table 4.2: Estimated OLS coefficients.

The highest coefficient absolute value is the one associated with the variable **sm1**. However, its contribute to the estimation is not so high, since the coefficient of **sm2** (the most correlated variable with **sm1**) tends to compensate the **sm1** effect. This phenomenon occurs when OLS deals with highly correlated variables: their coefficients tend to grow large in opposite directions compensating each others.

This happens with **sm3** and **sm4** too, but in this case the coefficients are both positive given the negative correlation between the variables.

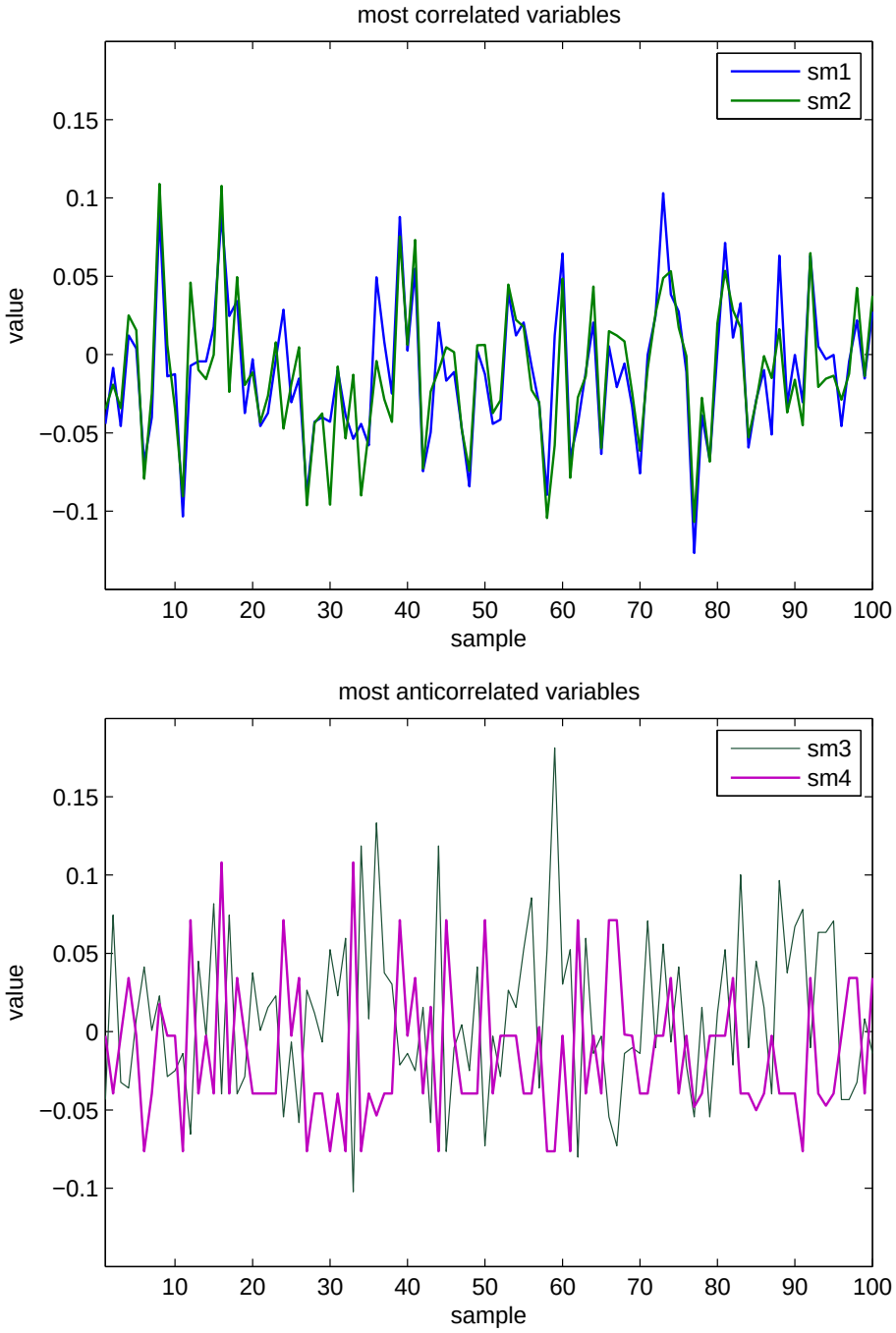

Figure 4.2: *Top*: Plot of the most correlated variables **sm1**(blue) and **sm2** (green). *Bottom*: Plot of the most anticorrelated variables **sm3** (grey) and **sm4** (magenta). Only the first 100 samples are shown.

4.4.2 Example 2 (Simulated data)

The reference data for this second example were generated by simulating glucose profiles with different time trends. From these profiles, twenty-seven input variables were obtained simulating Multisensor data, therefore exhibiting high correlation and including confounding processes such as body temperature. In particular, twenty-five signals were chosen for modeling MHz data (**MHz1**,..., **MHz25** which contain glucose information and are highly correlated), one signal was selected to mimic optical data (**Opt** very noisy and measures different kinds of confounding processes), and finally one signal for the effects of body temperature (**Temp**) on MHz signals.

The training data were simulated by using a sequence of three glucose profiles, each having a length of eight hours and showing one or two glycaemic peaks. While input variables had an elevated sampling frequency (3 sample/minute), reference data were collected approximately every 15 minutes.

Considering the linear model (4.2):

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}$$

and indicating with t_{end}, the time instant associated with the last available sample, the target vector y contains the simulate glucose samples (**SG**), collected as explained before:

$$\boldsymbol{y} = \left.\begin{bmatrix} SG(0) \\ SG(16.33) \\ SG(32.66) \\ \vdots \\ SG(t_{end}) \end{bmatrix}\right\} \textbf{SG}$$

while matrix $\boldsymbol{X}$ is formed using the Multisensor data (MHz-like, optical and temperature data); each column of the matrix $\boldsymbol{X}$ contains one of the input variables, while the rows correspond to the time-samples. The number of available input samples is greater than the number of reference samples, since the first are collected with higher frequency. However, only the input samples that have a

corresponding reference can be included in matrix $\boldsymbol{X}$:

$$\boldsymbol{X} = \begin{bmatrix} MHz1(0) & \dots & \overline{MHz25(0)} & Temp(0) & Opt(0) \\ MHz1(16.33) & \dots & MHz25(16.33) & Temp(16.33) & Opt(16.33) \\ MHz1(32.66) & \dots & MHz25(32.66) & Temp(32.66) & Opt(32.66) \\ \vdots & & \vdots & \vdots & \vdots \\ MHz1(t_{end}) & \dots & \underline{MHz25(t_{end})} & Temp(t_{end}) & Opt(t_{end}) \end{bmatrix} \begin{matrix} \Rightarrow time & 0 \\ \Rightarrow time & 16.33 \\ \Rightarrow time & 32.66 \\ & \\ \Rightarrow time & t_{end} \end{matrix}$$

$$\Downarrow$$

MHz25

Our aim is to determine the coefficients vector $\boldsymbol{\beta}$ that describes the influence of the simulated Multisensor data upon the glucose profile. Since the training data have been standardized (mean=0, sd=1), the offset parameter β_0 can be dropped and the unknown vector $\boldsymbol{\beta}$ has dimension twenty-seven:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{27} \end{bmatrix}$$

Since some MHz-simulated data were linear dependent, the matrix $\boldsymbol{X}$, containing the input variables, did not present full column rank, in particular its rank was 22 (< 27 columns). Consequently, the OLS solution was not unique and the closed form solution (4.8) did not exist. To solve this low rank issue the OLS estimation was calculated using QR decomposition.

Table 4.3 shows the estimated OLS coefficients.

	MHz1	**MHz2**	**MHz3**	**MHz4**	**MHz5**	**MHz6**	**MHz7**	**MHz8**	**MHz9**
coeff	39.974	0.000	0.000	0.000	0.000	3.381	-8.656	-1.951	-6.151
	MHz10	**MHz11**	**MHz12**	**MHz13**	**MHz14**	**MHz15**	**MHz16**	**MHz17**	**MHz18**
coeff	-14.849	40.897	26.916	-24.356	2.557	-5.082	-23.234	54.732	-53.089
	MHz19	**MHz20**	**MHz21**	**MHz22**	**MHz23**	**MHz24**	**MHz25**	**Temp**	**Opt**
coeff	-4.424	-27.901	0.000	1.410	4.014	1.543	-12.167	2.397	0.071

Table 4.3: Estimated OLS coefficients.

As discussed for the previous example, the coefficients associated to high correlated variables (**MHz**) show elevate magnitude with opposite signs.

In the following Figure two variables that compensate each others are shown.

Figure 4.3: Highly correlated variables of second simulated glucose profile: **MHz12** (blue) vs. **MHz13** (green) - first 250 minutes only.

4.5 Concluding Remarks

OLS is the most popular estimation method for linear regression models. The OLS solution is mathematically achieved by minimizing the residual sum of squares. This loss function has a quadratic form that allows to calculate the solution in a closed form in a very efficient way.

All these advantages make OLS an attractive estimator for linear models. However, it can lead to unsatisfactory results in several cases. First of all, the solution can not be calculated or could be calculated only with a small precision, when there is a strong correlation between two, or more, inputs variables. In this case, the most common solution is to remove the redundant variables. In addition, it may happen that a coefficient associated with a variable results very large, while another coefficient (associated with a variable correlated with the previous one) compensates it in the opposite direction (canceling the first variable's effect). As a consequence, the information carried by one variable is deleted by the other.

However, it may happen that the noise contained in each variable, instead of being canceled, adds up leading to an increased variance in the estimation.

Chapter 5

Partial Least Squares

As said in Chapter 3, algorithms for solving linear regression generally suffer from overfitting when they deal with high-dimensional datasets. This is the case of the OLS method described in the previous Chapter.

There are different kinds of estimation methods that try to deal with the overfitting problem. In this Chapter, we will present a regression technique based on the dimensionality reduction, i.e. it uses M ($\leq p$) new regressors $\boldsymbol{z}_k$ calculated from a linear combination of the original ones $\boldsymbol{X}_{jN}$. These new regressors are combined, using a linear model, to estimate the target $\boldsymbol{y}$:

$$\hat{\boldsymbol{y}} = \boldsymbol{Z}\boldsymbol{\theta} \tag{5.1}$$

where $\boldsymbol{Z}$ is a (N x M) matrix, whose columns contain the regressors $\boldsymbol{z}_k$ and $\boldsymbol{\theta}$ is the M dimensional vector of the related coefficients, which have to be estimated along with the new regressors $\boldsymbol{z}_k$.

This technique, which goes under the name of Partial Least Squares (PLS), can be implemented in several ways, depending on how the linear combinations are constructed and is usually suitable when data in matrix $\boldsymbol{X}$ include a large number of very correlated inputs. Since the value of M, describing how many new predictors are used in the regression problem, is fixed by the user, these methods can allow to tune model complexity according to the problem.

As the new regressors $\boldsymbol{z}_k$ are linearly related to the original ones, the model of (5.1) can still be expressed as a function of the original inputs $\boldsymbol{X}_{jN}$:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}^{PLS} \tag{5.2}$$

where $\hat{\boldsymbol{\beta}}^{PLS}$ is the vector of coefficients estimated using the PLS technique.

This Chapter will present the characterisation of PLS construction of the new regressors $\boldsymbol{z}_k$ and the related coefficients estimation procedure. Then, with the support of the same two simple examples of Chapter 4, advantages and drawbacks of PLS will be shown and the results will be compared with those obtained by using OLS. Part of this material can be referred to [52].

5.1 Definition of PLS

In this Section we will described the PLS estimation procedure. First its classical derivation will be present, followed by an alternative implementation.

5.1.1 Derivation of the PLS estimator

Consider a training set consisting of a reference vector $\boldsymbol{y}$ (N x 1), containing N samples of the target at different time instants, and the corresponding input matrix $\boldsymbol{X}$ (N x p), whose rows represent the input variables $\boldsymbol{X}_{ip}$ at the same time instants, while each column $\boldsymbol{X}_{jN}$ contains all the samples referred to the j-th variable (see Section 3.1).

Since PLS is not scale invariant, i.e. the estimates depend on the scaling of the inputs, before starting the construction of the M new regressors $\boldsymbol{z}_1$, $\boldsymbol{z}_2$,..., $\boldsymbol{z}_M$, the input variables $\boldsymbol{X}_{jN}$ have to be normalized, i.e. zero mean and unitary variance. To avoid the introduction of a new symbol below, we assume that each input variable $\boldsymbol{X}_{jN}$ is normalized.

As mentioned before, PLS iteratively constructs a set of linear combinations of the inputs, using both $\boldsymbol{X}$ and $\boldsymbol{y}$. For this construction the original inputs $\boldsymbol{X}_{jN}$ are weighted according to their univariate effect on $\boldsymbol{y}$.

Since PLS is an iterative procedure in which the input variables $\boldsymbol{X}_{jN}$ are updated at every iteration, it is useful to add a superscript to the notation indicating the iteration number. Hence, $\boldsymbol{X}_{jN}^{(k)}$ represent the j-th input variables at the k-th iteration and $\boldsymbol{X}_{jN}^{(0)}$ correspond to the original input variables $\boldsymbol{X}_{jN}$. The same superscript is added to the estimated target variable $\hat{\boldsymbol{y}}$, as it is also updated at every iteration. In particular, at first, $\hat{\boldsymbol{y}}$ equals the mean of the reference, represented using $\bar{y}$ ($\hat{\boldsymbol{y}}^{(0)} = \bar{y}$). Then, the estimate $\hat{\boldsymbol{y}}$ is adjusted during each iteration,

in which a new direction $\boldsymbol{z}_k$ is constructed.

PLS begins by computing the correlation $\hat{\varphi}_{1j}$ between the current input variables $\boldsymbol{X}_{jN}^{(0)}$ and the reference $\boldsymbol{y}$:

$$\hat{\varphi}_{1j} = \boldsymbol{X}_{jN}^{(0)} \boldsymbol{y} \tag{5.3}$$

where the first value of the subscript of $\hat{\varphi}$ indicates the iteration, while the second stands for the variable j.

Each current input variable $\boldsymbol{X}_{jN}^{(0)}$ is weighted by its corresponding correlation $\hat{\varphi}_{1j}$ in (5.3) to construct the first "derived" input $\boldsymbol{z}_1$ (N x 1):

$$\boldsymbol{z}_1 = \sum_{j=1}^{p} \hat{\varphi}_{1j} \boldsymbol{X}_{jN}^{(0)} \tag{5.4}$$

where $\boldsymbol{z}_1$ is called the first partial least squares direction.

Subsequently, the reference $\boldsymbol{y}$ is regressed on $\boldsymbol{z}_1$, obtaining the scalar coefficient $\hat{\theta}_1$:

$$\hat{\theta}_1 = \frac{\boldsymbol{z}_1^T \boldsymbol{y}}{\boldsymbol{z}_1^T \boldsymbol{z}_1} \tag{5.5}$$

which is the OLS solution to the regression problem where $\boldsymbol{y}$ is the reference and $\boldsymbol{z}_1$ is the (only) input variable, compare eq. (5.5) with eq. (4.8).

The coefficient $\hat{\theta}_1$ in (5.5) is used as the multiplier of $\boldsymbol{z}_1$ in (5.4) to update the reference estimate $\hat{\boldsymbol{y}}$:

$$\hat{\boldsymbol{y}}^{(1)} = \hat{\boldsymbol{y}}^{(0)} + \hat{\theta}_1 \boldsymbol{z}_1 \tag{5.6}$$

Using the coefficient $\hat{\theta}_1$, each current input variables $\boldsymbol{x}_{jN}^{(0)}$ is orthogonalized with respect to $\boldsymbol{z}_1$, i.e. its contribution to $\boldsymbol{z}_1$ is subtracted from it:

$$\boldsymbol{X}_{jN}^{(1)} = \boldsymbol{X}_{jN}^{(0)} - \gamma_j \boldsymbol{z}_1 \quad \text{where} \quad \gamma_j = \frac{\boldsymbol{z}_1^T \boldsymbol{X}_{jN}^{(0)}}{\boldsymbol{z}_1^T \boldsymbol{z}_1} \tag{5.7}$$

Then, the process continues until $M \leq p$ directions have been obtained.

Since the $\boldsymbol{z}_k$'s, with $k = [1, 2, \ldots, M]$, are linear in the original inputs (see eq. (5.4) and (5.7)), the reference estimation after M steps $\hat{\boldsymbol{y}}^{(M)}$ can be also computed as:

$$\hat{\boldsymbol{y}}^{(M)} = \boldsymbol{X} \hat{\boldsymbol{\beta}}^{PLS} \tag{5.8}$$

recovering the coefficients $\hat{\boldsymbol{\beta}}^{PLS}$ from the sequence of PLS transformation.

As for OLS, once the coefficients $\hat{\boldsymbol{\beta}}^{PLS}$ are estimated from the training set, they can be used in the linear model to predict unseen data through a linear

combination of the inputs. It is worth noting that, if $M = p$ (i.e. the number of the PLS directions $\boldsymbol{z}_k$ equals the number of the original input $\boldsymbol{X}_{jN}$), the PLS solution is equivalent to the OLS estimates.

5.1.2 Alternative implementation of PLS

Other algorithms have been developed allowing a direct estimation of the coefficients $\hat{\boldsymbol{\beta}}^{PLS}$. Without going into details, it is worth mentioning the SIMPLS algorithm [54] based on the input approximation using score and loading matrices:

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{X}_l^T + \boldsymbol{E} \tag{5.9}$$

In this case, $\boldsymbol{Z}$ is the (N x M) matrix of the M extracted score vectors (PLS directions $\boldsymbol{z}_k$), the (p x M) matrix $\boldsymbol{X}_l$ represents the matrix of loadings and $\boldsymbol{E}$ the matrix of residuals. The approximation of the target is like in (5.1):

$$\boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\theta} + \boldsymbol{e} \tag{5.10}$$

The key of this algorithm is that it directly estimates a matrix of weights $\boldsymbol{W}$, representing the relationship between the PLS direction in $\boldsymbol{Z}$ with the original matrix $\boldsymbol{X}$:

$$\boldsymbol{X}\boldsymbol{W} = \boldsymbol{Z} \tag{5.11}$$

Then, substituting (5.11) into (5.10), one gets:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{\theta} + \boldsymbol{e} \tag{5.12}$$

the approximation of the reference $\boldsymbol{y}$ is directly related to the original inputs $\boldsymbol{X}$. Hence, ignoring the contribution of the residual matrix $\boldsymbol{e}$, the PLS reference estimation $\hat{\boldsymbol{y}}$ is obtained as:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{\theta} \tag{5.13}$$

Comparing (5.13) with (5.8), one gets:

$$\hat{\boldsymbol{\beta}}^{PLS} = \boldsymbol{W}\boldsymbol{\theta} \tag{5.14}$$

Hence, the matrix of weight $\boldsymbol{W}$ allows to calculate directly the estimation of the PLS coefficients $\hat{\boldsymbol{\beta}}^{PLS}$, without recovering them from the sequence of PLS transformation by a back tracking. In fact, $\boldsymbol{W}$ describes how to combine the coefficients of the new regressors $\boldsymbol{z}_k$, contained in the matrix $\boldsymbol{\theta}$.

5.2 Geometrical Properties of PLS

It can be shown that PLS seeks directions that have high variance and high correlation with the response variable. Hence, the k-th PLS direction solves the problem:

$$\max_{\alpha} corr^2(\boldsymbol{y}, \boldsymbol{X}\alpha)var(\boldsymbol{X}\alpha) \tag{5.15}$$

with the two constraints:

$$\|\alpha\| = 1 \tag{5.16}$$

$$\alpha \boldsymbol{S} \hat{\varphi}_l = 0 \quad \text{with} \quad l = 1, 2, \ldots, k-1 \tag{5.17}$$

where $\boldsymbol{S}$ is the sample covariance matrix of $\boldsymbol{X}_{jN}$. The condition (5.17) ensures that the next direction $\boldsymbol{z}_k$ is uncorrelated with all the previous ones.

From (5.15), it can be observed that the first chosen PLS direction $\boldsymbol{z}_1$ coincides with the particular vector that lies in the $\boldsymbol{X}$ space, represented using S, and makes a compromise between its variation and its correlation with the response $\boldsymbol{y}$. Similarly, from (5.7) we notice that the next space $S^{(1)}$, spanned by the updated input variables $\boldsymbol{X}^{(1)}_{jN}$, is the subspace of S orthogonal to the first PLS direction $\boldsymbol{z}_1$. As before, the second PLS direction $\boldsymbol{z}_2$ is that maximising the (5.15) and lying in this subspace $S^{(1)}$. Successive directions $\boldsymbol{z}_k$'s are calculated in a similar manner, with the residual subspace $S^{(k-1)}$ determined by removing from the space S, the space defined by the previous PLS directions.

Example

In the simple case of two different input variables, i.e. $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$, each having three time samples, they span a plane S in the three-dimensional space. The first PLS direction $\boldsymbol{z}_1$ is that vector that lies on S and maximizes (5.15). Once this first direction has been estimated, the second one lies on the subspace of S orthogonal to $\boldsymbol{z}_1$. In this particular case, the previously defined subspace correspond to the line orthogonal to $\boldsymbol{z}_1$.

5.3 Implementation of PLS

The PLS estimator can be implemented by the following (Matlab-like notation) pseudo-code:

load X, y → load data

normalize X,y → normalize data

$\hat{y}^{(0)} \leftarrow 0$
$X^{(0)} \leftarrow X$
} initialization

for $k = 1$ to M **do**

$\hat{\varphi}_{kj} \leftarrow X_{jN}^{(k-1)T} y$

$z_k \leftarrow \sum_{j=1}^{p} \hat{\varphi}_{kj} X_{jN}^{(k-1)T}$

$\hat{\theta}_k \leftarrow \frac{z_k^T y}{z_k^T z_k}$

$\hat{y}^{(k)} \leftarrow \hat{y}^{(k-1)} + \hat{\theta}_k z_k$

$\gamma_j \leftarrow \frac{z_k^T X_{jN}^{(k-1)}}{z_k^T z_k}$

$X_{jN}^{(k+1)} \leftarrow X_{jN}^{(k)} - \gamma_j z_k$

end for

} main loop

The Matlab code used in this thesis to implement PLS is reported in Appendix A.2.

5.4 Tutorial Examples

The same examples used in the Chapter 4 will be used below to highlight the features of PLS in estimating linear regression models and to compare it with OLS estimation.

5.4.1 Example 1 (Diabetes data)

As said in Chapter 4, the data for this example examine the correlation between a number of clinical measures in diabetic patients and a measure of "diabetes progression" (**dp**). In Section 4.4.1, it was described how to form matrix $\boldsymbol{X}$ and vector $\boldsymbol{y}$ involved in the regression problem. In addition, OLS estimates were presented and commented. Here the PLS estimation will be illustrated and compared with OLS's. Hence, the division of the data into training and test set is the same as in Section 4.4.1.

As PLS estimation is not scale invariant, before applying the algorithm, the matrices $\boldsymbol{X}$ and $\boldsymbol{y}$ have to be normalized. Since in Section 4.4.1 the data were normalized as well, a direct comparison of the estimated coefficients will be possible.

Before estimating the PLS coefficients we have to choose model complexity, as fully described in Chapter 3. Briefly, the test error curve as a function of the model complexity (for PLS the number of new regressors or directions) has to be estimated. Hence, the model complexity is selected using the "one-standard error" rule (Section 3.3.2), which indicates as best model the most parsimonious one, whose error is less then the minimum plus one time its standard deviation.

The test error curve is estimated using 7-fold cross-validation, described in Section 3.3.2. Briefly, the training data are randomly split in 7 parts of approximately equal size (in this case sets of 40 samples are formed). Iteratively, one part is left aside to calculate the test error (using MSE), while the other 6 parts are used to "learn" the coefficients of the model. In this way a test error upon each 7-th part is calculated and, averaging these values, an estimation of the test error is obtained.

In Figure 5.1 the test curve is shown. In this case, instead of using the *mean* operator to average the MSE values obtained during the cross-validation, the *median* operator was used. This choice allows to obtain a better estimation of the MSE average, as only 7 sample of MSE are calculated during the 7-fold cross-validation. Similarly, the *mean absolute deviance*[1] was used instead of the *standard deviation*.

In this case, the selected model correspond to the minimum value of the test error curve at two PLS directions. Hence, the value of M is set to 2.

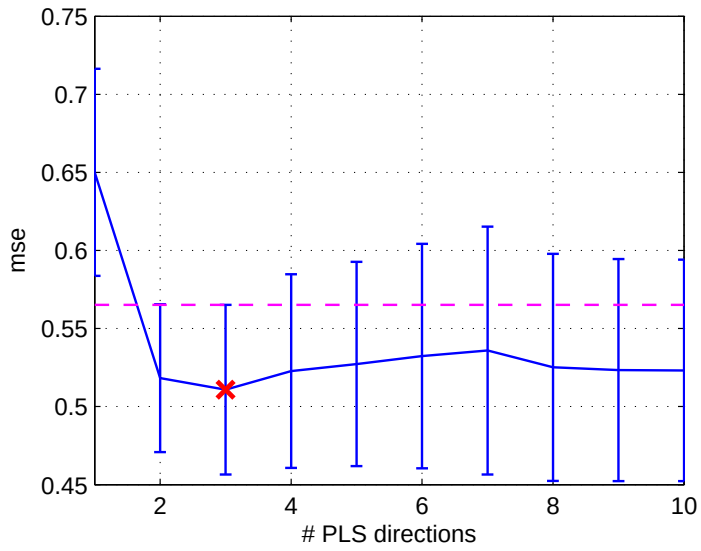

Figure 5.1: In blue the test error curve with its confidence intervals, obtained using 7-fold cross-validation. The red cross indicates the minimum value and the magenta dashed line represents the one standard error rule limit.

The training error curve cannot be used to estimate model complexity, given its monotonically decreasing behaviour. This can be seen in Figure 5.2, where training vs. test error curves are plotted allowing a direct comparison. However, the training error curve for this particular dataset has a L-form with an edge at $M = 2$, which agrees with the results obtained with the test error curve.

After having estimate M, the PLS estimates were computed. In Table 5.1 the

[1]Median Absolute Deviation of X corresponds to this formula: *median(abs(X-median(X)))*

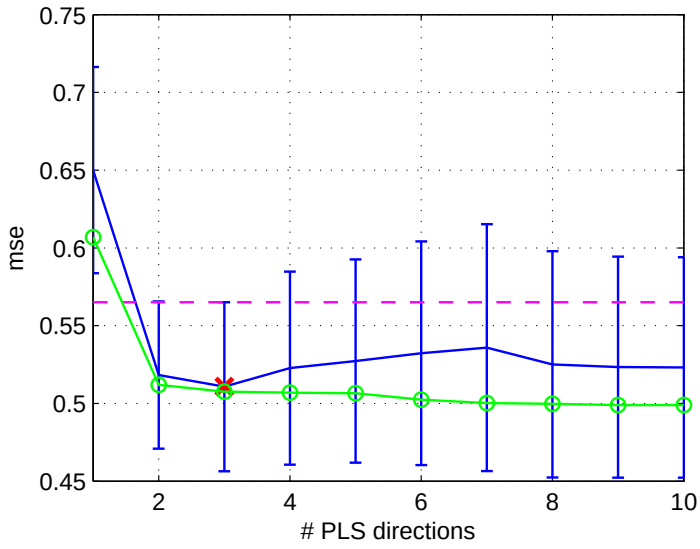

Figure 5.2: In green the training error curve, and in blue the test error curve with its confidence intervals, obtained using 7-fold cross-validation. The red cross indicates the minimum value and the magenta dashed line represents the one standard error rule limit.

values of the estimated $\boldsymbol{W}$ matrix (10 x 2) are shown. This matrix connects the original input $\boldsymbol{X}$ with the PLS direction $\boldsymbol{Z}$, using eq (5.11).

-0.005	-0.006
0.000	-0.025
-0.015	0.026
-0.011	0.014
-0.005	-0.024
-0.005	-0.027
0.011	-0.009
-0.011	-0.007
-0.014	0.013
-0.010	0.002

Table 5.1: Estimated values of the matrix $\boldsymbol{W}$.

However, instead of analysing the PLS directions, it is interesting to compare

the estimated OLS coefficients $\hat{\boldsymbol{\beta}}^{OLS}$ with the PLS ones $\hat{\boldsymbol{\beta}}^{PLS}$, as reported in Table 5.2:

	age	**sex**	**bmi**	**bp**	**sm1**	**sm2**	**sm3**	**sm4**	**sm5**	**sm6**
OLS	0.009	-0.167	0.318	0.190	-0.765	0.498	0.157	0.184	0.485	0.063
PLS	0.022	-0.126	0.297	0.192	-0.069	-0.088	-0.163	0.084	0.216	0.119

Table 5.2: Estimated OLS coefficients vs estimated PLS coefficients.

In Chapter 4 we noticed that, using the OLS estimator, the coefficients of the high correlated variables tend to grow large in opposite directions compensating each others. It was the case of **sm1** and **sm2**, which are positively correlated, and of **sm3** and **sm4**, which are negatively correlated. This is not the case using the PLS estimator, since the contributions of the previously mentioned variables add up.

From Table 5.2, we can also notice that the estimated PLS coefficients have, on average, a smaller absolute value of the OLS ones. This means that, using two PLS directions, some kind of regularisation has been performed.

As described in Chapter 3, to evaluate the performance of the two different methods, it is convenient to analyse their behaviour in predicting unseen data. Hence, the previously estimated coefficients are applied on inputs of the test set and the result is compared with the test reference. The predictions for both methods are plotted in Figure 5.3.

Visually, the methods have similar performances. However, it must be noted that PLS achieves results similar to OLS by using only two directions.

To quantify the performance of the two methods, some indicators must be used, as described in Chapter 3. In this case, to summarize how well the estimators predict the output, MSE indicator was used as shown in Table 5.3, confirming that the two estimators have similar performances.

	MSE
OLS	2842
PLS	2873

Table 5.3: MSE indicator for OLS and PLS on test data.

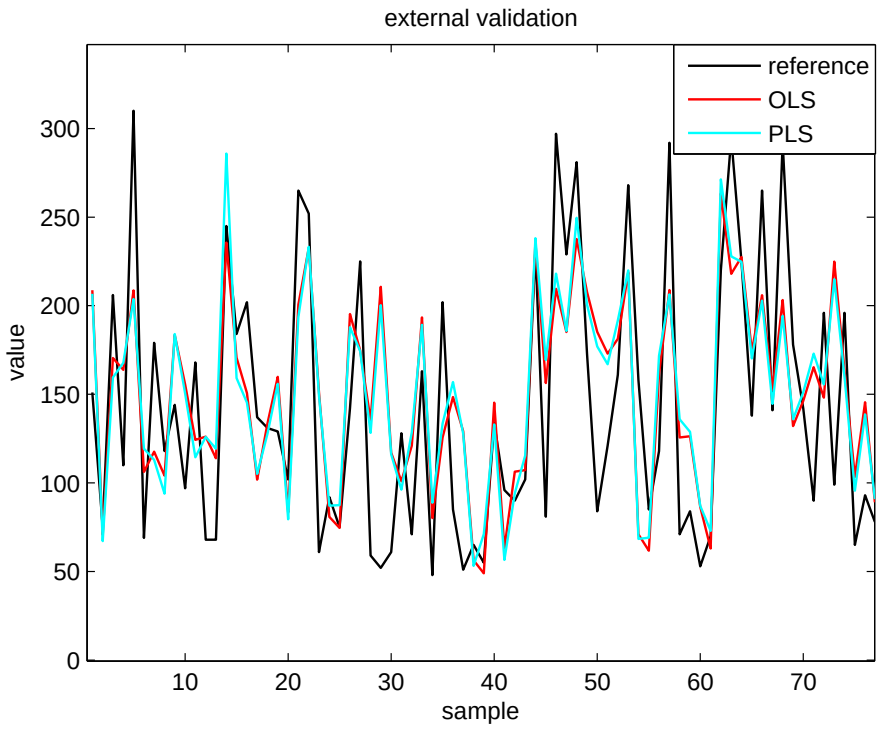

Figure 5.3: External validation using OLS and PLS (first 76 samples). In black the reference values, in red OLS predictions and in blue PLS predictions.

5.4.2 Example 2 (Simulated data)

As said in Chapter 4, the reference data for this second example were generated simulating glucose profiles with different time trends. From these profiles, twenty-seven input variables were obtained simulating Multisensor data, therefore exhibiting high correlation and including confounding processes such as body temperature.

The training data were simulated using a sequence of three glucose profiles, each having a length of eight hours and showing one or two *glycaemic* peaks. While input variables had an elevated sampling frequency (3 sample/minute), reference data were collected approximately every 15 minutes.

In Section 4.4.2 was described how to form the matrix $\boldsymbol{X}$ and the vector $\boldsymbol{y}$ involved in the regression problem. In addition, the OLS estimate was presented and commented. Here the PLS estimation will be illustrated and compared with OLS's.

As PLS algorithm is not scale invariant, before applying the algorithm, the matrices $\boldsymbol{X}$ and $\boldsymbol{y}$ have to be normalized. In Section 4.4.2 the data has been also normalised, allowing a direct comparison of the estimated coefficients.

Before estimating the PLS coefficients we have to choose model complexity. The model complexity was fixed using the test error curve, estimated from 3-fold cross-validation. Dividing the training set into 3 groups, each contains the same number of samples as one glucose profiles.

Figure 5.4 shows the test curve. As in Example 1, instead of using the *mean* operator to average the MSE values obtained during the cross-validation, the *median* operator was used. Similarly, the *mean absolute deviance* was used instead of the *standard deviation*.

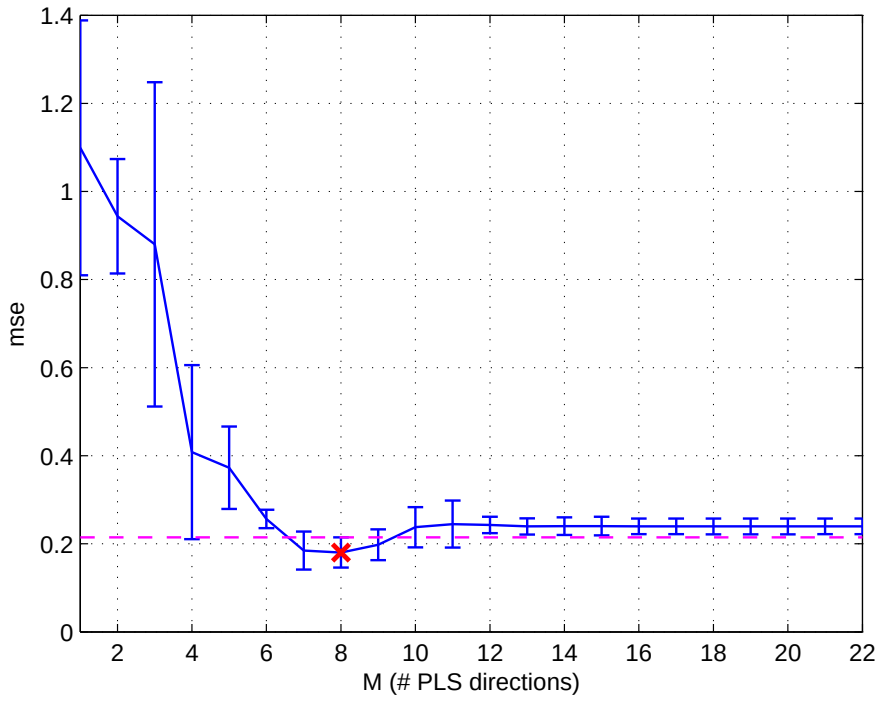

Figure 5.4: In blue the test error curve with its confidence intervals, obtained using 3-fold cross-validation. The red cross indicates the minimum value and the magenta dashed line represents the one standard error rule limit.

Notably, the maximum number M of PLS directions in Figure 5.4 is 22 and does not correspond to the number p (27) of original input variables. The reason of this choice lies in the rank of matrix $\boldsymbol{X}$. In this case this rank is 22, which means

that the space spanned by its columns has not dimension 27, but 22. As said in Section 5.2, the 23-th PLS direction lies in the subspace obtained removing from the column space of $\boldsymbol{X}$, the space spanned by the previous 22 PLS directions. Since the column space of $\boldsymbol{X}$ has dimension 22, removing the space spanned by the previous 22 PLS directions, one gets a subspace of dimension 0. Hence, it has not sense to calculate further PLS directions.

Going back to the choice of the model complexity, from Figure 5.4 it can be deduced that cross-validation selects 7 as the best value of M.

As for Example 1, instead of analysing the PLS directions, it is interesting to compare the estimated OLS coefficients $\hat{\boldsymbol{\beta}}^{OLS}$ against the PLS ones $\hat{\boldsymbol{\beta}}^{OLS}$, as reported in Table 5.4:

	MHz1	**MHz2**	**MHz3**	**MHz4**	**MHz5**	**MHz6**	**MHz7**	**MHz8**	**MHz9**
OLS	39.974	0.000	0.000	0.000	0.000	3.381	-8.656	-1.951	-6.151
PLS	-0.250	0.398	0.438	0.137	-0.636	-0.583	-0.337	0.002	1.471
	MHz10	**MHz11**	**MHz12**	**MHz13**	**MHz14**	**MHz15**	**MHz16**	**MHz17**	**MHz18**
OLS	-14.849	40.897	26.916	-24.356	2.557	-5.082	-23.234	54.732	-53.089
PLS	2.235	0.411	-0.168	-0.210	0.046	0.595	0.485	0.322	0.110
	MHz19	**MHz20**	**MHz21**	**MHz22**	**MHz23**	**MHz24**	**MHz25**	**Temp**	**Opt**
OLS	-4.424	-27.901	0.000	1.410	4.014	1.543	-12.167	2.397	0.071
PLS	-0.748	-1.357	-0.250	0.398	0.438	0.137	-0.636	2.429	0.074

Table 5.4: Estimated OLS coefficients vs estimated PLS coefficients.

As discussed in the previous example, the OLS coefficients associated to high correlated variables (**MHz**) show elevate magnitude with opposite signs. In addition, using the PLS estimator, a sort of regularisation is performed and PLS coefficients show significantly smaller magnitude.

The performance of the two different methods was evaluated using a test set, composed by two simulated glucose profiles. Hence, the previously estimated coefficients are applied on inputs of the test set and the result is compared with the test reference. The predictions for both methods are plotted in Figure 5.5.

Combining the measured inputs of the test set using the estimated coefficients, the prediction can be also calculated in the time instants that have not the corresponding reference. However, the prediction at these time instants cannot be used for calculating the indicators for the model assessment described in Chapter

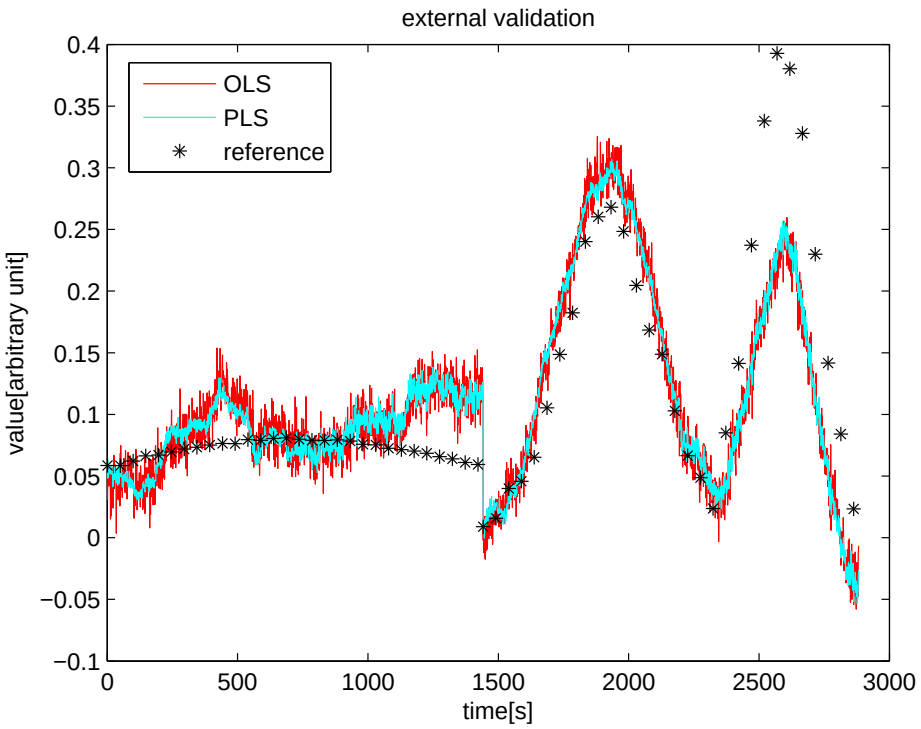

Figure 5.5: External validation using OLS and PLS. In black stars the reference values, in red OLS predictions and in blue PLS predictions.

3. In fact, each prediction value, used to calculate those indicators, must have the corresponding reference, since they quantify in different manners how well the prediction approximates the reference.

Visually, the methods have similar performances. However, it must be taken into account that PLS achieves results similar to OLS by using only 7 directions. In this case, even if each PLS direction is estimated combining all the original inputs, PLS is less noisy than OLS. This may be due to the shrinkage of the coefficients or to the fact that there is only one very noisy component, the optical one, and it has less influence on the construction of the PLS directions. This last observation is confirmed by observing the last row of the matrix $\boldsymbol{W}$:

$$\boldsymbol{W}(end,:) = \begin{bmatrix} -0.0009 & 0.0003 & -0.0034 & 0.0488 & -0.1094 & 0.0014 & 0.0016 \end{bmatrix}$$

which represents the weight of the optical component in the 7 PLS directions.

To quantify the performance of the two methods the MSE indicator was used as shown in Table 5.5.

	MSE
OLS	0.0032
PLS	0.0030

Table 5.5: MSE indicator for OLS and PLS on test data.

In this case, PLS has a better performance of the OLS. This confirms that probably OLS suffers from overfitting and PLS has partially solved the problem.

5.5 Concluding Remarks

PLS is a regression technique based on dimensionality reduction, which uses M new regressors, called PLS directions, calculated from a linear combination of the original input variables depending on their univariate influence on the target. The PLS solution is iteratively obtained and at each iteration a new PLS direction is estimated.

This technique for estimating linear model tries to avoid the OLS problem of overfitting, building orthogonal PLS direction. A further feature of the PLS directions is that they are estimated maximizing both their variance and the correlation with the reference. In this way, the PLS directions try to include the informative components of the original inputs, considering also their relationship with the reference. This may be an advantage, since, as noticed from the examples, much less PLS directions are sufficient to obtain similar or even better performance than OLS. However, since all the original input variables are included in each PLS direction, also some noise may affect it.

Chapter 6

Least Absolute Shrinkage and Selection Operator

In this Chapter we will present a regression technique that uses regularisation, adding a term to the function that has to be minimized. The methods using this technique are commonly called *regularisation methods* or *shrinkage methods*.

This Chapter will present the characterisation of the LASSO method. Then, the same two examples of Chapter 4 and 5 will be used to show advantages and drawbacks of LASSO. The use of the same examples allows a direct comparison with OLS estimation procedure presented in Chapter 4 and PLS estimation procedure presented in Chapter 5.

6.1 Definition of LASSO

In this Section we will described the LASSO estimation procedure. First the LASSO formulation will be presented, followed by an explanation of its problematic calculation.

6.1.1 *Rationale*

It is useful to recall that OLS estimates the coefficients of the linear regression model:

$$y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j = \beta_0 + \boldsymbol{X}_{ip}\boldsymbol{\beta} \tag{6.1}$$

by minimising the RSS:

$$\hat{\boldsymbol{\beta}}^{OLS} = \arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j \right)^2 \tag{6.2}$$

see eqs. (4.4), (4.8).

To avoid overfitting, shrinkage methods add a regularisation term $F(\boldsymbol{\beta})$ to the minimization term present in the cost function of (6.2), putting a price on $\boldsymbol{\beta}$ in order to avoid the coefficients to become, in absolute value, too big, as may happen with OLS (see Chapter 4). Hence, the function to minimise turns into:

$$L(\boldsymbol{\beta}) = RSS(\boldsymbol{\beta}) + F(\boldsymbol{\beta}) \tag{6.3}$$

and the estimated coefficients become:

$$\hat{\boldsymbol{\beta}}^{REG} = \arg\min_{\beta}(RSS(\boldsymbol{\beta}) + F(\boldsymbol{\beta})) \tag{6.4}$$

The result is that some coefficients (associated with less informative variables) shrink to zero. Typical regression techniques exploiting regularisation are Ridge Regression and Least Absolute Shrinkage and Selection Operator (LASSO)[52].

For example, Ridge Regression shrinks the coefficients $\boldsymbol{\beta}$ by imposing a penalty on their size, using a regularization term consisting in the coefficients sum of squares:

$$\hat{\boldsymbol{\beta}}^{RIDGE} = \arg\min_{\beta} \left(RSS(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \beta_j^2 \right) \tag{6.5}$$

while the regularization term of LASSO consists in the sum of coefficients absolute value:

$$\hat{\boldsymbol{\beta}}^{LASSO} = \arg\min_{\beta} \left(RSS(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j| \right) \tag{6.6}$$

$\lambda(\geq 0)$ is a complexity parameter that controls the amount of shrinkage; in particular, the larger λ, the greater the amount of shrinkage and the coefficients tend to zero.

The regularisation terms added in (6.5) and (6.6) are very similar. However, while in the case of ridge regression the minimisation function $L(\boldsymbol{\beta})$ has still a quadratic form allowing a closed form solution, in the case of LASSO the regularisation term introduces a non linearity that does not allow to recover a closed form expression for the solution.

6.1.2 Calculation of LASSO estimates

From eq. (6.6), LASSO estimate is:

$$\hat{\boldsymbol{\beta}}^{LASSO} = \arg\min_{\beta} \left(\sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right) \tag{6.7}$$

Problem (6.7) can also be formulated as a constrained optimization problem as follows:

$$\begin{aligned} \hat{\boldsymbol{\beta}}^{LASSO} &= \arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j \right)^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} |\beta_j| \leq t \end{aligned} \tag{6.8}$$

where t is inversely related to λ.

Because of the nature of the constrains, making t sufficiently small will cause some of the coefficients to be exactly zero, leading to a sparse solution.

To calculate the OLS solution in Chapter 4 we have set the first derivative of RSS to zero, in order to find the minimum of that function; thus, it seems reasonable to make the same for finding the LASSO solution. However, equation (6.7) is not differentiable when $\boldsymbol{\beta}$ contains zero values. Hence, an exact solution in closed form is not available and iterative methods are needed to compute the solution. As a consequence, computing the LASSO solution is a quadratic programming problem. This has lead to a wide variety of approaches proposed in the literature to solve the LASSO problem. In the next section, some algorithms for computing LASSO solution in an efficient way will be briefly described; particular attention will be given to the algorithm that has been chosen to analyze our data.

6.2 Some Numerical Methods for Computing LASSO Estimates

All the methods presented in this section compute the LASSO solution using Newton's method for unconstrained optimization. This method updates the vector of coefficients $\boldsymbol{\beta}$ at each iteration using a descent direction of the form:

$$\boldsymbol{\beta}_{k+1} \leftarrow \boldsymbol{\beta}_k - \alpha \boldsymbol{\nabla} L(\boldsymbol{\beta}_k) / \boldsymbol{\nabla}^2 L(\boldsymbol{\beta}_k) \tag{6.9}$$

where the subscript indicates the iteration.

Since the gradient $\boldsymbol{\nabla} L(\boldsymbol{\beta}_k)$ does not exist if some coefficients β_i are zero, different strategies were proposed to solve this problem. Here we propose a brief review of these methods, giving only the general flavour (see the references for the mathematical details). In Section 6.3, we will describe a modification of the Least Angle Regression (LAR) procedure for the LASSO implementation along with its interpretation.

6.2.1 Sub-gradient Methods

The optimization strategies of this kind use sub-gradients of the function at non-differentiable points. The absolute value function is differentiable everywhere except in zero. Its differential is given by the sign function $sgn(x)$, which takes on the sign of x. Hence, the minimisation function for LASSO:

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} X_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{6.10}$$

is differentiable except in zero. Considering that the sign function in zero can take any value between -1 and +1, the stationarity conditions (first derivative sets to zero) turns into:

$$\begin{cases} \boldsymbol{\nabla}_i RSS(\boldsymbol{\beta}) + \lambda sgn(\beta_i) = 0 & |\beta_i| > 0 \\ |\boldsymbol{\nabla}_i RSS(\boldsymbol{\beta})| \leq \lambda & \beta_i = 0 \end{cases} \tag{6.11}$$

and the steepest descent direction, which is used as sub-gradient in iteratively finding the best solution, becomes:

$$\boldsymbol{\nabla}_i^s L(\boldsymbol{\beta}) = \begin{cases} \boldsymbol{\nabla}_i RSS(\boldsymbol{\beta}) + \lambda sgn(\beta_i) & |\beta_i| > 0 \\ \boldsymbol{\nabla}_i RSS(\boldsymbol{\beta}) + \lambda & \beta_i = 0, \boldsymbol{\nabla}_i RSS(\boldsymbol{\beta}) < -\lambda \\ \boldsymbol{\nabla}_i RSS(\boldsymbol{\beta}) - \lambda & \beta_i = 0, \boldsymbol{\nabla}_i RSS(\boldsymbol{\beta}) > \lambda \\ 0 & \beta_i = 0, -\lambda \leq \boldsymbol{\nabla}_i RSS(\boldsymbol{\beta}) \leq \lambda \end{cases} \tag{6.12}$$

Without going into details, we simply mention that the algorithms based on sub-gradient can be classified in three different strategies, according to which variables are optimized at every iteration: coordinate descent methods[55][56], that optimize over one variable at a time, active set methods[57][58][59], that optimize all the non-zero variables at every iteration and orthant-wise descent methods[60], that are similar to the previous but adds two projection operators.

6.2.2 Unconstrained approximation methods

These methods replace the minimization function $L(\boldsymbol{\beta})$ with a twice differentiable surrogate objective function, whose minimiser is sufficiently close to the minimiser of $L(\boldsymbol{\beta})$. The main advantage of this approach is that, since the replaced function is twice differentiable, we can directly apply an unconstrained optimization method to minimize the function. For example, in [61] the L1-norm constrained is replaced with the multi-quadratic functions:

$$G(\boldsymbol{\beta}) = RSS(\boldsymbol{\beta}) + \lambda \sum_{i=1}^{p} \sqrt{\beta_i^2 + \epsilon} \tag{6.13}$$

whose limit for $\epsilon \to 0^+$ correspond to the function $L(\boldsymbol{\beta})$. Rather than replacing the regulariser with a fixed smooth function, at each iteration *bound optimization* methods replace it with a convex upper bounding function. For example, the absolute value function can be approximated using the upper bound of this inequality:

$$|\beta_i| \leq \frac{1}{2} \frac{\beta_i^2}{|\gamma_i|} + \frac{1}{2} |\gamma_i| \tag{6.14}$$

where γ_i represent the value of β_i at the previous iteration. However, this approximation is undefined if γ_i is zero. To solve this problem, the algorithm must be initialised with non-zero coefficients and a variable is typically removed from the problem if it becomes sufficiently close to zero [62][63].

6.2.3 Constrained optimization methods

These methods re-formulate problem (6.7) as a differentiable one with constraints. In this case, each variable β_i is represented as the sum of two variables:

$$\beta_i = \beta_i^+ - \beta_i^- \tag{6.15}$$

where $\beta_i^+ \geq 0$ and $\beta_i^- \geq 0$. In this formulation the absolute value function becomes:

$$|\beta_i| = \beta_i^+ + \beta_i^- \tag{6.16}$$

An obvious drawback of this approach is that it doubles the number of variables in the optimization problem.

Different methods are based on this approach, for instance: log-barrier[64], interior-point[65], projected Newton[66] and two-metric projection[67].

6.3 Least Angle Regression Method for Computing LASSO Estimates

LAR is an iterative method intimately connected with LASSO. In fact it provides an extremely efficient algorithm for computing the entire LASSO path, i.e. the behaviour of the coefficients $\boldsymbol{\beta}$ for different values of the complexity parameter λ.

Since this optimisation method will be later used to find the LASSO solution in our dataset, it will be described in detail. First the LAR procedure will be presented, then, the modification for the LASSO implementation along with its interpretation will also be illustrated.

6.3.1 The LAR procedure

It is useful to define the active set $\mathcal{A}_k$ (of dimension m) as the set of the non-zero coefficients at the k-th step. When it is used as a subscript for a matrix or a vector, it selects the values connected to the active variables at the k-th step. Hence, $\boldsymbol{X}_{\mathcal{A}_k}$ is the sub-matrix of $\boldsymbol{X}$ composed by the active variables and $\boldsymbol{\beta}_{\mathcal{A}_k}$ is the coefficient vector for these variables. To simplify the notation, the subscript k will be dropped, if it is clear that we are referring to the k-th step.

The LAR solution is computed following these steps:

1. set all the coefficients β_i to zero;

2. choose the variable $\boldsymbol{X}_{jN}$ most correlated with the reference $\boldsymbol{y}$;

3. move the correspondent coefficient β_j from zero towards its OLS value β_j^{OLS} (in this way the correlation of the variable $\boldsymbol{X}_{jN}$ with the current residual $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{X}_{jN}\beta_j$ decreases);

4. the process continues until another variable $\boldsymbol{X}_{lN}$ has as much correlation with the current residual as $\boldsymbol{X}_{jN}$ has;

5. the variable $\boldsymbol{X}_{lN}$ is added to the active set $\mathcal{A}_k$;

6. move the coefficients $\boldsymbol{\beta}_{\mathcal{A}_k}$ towards their OLS values, in such a way that their correlation with the current residual $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{X}_{\mathcal{A}_k}\boldsymbol{\beta}_{\mathcal{A}_k}$ continues to be the same;

7. repeat steps 4-6 until $\mathcal{A}_k$ has reached the desired dimension or until all the variables have been included to $\mathcal{A}_k$ (in this case the OLS solution is obtained).

Figure 6.1 shows an example of the progression of the absolute correlations during each step of the LAR procedure. The labels at the top of the plot indicate which variable enters the active set at each step.

By construction, the coefficients in LAR change in a piecewise linear fashion. Note that we do not need to take small steps and re-check the correlation in step 4. In fact, using the knowledge of the covariance of the predictors and the piecewise linearity of the algorithm, the exact step length can be calculated at the beginning of each step.

6.3.2 The LAR implementation

Now that we have understood the guidelines of the LAR algorithm, we can go into its mathematical details. First of all, let us define some useful notation. $\boldsymbol{X}_{s\mathcal{A}}$ is the same as $\boldsymbol{X}_{\mathcal{A}_k}$, but each regressor is multiplied by the sign s_j of its correlation with the current residual $\boldsymbol{r}$:

$$\boldsymbol{X}_{s\mathcal{A}} = \begin{bmatrix} \dots & s_j\boldsymbol{X}_{jN} & \dots \end{bmatrix} \tag{6.17}$$

where $\boldsymbol{X}_{jN} \in \mathcal{A}_k$. For simplicity, let's define $\boldsymbol{G}_{\mathcal{A}}$ (m x m) as:

$$\boldsymbol{G}_{\mathcal{A}} = \boldsymbol{X}_{s\mathcal{A}}^T\boldsymbol{X}_{s\mathcal{A}} \tag{6.18}$$

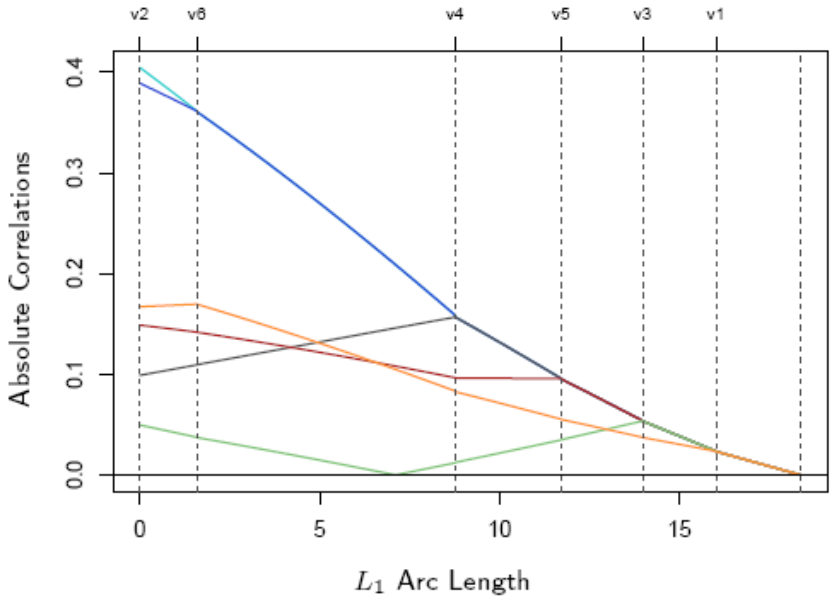

Figure 6.1: Progression of the absolute correlations during each step of the LAR procedure [52].

and the scalar $A_{\mathcal{A}}$ as:

$$A_{\mathcal{A}} = \left(\mathbf{1}_{\mathcal{A}}^T \mathbf{G}_{\mathcal{A}} \mathbf{1}_{\mathcal{A}}\right)^{-1/2} \tag{6.19}$$

where $\mathbf{1}_{\mathcal{A}}$ (m x 1) is a column vector of ones.

Since the LAR procedure is not scale invariant, data have to be normalized before starting the iterative procedure. Hence, the initial target estimation $\hat{\boldsymbol{y}}_0$ is set to zero. Let $\hat{\boldsymbol{y}}_k$ the current target estimation at the k-th step, the current correlation $\boldsymbol{c}$ (m x 1) of the predictors with the current residual can be written as:

$$\boldsymbol{c} = \boldsymbol{X}^T(\boldsymbol{y} - \hat{\boldsymbol{y}}_k) \tag{6.20}$$

The current active set $\mathcal{A}_k$ includes all the variables, whose absolute correlation correspond to the maximum of all the absolute correlations C_{max}:

$$\mathcal{A}_k = \{j : |c_j| = C_{max}\} \quad \text{where} \quad C_{max} = \max_j \{|c_j|\} \tag{6.21}$$

The solution at the next step updates as follows:

$$\hat{\boldsymbol{y}}_{k+1} = \hat{\boldsymbol{y}}_k + \gamma \boldsymbol{u}_{\mathcal{A}} \tag{6.22}$$

where $\boldsymbol{u}_{\mathcal{A}}$ is a versor ($\|\boldsymbol{u}_{\mathcal{A}}\| = 1$) defining the direction to which the current target estimation $\hat{\boldsymbol{y}}_k$ is moved. This direction is calculated in such a way that the correlation of each active variables with the current residual vector equals the correlation of the other active variables. Hence, it seem reasonable that the versor $\boldsymbol{u}_{\mathcal{A}}$ makes equal angles with the columns of $\boldsymbol{X}_{s\mathcal{A}}$. The versor $\boldsymbol{u}_{\mathcal{A}}$ is calculated as follows:

$$\boldsymbol{u}_{\mathcal{A}} = \boldsymbol{X}_{s\mathcal{A}}\boldsymbol{w}_{\mathcal{A}} \quad \text{where} \quad \boldsymbol{w}_{\mathcal{A}} = A_{\mathcal{A}}\boldsymbol{G}_{\mathcal{A}}^{-1}\boldsymbol{1}_{\mathcal{A}} \quad (m \text{ x } 1) \tag{6.23}$$

and, since it is an equiangular vector, it enjoys this property:

$$\boldsymbol{X}_{s\mathcal{A}}^T\boldsymbol{u}_{\mathcal{A}} = A_{\mathcal{A}}\boldsymbol{1}_{\mathcal{A}} \tag{6.24}$$

Instead, the coefficients updates as follows:

$$\hat{\boldsymbol{\beta}}_{k+1} = \hat{\boldsymbol{\beta}}_k + \gamma\boldsymbol{d}_{\mathcal{A}} \tag{6.25}$$

where $\boldsymbol{d}_{\mathcal{A}}$ (m x 1) is the vector equaling $s_j w_{Aj}$ for $j \in \mathcal{A}_k$ (note the connection with the versor $\boldsymbol{u}_{\mathcal{A}}$ in (6.23)) and zero elsewhere.

As said before, γ can be exactly computed as to update the variables to the point in which another variable enter the active set. In particular, γ is calculated as follows:

$$\gamma = \min_{j\in\mathcal{A}^c}{}^{+}\left\{\frac{C_{max} - c_j}{A_{\mathcal{A}} - a_j}, \frac{C_{max} + c_j}{A_{\mathcal{A}} + a_j}\right\} \quad \text{where} \quad a_j = \boldsymbol{X}_{jN}^T\boldsymbol{u}_{\mathcal{A}} \tag{6.26}$$

where $\min^+$ indicates the minimum between the positive values, being $\gamma > 0$.

The explanation of (6.26) is obtained by equalling the current correlation of a variable that is not in the active set with the correlation of the active variables. In particular, the current correlation of the j-th variable is:

$$c_j(\gamma) = \boldsymbol{X}_{jN}^T(\boldsymbol{y} - \hat{\boldsymbol{y}}_{k+1}) \tag{6.27}$$

then, substituting (6.22) in (6.27) one gets:

$$c_j(\gamma) = \boldsymbol{X}_{jN}^T(\boldsymbol{y} - \hat{\boldsymbol{y}}_k - \gamma\boldsymbol{u}_{\mathcal{A}}) \tag{6.28}$$

which using (6.20) and (6.26), becomes:

$$c_j(\gamma) = c_j - \gamma a_j \tag{6.29}$$

If the absolute value of (6.29) is referred to an active set variable, using (6.21) and (6.24), it becomes:

$$|c_j(\gamma)| = C_{max} - \gamma A_{\mathcal{A}} \tag{6.30}$$

then, equalling (6.29) with (6.30) one gets:

$$\begin{cases} C_{max} - \gamma A_{\mathcal{A}} &= c_j - \gamma a_j \\ -C_{max} + \gamma A_{\mathcal{A}} &= c_j - \gamma a_j \end{cases} \tag{6.31}$$

Solving the set of equations in (6.31) for γ, one obtains the values of γ for which the correlation of a variable that is not in the active set equals the correlation of the active variables. Since we search the minimum positive value of γ, corresponding to the step of the first non active variable equalling the correlation of the active ones, we get the (6.26).

6.3.3 LAR vs. LASSO

In Figure 6.2 the coefficient profiles are plotted as model complexity increases for both LAR (left) and LASSO (right). It can be noticed that the profiles are similar to each other, except when a non-zero variable hits zero (highlighted by a red circle in Figure 6.2). In fact, a small modification in LAR procedure allows implementing the LASSO path. The modification is the following: if a non-zero coefficient hits zero, drop its variable from the active set and recomputed the current joint least squares direction.

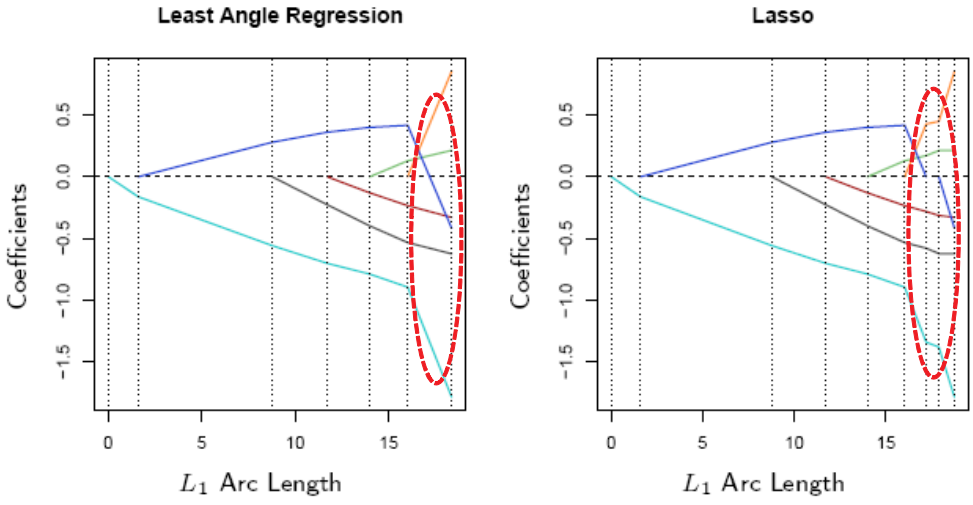

Figure 6.2: *Left*: LAR coefficients profile as the model complexity increases. *Right*: LASSO coefficients profile as the model complexity increases [52].

Below we explain why LAR and LASSO are so similar.

The correlation of an active set variable with the current residual can be expressed as:

$$\boldsymbol{X}_{jp}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \gamma s_j \quad \forall j \in \mathcal{A}_k \tag{6.32}$$

where $s_j \in \{-1, 1\}$ indicates the sign of the correlation and γ is the absolute value of the correlation.

Since the non-active variables are less correlated to the current residual than the active variables, we can write:

$$\left|\boldsymbol{X}_{lp}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right| \leq \gamma \quad \forall k \notin \mathcal{A}_k \tag{6.33}$$

The LASSO minimisation function:

$$L(\boldsymbol{\beta}) = \frac{1}{2}\left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\right\|^2 + \lambda\left|\boldsymbol{\beta}\right| \tag{6.34}$$

is differentiable for the active variables. For these variables the stationarity conditions (first derivative sets to zero) are:

$$\boldsymbol{X}_{jp}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \lambda sgn(\beta_j) \quad \forall j \in \mathcal{A}_k \tag{6.35}$$

which correspond to (6.32) if the sign of the correlation s_j matches the sign of the coefficients β_j. That is why the LAR algorithm and the LASSO start to differ when an active coefficient passes through zero. The LASSO condition (6.34) is violated for that variable, which is, thus, kicked out of the active set.

Finally, the stationarity conditions for the non-active variables are:

$$\left|\boldsymbol{X}_{lp}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right| \leq \gamma \quad \forall k \notin \mathcal{A}_k \tag{6.36}$$

which correspond to the LAR equation (6.33).

6.3.4 LASSO Implementation by a LAR modification

The only modification of the LAR procedure for implementing LASSO is a checking of the γ value calculated in (6.26). In fact, we have to make sure that during the LAR step none of the coefficients $\boldsymbol{\beta}$ changes its sign. In particular, starting from the updating of the coefficients in (6.26), here reported:

$$\hat{\boldsymbol{\beta}}_{k+1} = \hat{\boldsymbol{\beta}}_k + \gamma \boldsymbol{d}_{\mathcal{A}}$$

a β_j will change sign at:

$$\gamma_j = -\frac{\hat{\beta}_j}{d_j} \tag{6.37}$$

The first change occurs at:

$$\tilde{\gamma} = \min_{\gamma_j > 0} \{\gamma_j\} \tag{6.38}$$

corresponding to the $\tilde{j}$-th variable.

Hence, if $\tilde{\gamma} > \gamma$ calculated in (6.26), no sign change will occur and the LAR step does not violate any LASSO condition. Contrarily, if in (6.26) $\tilde{\gamma} \leq \gamma$, the updated coefficients $\hat{\boldsymbol{\beta}}_{k+1}$ cannot be a LASSO solution. To avoid this, the LAR step is not completed, but it is stopped at $\tilde{\gamma} = \gamma$. Then, the $\tilde{j}$-th variable is removed from the active set and a new equiangular direction in (6.23) is calculated.

6.4 Properties of LASSO

6.4.1 Geometrical Properties

As for OLS in Chapter 4, we now consider the case of two different input variables $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$, each having three time samples.

As described in the previous section, LAR builds up the estimates in successive steps, each step adding one regressor to the model, according to the value of its correlation with the target variable. In the case of two input variables, the current correlations $\boldsymbol{c}$ depend only on the projection $\bar{\boldsymbol{y}}$ of $\boldsymbol{y}$ into the plane spanned by $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$:

$$\boldsymbol{c} = \boldsymbol{X}^T \boldsymbol{y} = \boldsymbol{X}^T \bar{\boldsymbol{y}} \tag{6.39}$$

As show in Figure 6.3, makes a smaller angle with $\boldsymbol{X}_{13}$ than with $\boldsymbol{X}_{23}$, that corresponds to a greater correlation with $\boldsymbol{X}_{13}$ than with $\boldsymbol{X}_{23}$. Hence, the variable $\boldsymbol{X}_{13}$ enters the active set (step 2) and the solution moves in direction of $\boldsymbol{X}_{13}$, indicated in Figure 6.3 by the equiangular unit vector $\boldsymbol{u}_1$ (step 3-eq. (6.23)). Representing the moving solution of this first iteration with $\vec{\boldsymbol{y}}_1$, the current correlations $\boldsymbol{c}$ with the current residual becomes:

$$\boldsymbol{c} = \boldsymbol{X}^T (\bar{\boldsymbol{y}} - \vec{\boldsymbol{y}}_1) \tag{6.40}$$

From the Figure 6.3, we can see that the correlation of $\boldsymbol{X}_{13}$ with the current residual decreases. This process stops when the current residual is equally correlated with $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$ (step 4), that happens when the residual vector $(\bar{\boldsymbol{y}} - \vec{\boldsymbol{y}}_1)$ bisects the angle between $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$. Hence, the variable $\boldsymbol{X}_{23}$ is added to the active set (step 5). Now the solution moves in such a direction as to keep equal

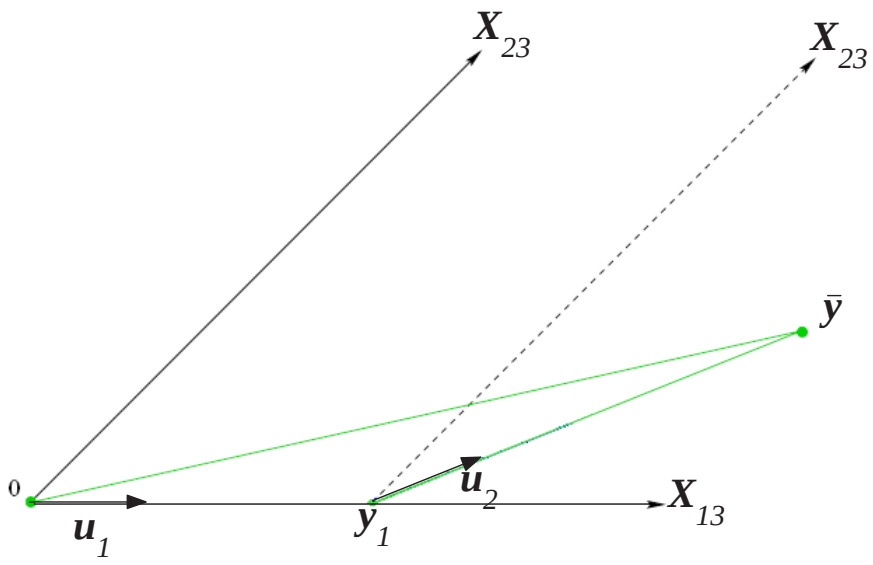

Figure 6.3: Geometrical interpretation of LASSO solution using LAR modification. Projection of the target vector $\bar{\boldsymbol{y}}$, input vectors $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$. Versor $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ indicating the equiangular vectors [53].

the two correlations (step 6). This direction is represented in Figure 6.3 by the equiangular unit vector $\boldsymbol{u}_2$ (eq. (6.23)), that corresponds to the bisector of the two vectors $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$.

In this case all the variables were added to the active set, hence at the next iteration the OLS solution is reached. Note that the OLS solution corresponds to $\bar{\boldsymbol{y}}$ (Section 4.2.1). In the general case, subsequent iterations are taken along equiangular vectors, generalizing the concept of the bisector $\boldsymbol{u}_2$.

6.4.2 Sparse Solution

As said in the Section 6.1.2, the regularisation term added to the minimisation function in LASSO yielded to a sparse solution. In this Section it will be described the reason why such a constraint lead to a sparse solution, using, for simplicity, the same example of two input variables $\boldsymbol{X}_{13}$ and $\boldsymbol{X}_{23}$.

From (6.8) the constraint region defined by LASSO is:

$$|\beta_1| + |\beta_2| \leq t \tag{6.41}$$

which is represented by a diamond area in the Cartesian space of the coefficients

(blue region in Figure 6.4). As a consequence, all the possible solutions of LASSO lie in this region.

Plotting in the same Cartesian space the OLS solution ($\hat{\boldsymbol{\beta}}$ in Figure 6.4) we can see how the OLS estimates, minimizing the RSS, falls in the center of the elliptical contours which represent the RSS behaviour for different estimates of $\boldsymbol{\beta}$.

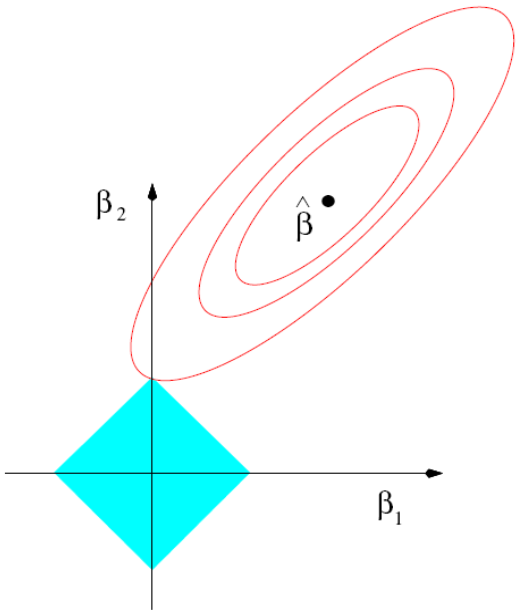

Figure 6.4: Interpretation of the sparse solution of LASSO. $\hat{\boldsymbol{\beta}}$ represents the OLS solution, the red ellipses are the contours of the residual sum of squares and the blue areas correspond to the constraint region $|\beta_1| + |\beta_2| \leq t$ (taken from [52]).

The LASSO solution is the first point where the elliptical contour hits the constraint region. Since the diamond region has got corners, it is probable that the solution occurs at a corner. In this case, one coefficient is exactly zero, in particular $\boldsymbol{\beta}_1$ in Figure 6.4. In addition, when there are more predictors, the diamond becomes a rhomboid, and has got many more corners and flat edges. As a consequence, there are many more opportunities for the estimated parameters to be zero.

A comparison with the Ridge constraints, mentioned in the introduction of

this Chapter, may help to understand the particularity of LASSO.

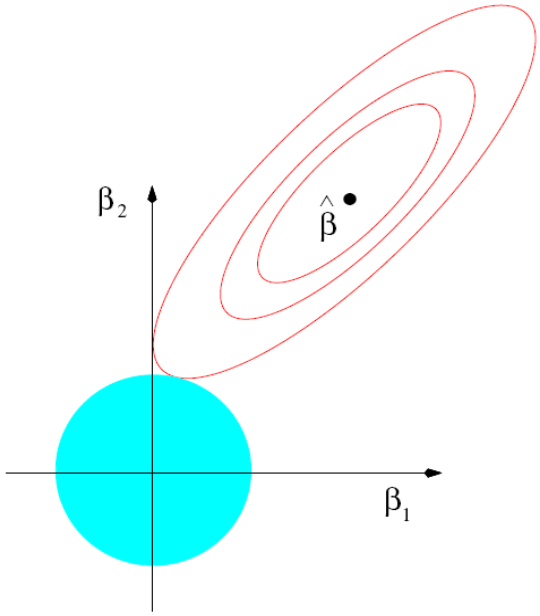

Figure 6.5: Ridge Regression regularised solution. As in 6.4, but here the blue area corresponds to the constraint $\beta_1^2 + \beta_2^2 \leq t$ (taken from [52]).

From (6.5) the constraint region defined by Ridge Regression is:

$$\beta_1^2 + \beta_2^2 \leq t \tag{6.42}$$

which is represented by a disk area in the Cartesian space of the coefficients (blue region in Figure 6.5). In this case, as the disk has no corners, there are fewer opportunities for one coefficient to be exactly zero.

6.5 LAR modification for the implementation of LASSO

The LASSO path can be estimated using the LAR modification. It can be implemented by the following (Matlab-like notation) pseudo-code (the updates of $\boldsymbol{u}_{\mathcal{A}}$, $\boldsymbol{d}_{\mathcal{A}}$ and $A_{\mathcal{A}}$ has not been reported):

load X, y → load data
normalize X, y → normalize data

$\hat{y}_0 \leftarrow 0;$
$\hat{\beta}_0 \leftarrow 0;$
$c \leftarrow X^T y;$
$C_{max} \leftarrow \max(c);$
$j \leftarrow \text{find}(c = C_{max});$
$A \leftarrow x_j$

} initialization

while active variables$< p$ **do**

$a = X^T u_A$
$\gamma = \min_{l \in A^c}{}^{+} \left\{ \frac{C_{max} - c_l}{A_A - a_l}, \frac{C_{max} + c_l}{A_A + a_l} \right\}$

(associated with X_l)
$\tilde{\gamma} = \min_j(-\hat{\beta}_k / d_A)$

if $\tilde{\gamma} < \gamma$ **then**
$\gamma = \tilde{\gamma};$
end if

} γ estimation (LASSO constraint)

$\hat{y}_{k+1} = \hat{y}_k + \gamma u_A$
$\hat{\beta}_{k+1} = \hat{\beta}_k + \gamma d_A$
$C_{max} = C_{max} - \gamma A_A$

} estimates update

if $\tilde{\gamma} < \gamma$ **then**
drop X_j from A
end if
$A \leftarrow X_l$

} active set update

update u_A, d_A and A_A
$c = X^T(y - \hat{y}_{k+1});$
end while

} main loop

The Matlab code used in this thesis to implement LASSO is reported in Appendix A.3.

6.6 Tutorial Examples

The same examples used in the previous Chapters will be used below to highlight the features of LASSO in estimating linear regression models and to compare it with the OLS and PLS estimator.

6.6.1 Example 1 (Diabetes data)

The data for this example examine the correlation between a number of clinical measures in diabetes patients and a measure of "diabetes progression" (**dp**). In Section 4.4.1 it was described how to form the matrix $\boldsymbol{X}$ and the vector $\boldsymbol{y}$ involved in the regression problem. In addition, in the previous Chapters OLS and PLS estimates were presented and commented. Here the LASSO estimation will be illustrated and compared with OLS and PLS.

As described before, the LAR procedure allows to create the entire LASSO path, i.e. the behaviour of the coefficients $\boldsymbol{\beta}$ as the model complexity increases.

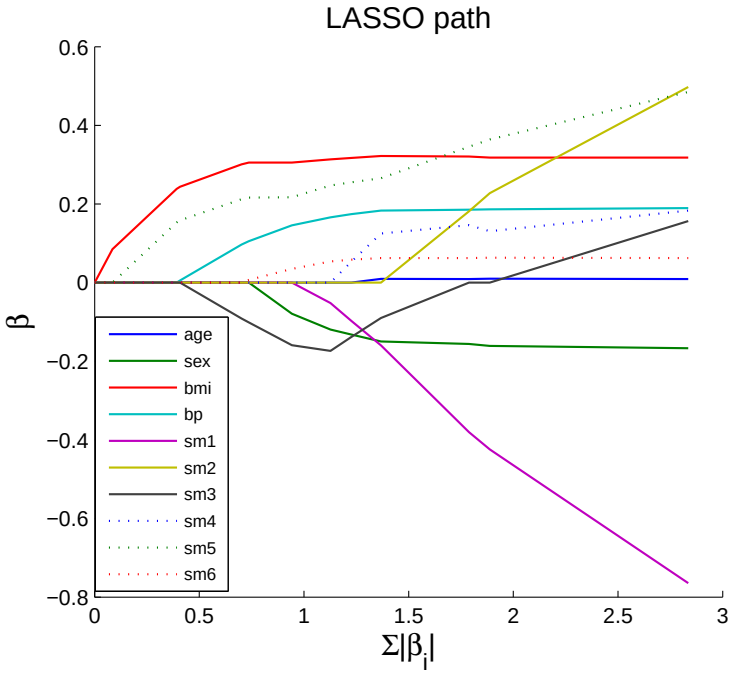

Figure 6.6: LASSO path for diabetes data in example 1.

As shown in Figure 6.6, at first all the parameters $\boldsymbol{\beta}$ are set to zero and enter in the active set according to their correlation with the current residual.

Hence, it may be useful to plot the behaviour of the correlation during the different iterations, as shown in Figure 6.7.

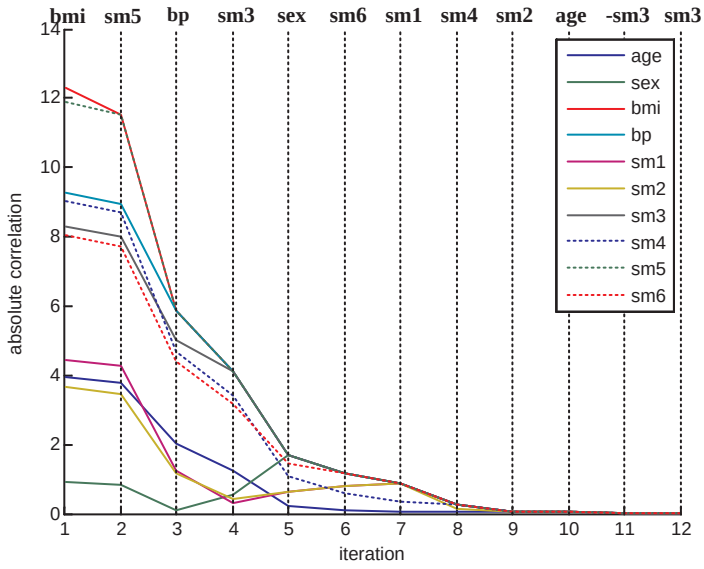

Figure 6.7: absolute correlations of the regressors with the current residual. At the top the name of the variable that are added (or dropped) from the active set is indicated.

Before describing how to choose the complexity of the model, it is worth to analyse the characteristics of the LASSO path.

As seen in the previous Chapter, the main problem of the OLS estimator was that the coefficients of highly correlated variables tend to grow large in opposite directions compensating each others. It is interesting to analyse the behaviour of such coefficients along the LASSO path. For convenience, the correlations between the variables are reported in Table 6.1.

To allow a direct comparison, we analyse the behaviour along the LASSO path of the variables, which were described in the previous Chapter.

The variables **sm3** and **sm4** show a high negative correlation. **Sm3** enters the active set before **sm4**; it is interesting to notice that, when **sm4** enters the active set, the coefficient of **sm3**, instead of growing larger, becomes smaller (see the big red ellipse in Figure 6.8 (left)). Hence, the compensation effect, detected in the OLS solution is avoided at that level of model complexity. However, as the model complexity increases (i.e. the solution moves towards the OLS one), the

	sex	bmi	bp	sm1	sm2	sm3	sm4	sm5	sm6
age	0.174	0.185	0.335	0.260	0.219	-0.075	0.204	0.271	0.302
sex	-	0.088	0.241	0.035	0.143	-0.379	0.332	0.150	0.208
bmi	-	-	0.395	0.250	0.261	-0.367	0.414	0.446	0.389
bp	-	-	-	0.242	0.186	-0.179	0.258	0.393	0.390
sm1	-	-	-	-	0.897	0.052	0.542	0.516	0.326
sm2	-	-	-	-	-	-0.196	0.660	0.318	0.291
sm3	-	-	-	-	-	-	-0.738	-0.399	-0.274
sm4	-	-	-	-	-	-	-	0.618	0.417
sm5	-	-	-	-	-	-	-	-	0.465

Table 6.1: correlation between the different input variables. Highlighter the most elevated correlations.

compensation problem reappears (see the small red ellipse in Figure 6.8 (left)).

The variables **sm1** and **sm2** show a high positive correlation. **Sm1** enters the active set before **sm2**. In this case the compensation problem affects the whole LASSO path (see Figure 6.8 (right)). However, we have to notice that these variables enter the active set after **sm3** and **sm4**. Hence, at that point we are nearer to the OLS than in the previous case.

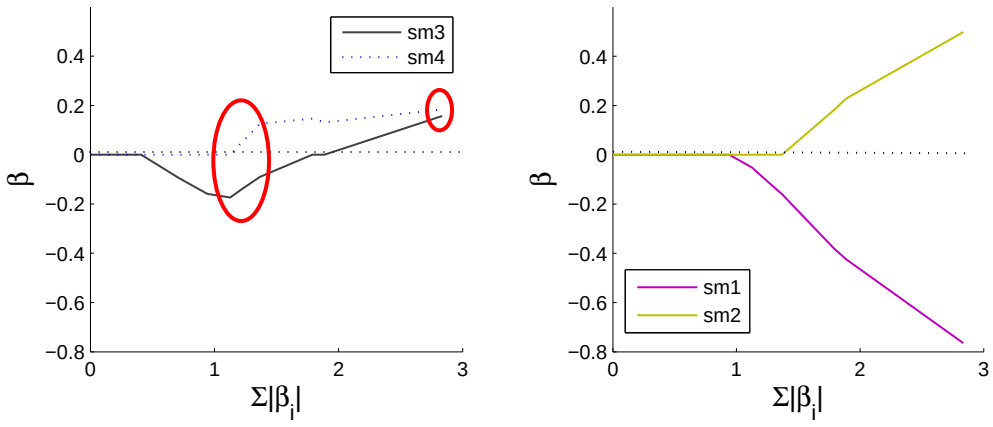

Figure 6.8: *Left*: LASSO path for **sm3** and **sm4** (remarkable point are highlighted by red ellipses). *Right*: LASSO path for **sm1** and **sm2**.

After this overview of the LASSO path, we move to the choice of the model complexity, as fully described in Chapter 3. In Figure 6.9 the "cross-validation"

curve is shown as a function of model complexity, obtained using 7-fold cross-validation as in Section 5.4.1. In short, the training data are randomly split in 7 parts of approximately equal size (in this case sets of 40 samples are formed). Iteratively, one part is left aside to calculate the test error (using MSE), while the other 6 parts are used to "learn" the coefficients of the model. In this way a test error upon each 7-th part is calculated and, averaging these values, an estimation of the test error is obtained.

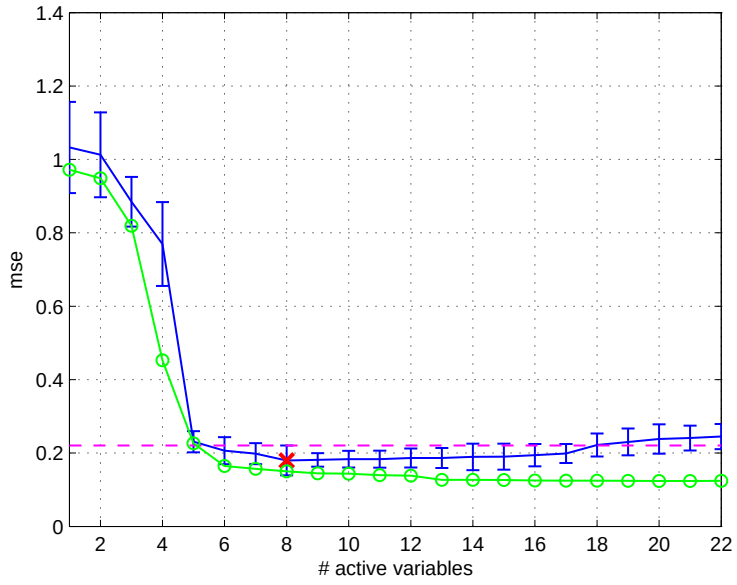

Figure 6.9: In blue the test curve with its confidence intervals, obtained using 7-fold cross-validation. The red cross indicates the minimum value and the magenta dashed line represents the one standard error rule.

As in the previous two Chapters, instead of using the *mean* operator to average the MSE values obtained during the cross-validation, the *median* operator was used. Similarly, the *mean absolute deviance* was used instead of the *standard deviation*.

The test curve is plotted as a function of the number of the active variables, instead of the most obvious λ (the complexity parameter). However, the number of the active variables is intuitively connected to the model complexity, but also to the degrees of freedom of the model (see [68] for more details).

The test error curve in Figure 6.9 has not a distinct minimum but it is very flat

in its correspondence. Since each point is estimated with an error, the minimum is not the best choice. The "one-standard error" rule (Section 3.3.2) is usually used in this case. It selects the most parsimonious model, whose error is less then the minimum plus its standard deviation. However, if the curve exhibits a well defined L-shape, it seems reasonable to select the model at its edge. However, if the curve shows a well defined elbow, it seems reasonable to select the model corresponding to this point. Hence, the final chosen model has four active variables.

It is interesting to compare the behaviour of the cross-validation curve using the LASSO estimator with the one obtained using the PLS estimator (see Section 5.4.1).

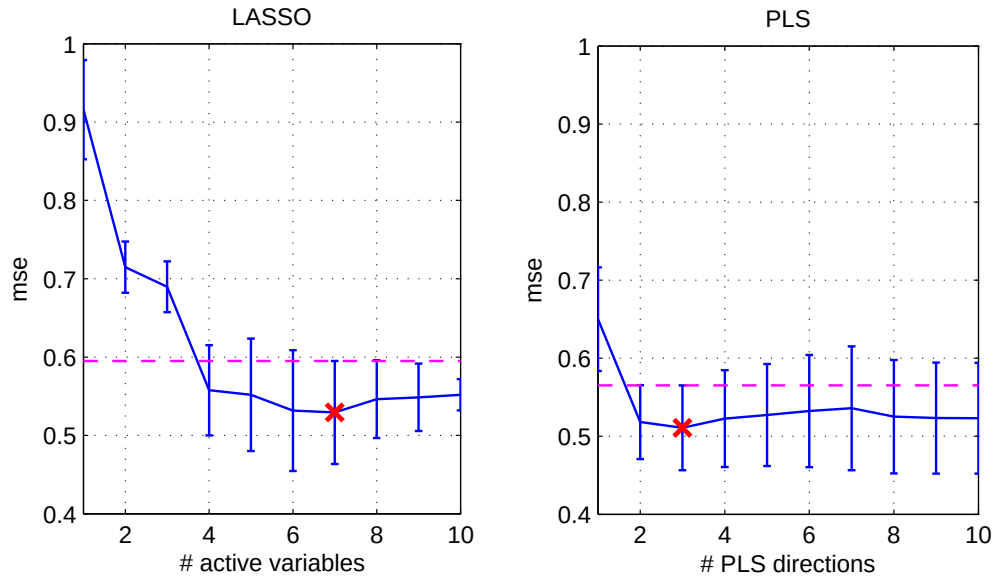

Figure 6.10: LASSO (*left*) vs PLS (*right*) test error curve. In blue the test curve with its confidence intervals, obtained using 7-fold cross-validation. The red cross indicates the minimum value and the magenta dashed line represents the one standard error rule limit.

From Figure 6.10, it can be noticed that LASSO error curve starts with an higher value then the OLS ones. The reason is that, while the LASSO only component is one of the original input variables, the first PLS direction may include the information contained in all the original input variables. With a similar reasoning, the LASSO curve is slower and need more steps to reach the minimum than the PLS curve.

A LASSO estimation of the vector parameters with four active variables has

been computed and its value shown in Table 6.2.

	age	**sex**	**bmi**	**bp**	**sm1**	**sm2**	**sm3**	**sm4**	**sm5**	**sm6**
OLS	0.009	-0.167	0.318	0.190	-0.765	0.498	0.157	0.184	0.485	0.063
PLS	0.022	-0.126	0.297	0.192	-0.069	-0.088	-0.163	0.084	0.216	0.119
LASSO	0	0	0.301	0.097	0	0	-0.091	0	0.210	0

Table 6.2: Estimated OLS, PLS and LASSO coefficients.

As said before, using the OLS estimator, the coefficients of the high correlated variables tend to grow large in opposite directions compensating each others. It was the case of **sm1** and **sm2**, which are positively correlated, and of **sm3** and **sm4**, which are negatively correlated. Using the LASSO estimation technique, **sm1** and **sm2** are not active variables, while only **sm3** is. Hence, we can conclude that LASSO selects the variables in such a way to avoid the inclusion of redundant information. Now that OLS (Chapter 4), PLS (Chapter 5) and LASSO estimator has been described, their performance on the test set may be compared.

As shown in Figure 6.11, there is not a significant difference between the predictions. This may be due to the small dimension of the data set and to the absence of strong correlated variables. However, it must be observed that LASSO has almost the same performance of OLS, using only four regressors (the active variables), while OLS uses all the variables. In the detail of the same Figure, we show an example of the regularisation introduced by LASSO, that usually exhibits smoother and flatter profiles then OLS and PLS.

To quantify the performance of the three methods, the MSE indicator is considered, as shown in Table 6.3. Table 6.3 confirms that the estimators have similar performances.

	MSE
OLS	2842
PLS	2873
LASSO	3021

Table 6.3: MSE indicator for OLS, PLS and LASSO on test data.

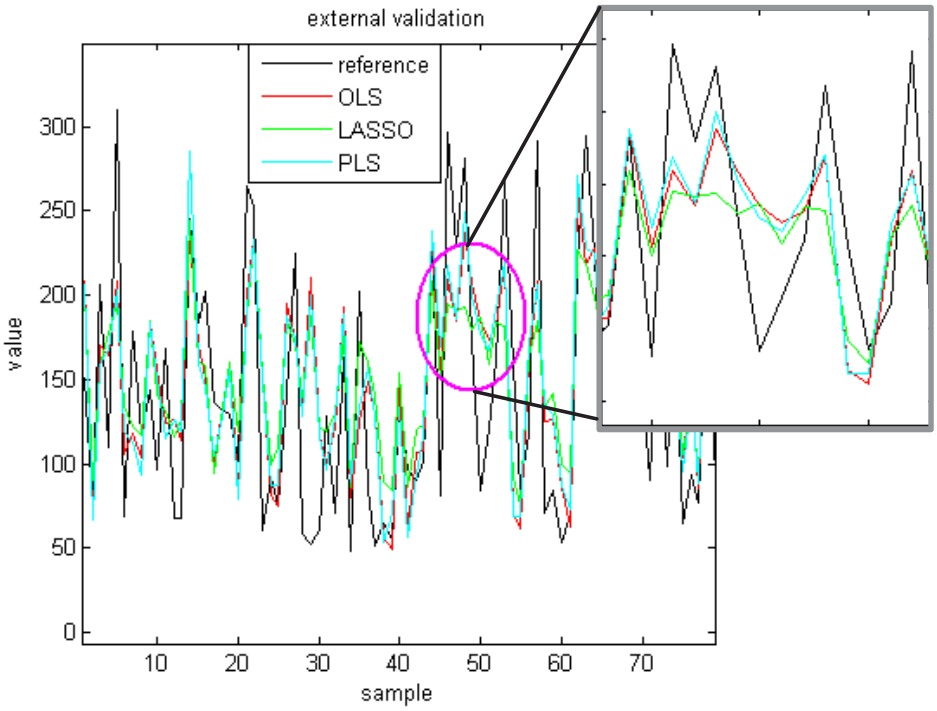

Figure 6.11: External validation using OLS, PLS and LASSO (first 76 samples and a detail in top right edge). In black the reference values, in red OLS predictions, in blue PLS predictions and in green LASSO predictions.

6.6.2 Example 2 (Simulated data)

As said in Chapter 4, the reference data for this second example were generated simulating glucose profiles with different time trends. From these profiles, twenty-seven input variables were obtained simulating Multisensor data, therefore exhibiting high correlation and including confounding processes such as body temperature.

The training data were simulated using a sequence of three glucose profiles, each having a length of eight hours and showing one or two *glycaemic* peaks. While input variables had an elevated sampling frequency (3 sample/minute), reference data were collected approximately every 15 minutes.

In Section 4.4.2, we described how to form the matrix $\boldsymbol{X}$ and the vector $\boldsymbol{y}$ involved in the regression problem. In the previous Chapters, OLS and PLS were presented and commented. Here the LASSO estimation will be illustrated and

compared with OLS and PLS.

As described before, the LAR procedure allows to create the entire LASSO path, i.e. the behaviour of the coefficients $\boldsymbol{\beta}$ as a function of the model complexity. In Figure 6.12, the first part of the LASSO path is shown.

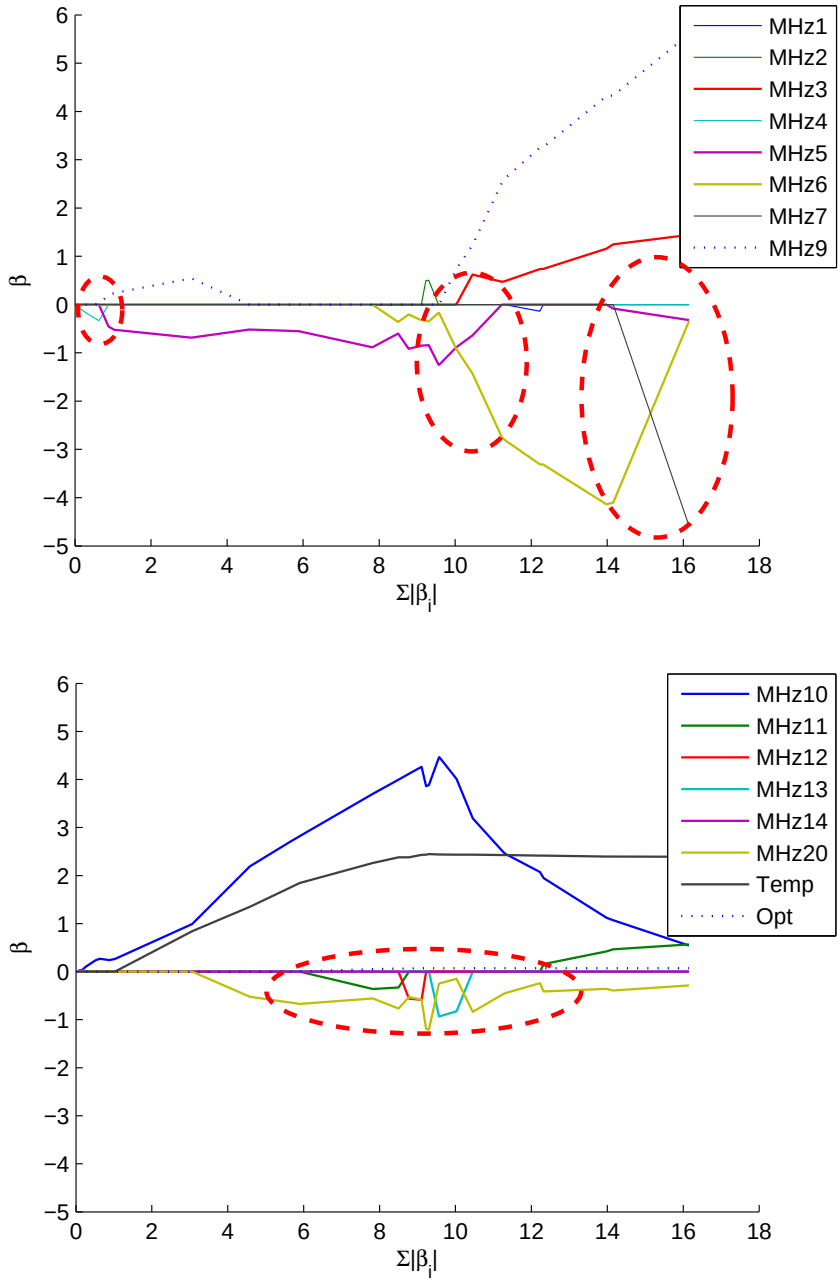

Figure 6.12: LASSO path for simulated data in example 2 (first 18 variables that enter the active set).

In the previous example, it were already noticed that the highly correlated variables are alternatively present, i.e. when the variable that enter the active set is highly correlated with another active variable, the absolute value of the coefficient associated with the last variable decreases, while the coefficient of the new active variable grows large. In Figure 6.12 some examples of this phenomenon are highlighted in the red ellipses.

As LASSO estimation is not scale invariant the matrices $\boldsymbol{X}$ and $\boldsymbol{y}$ have to be normalised before applying the algorithm. In the previous Chapters, the data has been also normalised, allowing a direct comparison of the estimated coefficients.

Before estimating the LASSO coefficients we have to choose the model complexity, as in Example 1. Model complexity was fixed by using the test error curve, estimated from 5-fold cross-validation.

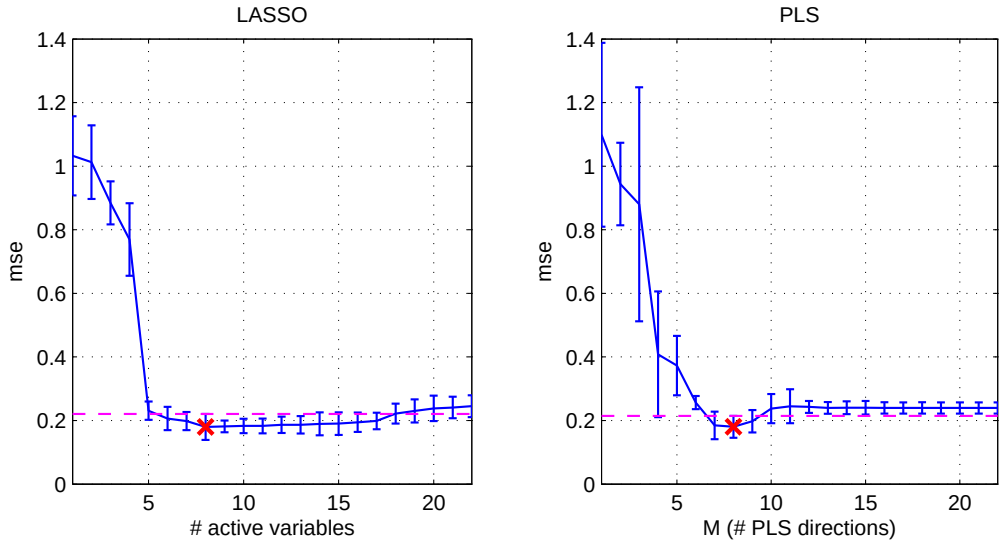

Figure 6.13: LASSO (*left*) vs PLS (*right*) test error curve. In blue the test curve with its confidence intervals, obtained using 5-fold cross-validation, the red cross indicates the minimum value and the magenta dashed line represents the one standard error rule limit.

In Figure 6.13 cross-validation curve is shown. As in Example 1, instead of using the *mean* operator to average the MSE values obtained during the cross-validation, the *median* operator and the *mean absolute deviance* are used.

Figure 6.13 compares the LASSO error curve with the PLS ones. In this case the two curves are very similar. However, while the LASSO curve presents a significant decrease when the fifth variable enters the active set, the PLS has a

more regular decreasing behaviour. Using the LASSO test error curve in Figure 6.13, the finally chosen model has six active variables.

A LASSO estimated vector with six active variables has been computed and its coefficients are shown in Table 6.4.

	MHz1	MHz2	MHz3	MHz4	MHz5	MHz6	MHz7	MHz8	MHz9
OLS	39.974	0.000	0.000	0.000	0.000	3.381	-8.656	-1.951	-6.151
PLS	-0.250	0.398	0.438	0.137	-0.636	-0.583	-0.337	0.002	1.471
LASSO	0.000	0.000	0.000	0.000	-0.887	0.000	0.000	0.000	0.000

	MHz10	MHz11	MHz12	MHz13	MHz14	MHz15	MHz16	MHz17	MHz18
OLS	-14.849	40.897	26.916	-24.356	2.557	-5.082	-23.234	54.732	-53.089
PLS	2.235	0.411	-0.168	-0.210	0.046	0.595	0.485	0.322	0.110
LASSO	3.702	-0.363	0.000	0.000	0.000	0.000	0.000	0.000	0.000

	MHz19	MHz20	MHz21	MHz22	MHz23	MHz24	MHz25	Temp	Opt
OLS	-4.424	-27.901	0.000	1.410	4.014	1.543	-12.167	2.397	0.071
PLS	-0.748	-1.357	-0.250	0.398	0.438	0.137	-0.636	2.429	0.074
LASSO	0.000	-0.556	0.000	0.000	0.000	0.000	0.000	2.262	0.052

Table 6.4: Estimated OLS, PLS and LASSO coefficients. Highlighted the six active variables found by LASSO.

As discussed in the previous example, the OLS coefficients associated to high correlated variables (**MHz**) show elevate magnitude with opposite signs. As noticed in the previous example, only few of these variables enter the active set with LASSO (four in our case). Using LASSO, the variables describing the confounding processes enter the active set as well. This observation proves that LASSO selects the variables avoiding to include unnecessary highly correlated variables and preferring those containing independent information.

The performance of the three different methods was evaluated using a test set, composed by two simulated glucose profiles. Hence, the previously estimated coefficients are applied on inputs of the test set and the result is compared with the test reference. The predictions for all methods are plotted in Figure 6.14.

The prediction can also be calculated in the time instants that have not the corresponding reference, combining the measured inputs of the test set using the

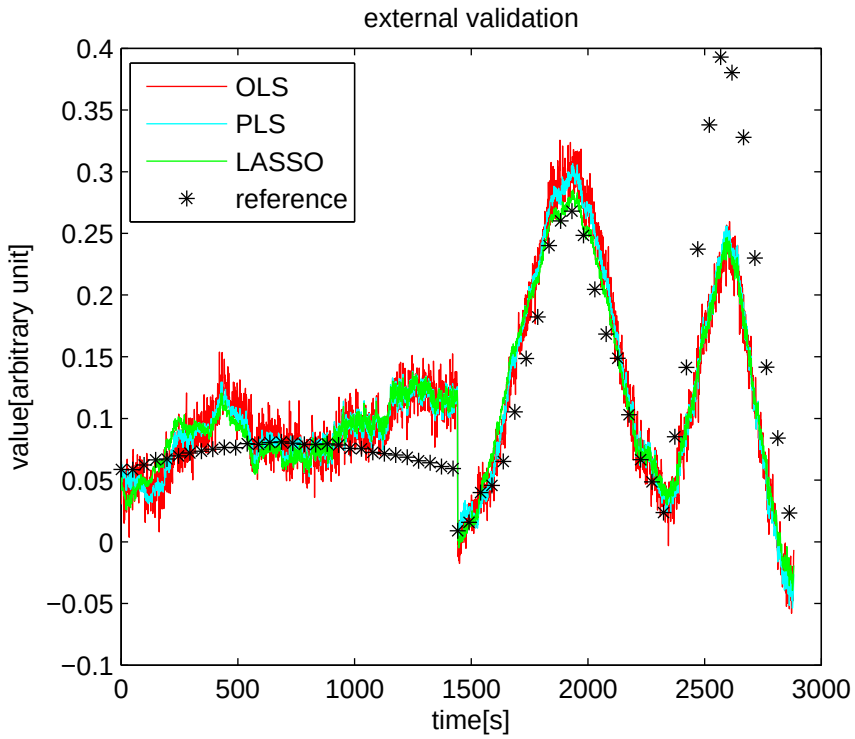

Figure 6.14: External validation using OLS (red), PLS (cyan) and LASSO (green). In black stars the reference values.

estimated coefficients. However, the prediction at these time instants cannot be used for calculate the indicators described in Chapter 3 for model assessment.

Visually, the methods have similar performances. However, it must be taken into account that LASSO achieves results similar to the other methods, using only 6 variables, even less than PLS (which employs 7 directions). In addition, LASSO is less noisy than OLS and fits the fist peak better than the other two methods.

To quantify the performance of the two methods the MSE indicator was used, as shown in Table 6.5.

	MSE
OLS	0.0032
PLS	0.0030
LASSO	0.0029

Table 6.5: MSE indicator for OLS, PLS and LASSO on test data.

In this case, LASSO has the best performance. This confirms that OLS likely suffers from overfitting, while LASSO seems to select the correct number of variables, allowing a better generalisation of the estimated model on unseen data.

6.7 Concluding Remarks

LASSO is a technique for the estimation of multivariate linear regression models based on regularisation In particular, the term added to the function that has to be minimized, prevents the model coefficients from assuming large values, since it penalises the sum of the coefficient absolute values. This particular kind of regularisation causes some of the coefficients to be exactly zero, leading both to sparse solutions and to an intrinsic method for variables selection.

Since the LASSO minimisation function is not differentiable, the solution cannot be computed in closed form. Hence, the solution has to be estimated using iterative methods. Different algorithms have been developed to compute the LASSO solution in an efficient way. In particular, with a small modification of the LAR algorithm, the entire LASSO path can be obtained in an extremely efficient way, making this algorithm an attractive method for solving the LASSO problem.

This regularisation technique tries to avoid the OLS problem of overfitting. As observed in the previous examples, in the initial part of the LASSO path, the cancellation effect of highly correlated variables is avoided. Using the OLS estimation technique, the coefficients of highly correlated variables tends to become large in opposite directions, cancelling their contribution to the target estimation. Contrarily, using LASSO estimator, the coefficient of the new active variable grows, while the coefficient of the correlated active variable tends to decrease to zero. At this point the variable is taken out from the active set. Hence, there is an exchange of the highly correlated variables. This advantage disappears as the complexity parameter λ moves to zero (i.e. the coefficients are less shrunk and more variables are allowed to enter the active set), approaching the OLS solution.

Chapter 7

Modeling the Solianis Multisensor Data: Experimental Protocol and Dataset Description

In Chapter 2 we introduced the Solianis Multisensor data and highlighted the necessity of a model for estimating glucose from Multisensor signals. In the successive Chapters, the regression problem and three candidate regression techniques, OLS, PLS and LASSO, for solving it were illustrated. In Chapter 8, these regression techniques will be applied for modeling Solianis Multisensor data, whose features are described in the present Chapter.

7.1 Acquisition Protocol and Dataset Composition

Data was acquired during an experimental clinical study that included four patients with Type 1 *Diabetes Mellitus* (T1DM), identified in our whole data analysis using the following labels: "AA06", "AA14", "AA17" and "AA18". Each subject performed four study visits. During each study visit, a different glucose profile was measured in controlled conditions using two different Solianis Multisensors, one attached to the upper left arm and the other to the upper right arm. Along with the Multisensor signals, some BGL references, collected using finger-stick methods, were recorded. Hence, for each subject, a total of 8 glucose profiles ("runs" of a duration of approximately 8/9 hours) were acquired. These

runs are somehow coupled, having the same reference but different Multisensor signals (one from right arm and the other from the left arm). While the Multisensor acquires the signals every 20 second, the BGL reference are collected approximately every 15 minutes and whenever necessary for medical purposes. The measurements in the first 75 minutes after the Multisensor was attached to the skin are not taken into account in our analysis since they suffer from artefacts due to physiological adjustment of the skin to the presence of the sensor.

As mentioned in Chapter 2, the Multisensor provides a set of measurements of different nature, mainly based on dielectric and optic sensors, for a total of more than 150 measured signals. Most of the signals come from the dielectric electrodes, showing a high correlation and exhibiting similar but not identical behaviour. Hence, there are two important characteristics of this dataset: it is a high-dimensional dataset and there are many correlated variables in it. Another feature to be taken into account for dataset handling is the presence of doubled references, i.e. the same reference is connected to two different Multisensor signals sets.

7.2 Organization of Data for Training and Validation

As said in Chapter 3, it is a good choice to evaluate the performance of the different methods using unseen data. Hence, it is worth describing how the available data have been handled.

Our choice was to use 16 runs as training set and the other 16 as test set. Since each experiment is present twice (with same reference BGL but different Multisensor signals), it may sound a good choice to use the left arm data in the training set and the right arm data in the test or *viceversa*. However, in this way a bias in the results would be introduced, since the data forming the test set would not be truly unseen. As a consequence, the runs from the Multisensors measured during the same experiment have been mostly assigned to the same set.

In this case, a global model is "learned" from 16 runs forming the training set. After this procedure, we obtain an estimation of the coefficients characterising the model, which is the same for every subject (global model). These coefficients are then used on the test set to predict the BGL values and to evaluate the

performance of the estimators.

A further analysis has been performed in order to assess if a global model is a good choice. In particular, the model is learned using the run associated with three patients (for a total of 24 runs) and the estimated model is applied on the fourth patient (8 runs), which is thus used as test set. In this way, we want to determine if the estimated model can be used for data of a totally new, unseen, subject. In the next Chapter, we will refer to this combination of training/test set as "cross-validation" with leave one subject out, see in particular Section 9.1.

7.3 *Rationale* of the Analysis

After having described how the data will be handled, in this Section we want to briefly illustrate the *rationale* of the procedure, which will be followed to analyse data and compare the results reported in Chapter 8.

In order to compare directly the coefficients estimated using the different methods, data have been normalized, since both PLS and LASSO are not scale invariant. Supposing that there exist a true value of the mean and the standard deviation of the signals measured by the Multisensor, they are estimated by considering only the training set, and will then be used to standardise both the training and the test set. In this way, the model prediction of the test set can be calculated without knowing the entire test signals, allowing an on-line prediction.

For the OLS method, the training set have been used to learn the model coefficients, which are then applied on the test set to estimate the BGL.

For both PLS and LASSO an additive step is needed, since before the learning procedure, the model complexity has to be selected. As a consequence, the training set is also used in K-fold cross-validation to estimate the model complexity (see Section 3.3.2). It is worth saying that our choice was to select K equal to the number of runs forming the training set. In this way, each group in the cross-validation procedure contained approximately the same number of samples as a single run. After having estimated the model complexity, both PLS and LASSO coefficients have been learned from the training set and applied on the test set for predicting the BGL values.

After this preliminary analysis, which is mainly used for comparing the performance of the different estimation techniques, an additional analysis is performed

in order to determine if a global model is sufficient or a specific model for each subject is needed. This further analysis is performed by using the best of the three previously considered techniques only and is implemented using the cross-validation with leave-one-subject-out.

Finally, we anticipate that two different calibration techniques, which may be used to improve the glucose estimates accuracy, will be also described and evaluated in Chapter 9.

Chapter 8

Application of OLS, PLS and LASSO to the Solianis Multisensor Data

In this Chapter the results of the application of OLS (Chapter 4), PLS (Chapter 5) and LASSO (Chapter 6) in modeling the Solianis Multisensor data will be shown, commented and compared.

8.1 OLS Results

In this Section we will discuss the results obtained applying the OLS estimator to the Solianis Multisensor data.

8.1.1 Internal Validation (Model Learning)

In Figure 8.1 and 8.2 is shown the target and the model estimates in internal validation. We can notice that the glucose estimate obtained with OLS well approximates the target. However, it is easy to see that estimates result very unstable and exhibit a lot of oscillations. Figure 8.3 shows a representative run where we can appreciate how the prediction reproduces the glucose profile and its oscillation between two target points. This phenomenon can be explained if we take into account that we are considering the training set, where the OLS coefficients are estimated to minimise the distance of the prediction from the reference.

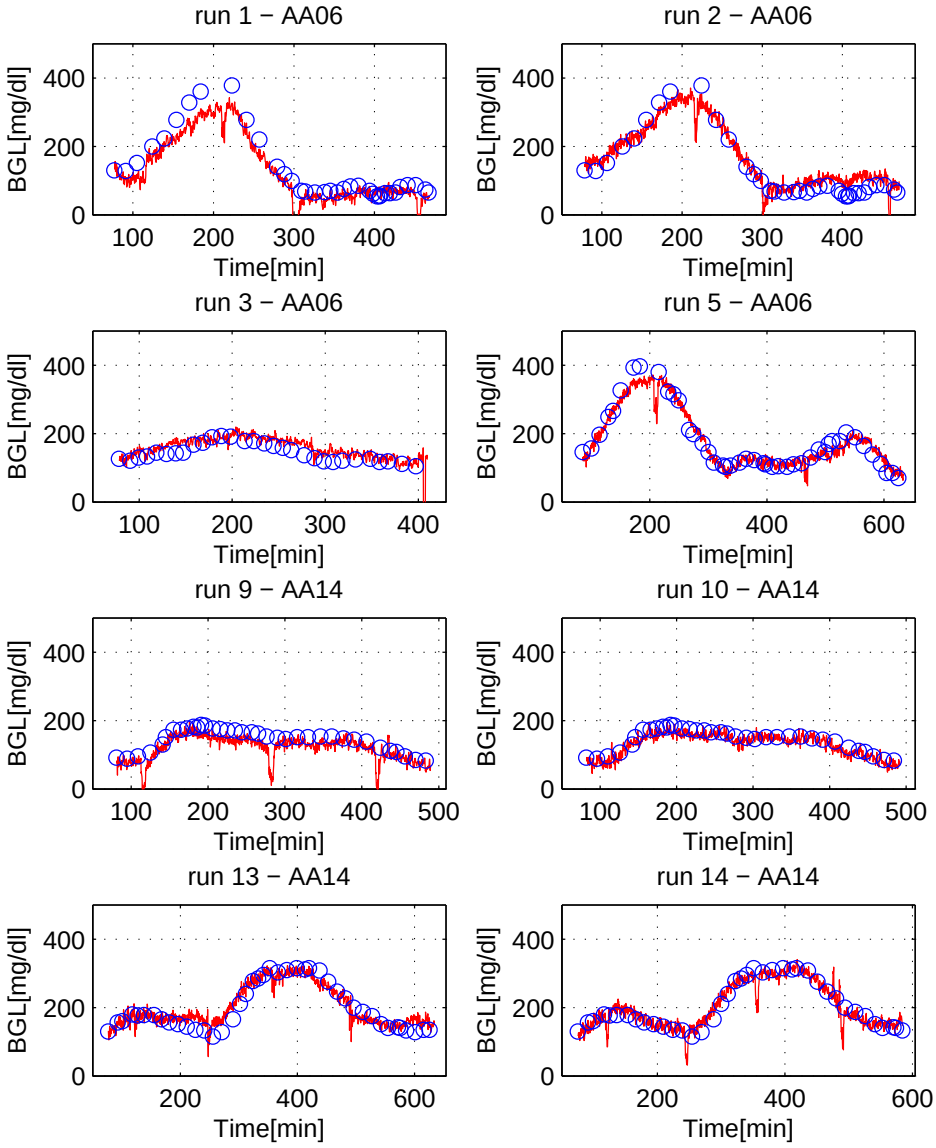

Figure 8.1: Internal validation for the OLS estimator. OLS model prediction(red) vs. reference BGL(blue circles). The first 8 runs of the training set are shown.

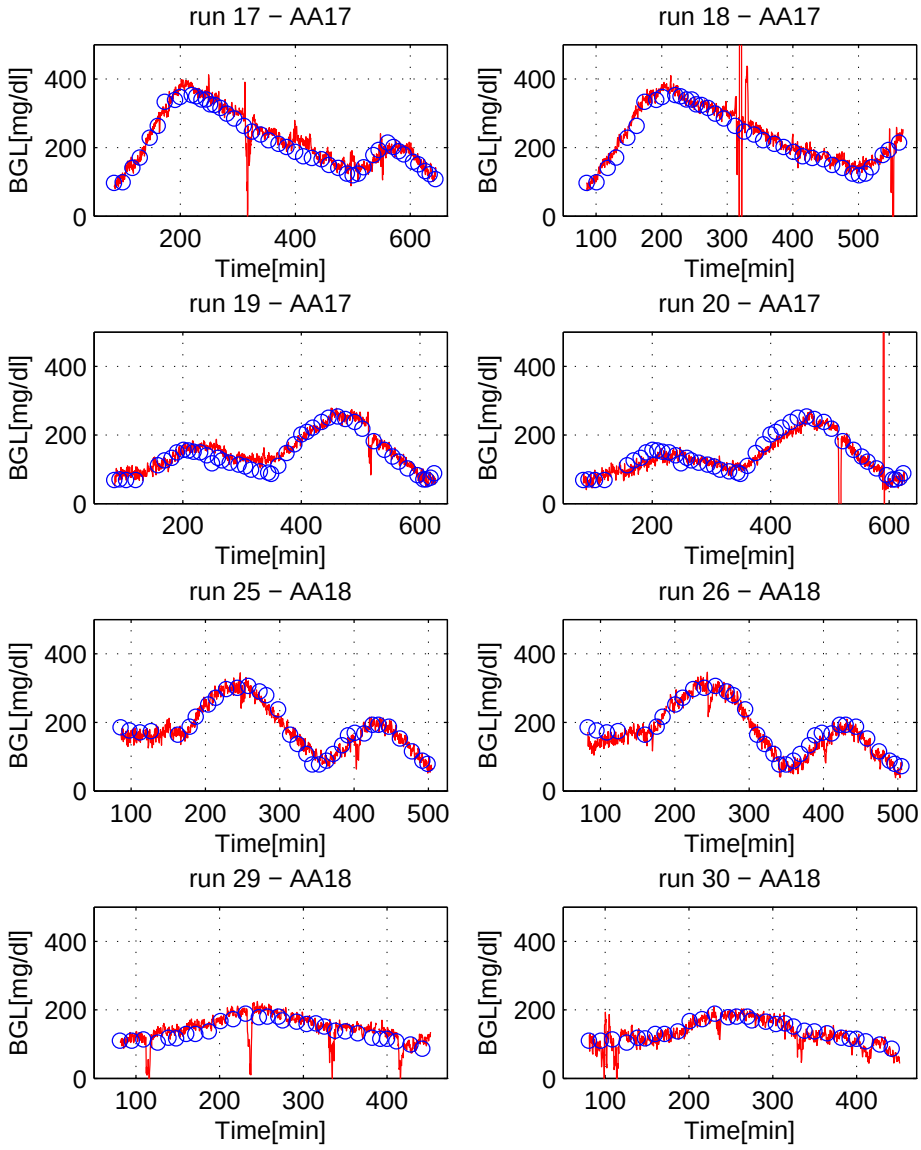

Figure 8.2: Internal validation for the OLS estimator. OLS model prediction(red) vs. reference BGL(blue circles). The last 8 runs of the training set are shown.

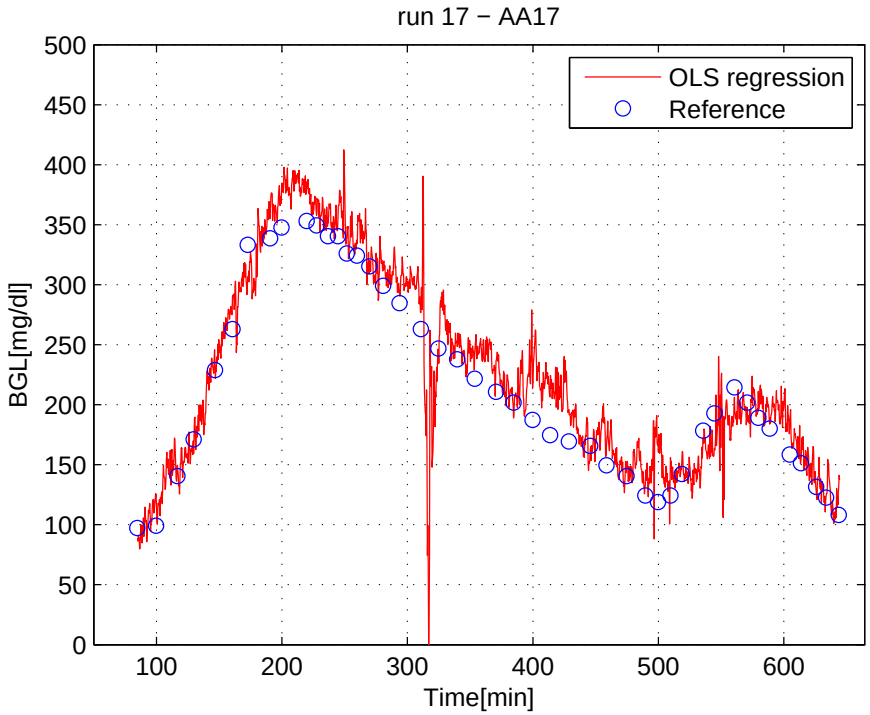

Figure 8.3: Representative run showing OLS in internal validation. OLS model prediction(red) vs. reference BGL(blue circles).

Since the Multisensor data are composed by a large number of variables, it is not convenient reporting all the estimated OLS coefficients. It is more meaningful to consider only those whose absolute values are greater then an arbitrary threshold. In this way, we can check to which variables are associated the larger coefficients. In Table 8.1 are reported the coefficients whose absolute values are larger then 10.

	GHz_el2_3	**GHz_el2_4**	**GHz_el2_7**	**GHz_el2_8**	**MHz_el1_7**
OLS	-44.113	47.876	-41.308	51.912	15.168
	MHz_el1_9	**MHz_el1_10**	**MHz_el1_21**	**MHz_el1_25**	**MHz_el1_26**
OLS	-32.792	29.433	-10.352	-10.515	13.524
	MHz_el2_8	**MHz_el3_5**	**MHz_el3_8**	**MHz_el3_9**	
OLS	11.373	-10.471	11.208	-10.990	

Table 8.1: OLS coefficients whose absolute values are greater then 10.

As shown in Table 8.1 the largest OLS coefficients have opposite signs and are associated to signals of the same type. This behaviour of the estimated model

coefficients obtained with OLS was observed in the tutorial examples in Chapter 4. Indeed, it was noticed that this behaviour arises when OLS deals with highly correlated variables.

An example of such compensation is reported in Figure 8.4, where **MHz_el1_10** and **MHz_el1_9** are plotted along with the BGL reference. In the Figure all the signals have been centered and normalised to allow a better visual comparison. As it can be noticed from Figure 8.4, the two Multisensor variables are almost identical and from Table 8.1 one can verify that their corresponding OLS estimated coefficients assume large absolute value and opposite sign, causing the cancellation of their contribution to the target estimation.

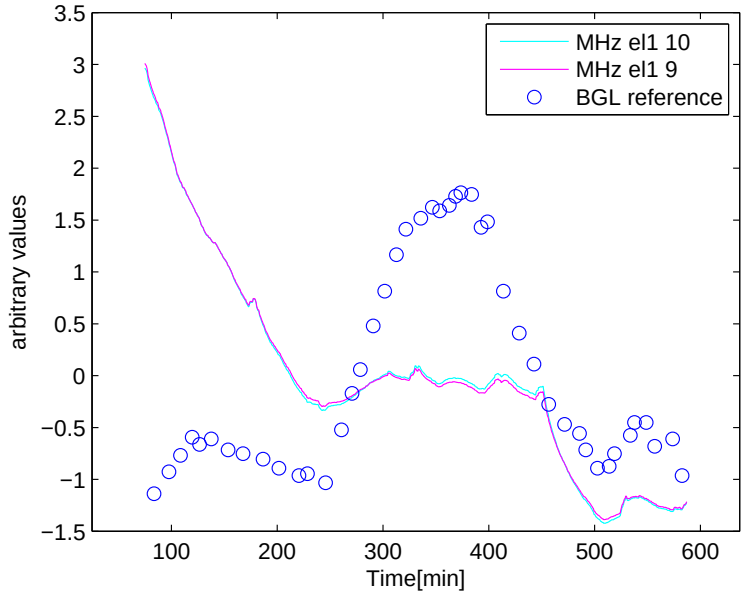

Figure 8.4: Example of (normalised) highly correlated variables(continuous lines), having OLS estimated coefficients with large absolute value and opposite sign.

8.1.2 External Validation (Model Test)

Another characteristic of OLS, which is more remarkable when dealing with high dimensional dataset, is the overfitting. In this case, the estimated predictive model fits not only the information yielded by the training set, but also to the

noise contained in it. To prove the predictive performance of the model, a test set is used and the results are shown in Figure 8.5 and 8.6.

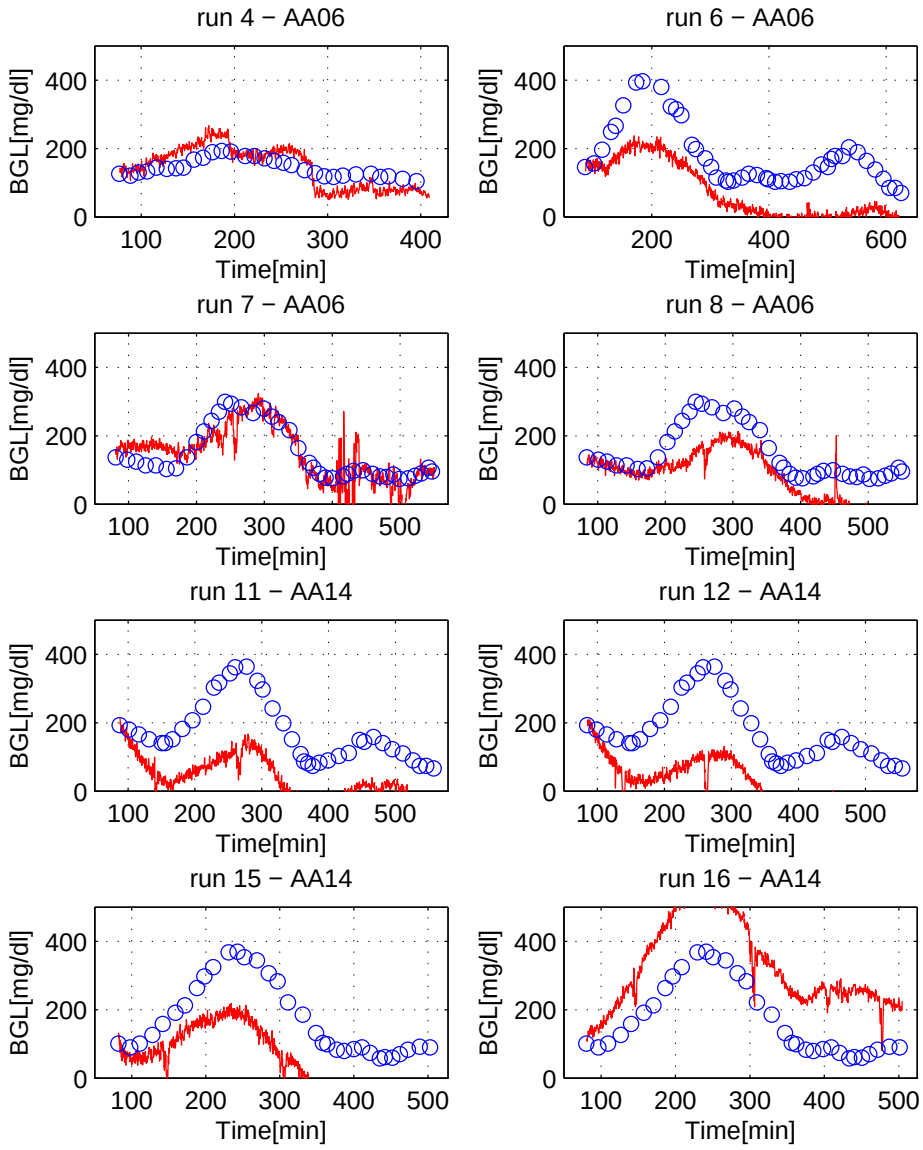

Figure 8.5: External validation of OLS. OLS model prediction(red) vs. reference BGL(blue circles). The first 8 runs of the training set are shown.

From Figure 8.5 and 8.6 it can be noticed that the model estimates using

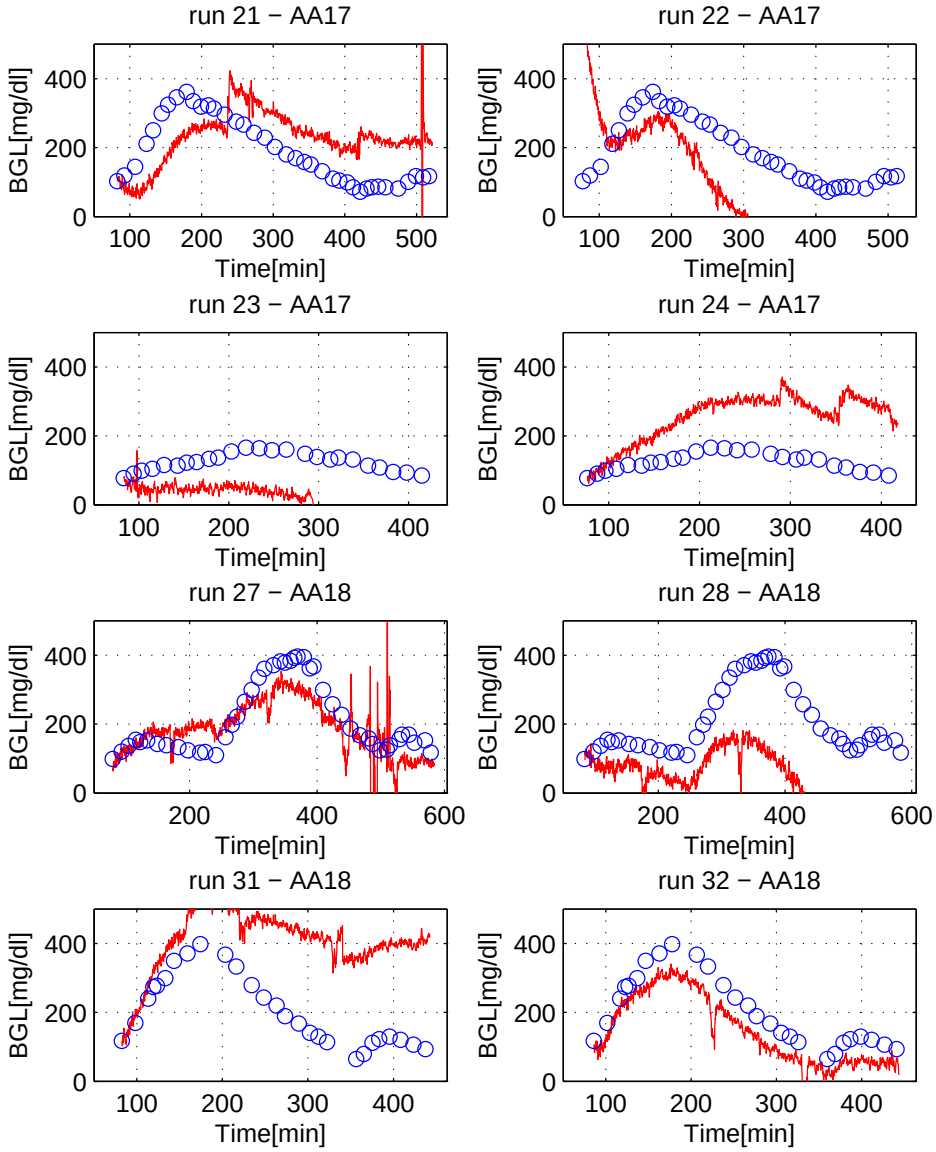

Figure 8.6: External validation of OLS. OLS model prediction(red) vs. reference BGL(blue circles). The last 8 runs of the training set are shown.

OLS fails in most cases in predicting unseen data, proving that probably OLS estimated coefficients are too much fitted to the training set.

8.2 PLS Results

In this section we will discuss the results obtained applying the PLS estimator to the Solianis Multisensor data.

8.2.1 Internal Validation (Model Learning)

In this case, the PLS cannot directly be applied since model complexity, represented by the number of PLS directions (i.e. M), has to be determined by using the cross-validation procedure. Before analysing the test error curve obtained using cross-validation, it is worthwhile to analyse the training error curve (internal validation).

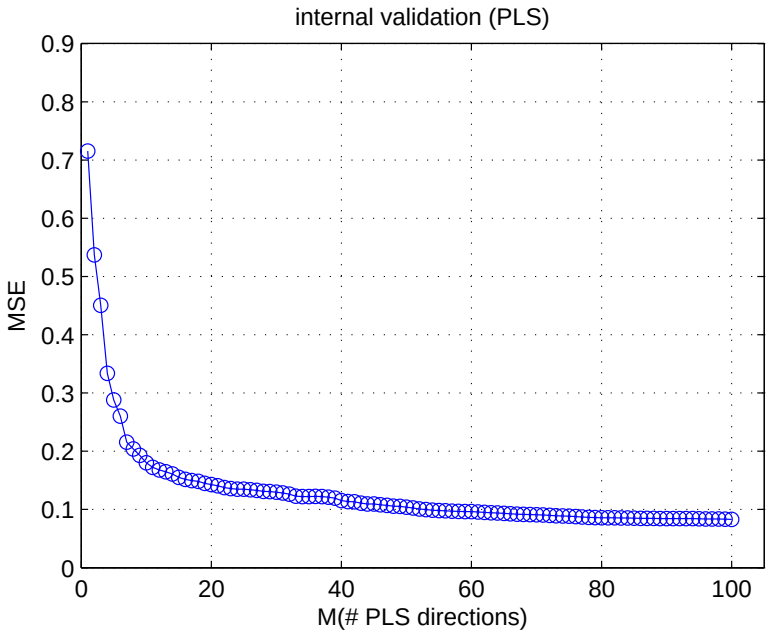

Figure 8.7: MSE on the training set as a function of the number of PLS directions (internal validation).

In Figure 8.7 we plot the behaviour of the training error as a function of the number of PLS directions. A considerable decreasing of the error in the initial part of the curve, followed by an almost flat zone can be noticed. As a consequence, it seems reasonable that only few PLS are sufficient to obtain an acceptable approximation of the training set. Hence, it may be sufficient to estimate the test error curve between 1 and 60 PLS directions.

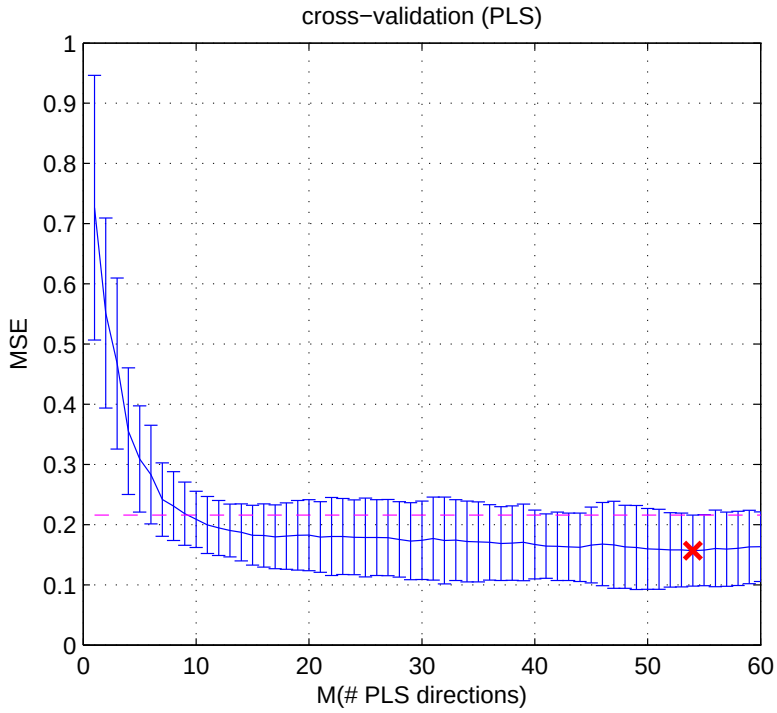

Figure 8.8: MSE on the test set as a function of the number of PLS directions (cross-validation). The red cross indicates the minimum value and the magenta dashed line represents the one standard error rule limit.

In Figure 8.8 the test error curve is shown, built using 16-fold cross-validation. In this case, the curve is similar to the one of Figure 8.7 (internal validation). The minimum is placed at 54. However, since the curve is very flat, it is reasonable to use the "one-standard error" rule, whose limit is represented by the magenta line. Hence, the most parsimonious model, whose error is less than the minimum plus its standard deviation, is positioned at 10.

Once the model complexity parameter has been fixed, its parameters can be estimated using PLS.

8.2.2 External Validation (Model Test)

To evaluate the performance of such model, we consider its capability in predicting unseen data. Hence, using the previously estimated parameters, the model is used to predict the test set target and the results are shown in Figure 8.9 and 8.10.

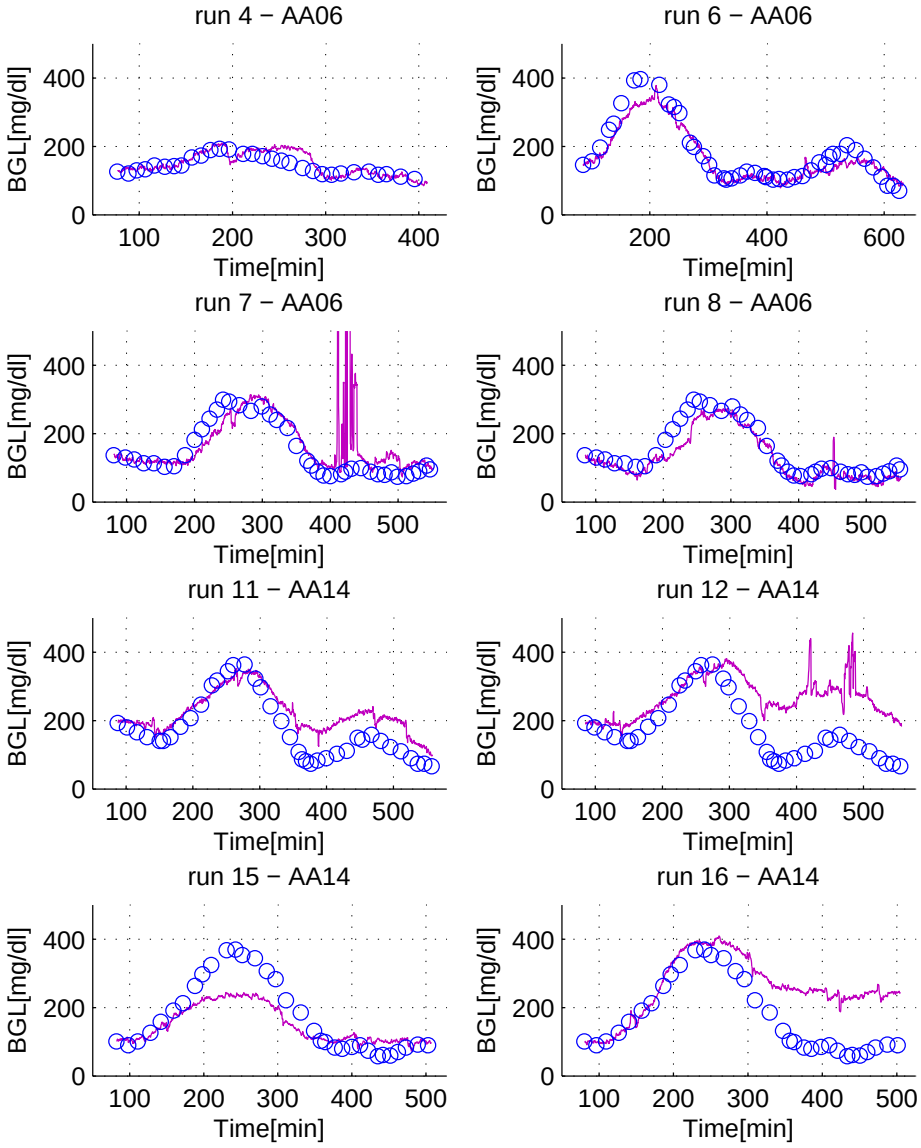

Figure 8.9: External validation of PLS. PLS model prediction(M=10, magenta) vs. reference BGL(blue circles). The first 8 runs of the training set are shown.

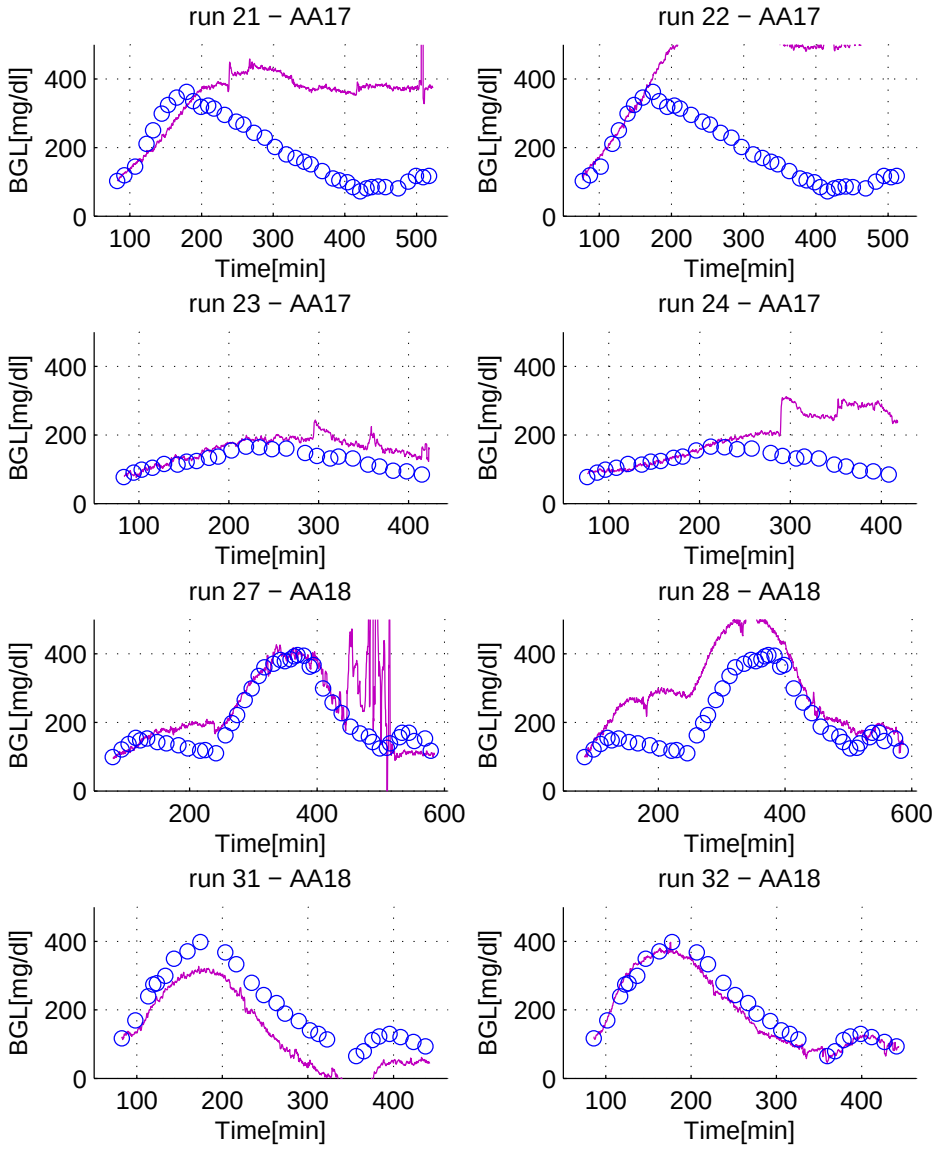

Figure 8.10: External validation of PLS. PLS model prediction(M=10, magenta) vs. reference BGL(blue circles). The last 8 runs of the training set are shown.

From Figure 8.9 and 8.10 it can be observed that, while some runs (for example # 6, 8 and 32) are properly predicted by PLS, other predicted profiles are very noisy (for example # 7, 12 and 27) or have an increasing trend (for example # 16, 21 and 2), demonstrating that selecting M =10 (number of PLS directions) does not allow to generalise over unseen data well.

In addition, for the same reasons explained in the previous Section, Table 8.2 shows the variables and their corresponding coefficients, whose absolute value are greater than 0.1. However, in this case the compensation effect between the variables is not present.

	GHz_el1_3	**GHz_el1_7**	**MHz_el3_22**	**Opt_10**	**Opt_11**	**Opt_12**
PLS	0.206	-0.227	-0.136	-0.117	0.141	-0.221
	Opt_16	**Opt_26**	**Opt_27**	**Ms_Hum**	**Ms_Temp**	**Skin_Temp**
PLS	0.101	-0.308	0.117	-0.206	0.117	0.600

Table 8.2: PLS coefficients whose absolute values are greater then 0.1.

8.3 LASSO Results

In this section we will discuss the results obtained applying LASSO estimator to the Solianis Multisensor data.

8.3.1 Internal Validation (Model Learning)

As for PLS, the LASSO estimator cannot directly be applied, since the model complexity, represented by the number of active variables, has to be determined using the cross-validation procedure. Before analysing the test error curve obtained using cross-validation, it is worthwhile to analyse the training error curve (internal validation).

In Figure 8.11 we plot the behaviour of the training error as a function of the number of active variables. As it happened for PLS, the initial part of the curve is characterised by a considerable decreasing of the error, followed by an almost flat zone. This phenomenon may be due to the presence of many highly correlated variables in the dataset. In fact, many variables bring similar information (as they are highly correlated). Hence, if another variable of this family is included in the active set, it is not able to improve the target approximation. As a consequence, it

seems reasonable that only few variables will be sufficient to obtain an acceptable approximation of the training set and the LASSO procedure may help in selecting the best ones.

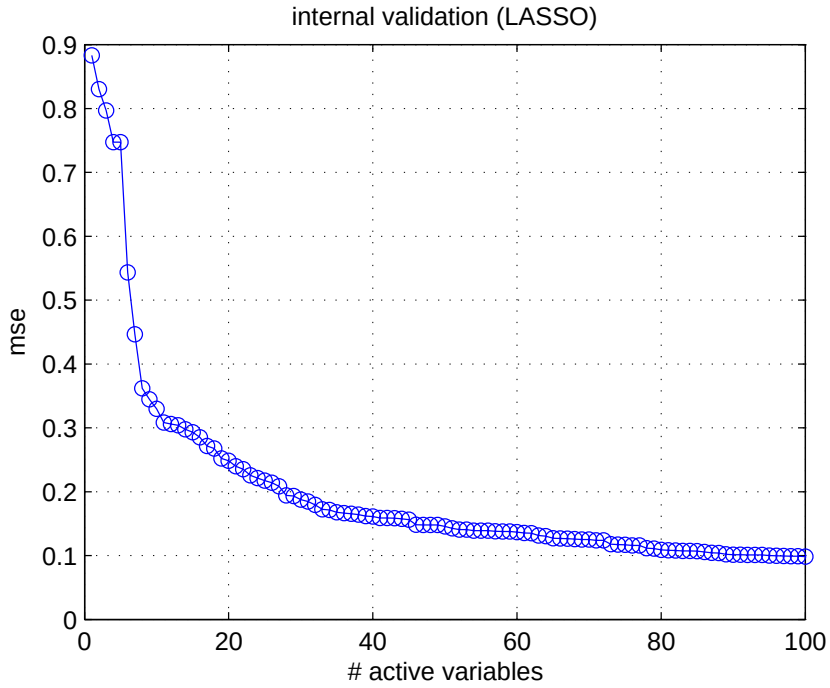

Figure 8.11: MSE on the training set as a function of the number of active variables (internal validation).

For the previous discussion, it seems reasonable to estimate the test error curve between 1 and 60 active variables.

In Figure 8.12 the test error curve, built using 16-fold cross-validation, is shown. In this case, the curve is similar to the one of internal validation (Figure 8.11). The minimum is placed at 55 active variables. However, since the curve is very flat, it is reasonable to use the "one-standard error" rule, whose limit is represented by the magenta line. Using this rule, the best model is the one having 27 active variables. However, at that point the curve is still very flat. Hence, it can be supposed that the best model is even more parsimonious, considering the error in estimating the cross-validation curve. For instance, since the curve exhibits an L-shape, its edge (near 9) can be a good choice.

At this point, it is convenient to consider the LASSO path till 27 variables have entered the active set. The result is shown in Figure 8.13.

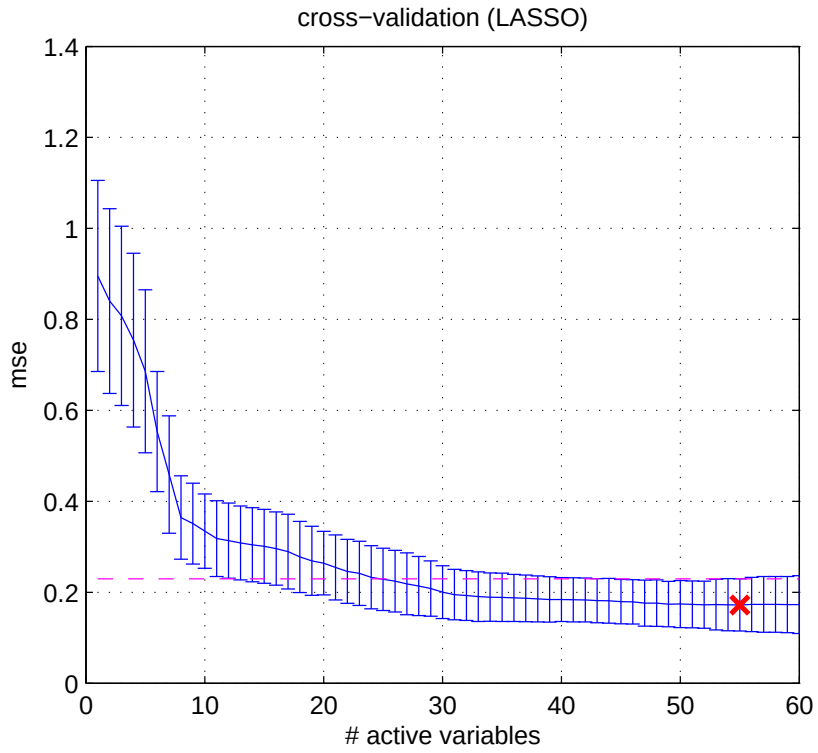

Figure 8.12: MSE on the test set as a function of the number of active variables (external validation). The red cross indicates the minimum value and the magenta dashed line represents the one standard error rule limit.

Along this initial part of the LASSO path, it can be noticed the exchange of highly correlated variables, highlighted in the red ellipses. In particular, this happens when the new variable, which enters the active set, is highly correlated with another of the active variables. In this case, the absolute value of the coefficient associated with the old active variable tends to zero, while the coefficient of the new active variable grows largely. This particularity of the LASSO path avoids the OLS problem, where highly correlated variables have the tendency to assume large absolute values with opposite signs, causing the cancellation of their contribute to the target estimation.

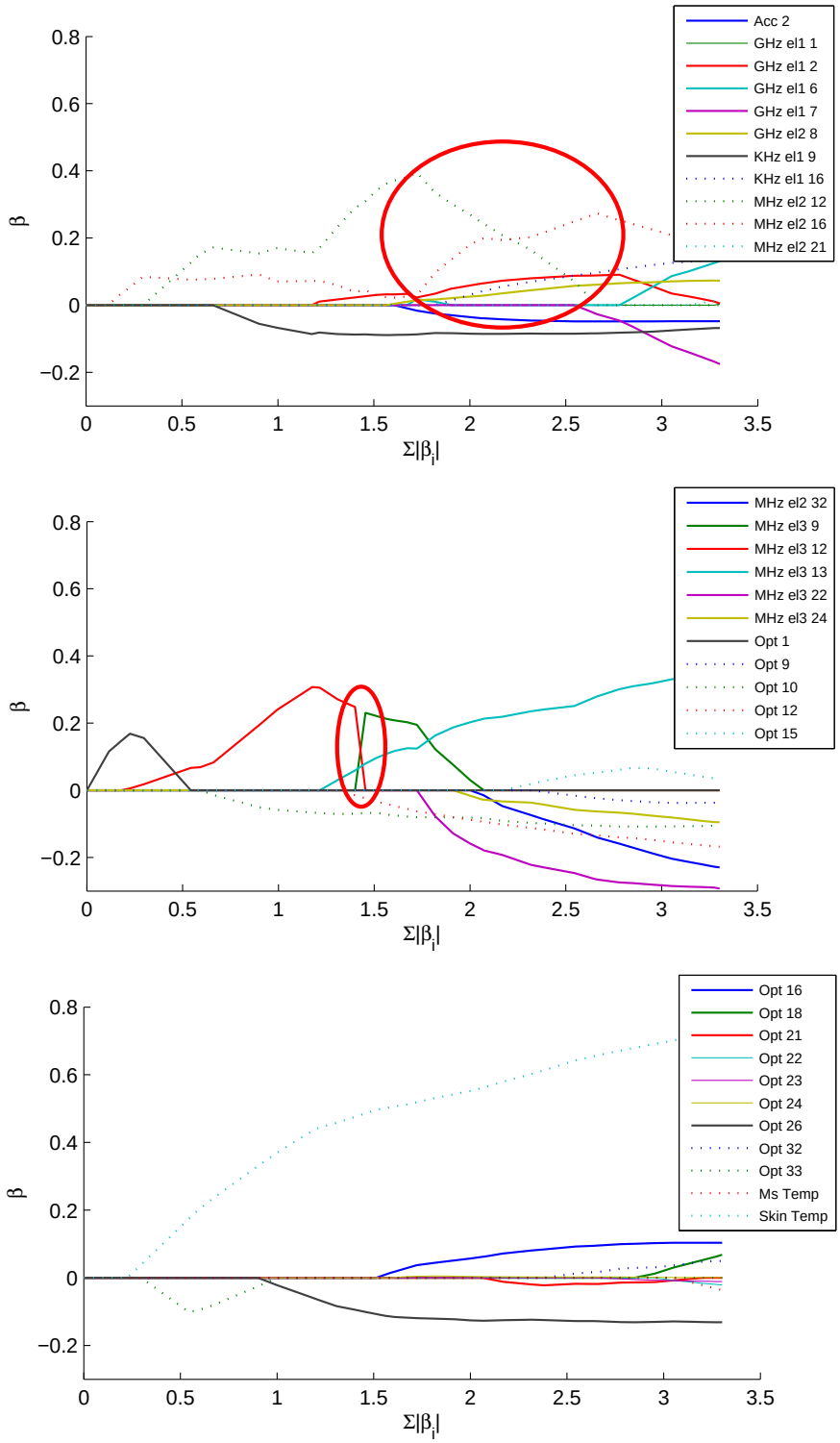

Figure 8.13: LASSO path till 27 variables have entered the active set. For simplicity, the path has been splitted in three subfigures, each containing 11 variables. The remarkable points are highlighted in the red ellipses.

In Figure 8.14 we show the time evolution for the representative run 14 of the variables highlighted in the first red ellipse in Figure 8.13, along with the corresponding BGL references. To allow a more direct comparison, all the signals have been centered and scaled. From Figure 8.14, it can be shown that the two MHz variables are almost identical and bring the same information about the BGL reference. Hence, their exchange along the LASSO path corresponds to a desirable behaviour of their coefficients.

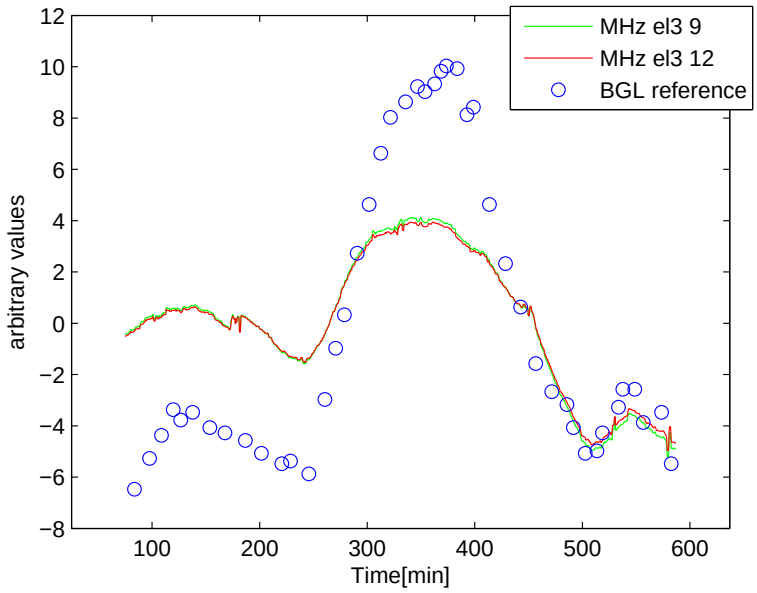

Figure 8.14: Example of highly correlated variables entering (MHz el3 9) and exiting (MHz el3 12) the active set during the LASSO path.

After having analysed the LASSO path, we move back to the choice of the best value for the model complexity parameter. Using LASSO, the parameters of both models (with 27 active variables and with 12 active variables) have been estimated and the results are shown in Figure 8.15 and 8.16.

It can be noticed that, with 27 active variables, the prediction is noisier than with 12. This proves that, in all likelihood, with 27 active variables the estimated model still suffers from overfitting. With 12 active variables, the estimated glucose profiles are very smooth and this may be due to the selection of the less noisy variables and to the greater amount of regularisation introduced in the model with 12 variables than in the model with 27.

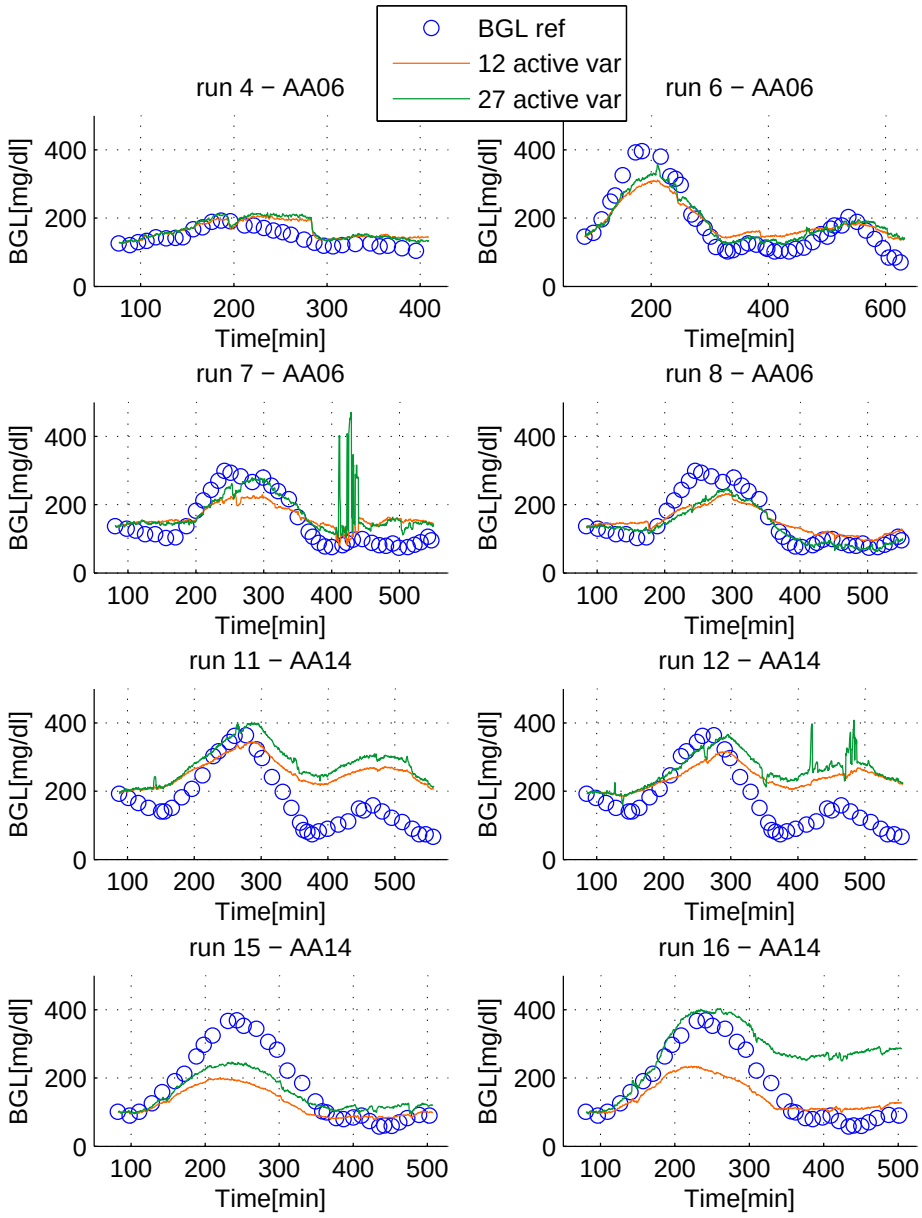

Figure 8.15: External validation for LASSO estimates. Glucose estimates with 27 active variables(green) and with 12 active variables(red). The first 8 runs of the test set are shown.

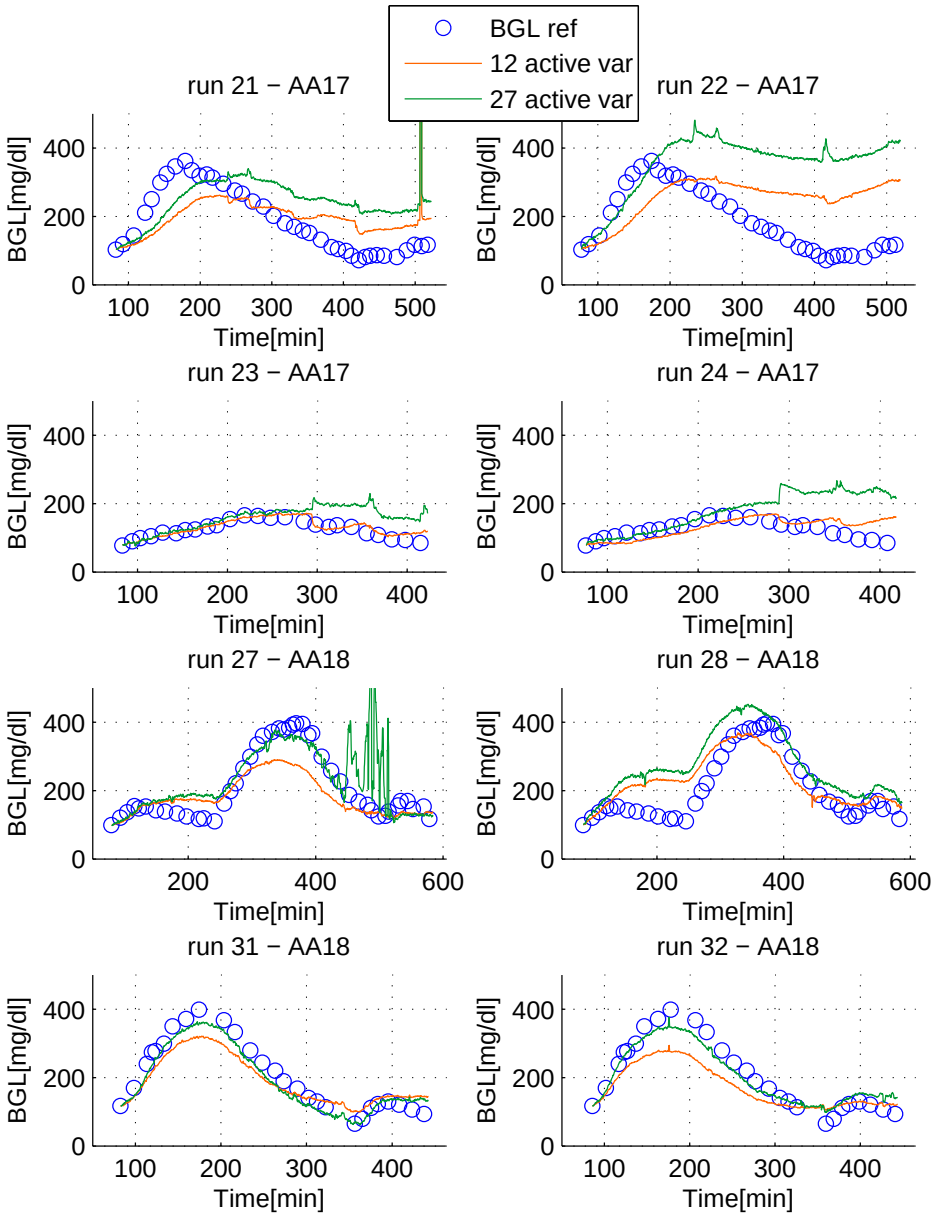

Figure 8.16: External validation for LASSO estimates. Glucose estimates with 27 active variables(green) and with 12 active variables(red). The last 8 runs of the test set are shown.

As a consequence, it seems reasonable to choose 12 as the *optium* number of active variables.

Once model complexity has been fixed, its parameters can be estimated using the LASSO procedure.

8.3.2 External Validation (Model Test)

To evaluate the performance of such model, we consider its ability in predicting unseen data. Hence, using the previously estimated parameters, the model is used to predict the test set target and the results are shown in Figure 8.18 and 8.18.

In this case, the estimated profiles with the LASSO model exhibit better performances by visual inspection, since the BGL estimates are rather smooth and are able to mimic the glucose fluctuations although their flatness.

	GHz_el1_2	**GHz_el2_8**	**KHz_el1_9**	**MHz_el2_12**
LASSO	0.032	0.002	-0.089	0.367
	MHz_el2_16	**MHz_el3_9**	**MHz_el3_13**	**Opt_10**
LASSO	0.023	0.211	0.113	-0.074
	Opt_12	**Opt_16**	**Opt_26**	**Skin_Temp**
LASSO	-0.044	0.015	-0.114	0.504

Table 8.3: LASSO coefficients of the active variables.

In Table 8.3 the estimated LASSO coefficients of the active variables are shown. Notably, no large absolute values are present.

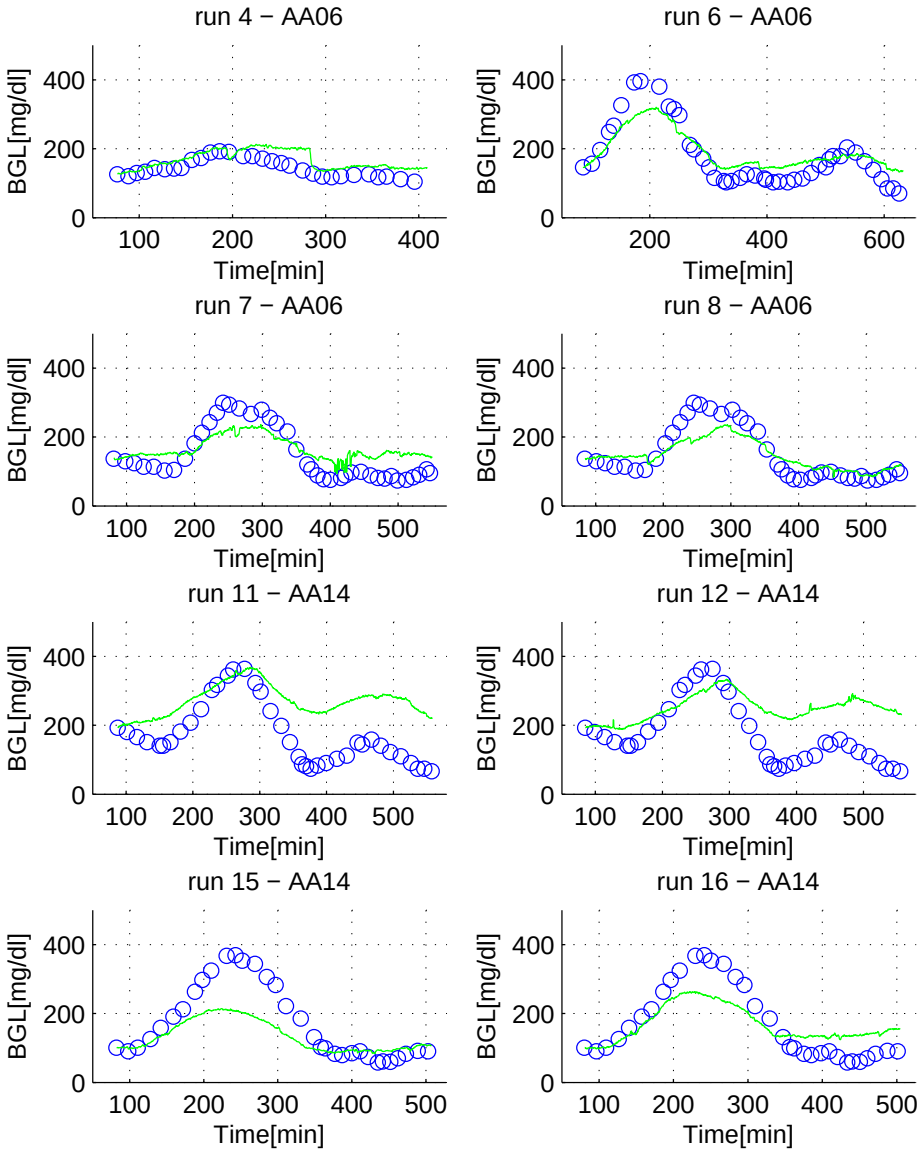

Figure 8.17: External validation of LASSO. LASSO model prediction(12 active variables, green) vs. reference BGL(blue circles). The first 8 runs of the training set are shown.

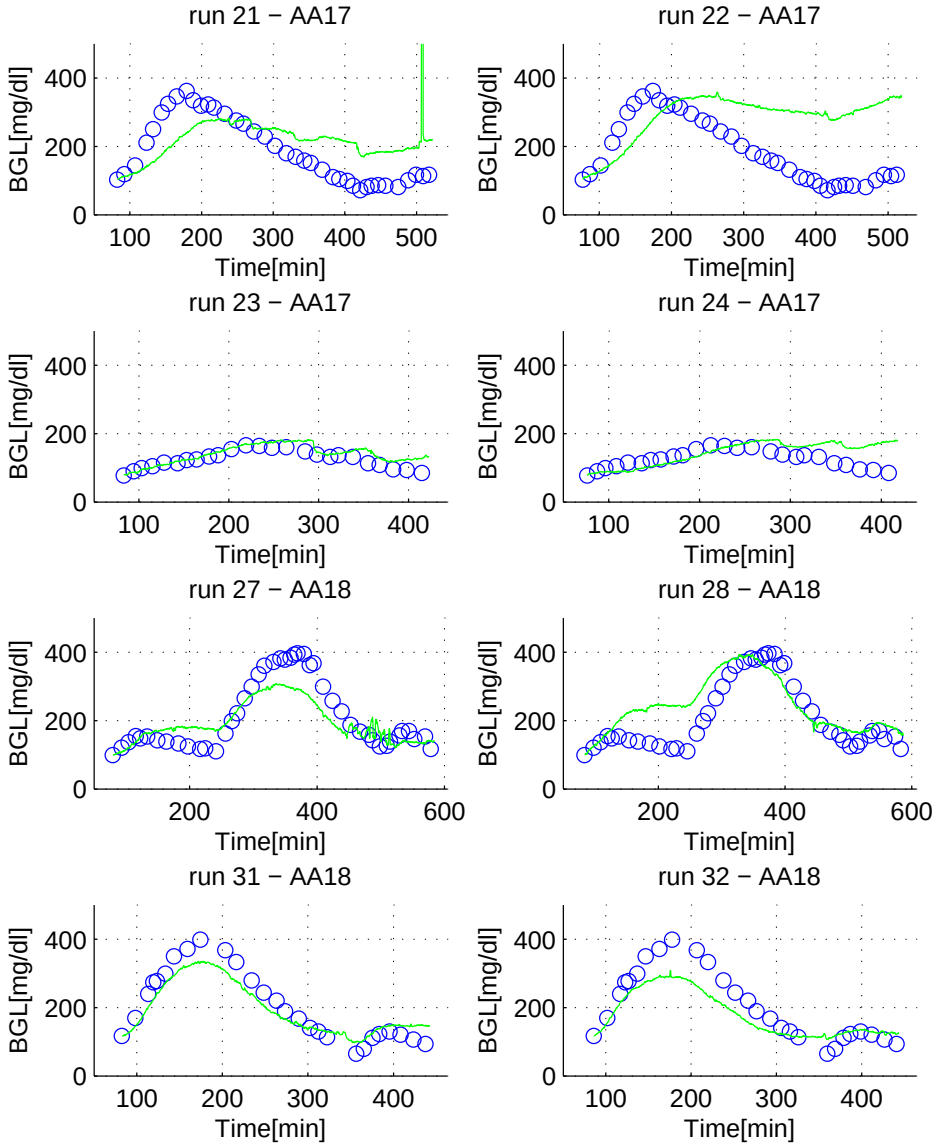

Figure 8.18: External validation of LASSO. LASSO model prediction(12 active variables, green) vs. reference BGL(blue circles). The last 8 runs of the training set are shown.

8.4 Performance Comparison for the three Methods

In this Section, we will compare the previously illustrated results obtained by applying OLS, PLS and LASSO in estimating linear regression models from the Solianis Multisensor data.

First of all it may be worthwhile to compare the estimated coefficients. In particular, since there are many different variables involved in our regression problem, we will focus our attention to those variables whose coefficients have the largest absolute value. Such variables are displayed in Table 8.4, along with their corresponding coefficients.

OLS			**PLS**			**LASSO**	
var name	coefficient		var name	coefficient		var name	coefficient
GHz_el2_3	-44.113		**GHz_el1_3**	0.206		**GHz_el1_2**	0.032
GHz_el2_4	47.876		**GHz_el1_7**	-0.227		**GHz_el2_8**	0.002
GHz_el2_7	-41.308		**MHz_el3_22**	-0.136		**KHz_el1_9**	-0.089
GHz_el2_8	51.912		**Opt_10**	-0.117		**MHz_el2_12**	0.367
MHz_el1_7	15.168		**Opt_11**	0.141		**MHz_el2_16**	0.023
MHz_el1_9	-32.792		**Opt_12**	-0.221		**MHz_el3_9**	0.211
MHz_el1_10	29.433		**Opt_16**	0.101		**MHz_el3_13**	0.113
MHz_el1_21	-10.352		**Opt_26**	-0.308		**Opt_10**	-0.074
MHz_el1_25	-10.515		**Opt_27**	0.117		**Opt_12**	-0.044
MHz_el1_26	13.524		**Ms_Hum**	-0.206		**Opt_16**	0.015
MHz_el2_8	11.373		**Ms_Temp**	0.117		**Opt_26**	-0.114
MHz_el3_5	-10.471		**Skin_Temp**	0.600		**Skin_Temp**	0.504
MHz_el3_8	11.208						
MHz_el3_9	-10.990						

Table 8.4: OLS, PLS and LASSO coefficients with greatest absolute value.

From Table 8.4, we can notice that the OLS coefficients are those presenting the largest absolute values. In addition, it is worth noting that these variables all belong to the dielectric type, which contains the most correlated variables. In fact, as observed several times throughout this thesis, using OLS on highly correlated variables results in large absolute values of the coefficients with opposite sign, leading to compensations among variables with the cancellation of their relative contribution to the target estimate. PLS and LASSO present significantly smaller

coefficients, due to their specific estimation procedure that, in different way, shrinks the number of variables included in the regression problem. However, there is a substantial difference in which variables provide the greater contribution to the target estimation. In fact, while in PLS there are many optical variables, more dielectric variables are selected by using LASSO . In particular, there are four impedance signals, which contain glucose information and are referred as the primary "glucose signals". Nevertheless, in both cases the largest absolute value is associated with the skin temperature.

The estimated coefficients have been used to predict the test BGL references. To allow a direct comparison, all the target estimates are plotted together in Figure 8.19 and 8.20. Visually, it can be noticed that the OLS predictions are the most sensitive to the noise contained in the data and are the most affected by increasing/decreasing trends. This proves that probably the OLS estimation suffers from overfitting. In fact, as observed in Section 8.1, the estimated coefficients allow a very good approximation of the training set, but fail in predicting unseen data. This is a typical consequence that the model during the learning procedure fits not only the informative part contained in the training set, but also the noise. This is not surprising if we consider the features of the dataset we are dealing with. In fact, as said in the previous Chapter, we are facing a high-dimensional dataset containing a lot of correlated variables, which tend to amplify the overfitting problem of OLS.

In the light of the previous observations, the choice of investigating the performance of other methods, trying to avoid the overfitting problem, seems reasonable.

The first presented method was PLS, which, through a linear combination of the original variables, forms a new set of regressors. Its main advantage is that the construction of the new regressors is based, not only on the information contained in each variables, but also on the correlation of the original variables with the target. Besides this, usually less new regressors are needed to obtain satisfactory approximations of the reference, introducing a sort of shrinkage. In this case, as it can be seen from Figure 8.19 and 8.20, the PLS estimator has a better performance than OLS. However, in some points the PLS predictions are still too noisy to be considered accurate (see for example run 3 and run 25).

The second presented method was LASSO, whose regularisation term prevents

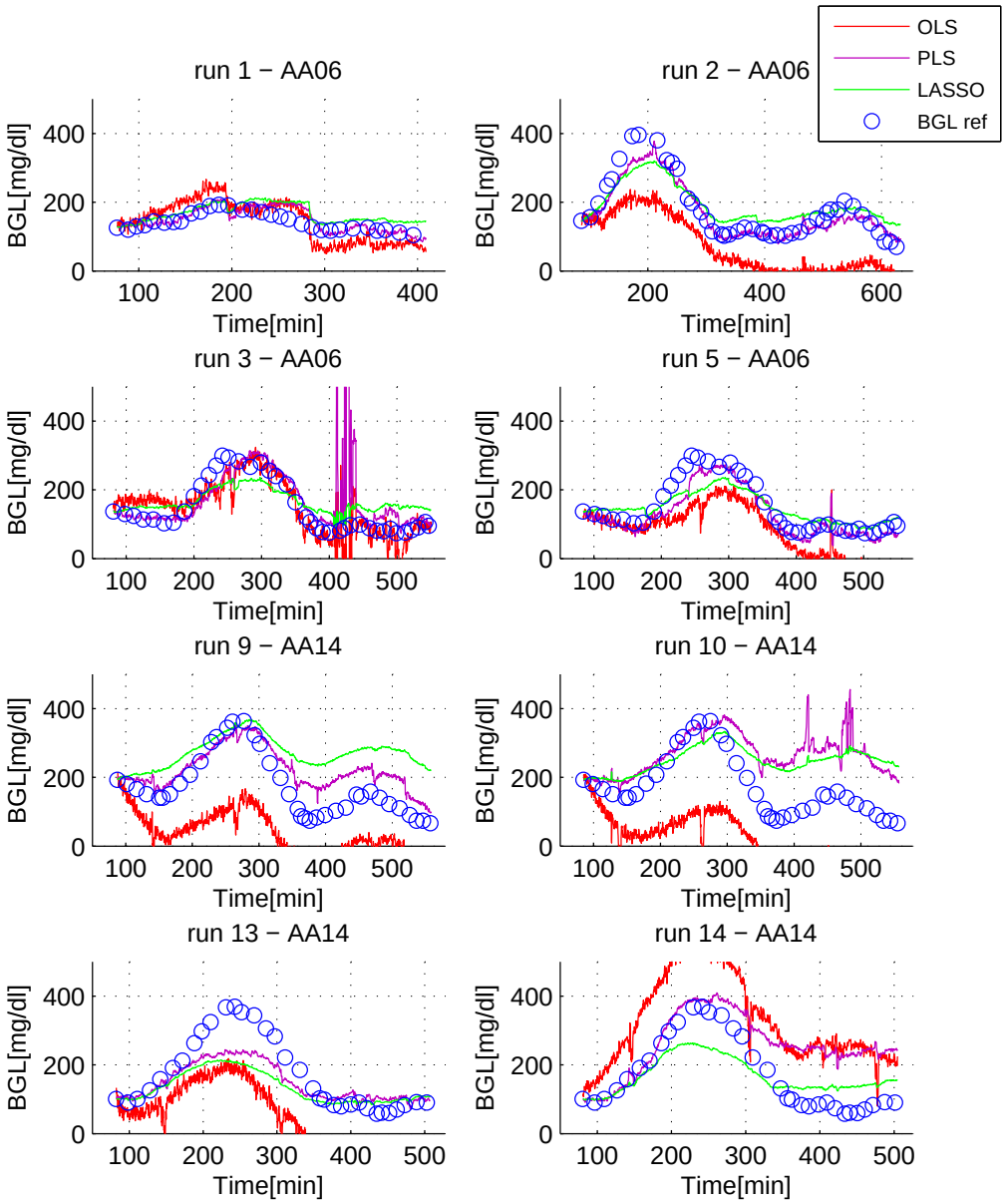

Figure 8.19: External validation for methods comparison. OLS(red), PLS(M=10, magenta) and LASSO(12 active variables, green) vs. reference BGL(blue circles). First 8 runs of the test set are shown.

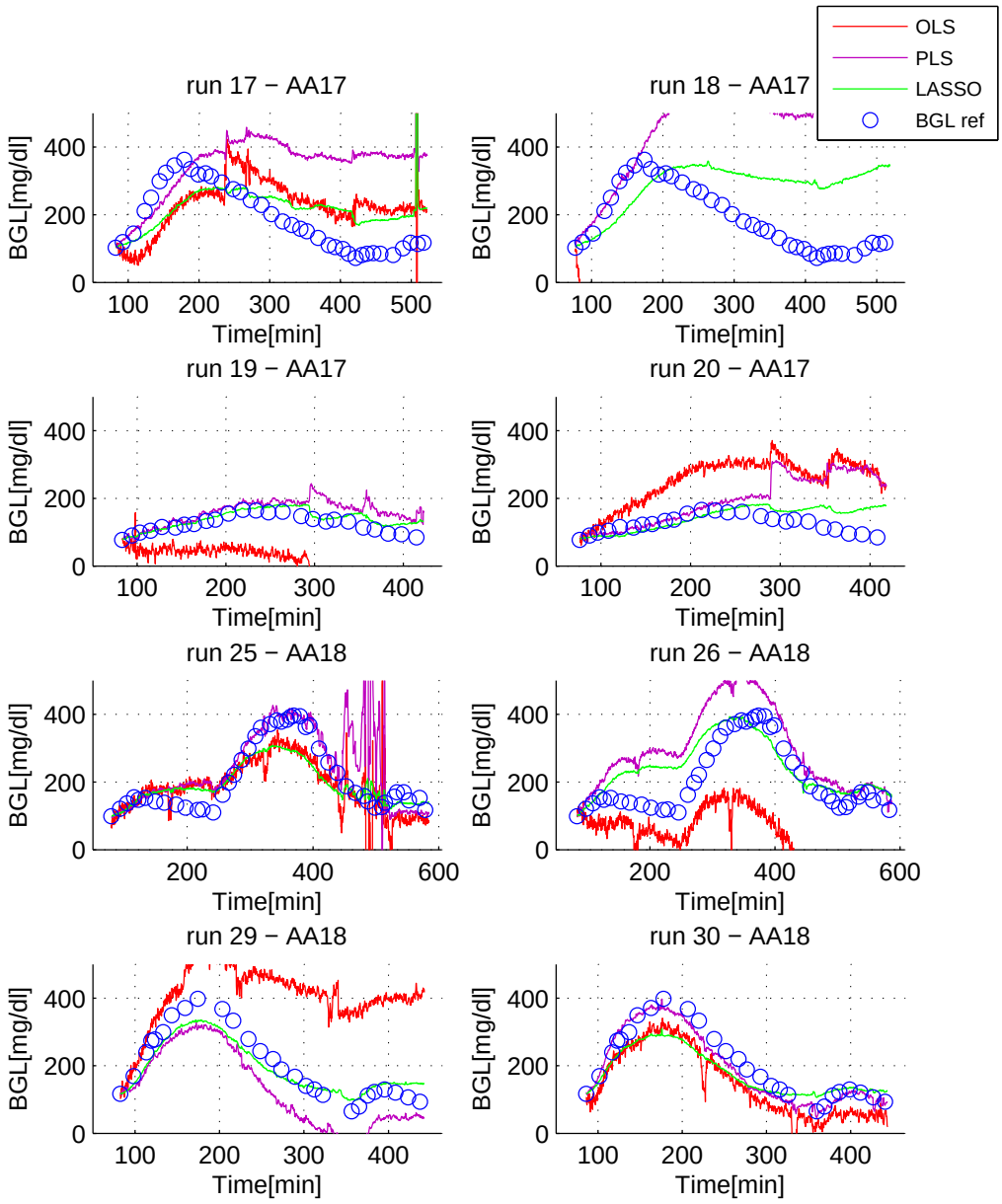

Figure 8.20: External validation for methods comparison. OLS(red), PLS(M=10, magenta) and LASSO(12 active variables, green) vs. reference BGL(blue circles). Last 8 runs of the test set are shown.

the estimated coefficients to assume too large values. Its main characteristic is to encourage sparse solutions, namely, if the regularisation term is appropriately weighted, some of the estimated coefficients are exactly zero. Selecting the model complexity using the cross-validation procedure, the best model results to have 12 active variables. In this case, as can be seen from Figure 8.19 and 8.20, the LASSO estimator is the one with the best performance, since its prediction are not very noisy proving that it selects in the proper way the active variables. Its good performance is due to different reasons: first, the regularisation term cause a prediction less sensitive too the noise and second, the correlated variables are not affected by the cancellation problem, even with more active variables. To prove this last statement, an example with 27 active variables is shown in Figure 8.21. All the variables have been centered and normalised, allowing a direct comparison.

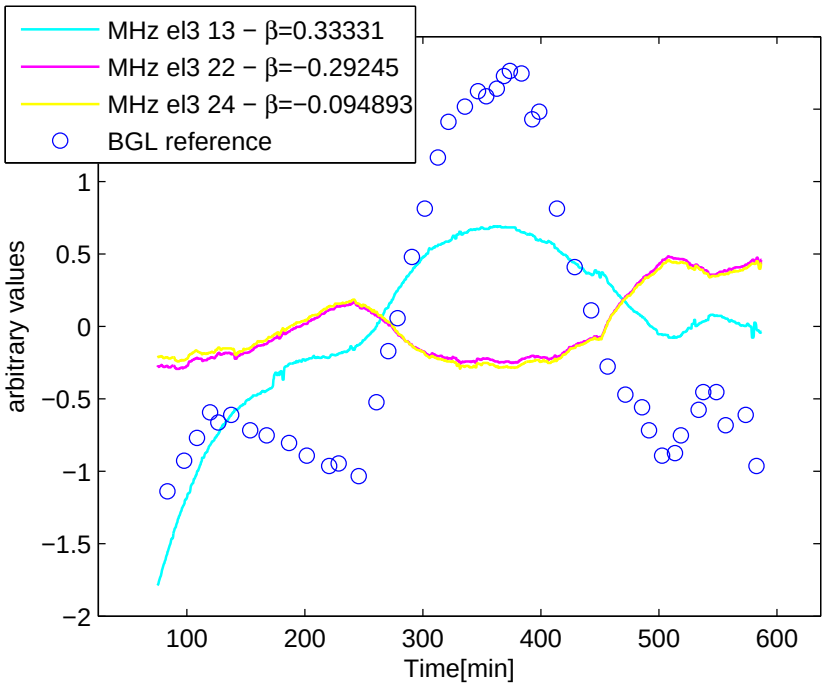

Figure 8.21: Example of highly correlated variables, which add up, compensating their effects on the target(blue circles).

Using the OLS estimator, the coefficients associated with the highly correlated variables tend to assume opposite signs, cancelling their contribution to the target estimates. With LASSO, the three variables shown in Figure 8.21 add up, when combined in the target estimation. In fact, as indicated in the legend, the light

blue variables have a positive coefficient, while the others(yellow and magenta lines) have a negative coefficient, as they are negative correlated to the first variable.

At this point, only a visual comparison between the methods have been performed. To quantify the performance of the different methods, some indicators are needed to summarise the behaviour of the predictions, as defined in Section 3.4.2.

INDICATOR	OLS	PLS	LASSO
RMSE	248.20	123.30	74.50
MAD	158.00	76.80	55.90
MARD	112.80	60.80	42.40
R^2	0.54	0.58	0.67

Table 8.5: Key-indicators for the external validation of OLS, PLS and LASSO.

The first three indicators represent a measure of how distant is the estimate from the reference. Hence, if the estimated model has a good performance, they will assume small values. While the last indicator, representing the squared Pearson correlation coefficient, denotes the approximation of the prediction in correspondence of the target oscillations and assumes values between 0 and +1. Hence, if the estimated model has a good performance, it will assume a value near +1.

In this case, all the indicators select LASSO as the best model estimator, which is in agreement with our visual observations.

In the next Chapter a deeper analysis of the LASSO performance will be performed. In particular, the performance of the so-called global model will be deeply evaluated. Then, two calibration methods, which have the aim to improve the glucose estimates accuracy, will be described.

Chapter 9

Further Topics and Margins of Improvement for Modeling Solianis Multisensor Data

In the previous Chapter we compared three different estimators (OLS, PLS and LASSO) for our linear regression problem and LASSO turned-out to be the one with the best performance. In this Chapter, a further analysis on the LASSO estimation ability is performed, using 12 active variables. First, it will be analysed if the global model is a good choice or if it is worth to investigate a subject-specific model to improve the glucose estimation. Then,two different calibration techniques, which may be used to improve the glucose estimates accuracy will be described and evaluated.

9.1 Assessment of the Global Model

In the previous Chapter the model was learnt from data including all the subjects and was tested on independent data from all the subjects. Hence, the performance of a "global" model was evaluated.

In this Section we want to determine if a global model is sufficient or a specific model for each subject is needed. The learning of a specific model would require to collect subject-specific data, on which to train the model, or, at least, to performe some kind of analysis on the subject to adapt the model to specific case. Hence, a global model has the advantage that, once it is estimated, it can be applied to

different subjects, without the necessity to adjust it to the specific subject.

For this purpose, we will evaluate if a model estimated from data of three subjects is suitable for predicting the BGL of another subject (the fourth subject of our dataset). Hence, a sort of cross-validation can be performed by leaving, iteratively, one subject out.

Implementing the previously described procedure, we obtain four different predictions, one for each subject, used as the test set. The results are summarized using the indicators shown in Table 9.1. The values obtained from the previous analysis are reported as a reference to assess the results.

	AA06	**AA14**	**AA17**	**AA18**	**ref values**
RMSE	54.70	144.30	59.20	47.20	74.50
MAD	43.60	121.50	44.20	38.10	55.90
MARD	36.50	78.70	31.90	21.90	42.40
R^2	0.57	0.33	0.48	0.73	0.67

Table 9.1: Indicators for cross-validation with leave one subject out. The last column is the reference values of the previous analysis in external validation.

From Table 9.1, we can notice that the model estimated on three subjects is able to well predict the glucose values of the fourth unseen subject, except for subject AA14. As representative example of the lucky case, the model predictions of subject AA18 are shown in Figure 9.1, from which we can observe that the model is able to mimic the glucose fluctuations and to approximate the glucose profiles correctly when applied on unseen subjects. Hence, in this case the estimated model has proved to generalise well. On the contrary, the bad model predictions in subject AA14 are shown in Figure 9.2, which demonstrate that in this case, a "global" model fails in approximating the glucose profiles. However, even in this unlucky case, the "global" model is still able to reproduce the glucose trends. From Figure 9.2 one can observe that a decreasing trend seems to be the common problem in the model predictions. Hence, it can be supposed that the generalisation failure of the estimated model may be due to a specific and common feature of the signals of subject AA14, instead of the model itself. To determine if the previous hypothesis is correct, further analysis on the signals of subject AA14 are needed.

However, from the previous observations we can conclude that, on average, a

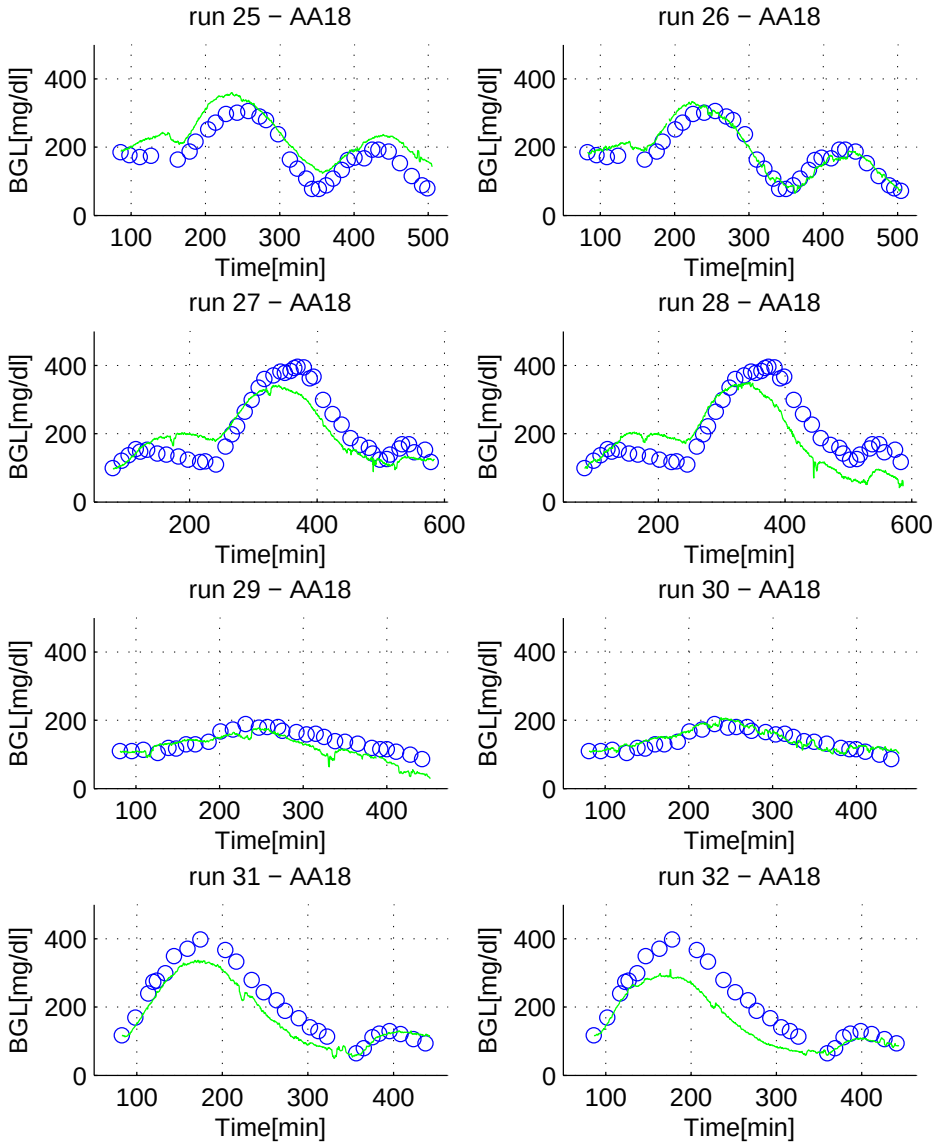

Figure 9.1: Cross-validation with leave subject AA18 out. LASSO predictions(12 active variables, green) vs. reference BGL(blue circles).

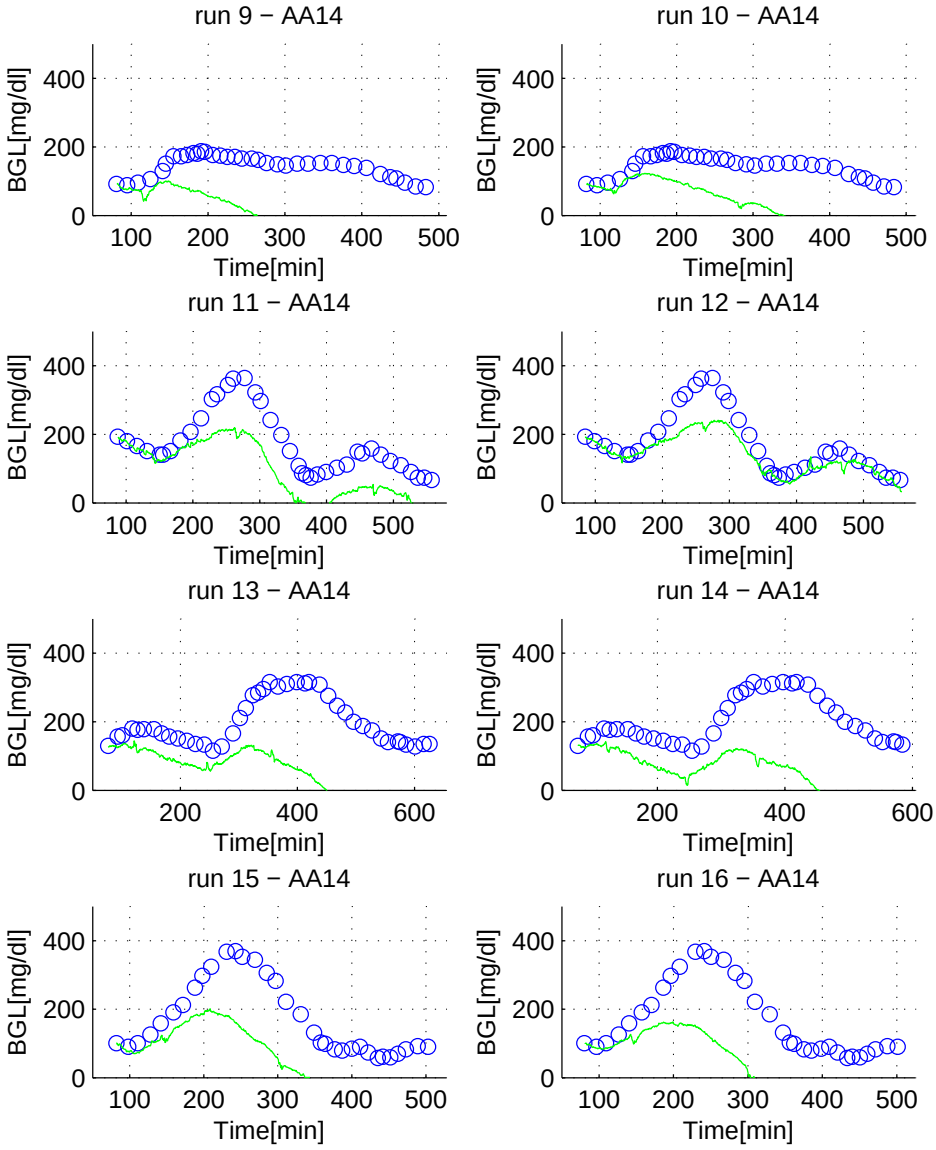

Figure 9.2: Cross-validation with leave subject AA14 out. LASSO predictions(12 active variables, green) vs. reference BGL(blue circles).

global model could be sufficient for predicting with acceptable accuracy glucose profile in "unseen" subjects.

9.2 Calibration Methods for Glucose Estimates Accuracy Improvement

The LASSO estimation procedure is not scale invariant. Hence, before applying the estimator for obtaining the linear model, the data have to be normalised. Then, to obtain the target estimate of unseen data (external validation), also the Multisensor data have to be normalised. However, since in the real application we do not have the entire Multisensor signals available, but only the sample at the current time step, we cannot obtain the normalisation parameters (mean and standard deviation) from the whole run. Hence, we can estimate the mean and the standard deviation from the training set and use them to normalise the unseen data of the test set.

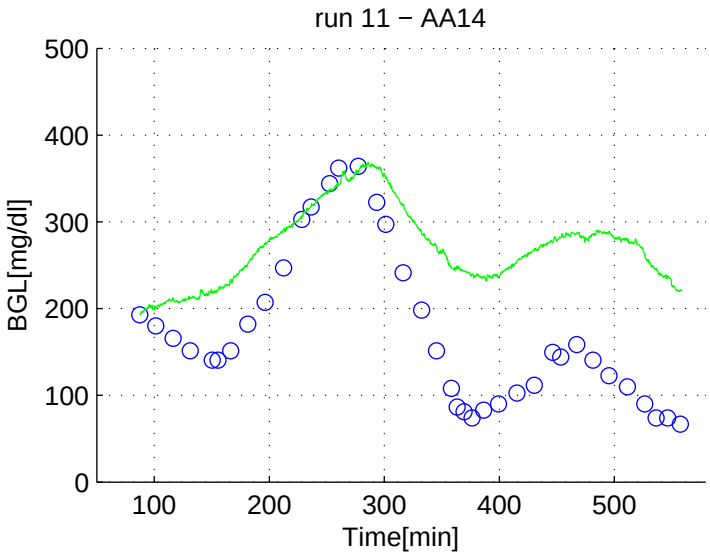

Figure 9.3: Representative run showing unsatisfactory predictions. LASSO model prediction(green) vs. reference BGL(blue circles).

Then, the model prediction has to be brought back to the original measuring units. For doing this, the reference mean and standard deviation, estimated from the training set, may be used. However, often this procedure leads to unsatisfactory results (see Figure 9.3).

In this Section we will describe two calibration methods, trying to improve the glucose estimates accuracy.

9.2.1 Calibration Method 1: Initial Baseline Adjustment

The initial baseline adjustment is a calibration technique, which uses only a glucose reference to set the estimation offset.

As said before, to re-convert the model estimation to the original measuring scale a reference sample may be helpful. Supposing that the estimated LASSO coefficients are available and we want to use them to predict new data. In this case, an initial BGL sample, obtained using finger-stick methods, may be used as a reference to set the prediction offset. In particular, indicating with $\hat{y}(t)$ the model prediction at the t-th time instant and supposing that the BGL sample is collected at the same time instant, the distance between the two can be used as the offset estimate:

$$offset = \text{BGL}(t) - \hat{y}(t)$$

and, adding such an offset to the model prediction $\hat{y}(t)$ we obtain the displayed value $y_d(t)$ (which coincides with the BGL sample):

$$y_d(t) = \hat{y}(t) + offset = \hat{y}(t) + \text{BGL}(t) - \hat{y}(t) = \text{BGL}(t)$$

The same offset is added to the successive model predictions $\hat{y}(t+1), \hat{y}(t+2), \ldots$ and has the aim to improve the accuracy of the displayed data. Hence:

$$y_d(t+1) = \hat{y}(t+1) + offset$$

This method has been used in the previous section to display the target estimates. In the next section, another simple method to calibrate the sensor will be described. Its application will concern only the glucose estimates obtained with LASSO and the results will be compared to those obtained using the simpler base-line adjustment.

9.2.2 Calibration Method 2: Offset Adjustment and Rescaling

This second type of calibration technique uses an additional glucose reference compared to the base line adjustment described in the previous section.

This procedure uses two parameters to calibrate the estimated glucose profiles. While one parameter represents the offset, as in the previous case, the

other denotes a "stretch quantity", introduced to suitably adjust the scale of the estimates[69][70].

As before, indicating with $\hat{y}(t)$ the model prediction at the t-th time instant and with $y_d(t)$ the corresponding transformed value, the calibration equation is:

$$y_d(t) = a\hat{y}(t) + b$$

where a is the above mentioned stretch parameter, used for changing the scale of the data, and b is the offset parameter.

In this case, as two different parameter have to be estimated, at least two BGL samples are needed. Indicating with n the number of BGL used in the calibration procedure, the parameter a and b are calculated using the OLS estimator, for solving the following system of equations:

$$\underbrace{\begin{bmatrix} \mathrm{BGL}(t_1) \\ \mathrm{BGL}(t_2) \\ \vdots \\ \mathrm{BGL}(t_n) \end{bmatrix}}_{\boldsymbol{R}} = \underbrace{\begin{bmatrix} \hat{y}(t_1) & 1 \\ \hat{y}(t_2) & 1 \\ \vdots & \vdots \\ \hat{y}(t_n) & 1 \end{bmatrix}}_{\boldsymbol{S}} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\boldsymbol{p}}$$

Representing with $\boldsymbol{R}$ (n x 1) the vector of the BGL samples, with $\boldsymbol{S}$ (n x 2) the matrix containing in the first column the model estimates and in the second a vector of ones and indicating with $\boldsymbol{p}$ (2 x 1) the parameters vector. The OLS solution for this problem is:

$$\boldsymbol{p} = \left(\boldsymbol{S}^T\boldsymbol{S}\right)^{-1}\boldsymbol{S}^T\boldsymbol{R}$$

Once the parameters have been estimated they can be used to improve the accuracy of the successive target estimates.

At this point we have not specified the suitable position for the second BGL reference. Suppose, as for the base line adjustment, that an initial BGL sample is available. To extract the information about the range of the possible BGL values, it seems reasonable to place the second BGL reference near a *glycaemic* peak. This can be easily reproduced quiet well in a practical context, collecting one sample before a meal and the second some time after the meal (typically 90 minutes).

For each run in the test set a different vector of parameters has been estimated and the LASSO predicted values have been calibrated. In Figure 9.4 and 9.5 the glucose prediction before and after the calibration are shown.

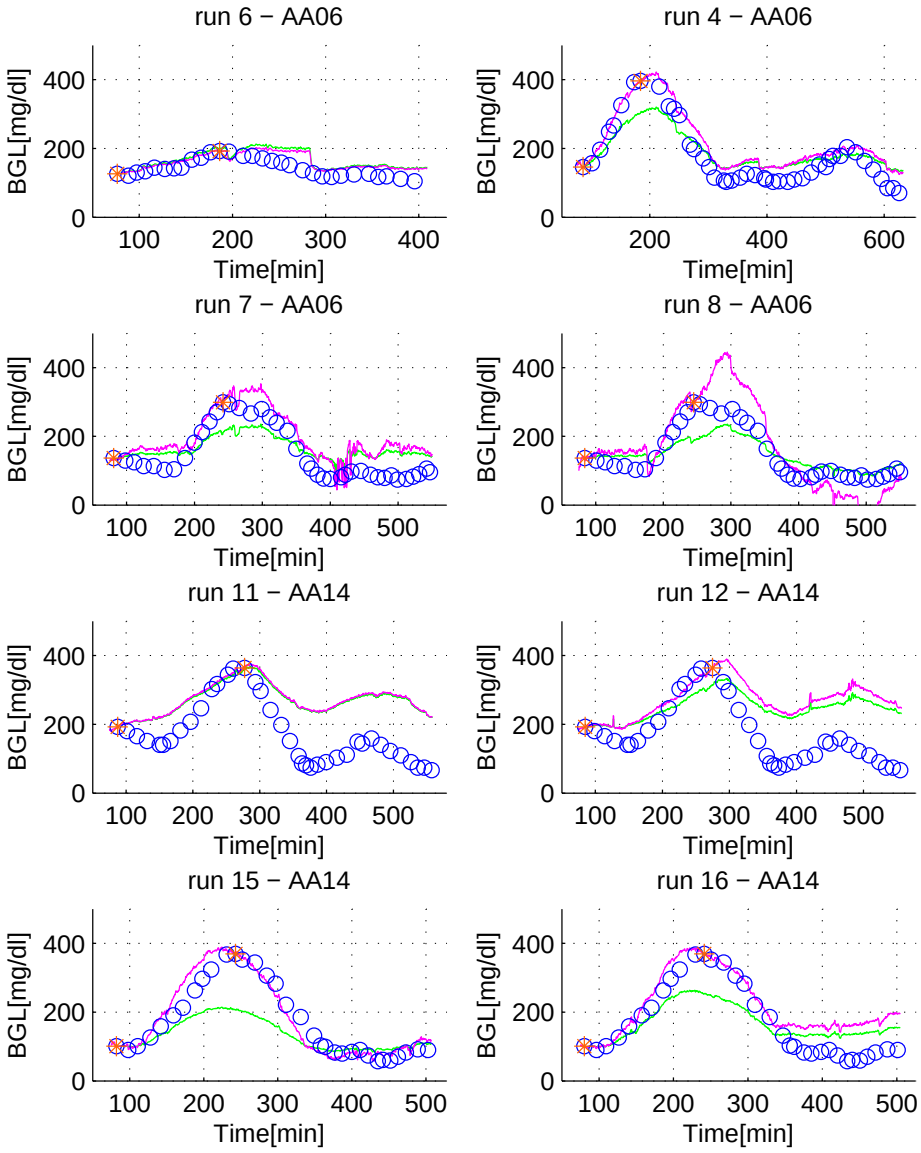

Figure 9.4: Calibration Method 2(magenta) vs. Calibration Method 1(green) for LASSO estimates. The red starts indicate the BGL samples used for the calibration procedure in Section 9.2.2. First 8 runs of the test set are shown.

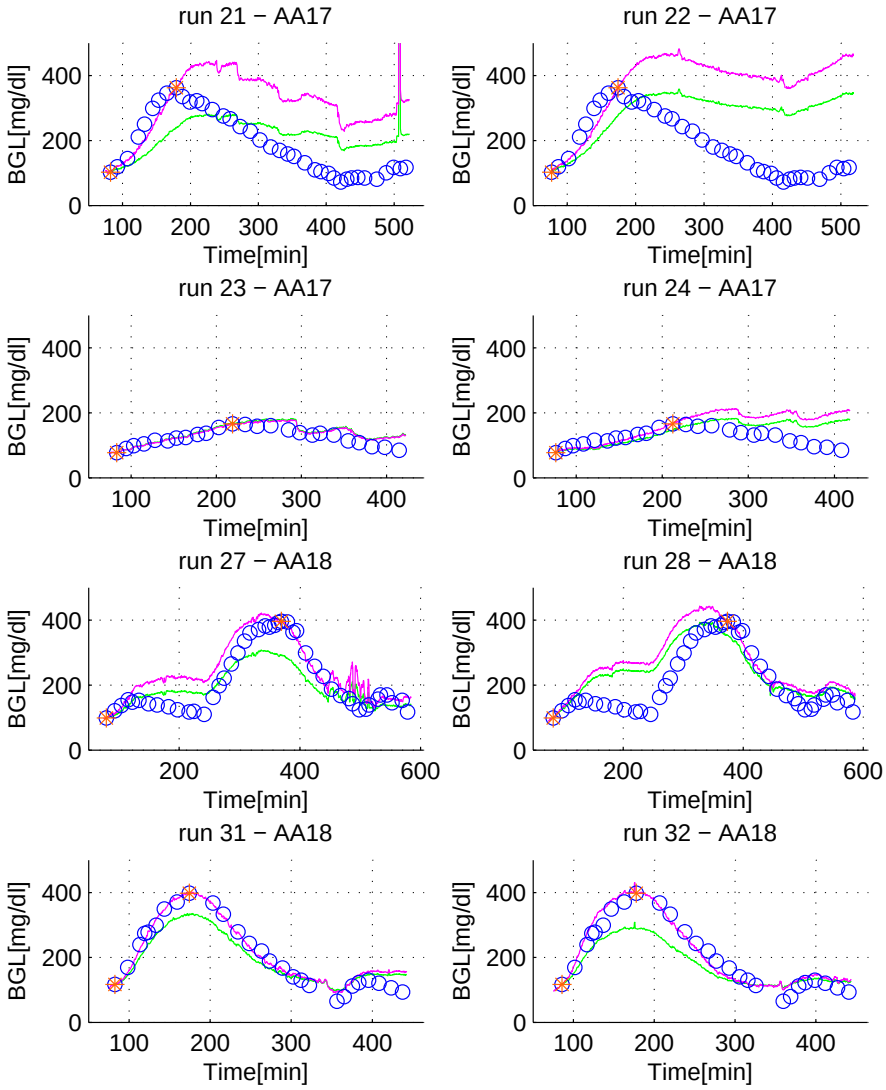

Figure 9.5: Calibration Method 2(magenta) vs. Calibration Method 1(green) for LASSO estimates. The red starts indicate the BGL samples used for the calibration procedure in Section 9.2.2. Last 8 runs of the test set are shown.

As shown in Figure 9.4 and 9.5, the performance of this calibration procedure is not the same for all the runs. In fact, in some cases (for example runs 4, 7, 15 and 31) the Calibration Method 2 significantly improves the accuracy of the glucose estimates, in other cases (for example runs 6, 11 and 23) the calibrated

estimates have the same performance of the Calibration Method 1. In the remaining cases (for example runs 8 e 21) the glucose estimates are worst than using the Calibration Method 1. For completeness, the use of a third BGL sample don't cause a further improvement in the accuracy, compared to the case of only two BGL samples. Hence, we limit our analysis to the case of two BGL samples.

We can conclude that in most cases Calibration Method 2 improves the accuracy of glucose estimates. When bad results occur, addictional problems affect in all likelihood the predictions. To verify it, we can focus our attention to the run 21 (shown in Figure 9.6).

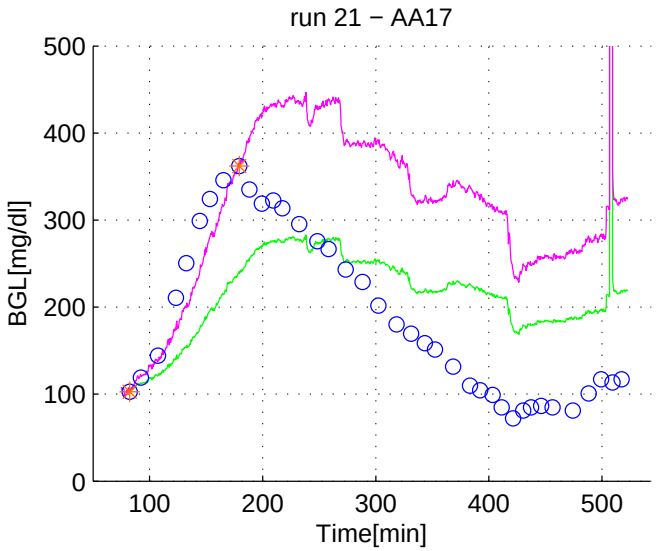

Figure 9.6: Representative run of Calibration Method 2(magenta) vs. Calibration Method 1(green) for LASSO estimates. The red starts indicate the BGL samples used for the calibration procedure in Section 9.2.2.

This specific run presents the minimum value after the *glycaemic* peak, which is lower with respect to the first BGL reference. In the prediction with the initial base line adjustment (green) we cannot notice the same characteristic feature. In particular, it seems that an increasing trend is overlapped to the true glucose profile. As a consequence, no calibration procedure can improve the estimates accuracy, unless the trend is removed. Hence, further analyse are needed to investigate the cause of this trend, in order to compensate it in order to allow a subsequent improvement of the glucose estimates by using the calibration procedure.

Chapter 10

Conclusions

Diabetes is a worldwide problem and the number of people with diabetes is constantly increasing due to several reasons including population growth, age, and increasing prevalence of obesity and physical inactivity. In particular, the long-term complications make diabetes a social and economical problem, since they have great impact on subject daily life and its management is financially expensive. As a consequence, considerable efforts have been made to control this disease also by using engineering technologies.

During the last decade, it has been proven that diabetes therapy can be improved by monitoring blood glucose levels by means of the so-called Continuous Glucose Monitoring (CGM) sensors. Different types of sensors, with different degrees of invasiveness, have been already developed in the literature and, at present time, new technologies are also under investigation. Among them, new completely Non-Invasive CGM sensors (NICGM) are very appealing for obvious practical reasons. In particular, Solianis Monitoring AG (Zurich, Switzerland) has recently proposed a NICGM sensor based on the multi-sensor concept, i.e. a system that includes several sensors (for impedance, optics, temperature, acceleration, ...) on one single substrate which can be attached to the human body in order to allow a broad characterisation of the skin and the underlying tissues (Caduff et all, Biosensors and Bioelectronics, pp. 2778-2784, 2009). Such Multisensor signals allow the indirect measurement of glucose level in the blood through a mathematical model.

The present work was performed under the aegis of a research agreement between the Department of Information Engineering of the University of Padova and

Solianis Monitoring AG. The scope of the project was the development and the assessment of a model for estimating glucose level from the Solianis Multisensor data. Specifically, in the present thesis three different methods for building a linear regression model to describe glucose data from Multisensor signals were investigated, assessed and compared: Ordinary Least Squares (OLS), Partial Least Squares (PLS) and Least Absolute Shrinkage and Selection Operator (LASSO).

OLS results suffered from overfitting, i.e. the estimated regression model fits not only the information yield by the data but also to the noise contained in them. Hence, the model was not able to generalize the data properly and failed in predicting unseen data.

PLS constructs new orthogonal predictors, starting from the original signals and makes a linear combination of a selected number them to calculate the regression. However, for the Solianis Multisensor data, the PLS estimate resulted too sensitive to the noise in the data. This phenomenon was probably due to the fact that the new regressors are constructed from a linear combination of all the original signals (also the noisier ones).

Finally, LASSO makes a linear combination of the signals, but penalizes the sum of absolute coefficients to prevent the multiplication coefficients from assuming too large values. In addition, LASSO has the characteristic to yield to sparse solutions, which means that some coefficients will be exactly zero. By applying LASSO to the Multisensor data, rather satisfactory estimates of the glucose profiles were obtained. In particular, LASSO, selecting only some of the original signals by means of the regularisation term, avoided the OLS overfitting problem.

Summarising, LASSO has shown the best performance in predicting the BGL from unseen Multisensor data.

Further analysis was then performed, by using the LASSO method, to assess if a global model could be a viable choice for the Solianis Multisensor. The global model would have the advantage that, once it is estimated, it can be applied to different subjects, without the necessity to adjust it to the specific subject. Hence, it was evaluated the capability of the model in predicting the glucose profiles of a subject, which was not used in the model learning. In this case, we concluded that on average, a global model was sufficient for predicting properly the glucose profile of new subjects. A second investigation concerned with the

possible usefulness of two calibration methods to improve the accuracy of glucose estimates. The so-called calibration method 2 showed, in most cases, a remarkable improvement of the accuracy of the estimated glucose profiles.

Future developments

From the previously described analysis, LASSO gave the best results in predicting the BGL from unseen Multisensor data, as it avoids overfitting which may easily occur with the ordinary techniques. Hence, it will be worthwhile to analyse the LASSO performance in modeling Solianis Multisensor data on a wider dataset, in order to obtain a deeper insight into its capability of generalising the data.

During the analysis a common limitation of LASSO estimates was due by the presence of some trends, which caused a reduction in the accuracy of the model in predicting the glucose profiles. Future analysis should have the aim of investigating the cause of these trends, in order to compensate them allowing a subsequent improvement of the glucose estimates.

Appendix A

Matlab code

In this Section we report the Matlab code for the implementation of the linear regression techniques presented in this thesis.

A.1 OLS

```
beta=X\y;

or

beta=inv(X'*X)*X'y
```

A.2 PLS

```
function [Xload,Yload,Xscores,Yscores,beta] = plsregress(X,Y,ncomp,varargin)

[n,dx] = size(X);
ny = size(Y,1);

% Center both predictors and response, and do PLS
meanX = mean(X,1);
meanY = mean(Y,1);
X0 = bsxfun(@minus, X, meanX);
Y0 = bsxfun(@minus, Y, meanY);

if nargout <= 2
    [Xload,Yload] = simpls(X0,Y0,ncomp);
```

```
elseif nargout <= 4
    [Xload,Yload,Xscores,Yscores] = simpls(X0,Y0,ncomp);

else
    % Compute the regression coefs, including intercept(s)
    [Xload,Yload,Xscores,Yscores,Weights] = simpls(X0,Y0,ncomp);
    beta = Weights*Yload';
    beta = [meanY - meanX*beta; beta];

end


%-----------------------------------------------------------------------
%SIMPLS Basic SIMPLS.  Performs no error checking.
function [Xload,Yload,Xscores,Yscores,Weights] = simpls(X0,Y0,ncomp)

[n,dx] = size(X0);
dy = size(Y0,2);

% Preallocate outputs
outClass = superiorfloat(X0,Y0);
Xload = zeros(dx,ncomp,outClass);
Yload = zeros(dy,ncomp,outClass);
if nargout > 2
    Xscores = zeros(n,ncomp,outClass);
    Yscores = zeros(n,ncomp,outClass);
    if nargout > 4
        Weights = zeros(dx,ncomp,outClass);
    end
end

% An orthonormal basis for the span of the Xload to make successive deflation
% X0'*Y0 simple - each new basis vector can be removed from Cov separately.
V = zeros(dx,ncomp);

Cov = X0'*Y0;
for i = 1:ncomp
    % Find unit length ti=X0*ri and ui=Y0*ci whose covariance,ri'*X0'*Y0*ci,
    % is jointly maximized, subject to ti'*tj=0 for j=1:(i-1).
    [ri,si,ci] = svd(Cov,'econ'); ri = ri(:,1); ci = ci(:,1); si = si(1);
    ti = X0*ri;
```

```
    normti = norm(ti); ti = ti ./ normti; % ti'*ti == 1
    Xload(:,i) = X0'*ti;

    qi = si*ci/normti; % = Y0'*ti
    Yload(:,i) = qi;

    if nargout > 2
        Xscores(:,i) = ti;
        Yscores(:,i) = Y0*qi; % = Y0*(Y0'*ti)
        if nargout > 4
            Weights(:,i) = ri ./ normti; % rescaled
        end
    end

    % Update the orthonormal basis with modified Gram Schmidt,(more stable),
    % repeated twice (ditto).
    vi = Xload(:,i);
    for repeat = 1:2
        for j = 1:i-1
            vj = V(:,j);
            vi = vi - (vi'*vj)*vj;
        end
    end
    vi = vi ./ norm(vi);
    V(:,i) = vi;

    % Deflate Cov, i.e. project onto the ortho-complement of the X loadings.
    % First remove projections along the current basis vector, then remove
    % any component along previous basis vectors that's crept in as noise from
    % previous deflations.
    Cov = Cov - vi*(vi'*Cov);
    Vi = V(:,1:i);
    Cov = Cov - Vi*(Vi'*Cov);
end

if nargout > 2
    % By convention, orthogonalize the Yscores w.r.t. the preceding Xscores,
    % i.e. XSCORES'*YSCORES will be lower triangular.  This gives, in effect,
    % only the "new" contribution to the Y scores for each PLS component. It
    % is also consistent with the PLS-1/PLS-2 algorithms, where the Yscores
    % are computed as linear combinations of a successively-deflated Y0.
```

```
    for i = 1:ncomp
        ui = Yscores(:,i);
        for repeat = 1:2
            for j = 1:i-1
                tj = Xscores(:,j);
                ui = ui - (ui'*tj)*tj;
            end
        end
        Yscores(:,i) = ui;
    end
end
```

A.3 LASSO

```
function [history, stopReason] = lars(yin, xin, XTX, type, ...
                                stopCriterion, regularization, trace, quiet)

global USING_CLUSTER;
global LARS_RESOL;
global REGULARIZATION_FACTOR;
lars_init();

regularization_factor = REGULARIZATION_FACTOR;

stopReason = {};

%%% Check parameters
if length(yin)==0 | length(xin)==0
    warning('\nInput or Output has zero length.\n');
    history.active_set = [];
    stopReason{1} = 'Parameter error';
    stopReason{2} = 0;
    return;
end
if size(yin,1) ~= size(xin,1)
    warning('\nSize of y does not match to that of x.\n');
    history.active_set = [];
    stopReason{1} = 'Parameter error';
    stopReason{2} = 0;
    return;
end
```

```
if ~strcmp(type,'lasso')&~strcmp(type,'lars')&~strcmp(type,'forward_stepwise')
    warning('\nUnknown type of regression.\n');
    history.active_set = [];
    stopReason{1} = 'Parameter error';
    stopReason{2} = 0;
    return;
end
if strcmp(type, 'forward_stepwise')
    warning('\nForward_stepwise is not implemented.\n');
    history.active_set = [];
    stopReason{1} = 'Parameter error';
    stopReason{2} = 0;
    return;
end


if exist('regularization','var') & ~isempty(regularization)
    regularization = 10;
else
    regularization  = 0;
end

if ~exist('trace','var') | isempty(trace)
    trace=0;
end

if ~exist('quiet','var') | isempty(quiet)
    quiet=0;
elseif quiet==1
    trace=0;
end

%-----------------------------------------------------------------------------
% Data preparation
% Program automatically centers and standardizes predictors.
if ~exist('XTX','var')
    XTX=[];
end
no_xtx = 0;
if ~isempty(XTX)
    if ~quiet & trace >=0
```

```
        fprintf('\nLars is using the provided xtx.\n');
    end
elseif size(xin,2)^2 > 10^6
    if ~quiet & trace >=0
        fprintf('Too large matrix (size(x,2)^2 > 10^6)');
    end
    no_xtx      = 1;
    XTX         = lars_getXTX(xin,no_xtx);
else
%     fprintf('\nCalculating xtx.\n');
    XTX         = lars_getXTX(xin);
end

x               = XTX.x;            % normalized xin
mx              = XTX.mx;           % mean xin
sx              = XTX.sx;           % length of each column xin
ignores         = XTX.ignores;      % indices for constant terms
all_cand        = XTX.all_cand;     % indices for all columns
if ~no_xtx
    xtx         = XTX.xtx;          % xtx matrix
    dup_columns = XTX.dup_columns;  % duplicated columns ignored
end

my              = mean(yin);
y               = yin-my;

n               = size(x,1);        % # of samples
m               = size(x,2);        % # of predictors

% Now, we can determine the maximum number of kernels.
existMaxKernels = 0;
existMSE        = 0;
for is = 1:size(stopCriterion,1)
    if strcmp(stopCriterion{is,1},'maxKernels')
        existMaxKernels = 1;
        stopCrit{is,2}=min(stopCrit{is,2},...
                   min(rank(xin),length(all_cand)));
        if stopCrit{is,2}<1
            warning('Max Kernel is less than 1.\n');
            stopCrit{is,2} = 1;
        end
```

```
    end
    if strcmp(stopCrit{is,1},'MSE')
        existMSE        = 1;
        if stopCrit{is,2}<1.0e-10
            warning('Maximum MSE is too small.');
            stopCrit{is,2} = 1.0e-10;
        end
    end
end
if ~existMaxKernels
    is = size(stopCrit,1);
    stopCrit{is+1,1} = 'maxKernels';
    stopCrit{is+1,2} = min(rank(xin), length(all_cand));
    % Stop when size of active set is data.maxKernels.
end
if ~existMSE
    is = size(stopCrit,1);
    stopCrit{is+1,1} = 'MSE';
    stopCrit{is+1,2} = 1.0e-10;
end

%--------------------------------------------------------------------------
% Initialization

active      = [];               % active set
inactive    = all_cand;    % inactive set

mu_a        = zeros(n,1);       % current estimate
mu_a_plus   = 0;                % next estimate
mu_a_OLS    = 0;                % OLS estimate

beta        = zeros(1,size(x,2));
beta_new    = beta;
beta_OLS    = beta;

history.active_set          = [];
history.add                 = [];
history.drop                = [];
history.beta_norm           = [];
history.beta                = [];
history.b                   = my;
```

```
history.mu                  = my;
history.beta_OLS_norm       = [];
history.beta_OLS            = [];
history.b_OLS               = my;
history.mu_OLS              = my*ones(size(yin));
history.MSE                 = sum(y.^2)/length(y);
history.R_square            = 0;
history.resolution_warning  = [];


if var(yin)==0
    stopReason{1} = 'zeroVarY';
    stopReason{2} = var(yin);
    return;
end

%-------------------------------------------------------------------------
% Main loop
%-------------------------------------------------------------------------
c               = 0;                % correlation vector
C_max           = max(abs(c));
C_max_ind       = [];
C_max_ind_pl    = [];
drop            = [];               % used for 'lasso'
k               = 1;                % iteration index
if ~quiet & trace >= 0
    fprintf('Active predictors/total : Current iteration\n');
end
while 1

    %-----------------------------------------------------------------
    % Exit Criterions
    %-----------------------------------------------------------------
    if exist('stopCrit','var')% If there is stop criterion
        % Any of these is satisfied, algorithm stops.

        for is = 1:size(stopCrit,1)
            % Default Criterions.Maximum number of consecutive drops.
            if strcmp(stopCrit{is,1},'maxDrops')
                drop_window = stopCrit{is,2}(1);
                if drop_window==0
```

```
            drop_window=k;
        end
        drop_n      = min(drop_window,stopCrit{is,2}(2));
        drop_vector = [];
        for z = max(k-drop_window+1,1):k
            drop_vector=[drop_vector, history(z).drop];
        end
        if length(drop_vector)>=drop_n
            stopReason{1} = 'maxDrops';
            stopReason{2} = drop_n;
            break;
        end
    end
    % Maximum number of kernels.
    if strcmp(stopCrit{is,1},'maxKernels')
        if length(active) >= min(stopCrit{is,2},...
                min(size(xin,1)-1, length(all_cand)))
            stopReason{1} = 'maxKernels';
            stopReason{2} = length(active);
            break;
        end
    end
    % Maximum number of iterations.
    if strcmp(stopCrit{is,1},'maxIterations')
        if k >= stopCrit{is,2}
            stopReason{1} = 'maxIterations';
            stopReason{2} = k;
            break;
        end
    end
    % MSE.
    if strcmp(stopCrit{is,1},'MSE')
        if history(k).MSE <= stopCrit{is,2}
            stopReason{1} = 'MSE';
            stopReason{2} = history(k).MSE;
            break;
        end
    end
    % User defined stop criterion.
    if strcmp(stopCrit{is,1},'userDefinedCriterion')
        fhandle = stopCrit{is,2}.fhandle;
```

```
            r_fhandle = fhandle(history, stopCrit{is,2}.data);
            if r_fhandle.stop
                stopReason{1} = 'userDefinedCriterion';
                stopReason{2} = r_fhandle;
                break;
            end
        end
    end
end % end of stop criterion checking

if length(stopReason)>0 % if there is any reason exit loop.
    break;
end
%-------------------------------------------------------------------------
% LARS Algorithm
%-------------------------------------------------------------------------

c                   = x'*(y-mu_a);
[C_max,C_max_ind]   = max(abs(c(inactive)));
C_max_ind           = inactive(C_max_ind);
% But because of machine limit, there can be multiple new predictors.
% This improves the overall precision of the result,
% and speeds up the whole process.
C_max_ind_pl        = abs(c(inactive))>C_max-LARS_RESOL;
C_max_ind_pl        = inactive(C_max_ind_pl);
active          = sort(union(active,C_max_ind_pl));
inactive        = setdiff(all_cand, active);
if strcmp(type,'lasso')
    if ~isempty(drop)&length(find(drop==C_max_ind))==0%If there is a drop
        if ~quiet & trace >=0
            fprintf('\n');
            warning('Dropped item and index of maximum corr');
            fprintf('\n                              ');
        end
        active(find(active==C_max_ind))=[];
    end
    if ~isempty(drop)
        C_max_ind   = [];
        C_max_ind_pl= [];
    end
    active          = setdiff(active,drop);
```

```
    inactive        = sort(union(inactive,drop));
end

s       = sign(c(active));
xa      = x(:,active).*repmat(s',n,1);
if ~no_xtx
    ga  = xtx(active,active).*(s*s');
else
    ga  = xa'*xa;
end
if regularization > 2
    ga  = ga+eye(length(ga))*regularization_factor;
    % This routine will make the test below
end
invga   = ga\eye(size(ga,1));
aa      = sum(sum(invga))^(-1/2);
wa      = aa*sum(invga,2);
ua      = xa*wa;

test_1      = xa'*ua;
test_2      = aa*ones(size(test_1));
test_1_2    = sum(sum(abs(test_1-test_2)));
test_3      = norm(ua) - 1;

history(k+1).resolution_warning=0;
if test_1_2>LARS_RESOL*100|abs(test_3)>LARS_RESOL*100
    if regularization <=2
        if ~quiet & trace>0
            fprintf('\n');
            warning('test failure.');
            fprintf('\n                          ');
        end
        regularization = regularization + 1;
        if regularization > 2
            if ~quiet & trace>0
                fprintf('\n');
                warning('Regularization be applied from now');
                fprintf('\n                            ');
            end
        end
    end
```

```
    history(k+1).resolution_warning=1;
end

a               = x'*ua;
tmp_1           = (C_max - c(inactive))./(aa - a(inactive));
tmp_2           = (C_max + c(inactive))./(aa + a(inactive));
tmp_3           = [tmp_1, tmp_2];
tmp             = tmp_3(find(tmp_3>0));
gamma = min(tmp);
if length(gamma)==0
%if this is the last step (i.e. length(active)==maxKernels)
    gamma = C_max/aa;
end

d               = zeros(1,m);
d(active)       = s.*wa;

if length(find(d(active)==0))
    fprintf('\n');
    warning('Something wrong with vector d:');
    fprintf('\n                              ');
end

tmp             = zeros(1,m);
tmp(active)     = -1*beta(active)./d(active);
tmp2            = tmp(find(tmp>0));

drop            = [];
gamma_tilde = inf;
if ~isempty(tmp2) & gamma >= min(tmp2)
    gamma_tilde = min(tmp2);
    drop        = find(tmp==gamma_tilde);
end

if strcmp(type, 'lars')
    mu_a_plus   = mu_a + gamma*ua;
    beta_new    = beta + gamma*d;
    drop        = [];
elseif strcmp(type, 'lasso')
    mu_a_plus   = mu_a + min(gamma, gamma_tilde)*ua;
    beta_new    = beta + min(gamma, gamma_tilde)*d;
```

```
        active      = setdiff(active,drop);
        inactive    = setdiff(all_cand,active);
        beta_new(drop) = 0;
    elseif strcmp(type, 'forward_stepwise')
        drop        = [];
        error('forward.stepwise has not been implemented yet.');
        return;
    end

    mu_a_OLS    = mu_a + C_max/aa*ua;
    beta_OLS    = beta + C_max/aa*d;
    MSE         = sum((y - mu_a_OLS).^2)/length(y);

    %-------------------------------------------------------------
    % update and save
    mu_a = mu_a_plus;
    beta = beta_new;
    % history with scale correction
    k = k+1;
    history(k).active_set   = active;
    history(k).drop         = drop;
    history(k).add      = C_max_ind_pl;
    history(k).beta_norm    = beta(active);
    history(k).beta     = beta(active)./sx(active);
    history(k).b        = my - sum(mx./sx.*beta);
    history(k).mu       = xin * (beta./sx)' + history(k).b;
    history(k).beta_OLS_norm= beta_OLS(active);
    history(k).beta_OLS = beta_OLS(active)./sx(active);
    history(k).b_OLS    = my - sum(mx./sx.*beta_OLS);
    history(k).mu_OLS   = xin*(beta_OLS./sx)'+ history(k).b_OLS;
    history(k).MSE      = MSE;
    history(k).R_square = 1-var(yin-history(k).mu_OLS)/var(yin);

    % exit if exact mathing is achieved.
    if abs(C_max/aa - min(gamma,gamma_tilde)) < LARS_RESOL
        stopReason{1}   = 'ExactMatching';
        stopReason{2}   = 0;
        break;
    end
end % end of while loop
return;
```

List of Abbreviations

BGL	Blood Glucose Levels
CGM	Continuous Glucose Monitoring
IDDM	Insulin Dependent Diabetes Mellitus
IS	Impedance Spectroscopy
LAR	Least Angle Regression
LASSO	Least Absolute Shrinkage and Selection Operator
MAD	Mean Absolute Difference
MARD	Mean Absolute Relative Difference
MSE	Mean Square Error
NICGM	Non Invasive Continuous Glucose Monitoring
NIDDM	Non-Insulin Dependent Diabetes Mellitus
OCT	Optical Coherence Tomography
OLS	Ordinary Least Squares
PLS	Partial Least Squares
RMSE	Root Mean Square Error
RSS	Residual Sum of Squares
SMBG	Self-Monitoring Blood Glucose

Bibliography

[1] A. Caduff, M. Talary, M. Mueller, F. Dewarrat, J. Klisic, M. Donath, L. Heinemann, and W. A. Stahel, "Non-invasive glucose monitoring in patients with type 1 diabetes: a multisensor system combining sensors for dielectric and optical characterization of skin," *Biosensors and Bioelectronics*, vol. 24, no. 9, pp. 2778–2784, 2009.

[2] A. R. Saltiel and R. Kahn, "Insulin signalling and the regulation of glucose and lipid metabolism," *Nature*, vol. 414, pp. 799–806, 2001.

[3] I. Stratton, A. Adler, H. Neil, D. Matthews, S. Manley, C. Cull, D. Hadden, R. Turner, and R. Holman, "Association of glycaemia with macrovascular and microvascular complications of type 2 diabetes: prospective observational study," *British medical journal*, vol. 321, no. 7258, p. 405, 2000.

[4] S. Davis and G. Lastra-Gonzalez, "Diabetes and Low Blood Sugar (Hypoglycemia)," *Journal of Clinical Endocrinology & Metabolism*, vol. 93, no. 8, 2008.

[5] S. Gambert and S. Pinkstaff, "Emerging epidemic: diabetes in older adults: demography, economic impact, and pathophysiology," *Diabetes Spectrum*, vol. 19, no. 4, p. 221, 2006.

[6] E. M. Benjamin, "Self-monitoring of blood glucose: The basics," *Clinical Diabetes*, vol. 20, no. 1, pp. 45–47, 2002.

[7] B. H. Ginsberg, "The current environment of continuous glucose monitoring technologies," *Journal of Diabetes Science and Technology*, vol. 1, no. 1, pp. 117–121, 2007.

[8] G. Sparacino, A. Facchinetti, and C. Cobelli, ""smart" continuous glucose monitoring sensors: On-line signal processing issues," *Sensors*, vol. 10, no. 6, pp. 6751–6772, 2010.

[9] A. Maran and et all., "Closed-loop artificial pancreas using subcutaneous glucose sensing and insulin delivery and a model predictive control algorithm: Preliminary studies in padova and montpellier," *Journal of Diabetes Science and Technology*, vol. 3, no. 5, pp. 1014–1021, 2009.

[10] E. Renard, "Implantable closed-loop glucose-sensing and insulin delivery pump therapy," *Current Opinion in Pharmacology*, vol. 2, no. 6, pp. 708–716, 2002.

[11] B. Beier, K. Musick, A. Matsumoto, A. Panitch, E. Naumann, and P. Irazoqui, "Toward a continuous intravascular glucose monitoring system," *Sensors*, vol. 11, no. 1, pp. 409–424, 2011.

[12] www.freestylenavigator.com.

[13] R. L. Weistein, S. L. Schwartz, R. L. Brazg, J. R. Bugler, T. A. Peyser, and G. V. McGarraugh, "Accuracy of the 5-day freestyle navigator continuous glucose monitoring system," *Diabetes Care*, vol. 30, no. 5, pp. 1125–1130, 2007.

[14] D. M. Wilson, R. W. Beck, W. V. Tamborlane, M. J. Dontchev, C. Kollman, P. Chase, L. A. Fox, K. J. Ruedy, E. Tsalikian, and S. A. Weinzimer, "The accuracy of the freestyle navigator continuous glucose monitoring system in children with type 1 diabetes," *Diabetes Care*, vol. 30, no. 1, pp. 59–64, 2007.

[15] www.dexcom.com/products/seven_difference.

[16] S. Garg, H. Zisser, S. Schwartz, T. Bailey, R. Kaplan, S. Ellis, and L. Jovanvic, "Improvement in glycemic excursion with a trancutaneous, real-time continuous glucose sensor," *Diabetes Care*, vol. 29, no. 1, pp. 44–50, 2006.

[17] www.minimed.com/products/guardian/index.html.

[18] J. Mastrototaro and S. Lee, "The integrated minimed paradigm real-time insulin pump and glucose monitoring system: Implications for improved patient outcomes," *Diabetes Technology & Therapeutics*, vol. 11, no. 1, pp. S37–S44, 2009.

[19] A. Maran, "Continuous subcutaneous glucose monitoring in diabetic patients," *Diabetes Care*, vol. 25, no. 2, pp. 347–352, 2002.

[20] T. Kubiak, B. Woerle, B. Kuhr, I. Nied, G. Glaesner, N. Hermanns, B. Kulzer, and T. Haak, "Microdialysis-based 48-hour continuous glucose monitoring with glucoday: clinical performance and patient's acceptance," *Diabetes Technology & Therapeutics*, vol. 8, no. 5, pp. 570–575, 2006.

[21] A. Tura, A. Maran, and G. Pacini, "Non-invasive glucose monitoring: Assessment of technologies and devices according to quantitative criteria," *Diabetes Research and Clinical Practice*, vol. 77, no. 1, pp. 16–40, 2007.

[22] K. Pitzer, S. Desai, T. Dunn, S. Edelman, Y. Jayalakshimi, J. Kennedy, J. A. Tamada, and R. O. Potts, "Detection oh hypoglycemia with the glucowatch biographer," *Diabetes Care*, vol. 24, no. 5, pp. 881–885, 2001.

[23] A. Tura, "Advances in the development of devices for noninvasive glycemia monitoring: who will win the race?" *Nutritional Therapy & Metabolism*, vol. 28, no. 1, pp. 33–39, 2010.

[24] S. Y. Rhee, S. Chon, G. koh, J. R. Paeng, S. Oh, J. Woo, S. W. Kim, J. Kim, and Y. S. Kim, "Clinical experience of an iontophoresis based glucose measuring system," *J Korean Med Sci*, vol. 22, no. 5, pp. 70–73, 2007.

[25] C. T. S. Ching, T. P. Sun, S. H. Huang, H. L. Shieh, and C. Y. Chen, "Mediated glucose biosensor incorporated with reverse iontophoresis function for noninvasive glucose monitoring," *Annals of Biomedical Engineering*, vol. 38, no. 4, pp. 1548–1555, 2010.

[26] B. M. Becker, S. Helfrich, E. Backer, K. Lovgren, A. Minugh, and J. Machan, "Ultrasound with topical anesthetic rapidly decreases pain of intravenous cannulation," *Acad Emerg Med*, vol. 12, no. 4, pp. 289–285, 2005.

[27] www.echotx.com/symphony-tcgm system.html.

[28] H. Chuang, M. Trieu, J. Hurley, E. J. Taylor, M. R. England, and S. A. Nasraway, "Pilot studies of transedrmal continuous glucose measurement in outpatient diabetic and in patients during and after cardiac surgery," *Journal of Diabetes Science and Technology*, vol. 2, no. 4, pp. 595–602, 2008.

[29] P. Zakharov, F. Dewarrat, A. Caduff, and M. Talary, "The effect of blood content on the optical and dielectric skin properties," *Physiological Measurement*, vol. 32, no. 1, pp. 131–151, 2011.

[30] C. S. Chen, K. K. Wang, M. Y. Jan, W. C. Hsu, S. P. Li, Y. Y. Wang-Lin, and J. G. Bau, "Noninvasive blood glucose monitoring using the optical signal of pulsatile microcirculation: a pilot study in subjects with diabetes," *Journal of Diabetes and Its Complications*, vol. 22, no. 6, pp. 371–376, 2008.

[31] O. Amir, D. Weinstein, S. Zilberman, M. Less, D. Perl-Treves, H. Primack, A. Weinstein, E. Gabis, B. Fikhte, and A. Karasik, "Continuous non invasive glucose monitoring technology based on "occlusion spectroscopy"," *Journal of Diabetes Science and Technology*, vol. 1, no. 4, pp. 463–469, 2007.

[32] R. A. Gabbay and S. Sivarajah, "Optical coherence tomography-based continuous noninvasive glucose monitoring in patients with diabetes," *Diabetes Technology & Therapeutics*, vol. 10, no. 3, pp. 188–193, 2008.

[33] N. S. Oliver, C. Toumazou, E. G. Cass, and G. Johnston, "Glucose sensors: a review of current and emerging technology," *Diabetic Medicine*, vol. 26, no. 3, pp. 197–210, 2009.

[34] G. Yosipovitch, E. Hodak, P. Vardi, I. Shraga, M. Karp, E. Sprecher, and M. David, "The prevalence of cutaneous manifestations in IDDM patients and their association with diabetes risk factors and microvascular complications," *Diabetes Care*, vol. 21, no. 4, pp. 506–509, 1998.

[35] www.inlughtsolutions.com/prod glu.html2.

[36] C. E. Ferrante do Amaral and B. Wolf, "Current development in non-invasive glucose monitoring," *Medical Engineering & Physics*, vol. 30, no. 5, pp. 541–549, 2008.

[37] D. A. Stuart, J. M. Yuen, N. Shah, O. Lyandres, C. R. Yonzon, M. R. Glucksberg, J. T. Walsh, and R. P. Van Duyne, "In vivo glucose measurement by surface-enhanced raman spectroscopy," *Analytical Chemistry*, vol. 78, no. 20, pp. 7211–7215, 2006.

[38] A. M. K. Enejder, T. G. Scecina, M. Hunter, W.-C. Shih, M. S. Feld, J. Oh, S. Sasic, and G. L. Horowitz, "Non invasive blood glucose monitoring using the optical signal of pulsatile microcirculation: a pilot study in subjects with diabetes," *Raman spectroscopy for noninvasive glucose measurements*, vol. 10, no. 3, pp. 031 114–1–031 114–9, 2005.

[39] www.orsense.com/Glucose.

[40] R. Badugu, J. R. Lakowicz, and C. D. Geddes, "A glucose-sensing contact lens: from bench top to patient," *Current Opinion in Biotechnology*, vol. 16, no. 1, pp. 100 – 107, 2005.

[41] H. Shibata, Y. J. Heo, T. Okitsu, Y. Matsunaga, T. Kawanishi, and S. Takeuchi, "Injectable hydrogel microbeads for fluorescence-based in vivo continuous glucose monitoring," *PNAS*, vol. 1, no. 5, pp. 1–5, 2010.

[42] B. H. Malik and G. L. Coté, "Real-time, closed-loop dual-wavelength optical polarimetry for glucose monitoring," *Journal of Biomedical Optics*, vol. 15, no. 1, pp. 017 002/1–017 002/6, 2010.

[43] C. D. Malchoff, K. Shoukri, J. I. Landau, and J. M. Buchert, "A novel noninvasive blood glucose monitor," *Diabetes Care*, vol. 25, no. 12, pp. 2268–2275, 2002.

[44] R. Weiss, Y. Yegorchikov, A. Shusterman, and I. Raz, "Non invasive continuous glucose monitoring using photoacoustic technology-results from the first 62 subjects," *Journal of Diabetes Science and Technology*, vol. 9, no. 1, pp. 68–74, 2007.

[45] A. Tura, S. Sbrignadello, D. Cianciavicchia, G. Pacini, and P. Ravazzani, "A low frequency electromagnetic sensor for indirect measurement of glucose concentration: In vitro experiments in different conductive solutions," *Sensors*, vol. 10, no. 6, pp. 5346–5358, 2010.

[46] A. Caduff, M. Talary, and P. Zakharov, "Cutaneous blood perfusion as a perturbing factor for non invasive glucose monitoring," *Diabetes Technology & Therapeutics*, vol. 12, no. 1, pp. 1–9, 2010.

[47] Y. Hayashi, L. Livshits, A. Caduff, and Y. Feldmann, "Dielectric spectroscopy study of specific glucose influence on human erythrocyte membranes," *Journal of Physics D: applied physics*, vol. 36, no. 4, pp. 369–374, 2003.

[48] A. Caduff, E. Hirt, Y. Feldman, Z. Ali, and L. Heinemann, "First human experiments with a novel non-invasive, non-optical continuous glucose monitoring system," *Biosensors and Bioelectronics*, vol. 19, no. 3, pp. 209–217, 2003.

[49] A. Caduff, F. Dewarrat, M. Talary, G. Stalder, L. Heinemann, and Y. Feldman, "Non-invasive glucose monitoring in patients with diabetes: a novel system based on impedance spectroscopy," *Biosensors and Bioelectronics*, vol. 22, no. 5, pp. 598–604, 2006.

[50] T. Forst, A. Caduff, M. Talary, M. Weder, M. Braendle, P. Kann, F. Flacke, C. Friedrich, and A. Pfuetzner, "Impact of environmental temperature on skin thickness and microvascular blood flow in subjects with and without diabetes," *Diabetes Technology & Therapeutics*, vol. 8, no. 1, pp. 94–101, 2006.

[51] R. Bellman, *Adaptive control processes: a guided tour*. Princeton University Press, 1961.

[52] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, 2nd ed. Springer Verlag, 2009.

[53] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. pp. 407–451, 2004.

[54] S. di Jong, "SIMPLS: an alternative approch to partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. pp. 251–263, 1993.

[55] W. Fu, "Penalized regressions: the bridge versus the lasso," *Journal of computational and graphical statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[56] S. Shevade and S. Keerthi, "A simple and efficient algorithm for gene selection using sparse logistic regression," *Bioinformatics*, vol. 19, no. 17, p. 2246, 2003.

[57] S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast, incremental feature selection by gradient descent in function space," *The Journal of Machine Learning Research*, vol. 3, pp. 1333–1356, 2003.

[58] M. Park and T. Hastie, "L_1-regularization path algorithm for generalized linear models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 659–677, 2007.

[59] S. Rosset, "Following curved regularized optimization solution paths," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds., 2005, pp. 1153–1160.

[60] G. Andrew and J. Gao, "Scalable training of L_1-regularized log-linear models," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 33–40.

[61] S. Lee, H. Lee, P. Abbeel, and A. Ng, "Efficient L_1-regularized logistic regression," in *proceedings of the national conference on artificial intelligence*, vol. 21, no. 1, 2006, p. 401.

[62] M. Figueiredo, "Adaptive sparseness for supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1050–1159, 2003.

[63] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 957–968, 2005.

[64] Y. Lee and O. Mangasarian, "SSVM: A smooth support vector machine for classification," *Computational optimization and Applications*, vol. 20, no. 1, pp. 5–22, 2001.

[65] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point Method for Large-Scale L_1-Regularized Least Squares," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 606–617, 2007.

[66] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.

[67] E. M. Gafni and D. P. Bertsekas, "Two-metric projection methods for constrained optimization," *SIAM Journal on Control and Optimization*, vol. 22, no. 6, pp. 936–964, 1984.

[68] B. H. Zou, T. Hastie, and R. Tibshirani, "On the "degrees of freedom" of the lasso," *The Annals of Statistics*, vol. 35, no. 5, pp. 2173–2192, 2007.

[69] A. Facchinetti, S. Guerra, G. Sparacino, G. D. Nicolao, and C. Cobelli, "Method to recalibrate continuous glucose monitoring data on-line," Nov. 2 2010, International Patent Application PCT/IB2010/054947.

[70] S. Guerra, A. Facchinetti, G. Sparacino, G. De Nicolao, and C. Cobelli, "Comparison of four methods for on-line calibration of CGM data," in *Proceedings of the 9th Diabetes Technology Meeting (DTM)*, November 2009, p. A51.