UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO ACADÊMICO EM CIÊNCIA COMPUTAÇÃO

JULIO ALBERTO SIBAJA RETTES

ROBUST ALGORITHMS FOR LINEAR REGRESSION AND LOCALLY LINEAR EMBEDDING

FORTALEZA
2017

JULIO ALBERTO SIBAJA RETTES

ROBUST ALGORITHMS FOR LINEAR REGRESSION AND LOCALLY LINEAR EMBEDDING

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. João Fernando Lima Alcantara

Co-Orientador: Prof. Dr. Francesco Corona

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R345a Rettes, Julio Alberto Sibaja.

Algoritmos robustos para regressão linear e locally linear embedding / Julio Alberto Sibaja Rettes. – 2017.

105 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2017.

Orientação: Prof. Dr. João Fernando Lima Alcântara.

Coorientação: Prof. Dr. Francesco Corona.

1. Outliers. 2. Estatística robusta. 3. Regressão linear. 4. Redução da dimensionalidade. 5. Locally Linear Embedding. I. Título.

CDD 005

JULIO ALBERTO SIBAJA RETTES

ROBUST ALGORITHMS FOR LINEAR REGRESSION AND LOCALLY LINEAR EMBEDDING

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. João Fernando Lima
Alcantara (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Francesco Corona (Co-Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. João Paulo Pordeus Gomes
Universidade Federal do Ceará (UFC)

Prof. Dr. Amauri Holanda de Souza Júnior
Instituto Federal de Educação, Ciência e Tecnologia
do Ceará (IFCE)

ACKNOWLEDGEMENTS

Writing this thesis was made possible by the financial support of the National Counsel of Technological and Scientific Development (CNPq). I would like to express my gratitude to my adviser Dr. Francesco Corona for his dedication and guidance in the entire process. His help, motivation and availability to work were essential to me. I want to thank Prof. João Paulo Pordeus for introducing me to the robust statistics field. I would also like to thank Prof. Carlos Brito for his exciting and interesting teaching methods. I am grateful to Dra. Michela Mulas for allowing me to work in her office, for all the support and for all the coffees.

Special thanks to my family for believing in me and for giving me all their support and love. I would like to extend my sincerest thanks to Margie Miller for her invaluable assistance in this work. I would like to thank my friends in the apartamiedo, my youths for visiting me and Romulo for the words of encouragement. I am also very grateful with Noemie for standing by my side with love and patience. Lastly, I thank to every person who makes me feel happy and inspired with their kindness, smiles and good wishes.

ABSTRACT

Nowadays a very large quantity of data is flowing around our digital society. There is a growing interest in converting this large amount of data into valuable and useful information. Machine learning plays an essential role in the transformation of data into knowledge. However, the probability of outliers inside the data is too high to marginalize the importance of robust algorithms. To understand that, various models of outliers are studied.

In this work, several robust estimators within the generalized linear model for regression framework are discussed and analyzed: namely, the M-Estimator, the S-Estimator, the MM-Estimator, the RANSAC and the Theil-Sen estimator. This choice is motivated by the necessity of examining algorithms with different working principles. In particular, the M-, S-, MM-Estimator are based on a modification of the least squares criterion, whereas the RANSAC is based on finding the smallest subset of points that guarantees a predefined model accuracy. The Theil Sen, on the other hand, uses the median of least square models to estimate. The performance of the estimators under a wide range of experimental conditions is compared and analyzed.

In addition to the linear regression problem, the dimensionality reduction problem is considered. More specifically, the locally linear embedding, the principal component analysis and some robust approaches of them are treated. Motivated by giving some robustness to the LLE algorithm, the RALLE algorithm is proposed. Its main idea is to use different sizes of neighborhoods to construct the weights of the points; to achieve this, the RAPCA is executed in each set of neighbors and the risky points are discarded from the corresponding neighborhood. The performance of the LLE, the RLLE and the RALLE over some datasets is evaluated.

Keywords: Outliers. Robustness. Linear Regression. Dimensionality Reduction. Locally Linear Embedding.

RESUMO

Na atualidade um grande volume de dados é produzido na nossa sociedade digital. Existe um crescente interesse em converter esses dados em informação útil e o aprendizado de máquinas tem um papel central nessa transformação de dados em conhecimento. Por outro lado, a probabilidade dos dados conterem outliers é muito alta para ignorar a importância dos algoritmos robustos. Para se familiarizar com isso, são estudados vários modelos de outliers.

Neste trabalho, discutimos e analisamos vários estimadores robustos dentro do contexto dos modelos de regressão linear generalizados: são eles o M-Estimator, o S-Estimator, o MM-Estimator, o RANSAC e o Theil-Senestimator. A escolha dos estimadores é motivada pelo principio de explorar algoritmos com distintos conceitos de funcionamento. Em particular os estimadores M, S e MM são baseados na modificação do critério de minimização dos mínimos quadrados, enquanto que o RANSAC se fundamenta em achar o menor subconjunto que permita garantir uma acurácia predefinida ao modelo. Por outro lado o Theil-Sen usa a mediana de modelos obtidos usando mínimos quadradosno processo de estimação. O desempenho dos estimadores em uma ampla gama de condições experimentais é comparado e analisado.

Além do problema de regressão linear, considera-se o problema de redução da dimensionalidade. Especificamente, são tratados o Locally Linear Embedding, o Principal ComponentAnalysis e outras abordagens robustas destes. É proposto um método denominado RALLE com a motivação de prover de robustez ao algoritmo de LLE. A ideia principal é usar vizinhanças de tamanhos variáveis para construir os pesos dos pontos; para fazer isto possível, o RAPCA é executado em cada grupo de vizinhos e os pontos sob risco são descartados da vizinhança correspondente. É feita uma avaliação do desempenho do LLE, do RLLE e do RALLE sobre algumas bases de dados.

Palavras-chave: Outliers. Estatística Robusta. Regressão Linear. Redução de Dimensionalidade. Locally Linear Embedding.

LIST OF FIGURES

LIST OF TABLES

LISTA DE ABREVIATURAS E SIGLAS

AE	Asymptotic Efficiency
BDP	Breakdown Point
CR	Croux and Ruiz-Gazen
IRLS	Iteratively Reweighted Least Squares
LLE	Locally Linear Embedding
LMS	Least Mean Squares
LTS	Least Trimmed Squares
MAD	Median Absolute Deviation
ML	Machine Learning
MSE	Mean Squared Error
MSS	Minimal Sample Set
OLS	Ordinary Least Squares
PCA	Principal Component Analysis
RBF	Radial Basis Functions
TC	Trustworthiness and Continuity

LIST OF SYMBOLS

x	Scalars are denoted by lower case Roman letters
$\mathbf{x}$	Vetors are denoted by a lower case bold Roman letters
x_j	Denotes the jth scalar element in the vector x
$\mathbf{M}^{\mathrm{T}}$	The transpose of a Matrix is denoted by superscript T
$\mathbf{M}$	Matrix are denoted by uppercase letters
$\mathbf{x}_j$	Denotes the jth vector of some matrix $\mathbf{M}$
$(w_1, ..., w_M)$	This notations represent a row vector with M elements
$\mathbb{R}^{n \times m}$	Denotes the vector space of matrices with entries in the real numbers that have an n by m rank
$[a, b]$	Represents a *closed* interval from a to b
(a, b)	Denotes the *open* interval excluding a and b
$f(x)$	Function f evaluating a variable x
$\mathbb{E}_x[f(x)]$	Expectation of a function$f(x)$ with respect to a random variable x
$P(x)$	Probability P of a random variable x
$\lvert \cdot \rvert$	Absolut value or cardinality of a set
$\lVert \cdot \rVert$	Euclidean Norm of a vector
$\mathbf{I}$	Represents the Identity matrix

CONTENTS

1 INTRODUCTION

1.1 Machine Learning

The main purpose of the Machine Learning (ML) is the implementation of algorithms capable of learning from data. For Murphy (2012, p. 1) the concept of ML is related with a constant effort to make automated methods for analyzing data. There is no doubt about the interdisciplinarity of the machine learning field, including probability and statistics, information theory, artificial intelligence and others (MITCHELL, 1997, 2).

In the machine learning universe, the word train is highly related with the word learn. According to Bishop (2006, p. 2), if someone uses a process that takes a data set and train to get the parameters of an adaptive model $\boldsymbol{\beta}$, then the person is using the machine learning approach. In the training phase the algorithm is learning from the data. The next phases will depend on the kind of the problem; one part of the problems that we treat in this work involve, in the last phase, the prediction of outcomes for new unknown data (see Chapter 2). Then generalization is our aim, and it happens when an ML algorithm is capable to recognize or categorize a new data which was not part of the data set used in the training phase (BISHOP, 2006, p. 2).

Preprocessing, such as dimensionality reduction, is a common technique executed in the data before any type of training; it will depend however on the nature of the problem, but more importantly, on the nature of the ML implementation algorithm (BISHOP, 2006, p. 2). Another relevant process is the design, which implies the completion of four steps (MITCHELL, 1997, p. 13): The first step is to determine the type of training experience (from whom or where will the algorithm learn?). The second step is to determine the objective function (What do you want to minimize?). The third step is to determine the representation of the model structure (i.e Linear, Polynomial, Neural Network). The last step is to determine the learning algorithm (How to minimize the objective function?).

There are parametric and non parametric processes in machine learning. The use of parameters makes the dependency on the data distribution assumptions stronger; Nevertheless, the computation of the model becomes faster. On the other hand, the non-parametric models are more flexible but computationally heavier (MURPHY, 2012, p. 16).

Probability theory has a really important role in machine learning. One of the most influential theorems in ML is the Bayes theorem (BISHOP, 2006, p. 15). Frequently in the probabilist machine learning, the maximum likelihood estimator that maximizes the function

$p(D|\tilde{\beta})$ is used, in which the model $\tilde{\beta}$ is chosen in regard to maximize the probability of the data D. Using the maximum likelihood approach, it is not reasonable to merely focus on the performance of the model over the training set, because overfitting (failure to generalize) is a frequent problem. To select the model that generalizes better, the data can be partitioned in two parts and using the biggest part to train and the other part, called the test set, to evaluate the performance of the model. Another option is to use cross-validation, the technique which splits the data into n folds, and then iterates using S_i as test set and $D \supsetneq S_i \ \forall i$ as the training set (BISHOP, 2006, p. 32-33).

Basically, learning paradigms can be divided into two principal types: the unsupervised learning or descriptive learning and the supervised learning or predictive learning (MURPHY, 2012, p. 2). We are talking about them in the Sections 1.1.1 and 1.1.2. The range of problems that can be treated with an unsupervised learning algorithm is wider than the range of problems which the supervised ML is capable to solve (MURPHY, 2012). The latter can be explained because of the inherent requirement of the supervised ML to train with labeled data. Some authors believe in the capacity to extract information coming out from the pure data itself, and also hold that the unsupervised ML is more related with the learning capacity of the animals (including the human species) than the supervised ML (MURPHY, 2012, p. 9-10). In contrast to this, the most widely used type is the supervised ML (MURPHY, 2012, 3).

There is a third type of machine learning called reinforcement learning that is related to learning based on the reward/punishment obtained for executing actions (MURPHY, 2012, 2). Therefore the objective of reinforcement learning is to learn which choices would maximize the absolute reward; its learning process typically uses a kind of trial/error technique (BISHOP, 2006, 3). Examples of problems using the reinforcement learning type are puzzle games, manufacturing problems and scheduling problems (MITCHELL, 1997, p. 367-368). The detailed treatment of reinforcement leaning lies beyond the scope of this work.

1.1.1 Unsupervised Learning

‘No outcome measure’ can be a representative phrase to describe unsupervised learning. In this type of machine learning the main goal is to discover patterns, associations or structures inside the data. In an unsupervised machine learning problem, we have a set of observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, and a density estimator can be used to accomplish the calculation of the model (FRIEDMAN *et al.*, 2001, p. 486).

The model $\boldsymbol{\beta}$ of parameters is made in the form of $p(\mathbf{x}_i|\boldsymbol{\beta})$. The intention is to infer the properties from the model $\boldsymbol{\beta}$ without having the right properties to correct the training process (FRIEDMAN *et al.*, 2001, p. 486). Examples of this type of learning are clustering and dimensionality reduction.

1.1.1.1 Cluster Analysis

The cluster analysis technique, also known as data segmentation, is one of the most popular unsupervised learning procedures. As its name suggest, it is related to finding, inside of the dataset, subgroups or segments called subsets or clusters (FRIEDMAN *et al.*, 2001, p. 501). In this technique, the elements of the dataset are identified by a set of properties and it is expected to obtain clusters, where all the elements inside each cluster have similar properties (FRIEDMAN *et al.*, 2001, 502).

1.1.1.2 Dimensionality Reduction

Dimensionality reduction can be useful in some contexts as a data pre-processing stage, still in other cases as a necessary procedure. Dimensionality reduction is a technique concerned with the transformation of the dimensionality of the data. As its name suggests this transformation is made to obtain a new meaningful dataset with less dimensionality than the original set (MAATEN *et al.*, 2009, p. 1).Some examples of real world datasets that commonly need a reduction of their dimensionality to be processed are digital photographies, speech signals and fMRI scans (MAATEN *et al.*, 2009, p. 1). The reduction of the dimensionality of data can also be helpful to achieve lossy data compression, data visualization and exploratory analysis (BISHOP, 2006, p. 561)(ROWEIS; SAUL, 2000, p. 1).

The processes of dimensionality reduction used in this context are the ones in which the dimension of the data is reduced (Feature Extraction) but not the number of observations. In this work, two methods of dimensionality reduction are explained: the Principal Component Analisys (PCA) and Local Linear Embedding (LLE), and some robust approaches of them, too. The theory about the algorithms is presented in detail in Chapter 3.

1.1.2 Supervised Learning

In supervised learning the objective is to learn from labeled data. This means that if we have n observations in the dataset, each element $\mathbf{x}_i$ has another element or label $\mathbf{t}_i$. In other words, the data consist of a set of input vectors $\mathbf{x}_i$ along with their corresponding output vector $\mathbf{t}_i\ \forall i$ (BISHOP, 2006, p. 3).

The input vector $\mathbf{x}_i$ can represent almost anything, from simple integers to phrases or audios. In the same way, the output can express a wide range of information, but it has to be represented into a categorical (qualitative, discrete) variable or nominal (quantitative, continue) variable. These conditions in the specification of the output lead to the categorization of the problem in two types: a classification problem when the output is categorical and a regression problem when the output is nominal. Besides classification and regression a special type of problem remains, it happens when the output presents qualitative and quantitative values. Then you can use a combination of methods used in the two first types if it is possible. (FRIEDMAN *et al.*, 2001, p. 10).

There exists a third type of supervised learning problem. It occurs when the output is an ordered variable (such as cold, warm, hot), but it is not in the scope of this work to explain the details of this problem (For details of this type see Friedman *et al.* (2001, p. 10)).

1.1.2.1 Classification

Classification supervised problem arises when the output is represented using qualitative variables. The most common and practical option is to encode all the possible outputs or categories using a numeric form. If there are only two categories (called binary classification), the simplest way is to encode using '1' to represent one category and '0' to represent the other category. On the other hand if there are $k > 2$ categories (called multiclass classification) a K binary variable scheme is normally used. In this scheme each output variable is represented in a vector $\mathbf{t} \in \mathbb{R}^k$, and all the values inside the vector are zero except the ith position that represents the number of the category, then $\mathbf{t}_i = 1$ and $\mathbf{t}_j = 0\ \ \forall j \neq i$ (FRIEDMAN *et al.*, 2001, 10).

Using the probabilistic approach, if C_k represents the class specified in the output t and D the dataset, the conditional probability function $p(C_k|\mathbf{x}_i, D)$ denotes the probability of C_k

over the input $\mathbf{x}$ (BISHOP, 2006, p. 180). Applying the Bayes theorem,

$$p(C_k|\mathbf{x}_i, D) = \frac{p(\mathbf{x}_i, D|C_k)p(C_k)}{p(\mathbf{x}_i, D)} \tag{1.1}$$

and using the maximum posterior to solve the classification problem, we say that the model represents the values y_i, such

$$y_i = \max_k p(C_k|\mathbf{x}_i, D) \ \forall i. \tag{1.2}$$

1.1.2.2 Regression

Regression is a type of supervised learning problem where the output variables are continuous. The inputs are also called explanatory variables and the outputs are the response variables. Using the probabilistic approach, regression is very similar to classification, since the main idea is to find the model in which

$$y_i = \max p(y_i|\mathbf{x}_i, D) \ \forall i. \tag{1.3}$$

In this work the models are represented by linear functions. The use of linear functions for regression is one of the most representative approaches and carries advantages for studying purposes (BISHOP, 2006, 137).

1.2 Outliers

The definitions of *Outlier* found in the literature are usually different from each other and there is no consensus about which of them is the most accurate. The word outlier literally means something that stands outside of somewhere. It is important to understand that the context of the problem that you are trying to solve makes one definition of outlier fit better than another.

Using a viewpoint more focused on the linear regression problems, Rousseeuw e Leroy (1987, p. 7) define Outlier as a point i where "$(\mathbf{x}_i, t_i)$ deviates from the linear relation followed by the majority of the data, taking into account both the explanatory variables and the response variable simultaneously". A broader definition can be found where Susanti *et al.* (2014, p. 1) used the word outlier like a synonym of extreme observation. Another explanation is provided in the work of Andersen (2008, p. 31), who says than an outlier is "a datum which sits away from the rest of the distribution for a particular variable, either response or explanatory". Analogous to Andersen, Freire e Barreto (2014, p. 1) treat outliers as inconsistent data points with the remainder points within the data set.

Anscombe (1960) mentioned the three main causes of variability in the data, that can arise in the occurrence of outliers. The first one is the measurement errors and are mostly related to instrument errors or the misuse of them. The second cause is the execution faults (e.g. changes in the system, mis-selection of some samples, design errors). Finally the third one is the intrinsic variability of the data. Generally, it is not an easy task to identify which of the last three causes is the origin of an outlier (BARNETT, 1978), and to then apply some method to deal with the spurious data.

One technique that can come to mind is to simply erase or remove the data which show similar patterns with some definition of outlier, but we cannot simply remove the data points from the dataset (HUBER; RONCHETTI, 2009, p. 4). The main reasons are:

1. It is not trivial to recognize the outliers (even more in multidimensional problems) in one step and then make the regression.
2. Even executing a process to clean the dataset of outliers, is common to make false rejections and false retentions. The resulting data set still cannot have normal distribution, and this can culminate in a more difficult problem. It is better to use robust methods than two-step (reject-regress) approaches.
3. In Hampbel experiments, the performance achieved by the best robust procedures looks to be higher than the best rejection procedures. In addition to that, the traditional rejection methods seem to suffer 'masking' when multiple outliers affect the data. (Studies made by Hampbel 1974,1985)

Some people may ask why they have to take care of the outliers, probably thinking that their dataset only has clear or right values. The occurrence of data with peculiarities or attributes far away from the bulk of data is almost present in each data set in the *real world*. Hampel (1973) joined the conclusions in a wide range of scientific works, and he stained that between "5-10% wrong values in a data set seem to be the rule rather than the exception".

In the Chapters 2 and 3, some algorithms designed to overcome the problem of outliers without (detecting and) deleting them from the dataset are described.

1.2.1 Models of Outliers

It is already known that usually the real world datasets contain outliers; on the other hand, if we are going to make artificial datasets with the goal of testing/comparing algorithms, it is probably a good decision to include a percentage of outliers, in the same way that a normal dataset would have so included. In this section are showed some models that describe the presence of outliers in one dataset; some are general models that can be used in almost any problem and others are more specific for linear regression problems.

Barnett (1978, p. ag248) proposes a list of model alternatives. He defines $\boldsymbol{H}$ as the hypothesis which claims that the data have an $\boldsymbol{F}$ distribution, so

$$\boldsymbol{H} : x_j \in \boldsymbol{F}\, \forall j.$$

One second hypothesis $\tilde{\boldsymbol{H}}$ determine the real distribution of the data, and at the same time, it is the explanation for the outliers present in the dataset. The following list explains the five alternatives of how $\tilde{\boldsymbol{H}}$ can be defined:

(B1) **Deterministic Alternative**

x_i is already known as an outlier, so

$$\tilde{\boldsymbol{H}} : x_j \in \boldsymbol{F}\, \forall\, (j \neq i).$$

(B2) **Inherent Alternative**

Here we have another distribution for all the data

$$\tilde{\boldsymbol{H}} : x_j \in \boldsymbol{G} \not\equiv \boldsymbol{F}\, \forall\, j.$$

(B3) **Mixture Alternative**

In this case we use λ, where $0 \leq \lambda \leq 1$, to make a mix of the **F** distribution from the original hypotesis **H** and another different distribution. Hence

$$\tilde{\boldsymbol{H}} : x_j \in (1-\lambda)\boldsymbol{F} + \lambda\boldsymbol{G}\, \forall j.$$

(B4) **Slippage Alternative**

This alternative is the most used,

$$\tilde{\boldsymbol{H}} : \begin{cases} x_j \in \boldsymbol{F}\, \forall\, (j \neq i), \\ x_i \in \boldsymbol{G}. \end{cases}$$

Additionally to that, Barnett (1978) saids that frequently $\boldsymbol{F} \equiv \boldsymbol{G}$, but, if $\boldsymbol{F} \sim (\mu, \sigma^2) \Rightarrow \boldsymbol{G} \sim (\mu + a, \sigma^2)$ or $\boldsymbol{G} \sim (\mu, b\sigma^2)$, where $(a > 0)$ and $(b > 1)$.

(B5) **Exchangeable Alternative**

This is an extension of the previous alternative, “assuming that the index i of the discordant value is equally likely to be $(1,2,...,n)$."

$$\tilde{\boldsymbol{H}}: \begin{cases} x_j \in \boldsymbol{F}\ \forall j \\ x_i \in \boldsymbol{G}, \\ p(i=j) = n^{-1}\ \forall j. \end{cases}$$

The following model is given by Horata *et al.* (2013). Suppose that we are working with simple linear regression, and therefore we have a target (output) vector **t**. For each element of the **t** vector we have an associated element **x** in the inputs; then either $\tilde{\mathbf{t}}$ is the target vector contaminated with 'one-sided' outliers or $\hat{\mathbf{t}}$ is the target vector contaminated with 'two-sided' outliers and the **o** vector is generated using normal distribution. Then we can define:

$$\tilde{t}_j = t_j + |o_j|,\ \forall j, \tag{1.4}$$

$$\hat{t}_j = t_j + o_j,\ \forall j. \tag{1.5}$$

The standard deviation σ of the **o** vector is an aspect not mentioned in the paper of Horata *et al.* (2013). It is possible to apply this generation model to other kind of problems, simply considering the contamination of other *target vectors* or variables. That is done in order to generate outliers in any dimension of the data.

The last outlier model described in this work is proposed by Wang *et al.* (2010). The explanation is made with the assumption of working with simple regression, but the same as in the last model, it can be applied in any dimension of any problem. Then we have a target vector **t** and the list of steps are:

1. Find out the maximum value *max* and the minimum value *min* of **t**, calculate it's mean μ and standard deviation σ too.
2. Calculate an upper M^{upper} margin and lower M^{lower} margin defined by

$$M^{upper} = ||max| - \sigma|, \tag{1.6}$$

$$M^{lower} = ||min| - \sigma|. \tag{1.7}$$

3. Generate k random values, where:

$$I \subseteq ([max, (max + M^{upper})] \cup [(min - M^{lower}), min]) \tag{1.8}$$

$$\text{and } |I| = k \tag{1.9}$$

In addition to the given models, a combination of the information provided by the context of the problem and statistics techniques can be found in literature. This is with the intention of providing an "adjusted to problem" model of outliers. For example, Ratcliff (1993, p. 511) knows that the common distribution of the time data for some chemical reactions is the ex-Gaussian with some specific parameters μ, σ and τ; then for the outliers he generates data with different sets of values for the parameters μ, σ and τ.

1.2.2 Leverage Points

A simple definition of leverage point is given by Rousseeuw e Yohai (1984, p. 257); they describe it as “an outlying in some $\mathbf{x}_i$, $i \in (1,...,n)$”. The type of effect that can be made by a leverage point differs from the effect produced by outliers in t_i (ROUSSEEUW; LEROY, 1987, p. 7). Generally the occurrence of leverage points can be explained if the $\mathbf{x}_i \exists i$ are not generated artificially; In other words, this happens mostly with data obtained from an observable real problem (ROUSSEEUW; ZOMEREN, 1990, p. 634).

Xin e Xiaogang (2009, p. 137) affirm that the points lying far *enough* from the spatial center of the other explanatory variables have leverage. This definition of leverage point does not include any relation with the response variable but rather with the explanatory. However, the relation between the two variables $(\mathbf{x}_i, t_i)$ is also important because that can define the type of the leverage points.

There are two types of leverage points. If the outlying $\mathbf{x}_i$ does not suggest a break in linear regression pattern followed by the bulk of the data, then we can affirm that it is a *good* leverage point; Otherwise it is a *bad* leverage point and represents a regression outlier (STUART, 2011, p. 6).

This work does not have as a goal to deal with leverage points, nor in analyzing the effects of them in the experiments, as we see in the Chapter 4. All the regressors chosen, except the ordinary least squares, protect in some degree from the outlying of the response variables.

1.3 Robustness

In this work the terms robust and resistant are treated as equivalents. Wrong modeling assumptions about the distribution of the residuals can cause serious problems. In the classical algorithms as least squares and principal component analysis (see Sections 2.1.2 and 3.2 respectively), the assumption is the normality of the residuals, and only one point “that does not follow the assumption of normallity” is capable to break the resultant model (ROUSSEEUW;

YOHAI, 1984, p. 256).

The notion of robustness in algorithms “signifies insensitivity to small deviations from the assumptions” (HUBER; RONCHETTI, 2009, p. 2) but involves more important concepts which persuades us not to trust totally in the assumptions made (STUART, 2011, p. 2). Thus, the resistance cannot be achieved by merely ignoring the points detected as spurious, like some people may think (ROUSSEEUW; LEROY, 1987, p. 8).

In the book *Robust Statistics* of Huber e Ronchetti (2009), the authors define a list with 3 principal features (efficiency, stability and breakdown) that a good robust statistical procedure should have. The first statement says that it must have a reasonably high efficiency, almost optimal; the second one is related to the repercussion of small deviations from the assumptions made, and stands to maintain low the asymptotic variance of the estimate; the last one emphasizes that if some big deviations appear from the assumptions made, it will not result in a catastrophe.

1.4 Motivation and Objectives

Many of the methods used in the classical statistic are known to be non robust. There are classic algorithms that are highly affected by deviations on the assumptions made over the structure of the data. Some of these methods are highly used until today for solving their related problems; they are also combined with other methods and that makes their scope larger. Nowadays, it is common to deal with big high-dimensional datasets; and it makes complicated to understand the underlying structure of the data and then make the correct assumptions to create the models.

There is a motivation to comprehend how the robust statistics works. The linear models, as Bishop says, have “nice analytical properties and are probably the best way to start if you want to understand more sophistical models” (BISHOP, 2006, 137). Hence, in this thesis, the linear regression experiments are designed to understand how some both classic and robust algorithms perform under wrong data modeling assumptions. To achieve that, some outlier generation methods and some robustness features in liner regression have to be explained. The objective is to analyze the performance of the ordinary least squares, the M-Estimator, the S-Estimator, the MM-Estimator, the RANSAC and the Theil-Sen Estimator under a wide range of experimental conditions.

Some of the data-driven lattice DR algorithms are simple techniques that finds solutions with global minimum. The locally linear embedding is one of them and its robust approaches are just a few. This leads to propose a new robust approach for the locally linear embedding method. To do that, a formal description of the Locally Linear Embedding (LLE), the RLLE, the RALLE, the Principal Component Analysis (PCA), the weighted PCA and the RAPCA is developed. The objective is to analyze the performance of the LLE, RLLE and RALLE under some experimental datasets.

The performance of the linear regressors is measured with the mean squared error. Meanwhile, the performance of the dimensionality reduction techniques is evaluated with the trustworthiness and continuity.

1.5 Overview and Organization of the Thesis

In the numerical datasets, the rows represent instances or elements and the columns are the attributes or variables. Spurious elements can be found in almost any dataset of the real word. The main algorithms proposed to deal with the linear regression problems and the dimensionality reduction problems are not developed to cope with these type of instances. Therefore, anyone can suggest the use of some process to analyze and find the anomalous elements inside the data; then erase these points from the dataset and continue with the typical procedure. This approach is sometimes employed, but as it was explained above, it can caries some additional problems and it can be also an inviable procedure to apply. It can be hard to perceive some underlying features within the data, but is even harder to understand them completely.

The robust statistics area has been developed since the middle of the last century (Tukey (1960)) and until today (DIAKONIKOLAS *et al.*, 2016) it is still developing techniques to cope with deviations from the assumptions made. In the scope of the thesis, the robust algorithms are designed to achieve the already mentioned problems without the application of some pre-processing phase. The treatment that the robust algorithms do to the data differs between each of them; and the performance also varies depending on the conditions that the spurious elements are disposed into the dataset.

This thesis is mainly organized into three parts; the first part involves the theory of the generalized linear regression models and some robust approaches of them; the second part of the work is related with the locally linear embedding and some methods to provide it with robustness; lastly, the third part describes the experiments made with synthetic and real-world data and also it is developed an analysis and a comparative of the performances of the algorithms.

The Chapter 2 begins with the definition of the generalized linear models and the use of basis functions. A set of robust features in linear regression used to analyze the estimators are explained. Besides that, the robust linear regression algorithms are detailed. These are the M-Estimator, the S-Estimator, the MM-Estimator, the RANSAC and the Theil Sen.

Chapter 3 treats dimensionality reduction topics. The locally linear embedding, the principal component analysis, the weighted principal component analysis, the RAPCA and the RLLE are explained. Additionally, a new robust approach for the locally linear embedding is formally presented; its name is RALLE.

Inside the Chapter 4, the entire experimentation process of the linear regression is described. The generation of the synthetic datasets and the description of the real-world data are detailed. The results of the experiments are analyzed and used to compare the algorithms. The Chapter 5 is similar to the previous one, except, that it is related to the locally linear embedding algorithms.

Finally, in Chapter 6, the obtained results are summarized and the thesis is concluded.

Part I

Robust Generalized Linear Regression

2 ROBUST GENERALIZED LINEAR REGRESSION

Knowing that every dataset commonly includes some percentage of outliers, the implementation of robust regression algorithms to make the models is probably the most reasonable decision. At the same time, this choice may be required if is known that the residuals are not normally distributed or have an unknown distribution (SUSANTI *et al.*, 2014, p. 351).

The main goal of regression is to predict continuous target y from an input vector $\mathbf{x}$, in that case we call it *simple* regression. If the prediction regards a target vector $\mathbf{y}$ the problem is called *multivariate* regression. In this work, we are concerned with simple linear regression.

Robust estimators of regression are the ones not highly affected by the outliers in the dataset (ROUSSEEUW; LEROY, 1987, p. 8). Huber e Ronchetti (2009, p. 8) mention that one goal of using robust methods “is to safeguard against deviations from the assumptions, in particular against those that are near or below the limits of detectability”.

2.1 Generalized Linear Models

We have a set of n observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, where each observation $\mathbf{x}_i \in \mathbb{R}^D$ is associated with a target value $t_i \in \mathbb{R}$. The D dimensional vectors $\mathbf{x}_i$ are called as input or explanatory variables because they are used to explain output or response variable $t = y(\mathbf{x}, \boldsymbol{\beta}) + \varepsilon$, in which ε is the zero mean Gaussian noise and $\boldsymbol{\beta}$ is the m dimensional vectors of parameters (XIN; XIAOGANG, 2009, p. 9).

The generalized linear regression model $y(\mathbf{x}, \boldsymbol{\beta})$ is given by

$$y(\mathbf{x}, \boldsymbol{\beta}) = \beta_0 + \sum_{j=1}^{m-1} \beta_j \phi_j(\mathbf{x}), \tag{2.1}$$

where $\phi_j(\mathbf{x})$ are known as basis functions and represent some transformation of the input $\mathbf{x}$; the $m-1$ value define the quantity of basis functions used in the problem (BISHOP, 2006). Classic linear regression is a particular case of this general model in which $\phi(\mathbf{x}) = \mathbf{x}$. Whatever the choice of the basis functions, the generalized regression model is linear in the parameters $\boldsymbol{\beta}$. The parameter β_0, also known as the *bias parameter*, corresponds to the output when all the inputs are 0. If we add to the basis functions the $\phi_0(\mathbf{x}) = 1$ term, then we can write the Equation 2.1 as follows

$$y(\mathbf{x}, \boldsymbol{\beta}) = \sum_{j=0}^{m-1} \beta_j \phi_j(\mathbf{x}) = \boldsymbol{\beta}^{\mathbf{T}} \phi(\mathbf{x}). \tag{2.2}$$

2.1.1 Basis Functions

The basis functions $\phi_j(\mathbf{x})$ can be non-linear, thus allowing $y(\mathbf{x},\boldsymbol{\beta})$ to be nonlinear in the inputs, but maintaining the linearity in the transformed inputs and in the parameters $\boldsymbol{\beta}$. Examples of basis functions are polynomials, splines, radial basis functions, wavelets, logarithmic transformations, and others (KOHN *et al.*, 2001, p. 139) (BISHOP, 2006).In the following we show some examples of basis functions.

The first example are Radial Basis Functions (RBF). Various types of RBFs are popular in neural networks, and in general they are characterized by the calculation of distances between the input vectors and some other point called center (FRIEDMAN *et al.*, 2001, p. 36). A commonly used type of RBF is the Gaussian Kernel

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{||\mathbf{x}-\mathbf{u}_j||}{2s_j^2}\right), \tag{2.3}$$

where the term $\mathbf{u}_j$ from $j = (1,...,m)$ represents the centers and s_j determines the scale of the space (FRIEDMAN *et al.*, 2001, p. 139) (BISHOP, 2006, p. 36). The norm of the 2 vectors can represent the Euclidean distance or some other measure of spatial distance.

Another common example of basis function is the sigmoid. This basis function is defined as

$$\phi_j(\mathbf{x}) = \sigma\left(-\frac{||\mathbf{x}-\mathbf{u}_j||}{s_j}\right), \tag{2.4}$$

where $\sigma(a)$ represents the logistic sigmoid function $\sigma(a) = 1/(1+\exp(-a))$. Note that the sigmoid uses the terms $\mathbf{u}_j$ and s_j in the same way as Gaussian Kernel does.

In the following, we assume that the set of observations $\mathbf{X} \in \mathbb{R}^{n\times m}$ have been already transformed by the ϕ_j functions, $\forall j$ and with $\phi_0 = 1$.

2.1.2 Classic Least Squares

The Least Squares algorithm is the standard method for estimating, from data, the parameters $\boldsymbol{\beta}$ in generalized linear regression. The most commonly used variant of this method is called Ordinary Least Squares (OLS) (SUSANTI *et al.*, 2014).

We define the input matrix as $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\mathbf{X} = [\mathbf{x}_1,....,\mathbf{x}_n]^{\mathrm{T}}$; it represents all the dataset coming from n points and m explanatory variables, and a target vector $\mathbf{t}$, then OLS is the method which obtains the model $\tilde{\boldsymbol{\beta}}$ that minimizes the sum of squared error $E(\boldsymbol{\beta})$,

$$E(\boldsymbol{\beta}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i)^2. \tag{2.5}$$

To obtain the model $\tilde{\boldsymbol{\beta}}$ which minimizes the sum of squares, we set the derivative

$$\nabla_{\boldsymbol{\beta}}E(\boldsymbol{\beta}) = \sum_{i=1}^{n}(t_i - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i)\mathbf{x}_i^{\mathrm{T}} \tag{2.6}$$

to 0, obtaining that

$$0 = \sum_{i=1}^{n}(t_i\mathbf{x}_i^{\mathrm{T}}) - \tilde{\boldsymbol{\beta}}^{\mathrm{T}}(\sum_{i=1}^{n}(\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}) \tag{2.7}$$

If we solve the Equation 2.7 for $\tilde{\boldsymbol{\beta}}$, then we have the normal equations for the least squares:

$$\tilde{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t} \tag{2.8}$$

We can use the $\mathbf{X}^{\dagger}$ Moore-Penrose pseudo-inverse of $\mathbf{X}$, in the Equation (2.8) which is defined as follows:

$$\mathbf{X}^{\dagger} \equiv (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}} \tag{2.9}$$

Minimizing the sum of squares error is equivalent to the maximization of the likelihood under the assumption of Gaussian noise distribution of r (BISHOP, 2006, p. 141).

The OLS, as a classical method, is one of the chosen regression algorithms in this work being evaluated and compared in the experiments made.

2.1.3 Mean Squared Error

The Mean Squared Error (MSE) of an estimator is a classic measure used to determine the performance of an estimate (XIN; XIAOGANG, 2009, p. 238). It is defined as:

$$MSE(\hat{\boldsymbol{\beta}}) = \mathbb{E}[(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^2] = \mathrm{Var}(\hat{\boldsymbol{\beta}}) + (\mathbb{E}[\hat{\boldsymbol{\beta}}] - \boldsymbol{\beta})^2. \tag{2.10}$$

This is known as the Bias-Variance trade off (MURPHY, 2012, p. 202). That is why even though the least squares is the one that achieves the lowest variance (among all the unbiased estimators), it will not necessarily reach the minimum MSE (FRIEDMAN *et al.*, 2001, p. 52). The MSE can be calculated for a test dataset as

$$MSE(\hat{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{t}_i - y(\mathbf{x}_i, \hat{\boldsymbol{\beta}})^2. \tag{2.11}$$

2.1.4 Robustness Features in Linear Regression

There are many robust methods to choose, and it can be a difficult task to determine which methods are better than others. Beyond the performance (in terms of error or outliers), there are other global measures to compare the methods. Thus we discuss in more detail what is Efficiency, Breakdown Point and Equivariance (STUART, 2011, p. 8).

2.1.4.1 Equivariance

Given an estimator S with $S(\mathbf{X},\mathbf{t}) = \boldsymbol{\beta}$, we can say that S is equivariant if it is possible to define linear transformations over some of the problem data and the model preserve consistency (ROUSSEEUW; LEROY, 1987, p. 116) (KOENKER; JR, 1978, p. 39). In other words an estimator can be considered equivariant if it can treat a problem as invariant under certain linear transformations (DAVIES *et al.*, 1993, p. 1861). The three types of equivariance and their linear transformations are (STUART, 2011, p. 9):

- The estimator is *regression (shift) equivariant* if an “additional linear dependence $\mathbf{t} \rightarrow \mathbf{t} + \mathbf{X}a$” can be reflected in $\boldsymbol{\beta} \rightarrow \boldsymbol{\beta} + a$:

$$T(\{(\mathbf{x}_i, t_i + \mathbf{x}_i\mathbf{v})\}) = T(\{(\mathbf{x}_i, t_i)\}) + \mathbf{v}, \quad \forall i. \tag{2.12}$$

- The estimator is *scale equivariant* if it guarantees independence over the scale of the response variable $\mathbf{t}$, so any transformation $\mathbf{t} \rightarrow a\mathbf{t}$ is reflected in $\boldsymbol{\beta} \rightarrow a\boldsymbol{\beta}$:

$$T(\{(\mathbf{x}_i, ct_i)\}) = cT(\{(\mathbf{x}_i, t_i)\}), \quad \forall i. \tag{2.13}$$

- The estimator is *affine equivariant* if guarantee independence over the transformations of the explanatory variables $\mathbf{X}$ or reparametrization of design, when $\mathbf{X} \rightarrow \mathbf{AX}$ is reflected in $\boldsymbol{\beta} \rightarrow \mathbf{A}^{-1}\boldsymbol{\beta}$:

$$T(\{(\mathbf{X}^{\mathrm{T}}\mathbf{A}, \mathbf{t})\}) = \mathbf{A}^{-1}T(\{(\mathbf{X}^{\mathrm{T}}, \mathbf{t})\}). \tag{2.14}$$

2.1.4.2 Breakdown Point of an estimator

Breakdown Point (BDP) is defined by the “smallest fraction of contamination that can cause the estimate to break down and no longer represent the trend in the bulk of the data” (STUART, 2011, p. 8). The first mathematical asymptotic definition was formulated by Hampbel

(1971), but it fell into disuse because Dohono and Huber (1983) introduced the finite-sample version.

In the least squares, only one point(outlier) can totally spoil the obtained model. However, this is not the case with others regressors that can handle a considerable portion of outliers in the data (ROUSSEEUW; LEROY, 1987, p. 9).

Rousseeuw e Leroy (1987, p. 10) describes the BDP as follows: If you have a dataset $(\mathbf{X}, \mathbf{t}) \in \mathbb{R}^{nxm}$ and take a sample $\mathscr{Z}$ where $\mathscr{Z} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{p \leq n}$, and the estimator S that produces the vector of parameters $\boldsymbol{\beta}$, where $S(\mathscr{Z}) = \boldsymbol{\beta}$. Then if you corrupt a quantity of points $j \leq p$ of $\mathscr{Z}$ with arbitrary values, to obtain a new sample $\mathscr{Z}'$. Thus, the maximum effect E caused in the resultant model by the corruption of $\mathscr{Z}$ is

$$E(j, S, \mathscr{Z}) = \sup ||S(\mathscr{Z}) - S(\mathscr{Z}')|| \tag{2.15}$$

So we define the breakdown point of T as

$$BDP(S, \mathscr{Z}) = \min(\frac{n}{j} : E(j, S, \mathscr{Z}) = \infty) \tag{2.16}$$

BDP is a simple but helpful measure to evaluate the resistance of an estimator. The best proportion (or resistance) that can be achieved is 1/2 (HAMPEL, 1973, p. 97). The explanation for that maximum BDP can be done if you imagine the case when your data presents 50% of outliers; in that situation the regressor cannot be able to distinguish which part of your set is the right and which is the wrong (STUART, 2011, p. 9). Another relevant point to remark is that BDP is not the unique measure to evaluate robustness; as BDP, efficiency, equivariance and others as well are important to have a complete view of the resistance of the estimator.

2.1.4.3 *Asymptotic Efficiency*

Given an estimator S where $S(\mathbf{X}, \mathbf{t}) = \boldsymbol{\beta}$, the efficiency $e(S)$ is a ratio calculated by the minimum possible variance divided by the actual variance of S. It is clear that the best possible ratio is when the two variances in the division are the same (ratio equal to 1). In our context (regression), when the assumption of Gaussian distribution of the residual r is satisfied, the minimum possible variance is achieved by the least squares estimator LS with $T(\mathbf{X}, \mathbf{t}) = \tilde{\boldsymbol{\beta}}$, then the efficiency of a generalized linear regression estimator S is (ANDERSEN, 2008, p. 9-10)

$$e(S) = \frac{\mathbb{E}\left[(t_i - \tilde{\boldsymbol{\beta}}^{\mathrm{T}} \mathbf{x}_i)^2\right]}{\mathbb{E}\left[(t_i - \boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i)^2\right]}. \tag{2.17}$$

To close the definition, the phrase *asymptotic efficiency* is used because the calculation of the efficiency is made over a sample with infinite size (STUART, 2011, p. 9).

2.2 M Estimator

Maximum Likelihood type estimators of M-Estimators was proposed by Huber(1973) with the intention of moderating the effect produced by the minimization of the squared error. The goal is to maintain the high efficiency of the OLS in the presence of normal error distribution and replace the objective function of the sum of squared residuals by some other robust function (ROUSSEEUW; LEROY, 1987, p. 148).

Using this type of estimators is possible to constrain the influence of values located ‘far away’ from the regression line (there the presence of *real* outliers is probably higher). The M-estimators employ some objective function that can decrease the influence in a smooth way, compared with the tough influence made by the distinction between the ‘good’ and the ‘bad’ observations of the classic rejection. Beyond that, it works better in the situations when the distribution is similar to normal but not normal (like a fatter distribution), because the rejection options are almost non-viable (HAMPEL, 1973, p. 10).

The M-Estimator principle is to find the model $\tilde{\boldsymbol{\beta}}$ which minimizes the $E(\boldsymbol{\beta})$ function, defined as

$$E(\boldsymbol{\beta}) = \sum_{i=1}^{n} \rho(r_i). \tag{2.18}$$

The residual $r_i = t_i - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i$, and the robust ρ function is recommended to be symmetric and with a unique minimum at 0 (ROUSSEEUW; LEROY, 1987, p. 12). To choose the best ρ function is a must to know the distribution of the errors, but that is commonly unknown. If $\rho(r_i) = r_i^2$ is used, the estimator is the same as OLS (STUART, 2011, p. 11-12). The original version of this algorithm is not scale equivariant, unless one change in the calculation of the function ρ be made (HUBER; RONCHETTI, 2009, p. 106). The Equation 2.18 has to be converted into

$$E(\boldsymbol{\beta}) = \sum_{i=1}^{n} \rho\left(\frac{r_i}{s}\right), \tag{2.19}$$

where s is calculated with an implementation of some scale estimator of the standard deviation of the data (see Section 2.2.1 for more details).

The M-Estimators have good flexibility properties because of the possibility to choose the ρ function, plus they “generalize straightforwardly to multiparameter problems” (HUBER; RONCHETTI, 2009, p. 45).

Now that the equivariant form of an M-Estimator was introduced, the process to calculate the parameters can be developed (STUART, 2011, p. 15-16). The first step is to derivate the $\tilde{\beta}$ with respect to the m parameters and set this derivate equal to 0; before we introduce the ‘score function’ $\psi(u)$, which represents the first derivate of $\nabla_u\rho(u)$, then we get

$$\nabla_\beta E(\tilde{\beta}) = \sum_{i=1}^{n} \mathbf{x}_{ij}\psi\left(\frac{r_i}{s}\right) = 0\ \forall j. \tag{2.20}$$

Then the second definition is introduced, called ‘weight function’, which is defined by

$$w_i = \frac{\psi\left(\frac{r_i}{s}\right)}{\frac{r_i}{s}}. \tag{2.21}$$

Now if we made some substitutions in 2.20 we have:

$$\begin{aligned}
\sum_{i=1}^{n} \mathbf{x}_{ij}\psi\left(\frac{r_i}{s}\right) &= \sum_{i=1}^{n} \mathbf{x}_{ij}\psi\left(\frac{r_i}{s}\right)\frac{\left(\frac{r_i}{s}\right)}{\left(\frac{r_i}{s}\right)} = 0\ \forall j\\
\sum_{i=1}^{n} \mathbf{x}_{ij}w_i\left(\frac{r_i}{s}\right) &= \sum_{i=1}^{n} \mathbf{x}_{ij}w_i(t_i - \tilde{\beta}^{\mathrm{T}}\mathbf{x}_i)\frac{1}{s} = 0\ \forall j\\
\sum_{i=1}^{n} \mathbf{x}_{ij}w_i\tilde{\beta}^{\mathrm{T}}\mathbf{x}_i &= \sum_{i=1}^{n} \mathbf{x}_{ij}w_i y_i\ \forall j
\end{aligned} \tag{2.22}$$

then we can represent 2.22 in vectorial form:

$$\begin{aligned}
\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\tilde{\beta} &= \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{Y}\\
\tilde{\beta} &= (\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{Y}.
\end{aligned} \tag{2.23}$$

Where $\mathbf{W} \in \mathbb{R}^{n\times n}$ as the diagonal matrix with the values of w_i, or

$$\mathbf{W} = diag(w) = \begin{pmatrix} w_1 & \dots & 0\\ \vdots & \ddots & \vdots\\ 0 & \dots & w_n \end{pmatrix} \tag{2.24}$$

This is not a closed form solution, because the value of the matrix $\mathbf{W}$ depends in the value $\tilde{\boldsymbol{\beta}}$ and vice versa. Then can be applied the IRLS (see Section 2.2.3) to find $\tilde{\boldsymbol{\beta}}$. In the first step, the β is commonly calculated with the OLS, and then an iterative process is used, which is stopped after a q quantity of iterations or after achieving one tolerance value e, where e represents the distance between the last two generated models.

The M-Estimators are better at generalizing if you compare with other robust estimators, because it is possible to constraint and shape the impact of the errors with the weight function (HUBER; RONCHETTI, 2009, p. 70). The implementation of redescending type of M-Estimators is recommended as well (see Section 2.2.2 for details). Huber e Ronchetti (2009, p. 70) are discordant. They belief that the use of redescending M-Estimators are overrated and they have to be employed with some precautions because the re-descending type can increase the minimax risk compensating no more than 'a few percent of the asymptotic variance' when extreme outliers are present. Consequently they suggest rejecting the most improbable data. (HUBER; RONCHETTI, 2009, p. 101).

2.2.1 Standard Deviation - Scale Estimate

In order to provide the model of the scale equivariance property, we have to implement some robust estimator of a scale (standard error), and include the resultant standard error s into the calculation of the objective function as we can see in the Equation 2.19. (HAMPEL, 1973). There are some types of Scale estimator, like pure scale, nuisance parameter and Studentizing.

In relation to the standard deviation of the M-Estimators the re-scaled Median Absolute Deviation (MAD) is one of the possible scale estimators to use.(HUBER; RONCHETTI, 2009, p. 106). The re-scaled MAD is

$$s = 1.4826\,\text{MAD}, \tag{2.25}$$

where

$$\text{MAD} = \text{Med}\{t_i - \boldsymbol{\beta}^{\text{T}}\mathbf{x}_i\ \forall i\} = \text{Med}\{r_i\ \forall i\} \tag{2.26}$$

and Med $\{u_a\ \forall a\}$ is the median of the numbers in $\{u_a : a \in A\}$.

2.2.2 Function ρ

In the case of the M-estimator (see Section 2.2), the S-Estimator (see Section 2.3) and the Weighted Principal Component Analysis (see Section 3.2.2), it is needed to select a ρ function that transforms the value of the residuals. There are some popular functions and, in this work we selected the Least Squares, the Huber (original propose), the Hampel and the Tukey bisquare (biweight) to compare their objective, score and weight functions, see table 2.

The redescending M-Estimators are one especial type of M-Estimators, because using some of them, it can be achieved the highest possible breakdown point in the M-Estimators (MÜLLER, 2004, p. 2). The redescending M-Estimators reduce the maximal asymptotic variance and it is indicated to use when the distribution can be long-tail or have extreme outliers (HUBER; RONCHETTI, 2009, p. 97).

One estimator is redescending if the score function or first derivative ψ of the objective function ρ satisfies $\lim_{a\to\pm\infty}\psi(u)=0$. Specifically you can write it in the following way:

$$\psi(u)=0 \text{ for } |u|\geq c, \tag{2.27}$$

with any $c>0$.

2.2.3 Iterative Reweighted Least Squares

The Iteratively Reweighted Least Squares (IRLS) uses the Newton-Raphson routine (FRIEDMAN *et al.*, 2001, p. 299). It is commonly employed if the applied basis functions have any hidden parameter, or more generally, if is not possible to obtain a closed form from the derivative of the objective function ρ.

If σ is any function, for the OLS it is the squared residual $\sigma(r_i)=r_i^2$, and $r_i=t_i-\beta^{\mathrm{T}}\mathbf{x}_i$ where $\beta\in\mathbb{R}^m$. Then we want to minimize the function

$$E(\tilde{\beta})=\sum_{i=1}^{n}\sigma(r_i). \tag{2.28}$$

So the Newton-Rapshon takes the form:

$$\beta^{(c)}=\beta^{(c-1)}-\mathbf{H}^{-1}\nabla\mathbf{E}(\beta), \tag{2.29}$$

$$\nabla\mathbf{E}(\beta)=\sum_{i=1}^{n}\psi(r_i)\mathbf{x}_{ij}^{\mathrm{T}}\ \forall\ j=(1,...,m),\ \text{where, } \psi(r_i)=\nabla\sigma(r_i) \tag{2.30}$$

Table 2 – Popular functions for M-Estimators

	Objective Function	Score Function	Weight Function
Cauchy	$\frac{c^2}{2}\log\left(1+\frac{x}{c}\right)^2$	$\frac{x}{1+\left(\frac{x}{c}\right)^2}$	$\frac{1}{1+\left(\frac{x}{c}\right)^2}$
Hampel	$\begin{cases} \frac{1}{2}u^2 & \text{if } \lvert u\rvert < a \\ a\lvert u\rvert - \frac{1}{2}a^2 & \text{if } a \le \lvert u\rvert < b \\ a\frac{c\lvert u\rvert - \frac{1}{2}u^2}{c-b} - \frac{7a^2}{6} & \text{if } b \le \lvert u\rvert \le c \\ a(b+c-a) & \text{if } \lvert u\rvert > c \end{cases}$	$\begin{cases} u & \text{if } \lvert u\rvert < a \\ a \text{ sign } u & \text{if } a \le \lvert u\rvert < b \\ a\frac{c \text{ sign } u - u}{c-b} & \text{if } b \le \lvert u\rvert \le c \\ 0 & \text{if } \lvert u\rvert > c \end{cases}$	$\begin{cases} 1 & \text{if } \lvert u\rvert < a \\ \frac{a}{\lvert u\rvert} & \text{if } a \le \lvert u\rvert < b \\ a\frac{\frac{c}{\lvert u\rvert} - 1}{c-b} & \text{if } b \le \lvert u\rvert \le c \\ 0 & \text{if } \lvert u\rvert > c \end{cases}$
Huber 1973	$\begin{cases} \frac{1}{2}u^2 & \text{if } \lvert u\rvert < a \\ a\lvert u\rvert - \frac{1}{2}a^2 & \text{if } \lvert u\rvert \ge a \end{cases}$	$\begin{cases} u & \text{if } \lvert u\rvert < a \\ a \text{ sign } u & \text{if } \lvert u\rvert \ge a \end{cases}$	$\begin{cases} 1 & \text{if } \lvert u\rvert < a \\ \frac{a}{\lvert u\rvert} & \text{if } \lvert u\rvert \ge a \end{cases}$
Least Squares	$\frac{1}{2}u^2 \quad -\infty \le u \le \infty$	u	1
L_1	$\lvert u\rvert$	$\text{sign } u$	$\frac{1}{\lvert u\rvert}$
Tukey bisquare	$\begin{cases} \frac{a^2}{6}\left(1-\left(1-\left(\frac{u}{a}\right)^2\right)^3\right) & \text{if } \lvert u\rvert \le a \\ \frac{1}{6}a^2 & \text{if } \lvert u\rvert > a \end{cases}$	$\begin{cases} u\left(1-\left(\frac{u}{a}\right)^2\right)^2 & \text{if } \lvert u\rvert \le a \\ 0 & \text{if } \lvert u\rvert > a \end{cases}$	$\begin{cases} \left(1-\left(\frac{u}{a}\right)^2\right)^2 & \text{if } \lvert u\rvert \le a \\ 0 & \text{if } \lvert u\rvert > a \end{cases}$

and

$$\mathbf{H} = \nabla\nabla\mathbf{E}(\beta), \tag{2.31}$$

where c represents the number of the actual iteration. The process is stopped when c is equal to a q quantity of iterations or after the distance between the last two generated models $\beta^{(c)}$ and $\beta^{(c-1)}$ achieve one tolerance value e. For the M-Estimators, fortunately the Equation 2.23 is already in the format of a weighted least squares, then the calculation of

$$\boldsymbol{\beta}^{(c)} = (\mathbf{X}^{\mathrm{T}}\mathbf{W}^{(c-1)}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{W}^{(c-1)}\mathbf{Y}, \tag{2.32}$$

where $W^{(c-1)}$ is determined applying the Equations 2.21 and 2.24 and using the vector of parameters $\boldsymbol{\beta}^{(c-1)}$. It is important to mention that the first iteration (when $c = 1$), $\boldsymbol{\beta}^{(c=1)}$ is calculated with the OLS or that which is the same using $\mathbf{W}^{(c-1=0)} = \mathbf{I}$.

2.3 S Estimator

The Robust Regression by means of S-Estimator is a linear regression algorithm proposed by Rousseeuw and Yohai (1984) with the main objective to minimize the residual scale (standard error) of the M-estimators (SUSANTI *et al.*, 2014, p. 354). Its name comes from the *Scale*-Estimator (ROUSSEEUW; YOHAI, 1984, p. 260). In order to introduce the S-estimator the estimator of scale $s(r_1,...,r_n)$ is presented, defined by the solution of

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{r_i}{s}\right) = K, \tag{2.33}$$

where $r_i = t_i - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i$, $r_i \in \mathbb{R}\ \forall\ i$, K is expected to be $\mathbb{E}_{\Phi}[\rho]$, Φ the standard normal and n the number of elements of the dataset (ROUSSEEUW; YOHAI, 1984, p. 260). If there are multiple solutions to Equation 2.33, then choose $\sup s$; if there are no solution then $s = 0$.

The chosen function ρ has to satisfy the following three conditions (STUART, 2011, p. 24)(the first two are mandatory and the third one is to reach the 50% of BDP):

1. The function ρ is symmetric, continuously differentiable and $\rho(0) = 0$.
2. The function ρ is redescending, particularly exist some $a > 0$ where $\rho(a)$ is strictly increasing in the interval $[o,a)$, and constant on the $[a,\infty)$.
3. $\dfrac{\mathbb{E}_{\Phi}[\rho]}{\rho(a)} = \dfrac{K}{\rho(a)} = \dfrac{1}{2}$.

Then the S-estimator is used to obtain the model parameters $\tilde{\boldsymbol{\beta}}$, that minimize the scale s, defined in the Equation 2.33, and is expressed by

$$\tilde{\boldsymbol{\beta}} = \min_{\beta} s(r_1(\boldsymbol{\beta}),...,r_n(\boldsymbol{\beta})), \tag{2.34}$$

and the scale estimator is

$$\tilde{\sigma} = s(r_1(\tilde{\boldsymbol{\beta}}),...,r_n(\tilde{\boldsymbol{\beta}}). \tag{2.35}$$

“It would be wrong to say that S-estimators are M-estimators, because their computation and breakdown point are completely different, but they do satisfy similar first-order necessary conditions”(ROUSSEEUW; LEROY, 1987, p. 141). The S-Estimators construct the model by a different approach than the M-estimator. S-Estimators search for the minimum scale of the residuals, and share with the M-Estimators the function that constrains the impact of the gross errors or the wrong assumptions. The main goal of Rousseeuw e Yohai (1984, p. 259) was to propose one estimator with a higher breakdown point than the M-Estimators, but at the same time share with them the "flexibility and nice asymptotic properties".

If the selections of the function ρ, the constant K and the parameter a are made properly, the 50% of BDP can be obtained, as demonstrated in the original proposal, when the authors (ROUSSEEUW; YOHAI, 1984, p. 261) use the Tukey Biweight (Bisquare) taking $a = 1.547$. Moreover in the same work, Rousseeuw e Yohai (1984, p. 262) generalize the calculation of the BPD for another combination of constant K and parameter a (and keeping tukey bisquare as the ρ function), and then defining $0 \leq \lambda \leq \frac{1}{2}$, they stand:

$$\frac{\mathbb{E}_{\Phi}[\rho]}{\rho(a)} = \frac{K}{\rho(a)} = \lambda \Longrightarrow \lim_{n \to \infty} \text{BDP} = \lambda. \tag{2.36}$$

They also explain that if the value of a is increased it will yield to an estimator with better asymptotic efficiency at Guassian Model, but as a consequence, the BDP will decrease.

The advantage of using the S-Estimator to estimate the model $\tilde{\beta}$ over other estimators is the capacity to achieve the best BDP; also Least Mean Squares (LMS) and Least Trimmed Squares (LTS) can reach the 50% of BDP (ROUSSEEUW; LEROY, 1987, p. 144-145).

2.3.1 Efficiency

Asymptotic Efficiency (AE). "There is a trade-off between robustness and efficiency for M and S-scale estimators" (AELST *et al.*, 2013, p. 280). In the next table we show how the S-estimators cannot estimate with a high BDP and at the same time with a high efficiency (under the error normal distribution assumption) (ROUSSEEUW; LEROY, 1987, p. 142).

Table 3 – (ROUSSEEUW; YOHAI, 1984, p. 268) BDP and AE of an S-Estimator

BDP	AE	a	K
50%	28.7%	1.547	0.1995
45%	37.0%	1.756	0.2312
40%	46.2%	1.988	0.2634
35%	56.0%	2.251	0.2957
30%	66.1%	2.560	0.3278
25%	75.9%	2.937	0.3593
20%	84.7%	3.420	0.3899
15%	91.7%	4.096	0.4194
10%	96.6%	5.182	0.4475

Some authors criticize the low efficiency, the instability, and the tradeoff BDP/efficiency, reaching the point of saying that the algorithm does not deserve to be called as “Robust” and it would be more appropriate to call it as just high BDP estimator (HUBER; RONCHETTI, 2009, p. 197). They also remark that the instability of the high breakdown point estimators is a known problem.

2.4 MM Estimator

This estimator was proposed by Yohai (1987) with the main intention of providing an estimator with 50% of BDP and at the same time with high asymptotic efficiency, under the assumption of normal distribution of the residuals. It is the first robust estimator proposed to have both high AS and BDP (STUART, 2011). The MM-estimator is a combination of one estimator with a high BDP (desirable 50%), an M-scale of the residuals and a computation of an M-Estimator.

The algorithm is defined in three steps:

1. Estimate the parameters $\hat{\boldsymbol{\beta}}$ using a high BDP estimator, a 50% of breakdown point is recommended. Examples of 50% BDP estimators are S-Estimator, the Least Trimmed Squares and the Least Median Squares (see Rousseeuw e Leroy (1987) for details).
2. Use an S-Estimator with the parameters $\hat{\boldsymbol{\beta}}$, to calculate the scale s_r (see Equation 2.35) of the residuals. The objective function used in this step is called ρ_0. The S-Estimator has to be tuned (objective function ρ_0 and its parameters) to produce an estimate with 50% BDP.
3. This stage uses the M-estimator described in the Section 2.2 with small modifications. As equal to M-estimator a function ρ has to be chosen. This ρ function has to be redescending (see Section 2.2.2) and satisfy that $\rho(u) \leq \rho_0(u)\ \forall\ u \in \mathbb{R}$. Then the goal of the MM-estimator is to find the parameters $\tilde{\boldsymbol{\beta}}$ which minimizes the $E(\boldsymbol{\beta})$function, defined as

$$E(\boldsymbol{\beta}) = \sum_{i=1}^{n} \rho \left(\frac{t_i - \boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i}{s_r} \right). \tag{2.37}$$

It is opportune to note that in this stage the value of the scale s_r will continues unchanged until the end of the algorithm. The score function $\psi(u)$ represents $\nabla_u \rho(u)$. Then to find the $\tilde{\boldsymbol{\beta}}$ we set $\nabla_{\beta} E(\tilde{\boldsymbol{\beta}}) = 0$ which is

$$\sum_{i=1}^{n} \mathbf{x}_{ij} \psi \left(\frac{r_i}{s_r} \right) = 0\ \forall j, \tag{2.38}$$

and must satisfy $E(\tilde{\boldsymbol{\beta}}) \leq E(\hat{\boldsymbol{\beta}})$. From here the process is the same as that in the M-estimator algorithm, defining the same weight function (Equation 2.21).

Note that the first and the second stage can be joined as one stage if you implement an S-Estimator (tuned to be 50% BDP) to get the parameters $\hat{\boldsymbol{\beta}}$ and utilize the scale obtained in that process. In the S-Estimator stage ρ_0 can be the Tukey Biweight function with $a = 1.547$ to provide to the S-estimator resistance to almost 50% of outliers (50% BDP). In the third stage, the ρ of the M-Estimator can be again the Tukey Bisquare. The parameter $a = 4.685$ guarantees a 95% of asymptotic efficiency in the final estimator for the Tukey bisquare or the chosen of $a = 2.697$ guarantees a 70% of asymptotic efficiency (VERARDI; CROUX, 2009, p. 5).

2.5 RANSAC

The RANdom SAmple Consensus (RANSAC) algorithm proposed by Fischler e Bolles (1981) is an estimator developed by the computer vision community. In their work, they stand that the smooth treatment made in the M, SS, MM and other robust estimator by the ρ function is not always the best approach to deal with outliers (FISCHLER; BOLLES, 1981, p. 1).

If the dataset contains the input variables $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]^\mathrm{T}$ and the output variables $\mathbf{t} \in \mathbb{R}^n$, RANSAC is an iterative parametric algorithm, also non-deterministic, that uses the minimal quantity of points needed to make the m parameters of the model $\boldsymbol{\beta}$. This quantity is called the Minimal Sample Set (MSS). If the vector of the parameters $\boldsymbol{\beta} \in \mathbb{R}^m$, then the MSS has to be equal to m. The threshold δ is a parameter value of the algorithm and is required to calculate the consensus set(CS) $\mathscr{S}$ where

$$\mathscr{S} = cs(\boldsymbol{\beta}) = \{\mathbf{x}_i : ||t_i - \boldsymbol{\beta}^\mathrm{T}\mathbf{x}_i|| \leq \delta, \quad \forall i\}. \tag{2.39}$$

The $|\mathscr{S}|$ represents the number of inliers and is a measure of how many points lie on the line described by $\boldsymbol{\beta}$ with a δ threshold. The distance between the line and a point is normally calculated with the Euclidean Norm. If a point $\mathbf{x}_i \subset \mathbf{S}$ then it is said that $\mathbf{x}_i$ is an inlier of the vector of parameters $\boldsymbol{\beta}$.

The RANSAC algorithm can be summarized in the following list of steps (HARTLEY; ZISSERMAN, 2004, p. 118):

1. Select a sample $\mathscr{R}$ of points stochastically from the dataset, with $|\mathscr{R}| \geq$ MSS.

2. Make the vector of parameters β using some estimator and the sample $\mathscr{R}$, commonly the ordinary least squares is used.
3. Using the Equation 2.39, calculate the consensus set $\mathscr{S}$ and determine its number of inliners.
4. If the percentage of number of inliners $\dfrac{|\mathscr{S}|}{n} \geq \tau$ reestimate the parameters of the model but now using the sample $\mathscr{S}$ to obtain the final vector of parameters $\tilde{\beta}$ and terminate. τ is a parameter of the algorithm, thus the value has to be previously defined.
5. If the percentage of number of inliners $\dfrac{|\mathscr{S}|}{n} < \tau$ and the iterations made of this steps are more than N, select the consensus set obtained with the greatest quantity of inliners. Otherwise, make another iteration starting from the step 1.

2.5.1 Parameters

To use the RANSAC algorithm, it has to be defined some parameters. This Section is about how to calculate the values of these parameters. The first approach described is given by the original paper by Fischler e Bolles (1981) and the parameters are δ, τ and N.

δ is a threshold which represents the tolerance of the residuals. The points that have a distance closer than δ from the model are called inliners. To obtain this threshold it is required to calculate the standard error σ_s between the model β and the entire dataset. Then the original paper proposed to set $\sigma_s \leq \delta \leq 2\sigma_s$ (FISCHLER; BOLLES, 1981, p. 383).

τ is a percentage representing how many inliers you want to have in your consensus set, and then to terminate the execution of the algorithm. A high value of τ can be a good choice, to assure that the model β represents the bulk of the data and the points inside the consensus help to build a good $\tilde{\beta}$ as well. It is assumed that the probability of one point being in the consensus set of any wrong model is y. We expect than no more than half of the data to be outliers, so we assume that $y < 0.5$. Then we have to make the $y^{\tau n - MSS}$ very low.

N represents the maximum number of iterations. p is commonly chosen to be 0.99 and represents the probability of taking a series of minimum sample sets $\mathscr{M}$ and that at least one of the sample sets be clean of outliers. Supposing that u is the probability to choose an inlier from the dataset, thus $(1-u)$ is the probability to choose an outlier. Then are required at least N

selections to achieve $(1-u^{|\mathscr{M}|})^N = 1-p$, doing manipulations

$$N = \frac{\log(1-p)}{\log(1-u^{|\mathscr{M}|})} \tag{2.40}$$

The second approach is more probabilistic (HARTLEY; ZISSERMAN, 2004, p. 118). The parameters are calculated, but it is needed two probabilities; the p has the same interpretation made in the first approach. For δ_p, let

$$\begin{aligned}\delta &= p(t_i - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i \leq \delta) \\ &= p\left(\frac{r_i^2}{\sigma_s^2} \leq \frac{\delta^2}{\sigma_s^2}\right) \forall i,\end{aligned} \tag{2.41}$$

and since $\dfrac{r_i}{\sigma_s}$ is normal distributed, then $\dfrac{\delta^2}{\sigma_s^2}$ has chi-square distribution. Then

$$\delta = \sigma_s \sqrt{\chi^2_{k,\delta_p}}, \tag{2.42}$$

where k are the degrees of freedom (dimensionality of the data) and δ^p is a probability to get just inliers inside the threshold δ. To calculate the quantity of iterations described in the Equation 2.40, the denominator of the fraction is recalculated, then

$$u^{|\mathscr{M}|} = \prod_{i=1}^{m} \frac{\tilde{n}}{n} \approx \left(\frac{\tilde{n}}{n}\right)^m, \tag{2.43}$$

where $\tilde{n}$ is the largest number of inliers found until that iteration.

2.6 Theil-Sen

Theil-Sen estimadors, was proposed by Theil (1950) and Sen (1968) and was originally a simple linear regression estimator. The original Theil-Sen is as an estimator of slope, and it has a BDP of 29,3% and high asymptotic efficiency (ROUSSEEUW; LEROY, 1987, p. 67).

We define the input vector $\mathbf{x} = [x_1, ..., x_n]^{\mathrm{T}} \in \mathbb{R}^n$; it represents some dataset coming from n points, and a target vector $\mathbf{t} \in \mathbb{R}^n$. The original Theil-Sen defines the slope as

$$\beta_1 = \operatorname{Med}\left\{ b_{i,j} = \frac{t_i - t_j}{x_i - x_j} : x_i \neq x_j, 1 \leq i \leq j \leq n \right\}, \tag{2.44}$$

where Med $\{u_a : a \in A\}$ is the median of the numbers in $\{u_a : a \in A\}$. Then bias parameter is calculated as

$$\beta_0 = \operatorname{Med}\{t_i - \beta_1 x_i : 1 \leq i \leq n\}, \tag{2.45}$$

For the multivariate linear regression estimator, the input defined as a matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]^{\mathrm{T}} \in \mathbb{R}^{n \times m}$ represents some dataset coming from n points and m explanatory variables, and the output stays as a vector $\mathbf{t} \in \mathbb{R}^n$. In the multivariate linear regression, the solution for a model with m parameters needs at least m points or equations. Then there has to be defined a set $\mathscr{K}$ where $\mathscr{K} \subset \{1, ..., n\}$ and $|\mathscr{K}| = m$. Therefore we can determine the parameters of the vector $\boldsymbol{\beta}_k$ as the resultant model of executing the least squares method with input matrix $\mathbf{X}_k = \{\mathbf{x}_i : \forall i \in \mathscr{K}\}$ and output vector $\mathbf{t}_k = \{t_i : \forall i \in \mathscr{K}\}$. Then the Theil-Sen estimator for multiple linear regression is defined as

$$\tilde{\boldsymbol{\beta}} = \mathrm{Mmed}\{\boldsymbol{\beta}_k : \ \forall k \in A\}, \tag{2.46}$$

where the set $A = \{\mathscr{K} : \det(\mathbf{X}_k) \neq 0\}$. The function $\det(\mathbf{A})$ is the determinant of the matrix $\mathbf{A}$ and the Mmed $\{\boldsymbol{\beta}_a : a \in U\}$ is the multivariate median of the vectors $\{\boldsymbol{\beta}_a : a \in U\}$.

Another extension to the Theil-sen estimator for multiple linear regression is the algorithm developed by Zhou e Serfling (2008). It uses the generalizations made in the algorithm that describes the parameters vector in the Equation 2.46 combined with a theory of spatial U-quantiles. We define N as all the combinations of pairwise differences of points in the data set, where the N pairwise differences redefine the regression model as

$$t_i - t_j = y(x_i - x_j, \boldsymbol{\beta}), \ \forall \ 1 \leq i < j \leq n. \tag{2.47}$$

Then we can define a generic pair (i, j) set $\mathscr{P} \in (\mathbb{R}, \mathbb{R})^m$, where $\mathscr{P} \subset \{(i, j)_1, ..., (i, j)_N\}$. Therefore we can define the parameters of the vector $\boldsymbol{\beta}_P$ as the OLS for which the input matrix is $\mathbf{X}_P = \{(\mathbf{x}_i - \mathbf{x}_j) : \forall (i, j) \in \mathscr{P}\}$ and the output vector is $\mathbf{t}_P = \{\mathbf{t}_i - \mathbf{t}_j : \forall (i, j) \in \mathscr{P}\}$. Then the vector of parameters of the Theil-Sen estimator for multiple linear regression is

$$\hat{\boldsymbol{\beta}} = \mathrm{Smed}\{\boldsymbol{\beta}_P : \forall \mathscr{P} \in A\}, \tag{2.48}$$

where the set $A = \{\mathscr{P} : \det(\mathbf{X}_P) \neq 0\}$ and Smed $\{\boldsymbol{\beta}_a : a \in U\}$ is the spatial median of the vectors $\{\boldsymbol{\beta}_a : a \in U\}$ (ZHOU; SERFLING, 2008, p. 7-8).

Part II

Robust Locally Linear Embedding

3 ROBUST LOCALLY LINEAR EMBEDDING

As well as the linear regression problem and knowing the issue that can represent the presence of outliers in our databases, the implementation of robust dimensionality reduction algorithms can improve the results (CHANG; YEUNG, 2006, p. 1). In this Chapter, two main dimensionality reduction methods are described; these are the locally linear embedding and principal component analysis. Additionally to that, some robust modifications of them are also explained and a new modification is proposed.

The type of reduction considered in this work is the feature extraction kind, in which the quantity of variables is maintained (MAATEN *et al.*, 2009, p. 1). Formally, if there is a set of n observations $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, where each observation $\mathbf{x}_i \in \mathbb{R}^D$, then a dimensionality reduction process can be represented by a function ρ, in which the new low-dimensional set is stored in some matrix $\mathbf{Y} = [\rho(\mathbf{x}_1), ..., \rho(\mathbf{x}_n)]^\mathrm{T}$. Each observation $\rho(\mathbf{x}_i) \in \mathbb{R}^K$, where $K \ll D$. It is desirable that the value of K correspond with the intrinsic dimensionality of the data. This means the minimum dimension in which the properties of the data are maintained (MAATEN *et al.*, 2009, p. 1).

3.1 Locally Linear Embedding

Locally linear embedding (LLE) uses the neighborhood of each point as a linear combination to reconstruct (represent) the point in its original high-dimensional space. As the principal component analysis method, LLE uses simple linear algebraic techniques and also it does not involve local minima (SAUL; ROWEIS, 2000, p. 2). Even though PCA is one of the most used techniques for dimensionality reduction, its accuracy in some transformation of non-linear data presents some issues. For that reason, other algorithms as locally linear embedding were proposed (MAATEN *et al.*, 2009, p. 1).

Considering the set $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, where the data is sampled for some underlying manifold and each observation $\mathbf{x}_i \in \mathbb{R}^D$. The classic locally linear embedding aims to use the local geometry properties between each point and its neighbors, for projecting the data in some K dimensional space. In LLE it is expected that the local geometry properties will be preserved in the less dimensional space as well. It uses the assumption that each point and its neighbors lies “on or close to a locally linear patch of the manifold” (ROWEIS; SAUL, 2000, p. 2323).

3.1.1 Formulation of the LLE

The main idea is to obtain i sets of linear coefficients that can reconstruct every point $\mathbf{x}_i$. That local geometry is obtained using a group of k neighbors. The selection of the neighborhoods is commonly achieved by choosing the k nearest neighbors of each point (using the Euclidean distance). However the selection of the neighbors can be made using another procedure or set of rules (SAUL; ROWEIS, 2000, p. 3). The reconstruction errors is the cost function defined by

$$\varepsilon = \sum_{i=1}^{n} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||^2, \tag{3.1}$$

where $\tilde{\mathbf{x}}_i$ is the reconstruction made by the set i of linear coefficients and the neighbors of the point $\mathbf{x}_i$.

The vector $\mathbf{n}_{ij}$ contains the value of the jth neighbor of the point i. Additionally the $\mathbf{W}$ matrix contains the linear coefficients (each row represents one point and each column represents the neighbor). Considering that the errors can be minimized independently, then we can rewrite the Equation 3.1 using the error for just one point of the dataset as

$$\begin{aligned}\varepsilon_i(\mathbf{W}_i) &= ||\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{n}_{ij}||^2 = ||\sum_{j=1}^{k} \mathbf{W}_{ij}(\mathbf{x}_i - \mathbf{n}_{ij})||^2 \\ &= \sum_{j=1}^{k}\sum_{m=1}^{k} \mathbf{W}_{ij}\mathbf{W}_{im}(\mathbf{x}_i - \mathbf{n}_{ij})(\mathbf{x}_i - \mathbf{n}_{im}) \\ &= \sum_{j=1}^{k}\sum_{m=1}^{k} \mathbf{W}_{ij}\mathbf{W}_{im}\mathbf{S}_{jm} = \mathbf{W}_i^{\mathrm{T}}\mathbf{S}\mathbf{W}_i,\end{aligned} \tag{3.2}$$

where the local covariance matrix $\mathbf{S}_{jm} = (\mathbf{x}_i - \mathbf{n}_{ij})(\mathbf{x}_i - \mathbf{n}_{im})$.

One important detail in this algorithm is the introduction of the constraint for the W coefficients or weights, where

$$\sum_{j=1}^{k} \mathbf{W}_{ij} = 1 \;\; \forall i. \tag{3.3}$$

This constraint has the objective of giving the property of regression (shift) equivariance or, what is the same, invariance to translations. More properties are implicit in the formulation of the cost function; these are the invariance to rescaling and the invariance to translations, equivalently called scale equivariance and affine equivariance respectively (see Section 2.1.4.1 for details). The symmetry in the weights, as a result of these three invariances, helps to maintain the geometric properties within the data in the new projected space (ROWEIS; SAUL, 2000, p. 2324).

It is desirable to calculate the weights or coefficients $\mathbf{W}_i$ that minimize the cost function ε. Using Lagrange to enforce the constraint of the weights sum, the new function

$$\tilde{\varepsilon}_i(\mathbf{W}_i) = \mathbf{W}_i^{\mathrm{T}}\mathbf{S}\mathbf{W}_i - \lambda(\mathbf{1}^{\mathrm{T}}\mathbf{W}_i - 1), \tag{3.4}$$

where $\mathbf{1} \in \mathbb{R}^k$ is the vector of ones. Then

$$\nabla_{\mathbf{W}_i}\tilde{\varepsilon} = 2\mathbf{S}\mathbf{W}_i - \lambda\mathbf{1} = 0 \quad \Longrightarrow \quad \mathbf{S}\mathbf{W}_i = \frac{\lambda}{2}\mathbf{1}, \tag{3.5}$$

where $\lambda/2 = 2/(\mathbf{1}^{\mathrm{T}}\mathbf{S}_i^{-1}\mathbf{1})$ to ensure that the sum of the coefficients inside $\mathbf{W}_i$ are equal to 1. Instead of doing that, the system of equations $\mathbf{S}\mathbf{W}_i = \mathbf{1}$ can be solved and then normalize the vector $\mathbf{W}_i$ (CHANG; YEUNG, 2006, p. 3). If for some reason (for example when $k > D$) the matrix of local covariance is singular or nearly singular, it can be solved by the addition of some small multiple of the identity matrix; then

$$\mathbf{S} = \mathbf{S} + \frac{\alpha}{k}I, \tag{3.6}$$

where α is a kind of hidden parameter of the algorithm and it can be some small fraction of the trace of the $\mathbf{S}$ matrix (SAUL; ROWEIS, 2000, p. 10). This penalizes “large weights that exploits correlations beyond some level of precision in the data” (LEE; VERLEYSEN, 2007, p. 155).

The second part of the locally linear embedding algorithm is the reconstruction of the points in the new less-dimensional space, using the coefficients inside the $\mathbf{W}$ matrix. To do that, each point $\mathbf{x}_i$ is now mapped into a new vector $\mathbf{y}_i \in \mathbb{R}^K$. The cost function for measuring the error of the reconstruction is

$$\begin{aligned}
\phi(\mathbf{Y}) &= \sum_{i=1}^{n} ||\mathbf{y}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{y}_j||^2 \\
&= \sum_{i=1}^{n} \mathbf{y}_i^2 - \mathbf{y}_i\left(\sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{y}_j\mathbf{y}_j\right) - \left(\sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{y}_j\right)\mathbf{y}_i + \left(\sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{y}_j\right)^2 \\
&= \mathbf{Y}^{\mathrm{T}}\mathbf{Y} - \mathbf{Y}^{\mathrm{T}}(\mathbf{W}\mathbf{Y}) - (\mathbf{W}\mathbf{Y})^{\mathrm{T}}\mathbf{Y} + (\mathbf{W}\mathbf{Y})^{\mathrm{T}}(\mathbf{W}\mathbf{Y}) \\
&= ((\mathbf{I} - \mathbf{W})\mathbf{Y})^{\mathrm{T}}((\mathbf{I} - \mathbf{W})\mathbf{Y}) \\
&= \mathbf{Y}^{\mathrm{T}}(\mathbf{I} - \mathbf{W})^{\mathrm{T}}(\mathbf{I} - \mathbf{W})\mathbf{Y} \\
&= \mathbf{Y}^{\mathrm{T}}\mathbf{M}\mathbf{Y},
\end{aligned} \tag{3.7}$$

where $\mathbf{M} = (\mathbf{I} - \mathbf{W})^{\mathrm{T}}(\mathbf{I} - \mathbf{W})$.

At this stage two constraints are introduced to the problem, both are related with the future projections y_i and their purpose is to make the problem well-posed (ROWEIS; SAUL, 2000, p. 2324). The first one stands for the mean of the data to be zero, then $\sum_{i=1}^{n} \mathbf{y}_i = 0$. The second constraint is that $\frac{1}{n}(\mathbf{Y}\mathbf{Y}^{\mathrm{T}}) = \mathbf{I}$. The objective function is modified to include the two constraints, using Lagrange multipliers

$$\tilde{\phi} = \mathbf{Y}^{\mathrm{T}}\mathbf{M}\mathbf{Y} - \boldsymbol{\lambda}(\frac{1}{n}\mathbf{Y}\mathbf{Y}^{\mathrm{T}} - \mathbf{I}), \tag{3.8}$$

and setting the derivative to zero,

$$\nabla_{\mathbf{Y}}\tilde{\phi} = 2\mathbf{M}\mathbf{Y} - \frac{2\boldsymbol{\lambda}}{n}\mathbf{Y} = 0 \quad \Longrightarrow \quad \mathbf{M}\mathbf{Y} = \frac{\boldsymbol{\lambda}}{n}\mathbf{Y}. \tag{3.9}$$

The resultant linear equations can be solved as an eigenvectors and eigenvalues problem. The smallest eigenvalue is supposed to be zero, and the corresponding eigenvector is to be a unit vector and is discarded to constraint the embedding with zero mean. Then the remaining K eigenvalues are the ones that provide the values of the projected data (SAUL; ROWEIS, 2000, p. 4). The proof of this statement will be developed in the maximum variance PCA formulation (Section 3.2.1.1). Therefore it is necessary to left multiply Equation 3.9 by $\mathbf{Y}^{\mathrm{T}}$ to obtain

$$\mathbf{Y}^{\mathrm{T}}\mathbf{M}\mathbf{Y} = \frac{\boldsymbol{\lambda}}{n} = \phi(\mathbf{Y}). \tag{3.10}$$

3.2 Principal Component Analysis

Discovering the latent factors, also known as latent variables, is based on the main idea of studying the variability among the variables in a dataset. This is with the intention of transforming the data in some less dimensional new dataset that carries the fundamental variability of the original data, called the latent factors (MURPHY, 2012, p. 11).

The Principal Component Analisys (PCA), also called Karhunen-Loève transform, is a linear method very popular in dimensionality reduction. PCA finds a set of linear orthogonal basis in which the variance in the data is maximized (MAATEN *et al.*, 2009, p. 3). It uses linear algebraic techniques that makes it simple to understand and implement, besides the absence of local minima inside the optimization (CHANG; YEUNG, 2006, p. 2).

3.2.1 Formulations

There are 2 approaches of how classic PCA can be defined. These approaches are the Maximun variance formulation and the Minimum Error formulation; both formulations will lead you to the same basic algorithm and they are covered in the following (BISHOP, 2006, p. 561). Besides the basic formulations, two versions of PCA designed to be outlier robust are discussed in the Sections 3.2.2 and 3.2.3, known as Weighted PCA and RAPCA respectively.

3.2.1.1 Maximum Variance

In this definition of principal component analysis, the variance of the low-dimensional projection is expected to be maximized. If it is considered a set of n observations $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ where each observation $\mathbf{x}_i \in \mathbb{R}^D$, then PCA is designed to reduce de dimensionality of the data and project it into a new K dimensional space maximizing the variance of the projected data and where $K \ll D$. To do that, it is build an orthonormal matrix $\mathbf{M} \in \mathbb{R}^{D \times K}$, thus $\mathbf{M}^{\mathrm{T}}\mathbf{M} = \mathbf{I}$. The $\mathbf{M}$ matrix defines a series of K directions, that project every point $\mathbf{x}_i$ in the less-dimensional space, where $\mathbf{M}^{\mathrm{T}}\mathbf{x}_i$ is the value of the projected point (BISHOP, 2006, p. 561).

The first stage to obtain the matrix $\mathbf{M}$ is the calculation of the mean $\overline{\mathbf{x}}$ of the original data, given by

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i, \tag{3.11}$$

consequently the covariance matrix $\mathbf{S}$ is calculated as

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^{\mathrm{T}}, \tag{3.12}$$

therefore the mean over the less-dimensional data can be calculated by $\mathbf{M}^{\mathrm{T}}\mathbf{x}_i$ and its covariance is given by

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{M}^{\mathrm{T}}\mathbf{x}_i - \mathbf{M}^{\mathrm{T}}\overline{\mathbf{x}})^2 = \mathbf{M}^{\mathrm{T}}\mathbf{S}\mathbf{M}. \tag{3.13}$$

In the second stage, the variance maximization over the less-dimensional data is executed. This can be achieved by the maximization of $\mathbf{M}^{\mathrm{T}}\mathbf{S}\mathbf{M}$ with respect to $\mathbf{M}$, and as $\mathbf{M}^{\mathrm{T}}\mathbf{M} = \mathbf{I}$, then the maximization is a constrained optimization problem. The set of linear equations that determines the maximization of the variances is given by

$$\mathbf{M}^{\mathrm{T}}\mathbf{S}\mathbf{M} + \lambda(I - \mathbf{M}^{\mathrm{T}}\mathbf{M}) = 0 \quad \Longrightarrow \quad \mathbf{S}\mathbf{M} = \lambda_{\boldsymbol{I}}\mathbf{M}. \tag{3.14}$$

The equations already include the Lagrange multiplier to ensure the constrain condition of the orthonormality of **M** and it can be treated as an eigenvectors and eigenvalues problem. The matrix $\boldsymbol{\lambda_I}$ is a diagonal matrix constructed from the corresponding eigenvalues of the problem. Left-multiplying the Equation 3.14 by $\mathbf{M}^{\mathrm{T}}$ can be done to prove that the largest eigenvalues are the ones that maximizes the variance of the projected data (Equation 3.13); then

$$\mathbf{M}_i^{\mathrm{T}}\mathbf{S}_i\mathbf{M}_i = \lambda_i, \tag{3.15}$$

where λ_i is the *ith* value inside the diagonal of $\boldsymbol{\lambda}_I$. All the principal components are the $\mathbf{M}_i$ eigenvectors, hence the first principal component is the one associated with the largest λ_i eigenvalue. That order relation is also applied for the next principal components and their respective eigenvalues. The eigenvector and eigenvalue decomposition can be made for any value of $K \leq D$ (BISHOP, 2006, p. 562).

3.2.1.2 Minimum Error

The second alternative that defines principal component analysis aims to minimize the error (distance) between the projection set and the original data set $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, where each of its n observations $\mathbf{x}_i \in \mathbb{R}^D$. Likewise the maximum variance formulation, it is defined the orthonormal matrix $\mathbf{M} \in \mathrm{R}^{D\times D}$, where $\mathbf{M}_j$ represents a column of the matrix as well as a basis orthonormal vector. To simplify the calculations and the notation, it is supposed that the original data has zero-mean.

In order to obtain the rotations between the original coordinates system and the new one, and using the properties of the matrix **M**, the inner product $\mathbf{C} \in \mathbb{R}^{n\times D}$ between the points in the dataset and **M** is defined by

$$\mathbf{C}_{ij} = \mathbf{x}_i^{\mathrm{T}}\mathbf{M}_j \ \forall i \ \ \forall j. \tag{3.16}$$

Then to reconstruct the original dataset

$$\mathbf{x}_i = \sum_{j=1}^{D} \mathbf{C}_{ij}\mathbf{M}_j \ \forall i. \tag{3.17}$$

This can be interpreted as a linear combination of the basis vector. Hence, using the Equation 3.16 to replace C in the Equation 3.17, it can be obtained that

$$\mathbf{x}_i = \sum_{j=1}^{D} (\mathbf{x}_i^{\mathrm{T}}\mathbf{M}_j)\mathbf{M}_j \ \forall i. \tag{3.18}$$

It is important to note that the Equations 3.16, 3.17 and 3.18 are using some matrix $\mathbf{C} \in \mathbb{R}^{n \times D}$, but for the process of dimensionality reduction the quantity of rotations that can be reconstructed have to be some $K < D$. Then a problem comes up, because the reconstruction of $\mathbf{x}_i$ takes only the first K rotations (vector) within the basis Matrix $\mathbf{M}$ (BISHOP, 2006, 563). For the others dimensions the value b_j is fixed. The $\tilde{\mathbf{x}}_i$ approximation of $\mathbf{x}_i$ is now defined by

$$\tilde{\mathbf{x}}_i = \sum_{j=1}^{K} a_{ij}\mathbf{M}_j + \sum_{j=M+1}^{D} b_j\mathbf{M}_j. \tag{3.19}$$

The values of a_{ij} and b_j can be chosen to minimize the function

$$\phi = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||^2. \tag{3.20}$$

This ϕ is the objective function to minimize, because it represents the error (Euclidean distance) between the projection and the original data. Thus the derivate of ϕ with respect to each a_{ij},

$$\nabla_{a_{ij}}\phi = \frac{2}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \sum_{j=1}^{K} a_{ij}\mathbf{M}_j + \sum_{j=M+1}^{D} b_j\mathbf{M}_j)\sum_{j=1}^{K}\mathbf{M}_j = 0$$

$$a_{ij} = \mathbf{x}_i^{\mathrm{T}}\mathbf{M}_j \;\; \forall j = (1, ..., K). \tag{3.21}$$

Additionally, the derivate of ϕ with respect to each b_i,

$$\nabla_{b_j}\phi = \frac{2}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \sum_{j=1}^{K} a_{ij}\mathbf{M}_j + \sum_{j=M+1}^{D} b_j\mathbf{M}_j)\sum_{j=M+1}^{D}\mathbf{M}_j = 0$$

$$b_j = \bar{\mathbf{x}}^{\mathrm{T}}\mathbf{M}_j \;\; \forall j = (K+1, ..., D). \tag{3.22}$$

Now using the values obtained for a_{ij}, b_j and the Equation 3.17 inside 3.20

$$\phi = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=K+1}^{D}(\mathbf{x}_i^{\mathrm{T}}\mathbf{M}_i - \bar{\mathbf{x}}\mathbf{M}_i)^2 = \sum_{j=K+1}^{D}\mathbf{M}_j^{\mathrm{T}}\mathbf{S}\mathbf{M}_j. \tag{3.23}$$

The next step is to calculate the minimization of the cost function ϕ. Considering the orthonormality of all the basis vectors of the matrix M and using the Lagrange multiplier to enforce the constraint of the $\mathbf{M}_j^{\mathrm{T}}\mathbf{M}_j = 1$ condition, the new cost function

$$\tilde{\phi} = \mathbf{M}^{\mathrm{T}}\mathbf{S}\mathbf{M} + \lambda(I - \mathbf{M}^{\mathrm{T}}\mathbf{M}). \tag{3.24}$$

Minimizing the $\tilde{\phi}$ function with respect to each M_j is equal to

$$\nabla_{\mathbf{M}_j}\tilde{\phi} = \mathbf{SM}_j - \lambda_j\mathbf{M}_j = 0 \quad \Longrightarrow \quad \mathbf{SM}_j = \lambda_j\mathbf{M}_j \ \forall j. \tag{3.25}$$

This can be solved as an eigenvector and eigenvalue problem, as well as the last formulation. The cost function ϕ is

$$\phi = \sum_{j=K+1}^{D} \mathbf{M}_j^{\mathrm{T}}\lambda_j\mathbf{M}_j = \sum_{j=K+1}^{D} \lambda_j \tag{3.26}$$

and finally it can be concluded that the best option to define the new subspace is choosing the K eigenvectors $\mathbf{M}_j$ that correspond with the largest K eigenvalues λ_i. This is as a result of the selection of the $D-K$ smallest eigenvalues that will minimize the function ϕ in the Equation 3.26 (from $j = K+1$ to D) (BISHOP, 2006, 565).

3.2.2 Weighted PCA

In this version, the first modification introduced is the absence of the assumption that the data is centralized; in other words, it does not have zero mean considering that we want to find some robust data mean. This modification is reflected in the original ϕ function by the addition of the weighted mean vector $\mathbf{u}_w$. The second variation is the transformation of the squared error (squared distance) between the original data and the projection to a robust function ρ. The function

$$\phi_w = \frac{1}{n}\sum_{i=1}^{n}\rho(||\mathbf{x}_i - \mathbf{u}_w - \tilde{\mathbf{x}}_i||) = \frac{1}{n}\sum_{i=1}^{n}\rho(||\mathbf{x}_i - \mathbf{u}_w - \mathbf{Mz}_i||) \tag{3.27}$$

is the new objective function to minimize (CHANG; YEUNG, 2006, p. 8), where $\mathbf{M}$ is the matrix containing all the eigenvectors and $\mathbf{z}_i$ represents the projected data.

The $\rho(e_j)$ function is expected to be robust; it is also called objective function (see Section 2.2.2 for details). The function is applied over the error (distance) e_j, between the $\mathbf{x}_j$ point and its projection. The Score function $\psi(e_j)$ is $\nabla_{e_j}\rho(e_j)$ and the Weight function $w(e_j)$ is equal to

$$a_j = w(e_j) = \frac{\psi(e)}{e} \tag{3.28}$$

See Table 2 for details of the possible functions to choose. The recommendation by Chang e Yeung (2006, p. 10) is to use the Huber function with

$$c = \frac{1}{2n}\sum_{j=1}^{n} e_j.$$

The calculations of the new weighted mean $\mathbf{u}_w$ and the weighted covariance matrix '$\mathbf{S}_w$ are

$$\mathbf{u}_w = \frac{\sum_{i=1}^{n} a_i \mathbf{x}_i}{\sum_{i=1}^{n} a_i} \tag{3.29}$$

and

$$\mathbf{S}_w = \frac{1}{n} \sum_{i=1}^{n} a_i (\mathbf{x}_i - u_w)(\mathbf{x}_i - u_w)^{\mathrm{T}}. \tag{3.30}$$

The weighted PCA algorithm uses the iterative procedure called iterative reweighted least squares (IRLS), detailed in the Section 2.2.3, because it is impossible to find a closed solution considering the mutual dependency between the a_j weights and the error $||\mathbf{x}_i - \mathbf{u}_w - \tilde{\mathbf{x}}_i||$ (due to $\mathbf{u}_w$ and $\mathbf{S}_w$). The iterative algorithm can be summarized in the following list of steps (CHANG; YEUNG, 2006, p. 9):

1. The standard PCA is executed to find the initial values of $\mathbf{M}^{t=0}$ and $\mathbf{u}_w^{t=0}$. Set $t = 1$.
2. Using the $\mathbf{M}^{t-1}$ and $\mathbf{u}_w^{t-1}$ calculate of the error (distance) value $e_j = ||\mathbf{x}_j - \mathbf{u}_w - \tilde{\mathbf{x}}_j|| \;\; \forall j$.
3. Calculate the weight value $a_j^t = w(e_j) \;\; \forall j$.
4. Using the set of weights a_j^t, execute an weighted PCA to obtain the news $\mathbf{M}^t$ and $\mathbf{u}_w^t$ values.
5. If $||\mathbf{M}^{t-1} - \mathbf{M}^t|| > \alpha_M$ or $||\mathbf{u}_w^t - \mathbf{u}_w^{t-1}|| > \alpha_w$, set $t = t + 1$ and start again at Step 2. Otherwise the algorithm ends.

3.2.3 RAPCA

The reflection-based algorithm for principal component analysis, also known as RAPCA, was formally defined by Hubert *et al.* (2002). It adopts a robust approach to do the dimensionality reduction of high-dimensional datasets. It is suggested to use RAPCA when the dimension D of the data is higher than the quantity n of elements inside the dataset. The RAPCA method was designed as a response to others robust PCA approaches (i.e. the Robust Covariance PCA), in which $n > D$ is needed. (HUBERT *et al.*, 2002, p. 102).

This robust procedure can be divided into two stages and uses a combination of three different methods to achieve its objective. One standard principal component analysis is executed in the first stage, meanwhile in the second stage the Croux and Ruiz-Gazen (CR) is implemented and improved with one algorithm called the R-Step (R of reflexion). The two stages in the RAPCA algorithm are detailed as follows:

1. In this first stage, a classical PCA is executed over the data. The reason for that execution, is because the two algorithms of the next stage become computationally heavy as the dimension of the data increases. Then it is important to obtain a secondary dataset with the r-dimensional affine subspace spanned by the elements of the original dataset.
   The new re-projected data (scores matrix) is denoted as $\widehat{\mathbf{Z}} \in \mathbb{R}^{n\times r}$ and the resultant matrix of eigenvectors as $\widehat{\mathbf{M}} \in \mathbb{R}^{D\times r}$. Then, without loss of information

   $$\mathbf{X} - \mathbf{1}\hat{\mathbf{u}}^{\mathrm{T}} = \widehat{\mathbf{Z}}\widehat{\mathbf{M}}, \tag{3.31}$$

   where $\mathbf{1} \in \mathbb{R}^D$ is the vector of ones, and $\hat{\mathbf{u}}$ is the classical mean vector of the data. It is relevant to note that the value of $r \leq n-1$ is the rank of the original centered data obtained by $\mathbf{X} - \mathbf{1}\mathbf{u}^{\mathrm{T}}$ (HUBERT *et al.*, 2002).
2. This second stage is the one that gives the robustness to the RAPCA algorithm. The main idea is to use the CR method. It is necessary to first calculate a robust mean (spatial median) $\mathbf{u}$ of $\widehat{\mathbf{Z}}$, then the new centered data

   $$\widetilde{\mathbf{Z}} = \widehat{\mathbf{Z}} - \mathbf{1}\mathbf{u}^{\mathrm{T}}. \tag{3.32}$$

   Likewise with the maximal variance formulation of the classical principal component analysis, the objective is to find the k eigenvectors that maximizes the variance observed in the data. However, in this case each eigenvector $\mathbf{M}_l$ is calculated using the next iterative process (HUBERT *et al.*, 2002, p. 110):
   a) The variance of the data is calculated with a robust process called $\mathscr{Q}_n$ estimator where

      $$\mathscr{Q}_n(\mathbf{z}_1, ..., \mathbf{z}_n) = 2.2219 \times c_n \times \{||\mathbf{z}_i - \mathbf{z}_j||\ \forall i < j\}_k, \tag{3.33}$$

      with k as the first quartile of the pairwise differences and c_n as a constant. The data $\widetilde{\mathbf{Z}}^{(l)} = [\mathbf{z}_1, ..., \mathbf{z}_n]^{\mathrm{T}}$ and $\widetilde{\mathbf{Z}}_i^{(l)} = \mathbf{z}_i$
   b) The data is transformed by means of reflection, $\mathbf{U}^{(l)}(\mathbf{M}_l) = (1, 0, ..., 0) \in \mathbb{R}^{D-l+1}$, then the data $\widehat{\mathbf{Z}}_i^{(l+1)} = \mathbf{U}^{(l)}(\widetilde{\mathbf{Z}}_i^{(l)})$
   c) The new data, transformed by the orthogonal complement of $\mathbf{U}^{(l)}(\mathbf{M}_l)$, is finally converted, omitting the first dimension of $\widehat{\mathbf{Z}}_i^{(l+1)}$. Obtaining the data $\widetilde{\mathbf{Z}}_i^{(l+1)}$

   To transform any eigenvector $\mathbf{M}_l$ into the $\mathbb{R}^{D-1+l}$ dimensional space, simply use the inverse of the reflection $\mathbf{U}^{(l-1)}$. Lastly using the Equation 3.32

   $$\widetilde{\mathbf{Z}}\mathbf{M}^{\mathrm{T}} = \mathbf{Z}, \tag{3.34}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times k}$ is the final projected data, and $k \leq r$ is the desired final dimension.

3.2.4 T2 and Q statistics for PCA

The T2 (score distance) and Q (orthogonal distance) statistic measures can be applied to the model obtained in the execution of any PCA type into the new less-dimensional data (HUBERT *et al.*, 2005, p. 6). Using these measures to calculate some cut-off values with some confidence parameter, it is possible to know how the model fits each point. In other words, taking some fixed probability the limit value where the points turns into outliers can be know.

The scored distance of one point i is defined as

$$T2_i = \sqrt{\sum_{j=1}^{k} \frac{\mathbf{Z}_{ij}^2}{\lambda_k}}, \tag{3.35}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times k}$ is the matrix within the projected data. It can be interpreted as the norm of the projected point normalized by its eigenvalue. To calculate the T2 cut-off of some PCA model using some probability ρ is

$$T2_c f = \sqrt{\chi^2_{k,\rho}}, \tag{3.36}$$

considering the assumption that the scores or projections $\mathbf{Z}$ are normally distributed, its square is Chi-Squared distributed (HUBERT *et al.*, 2005, p. 6). Moreover, the orthogonal distance Q_i, is the reconstruction error for one point i, then

$$Q_i = ||\mathbf{x}_i - \mathbf{u}_w - \tilde{\mathbf{x}}_i||. \tag{3.37}$$

To estimate the Q cut-off of some classical PCA model using the probability parameter θ, and considering the assumption that squares of the cube roots of the orthogonal distances are normally distributed. Then the Q cut-off is the normal inverse distribution and where the mean $\mu_Q = 1/n \sum (Q_i)^{2/3}$ and $\sigma_Q = \sqrt{1/n \sum (\mathbf{x}_i - \mu_Q)^2}$ (HUBERT *et al.*, 2005, p. 6). The correspondent Q cut-off of the RAPCA models is estimated using some μ and σ obtained in the execution of an univariate minimum covariance determinant (HUBERT; DEBRUYNE, 2010).

3.3 Robust Locally Linear Embedding

3.3.1 RLLE

This version of robust locally linear embedding was developed by Chang e Yeung (2006) in 2005 with the intention to reduce the influence of outliers into the LLE algorithm. The approach used by the authors is to execute a weighted principal component analysis (see 3.2.2 for details) into every first group of neighbors of each point to make a score of the points. Then that score is used to select another *better* group of new-neighbors for doing the construction of the weights or coefficients. The score is also adopted to make a weighted reconstruction of the data.

It is defined a set of n observations $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, where the data is sampled for some underlying manifold and each observation $\mathbf{x}_i \in \mathbb{R}^D$. The stages of the robust locally linear embedding and its details are described in the following:

1. As in the locally linear embedding, a set $\mathscr{N}_i$ of k neighbors is chosen for each point $\mathbf{x}_i$, the vector $\mathbf{n}_{ij}$ represent the j neighbor of the point i.

   For each set of neighbors, a weighted principal component analysis is executed independently. In other words, a weighted PCA is performed over each set $\mathscr{V}_i = \mathbf{n}_{i1}, ..., \mathbf{n}_{ik}\ \forall i$. The resultant vectors of weights from each weighted PCA are stored in the matrix $\mathbf{A}_{ij}$, where the row i represent the set coming from the point $\mathbf{x}_i$ and j column is the jth neighbor of the point.

   A normalization is executed in each row of $\mathbf{A}$, computed as

   $$\mathbf{A}_i^* = \frac{\mathbf{A}_i}{\sum_{j \in \mathscr{N}_i} \mathbf{A}_{ij}}. \tag{3.38}$$

   Lastly a reliability score $\mathbf{s}$ is calculated for every point, where $\mathbf{s}_m$ is the sum of the weights $\mathbf{A}_{ij}^*$ obtained by a point m that is the j neighbor of any i point. (CHANG; YEUNG, 2006, p. 10).

2. In the second stage, we have to 'separate' the database into two subsets. A threshold ε needs to be chosen and then the subset $\mathscr{X}^I = \{\mathbf{x}_i : \mathbf{s}_i > \varepsilon\}$.

   For the process of reconstruction, a small change is introduced on the classical LLE algorithm. The k nearest neighbors of each point $\mathbf{x}_i$, have to be chosen exclusively from the set $\mathscr{X}^I$. The construction of the weights is made minimizing the same cost function in the Equation 3.1, but using the new neighborhood selection.

To do the computation of the K-dimensional embedding for **X**, a new cost function is introduced.

$$\begin{aligned}\phi(Y) &= \sum_{i=1}^{n} \mathbf{s}_i||\mathbf{y}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{y}_j||^2 \\ &= \mathbf{S}((\mathbf{I}-\mathbf{W})\mathbf{Y})^{\mathrm{T}}((\mathbf{I}-\mathbf{W})\mathbf{Y}) \\ &= \mathbf{Y}^{\mathrm{T}}\mathbf{S}(\mathbf{I}-\mathbf{W})^{\mathrm{T}}(\mathbf{I}-\mathbf{W})\mathbf{Y} \\ &= \mathbf{Y}^{\mathrm{T}}\mathbf{SMY},\end{aligned} \tag{3.39}$$

where **S** is the diagonal matrix build with the values of **s**, or $\mathbf{S}_{lm} = \mathbf{s}_m\delta_{lm}$.

This can be solved in the same way as the eigenvectors and eigenvalues problem from the classic LLE, with the same constraint also. Thus the Equation 3.9 is transformed to

$$\mathbf{MY} = \frac{\lambda}{n}\mathbf{S}^{-1}\mathbf{Y} \tag{3.40}$$

3.3.2 RALLE

Presented in this work as an alternative to provide with robustness to the locally linear embedding. Using the main idea that is not necessary to use all the quantity of neighbors indicated as a parameter, if the confidence of them to be outliers is high enough. As in the work of Chang e Yeung (2006), the application of some algorithm to determine the probability of each neighbor to be an outlier is needed. Besides that, a score value is assigned for each point and is used to make a weighted projection.

A set of *n* observations $\mathscr{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ is defined, where the data is sampled for some underlying manifold and each observation $\mathbf{x}_i \in \mathbb{R}^D$. The stages of the robust locally linear embedding and its details are described in the following:

1. As in the locally linear embedding, a set $\mathscr{N}_i$ of *k* neighbors is chosen for each point $\mathbf{x}_i$, the vector $\mathbf{n}_{ij}$ represents the *j* neighbor of the point *i*. For each set of neighbors, RAPCA is executed. All the neighbors are measured with the T2 and Q methods, and also some cut off values are calculated with the parameters α^t and α^q. If some neighbor *j* is not inside the T2 and Q cut off values, it is discarded from that set of neighbors without being replaced with another new point. Just in the case that the number of *l* of resultant neighbors is lower than the value *k*, then the nearest $k-l$ rejected neighbors are included to the set $\mathscr{N}_i$.

The reconstruction is made with the resultant neighbors and minimizing the same cost function expressed in the Equation 3.1. A vector of scores is also made, in which the value $\mathbf{s}_i$ is equal to the quantity of times that the point i is used as a neighbor of the others $n-1$ points of the dataset.

2. For the computation of the K-dimensional embedding of $\mathbf{X}$, the cost function defined in the Equation 3.39 is maintained. The value $\mathbf{S}$ represents the diagonal matrix build with the values of $\mathbf{s}$, or equivalently $\mathbf{S}_{lm} = \mathbf{s}_m \delta_{lm}$ (when $s_m = 0$, then it is replaced with some small value). Therefore the solution can be found using the same procedure as the RLLE; that is solving the Equation

$$\mathbf{MY} = \frac{\lambda}{n}\mathbf{S}^{-1}\mathbf{Y} \tag{3.41}$$

as an eigenvectors and eigenvalues problem.

3.4 Trustworthiness and Continuity Measures

The Trustworthiness and Continuity (TC) are two related quality measures, in which the neighborhood of every point is analyzed, both in the original space and in the embedding space. The neighborhoods of each point, inside the original and projected datasets, are build choosing the k nearest elements (using some measure of distance). If the function

$$\mathrm{M} = 1 - \frac{2}{nk(2n-3k-1)}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}(r(i,j)-k) \tag{3.42}$$

is defined (VENNA; KASKI, 2005, p. 696), then in

- **Trustworthiness**: the set $\mathcal{N}_i$ represents the points that are in the neighborhood of some point i in the embedding space but not in the neighborhood in the original space. The $r(i,j)$ is a rank function; then it gives the distance order (within the original space) from the point i as origin and between some other point j taking into account all the other points in the dataset.
- **Continuity**: the set $\mathcal{N}_i$ represents the points that are in the neighborhood of some point i in the original space but not in the neighborhood in the embedding space. The $r(i,j)$ is a rank function; then it gives the distance order (within the embedding space) from the point i as origin and between some other point j taking into account all the other points in the dataset.

Part III

Experiments and Results

4 EXPERIMENTS ROBUST LINEAR REGRESSION

The main goal in this Chapter is to analyze the performance of the Classic Least Squares, the M-Estimator, the S-Estimator, the MM-estimator and the RANSAC Estimator when fitting a generalized linear regression model to a list of datasets. In order to accomplish the main goal, the theoretical information of the previous chapters is used in combination with the set of experiments executed in a controlled environment (Synthetic Dataset) as well as with a real problem dataset.

The Theil-Sen Estimator was excluded from the experiment because of its computational cost. The quantity of model estimations that are required is a dimensional-combination of the number of elements inside the dataset. Additionally to that, the calculation of the spatial median of all the parameters of the models is required. The Theil-Sen is non-viable over the configurations of the datasets used in the experiments.

4.1 Methodology of the experiments

Each experiment consists in the estimation of the parameters of the model using a training dataset. After the estimation, a test dataset is used to evaluate the grade of generalization reached in the estimation (mean squared error). It is possible to identify three common stages that are present in every single experiment made:

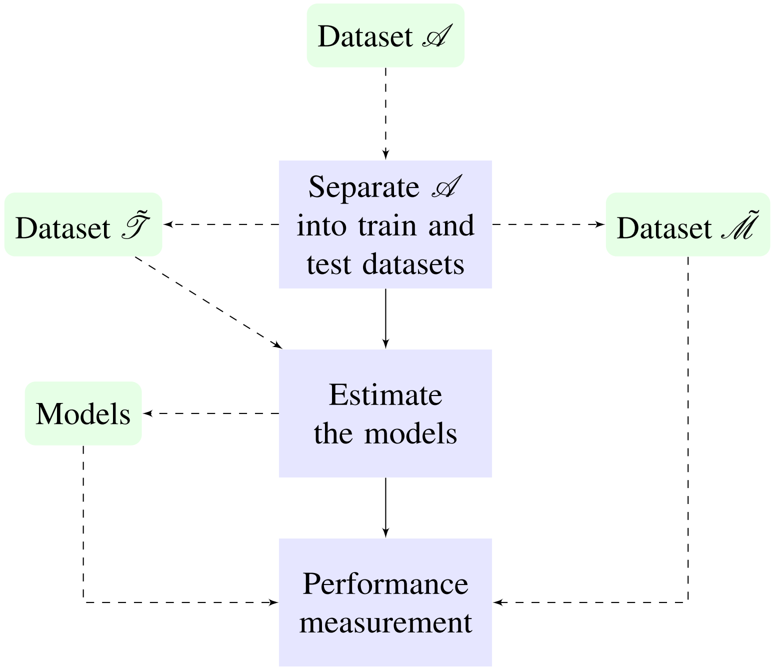

Dataset $\mathscr{A}$
Separate $\mathscr{A}$ into train and test datasets
Dataset $\tilde{\mathscr{T}}$
Dataset $\tilde{\mathscr{M}}$
Estimate the models
Models
Performance measurement

Figure 1 – Process of a single linear regression experiment

1. The main dataset is separated into two subsets. Let A be the original dataset, such $\mathscr{T}$ and $\mathscr{M}$ are two subsets, where $\mathscr{T} \cup \mathscr{M} = \mathscr{A}$ and $\mathscr{T} \cap \mathscr{M} = \varnothing$.
2. The estimators are executed to get the parameters of the models, using the new dataset $\tilde{\mathscr{T}}$.
3. The $\tilde{\mathscr{M}}$ subset is used to assess the performance of the estimates obtained in the previous step.

The configuration of the experiments varies between the synthetic data and the real data. Knowing that it is possible to define some extra conditions over the synthetic datasets, their methodology is going to be explained first.

4.1.1 Synthetic datasets

There are two types of datasets generated, the linear datasets are created using some linear model and $\mathbb{R}^5$ is its dimensional space. The non-linear dataset is in $\mathbb{R}^3$; the explanatory variables are randomly generated with uniform distribution and with $t_i = \sin(2\pi \mathbf{x}_{i1})\sin(2\pi \mathbf{x}_{i2})$.

The cardinality of the subset $\tilde{\mathscr{T}}$ is equal to the 67% of the quantity of elements in $\mathscr{A}$. The remaining 33% is assigned for measuring the performance into the subset $\tilde{\mathscr{M}}$. The set of outliers percentages $\mathscr{P}$ =$\{$0%, 4%, 8%, 12%, 16%, 20%, 24%, 28%, 32%, 38%, 42%, 46%, 50%, 64%$\}$. Each element in $\mathscr{P}$ represents the percentage of response variables in the dataset $\mathscr{T}$ that are spoiled with outliers. The remaining data in $\mathscr{T}$ is contaminated with white noise. The contamination vectors can be created with diverse values of standard deviation defined inside some set $\mathscr{S}$ and described in the the Section 4.2.

Table 4 – Types of outliers: defines the percentage of outliers that was taken from the *min* and *max* components

	Type I	Type II	Type III	Type IV	Type V	Type VI	Type VII
min	0%	20%	40%	50%	60%	80%	100%
max	100%	80%	60%	50%	40%	20%	0%

The synthetic datasets contain three components of outliers (See Section 4.2 for details), the *min* outliers, the *max* outliers and the extreme outlier. The set of pairs $\mathscr{O}$ =$\{$(0%,100%), (20%,80%), (40%,60%), (50%,50%), (60%,40%), (80%,20%), (100%,0%)$\}$ define the type; each element in $\mathscr{O}$ represents which percentages, from the 100% of the outliers, belongs to the *min* group and which to the *max* group respectively. This also defines the nomenclature for the *type of outliers*, the Table 4 explains which was the percentage taken from each component to make the outliers. The third component just indicates the presence or absence of one extreme point in the first element of the response variable. Choosing one configuration can determine the topology of the outliers.

Two more configurations are possible when Gaussian basis functions are used. These are the quantity of centroids and its standard deviation. The centroids quantity of the synthetic dataset is defined in the set $\mathscr{C} = \{1, 3, 7, 15, 31\}$, and the elements inside the set $\mathscr{D} = \{1, 0.5, 0.1\}$ define the standard deviations.

All the combinations between the elements in $\mathscr{S}$, $\mathscr{P}$, $\mathscr{O}$ and the presence or absence of the extreme outlier determine all the experiments that can be executed over one single dataset $\mathscr{A}$. Additionally, the combinations of the elements in $\mathscr{C}$ and $\mathscr{D}$ can also be joined to the set of feasible experiments when the datasets use Gaussian basis functions. Lastly, one single configuration is executed 10 times using different stochastic sets of $\tilde{\mathscr{T}}$ and $\tilde{\mathscr{M}}$.

4.1.2 Real dataset

The Real dataset presents a small quantity of combinations of possible configurations if you compare with the synthetic data. For measuring the performance, the quantity of elements in $\tilde{\mathscr{M}}$ is 20% of the cardinality of the set $\mathscr{A}$.

The real dataset uses the Gaussian basis function to transform its original data. Then the centroids quantity are defined in the set $\mathscr{C} = \{1, 3, 7, 15\}$ and the elements inside the set $\mathscr{D} = \{1, 0.5, 0.1, 0.05, 0.01\}$ define the standard deviations.

All the combinations between $\mathscr{C}$ and $\mathscr{D}$ determine all the possible configurations in this dataset. Each configuration is executed 10 times using different stochastic sets of $\tilde{\mathscr{T}}$ and $\tilde{\mathscr{M}}$.

4.1.3 Configuration

Most of the estimators chosen in this work need to specify parameters to their execution. Only the classic least squares is a non-parametric algorithm. This Section stands for the specification of all the parameters used. However it is worth mentioning that all the values chosen are the default values recommended in their common implementations (normally to achieve some feature such as high BDP).

Table 5 – Parameters used in the execution of the estimator

M Estimator	**S Estimator**
ρ **function** = tukey bisquare. a **value** = $4,685$.	ρ **function** = tukey bisquare. **Breakdown point** $= 50\%$
MM Estimator	**RANSAC**
50% **BDP estimator:** S Estimator **M Estimator:** ρ **function** = tukey bisquare. **Asymptotic efficiency** $= 95\%$.	**Using 2nd approach of 2.5.1:** δ_p =0.99 $1-p = 1e-3$

4.2 Synthetic Datasets

The generation of artificial datasets gives the opportunity to specify the shape and quantity of the outliers inside the dataset. The major advantage of the controlled environments is the possibility to accurately compare the behavior of the estimators accordingly to the modifications on the shape or on the percentage of the outliers within the data. In this work, the presence of outliers in the synthetic datasets is designed to be exclusively inside the response variables.

4.2.1 Generation

The quantity of points is defined to be 1500, in other words $|\mathscr{A}| = 1500$. For the linear regression problem where the basis function $\phi(\mathbf{x}) = \mathbf{x}$, it is generated a dataset $\mathscr{A} = \{\mathbf{X}, \mathbf{t}\}$ using some linear model $\boldsymbol{\beta}$ with 4 parameters and without noise. The matrix $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_{1500})^{\mathrm{T}}$, and each observation $\mathbf{x}_i \in \mathbb{R}^4$ is associated with a target value $\mathbf{t}_i = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i \in \mathbb{R}$. If the datasets use Gaussian basis function, then $\mathbf{x}_i \in \mathbb{R}^2$ is generated using the continuous uniform distribution between 0 and 1 and $\mathbf{t}_i = \sin(2\pi\mathbf{x}_{i1})\sin(2\pi\mathbf{x}_{i2})$.

The unidimensional data vector $\mathbf{g} \in \mathbb{R}^n$ of Gaussian data (with $\mu_{\mathbf{g}} = 0$ and $\sigma_{\mathbf{g}} = \mathscr{S}$) is created for generating the white noise. Additionally, two one-dimensional Gaussian data vectors called $\mathbf{o}_{max} \in \mathbb{R}^n$ and $\mathbf{o}_{min} \in \mathbb{R}^n$ are built for generating the outliers. The two vectors are created using different sets of μ and σ (i.e μ_{max} and σ_{max} for $\mathbf{o}_{max}$) and it will depend on the features needed for the experiments (see Section 1.2.2 for details).

The error vector $\mathbf{e}$ is built with a stochastic combination of u values from the vector $\mathbf{g}$ and v values from the vector $\mathbf{o}$, where $\mathbf{e} \in \mathbb{R}^{u+v}$ and $u + v = |\mathscr{T}|$. The subset $\mathscr{T}$ is modified with the intention to contaminate it with outliers. These outliers will be placed in the output vector and not as leverage points (see Section 1.2). Taking $\hat{\mathbf{t}}$ as the new output vector, the $\tilde{\mathbf{t}} = \hat{\mathbf{t}} + \mathbf{e}$ and the new subset $\tilde{\mathscr{T}} = \{\hat{\mathbf{X}}, \tilde{\mathbf{t}}\}$.

Table 6 – Parameters of the Noise/Outliers creation

Dataset	$\sigma_{\mathbf{g}}$	σ_{min}	μ_{min}	σ_{max}	μ_{max}	Extreme Outlier
1	{1e-2,1e-1}	$0.5\sigma_{\mathbf{g}}$	$-15\sigma_{\mathbf{g}}$	$0.5\sigma_{\mathbf{g}}$	$15\sigma_{\mathbf{g}}$	
2						$300\sigma_{\mathbf{g}}$
3			$-4\sigma_{\mathbf{g}}$		$4\sigma_{\mathbf{g}}$	
4						$300\sigma_{\mathbf{g}}$
5				$0.4\sigma_{\mathbf{g}}$		
6						$300\sigma_{\mathbf{g}}$
7			$\text{Min(g)} + \sigma_{\mathbf{g}}$		$\text{Max(g)} - \sigma_{\mathbf{g}}$	$300\sigma_{\mathbf{g}}$
8						
9				$0.5\sigma_{\mathbf{g}}$		$300\sigma_{\mathbf{g}}$
10						
11		$\sigma_{\mathbf{g}}$		$\sigma_{\mathbf{g}}$		$300\sigma_{\mathbf{g}}$
12						

The $\mathscr{M}$ subset is also modified, but in this case the Gaussian noise from the **g** vector is only used. A stochastic subset of **g** with $|\mathscr{M}|$ size for contamination purposes was taken; the subset is stored in the vector $\mathbf{r} \in \mathbb{R}^{|\mathscr{M}|}$. Then, if we define $\bar{\mathbf{X}}$ as the input matrix of $\mathscr{M}$ and the output vector as $\check{\mathbf{t}}$, therefore $\mathbf{t}' = \check{\mathbf{t}} + \mathbf{r}$ and the new subset $\tilde{\mathscr{M}} = \{\bar{\mathbf{X}}, \mathbf{t}'\}$.

It is important to note that the cardinality of the subsets $\mathscr{T}$ and $\mathscr{M}$ and the u and v values are chosen in each experiment. The set of experiments includes the generation of different datasets using distinct models of outliers for the vector **o** (see Section 1.2.1).

4.2.2 Results

Four synthetics were selected after the execution of the planned experiments. This is because, taking into account the list of graphics and information generated and analyzed, four datasets can represent the most important findings and results. The outliers scheme chosen are the 2, 3 and 10 for the linear datasets and the 4 for the non-linear (it is called 4B); the election was based on the differences obtained by variations on the parameters inside each generation.

The asymptotic efficiency-breakdown point trade-off is clear over all the experiments; the RANSAC and LS algorithms have high asymptotic efficiency but low breakdown point; the S-Estimator has the highest BDP but the lowest asymptotic efficiency; lastly, the MM-Estimator stays between the S and the M-Estimator in most of the executions.

4.2.2.1 Dataset 2

The dataset 2 contains the extreme outlier; it is one of the most important features in this dataset. As shown in Figure 2 the 3, the mean squared error of the least squares algorithm was higher, in the majority of the cases, than the other algorithms; from type I to type IV all the estimators maintain a better performance than LS, at least until their breakdown point.

The extreme outlier has a considerable influence inside the LS estimator; for each algorithm within the Figure 4, the symmetry (between types) of the MSE values is clear. There is another symmetry in the Figure 5. This applies for all the algorithms except for the least squares; it is easy to see how the values of the right side (from Type V to VII) of the LS graphics are lower than the left side.

There is another curious influence of the extreme outlier. It was previously mentioned that the extreme point is not a leverage point; then its repercussion can be exclusively observable over the least squares. The 0% graph inside the Figure 5 shows that LS does not need a significant number of outliers to break. However, the performance of the LS rose from the outlier type V to the outlier type VII. The explanation of that is the ‘positive’ effect that the extreme outlier can bring when located in the other side of the greater part of outliers. The percentages of *min* outliers are higher than the percentages of the *max* outliers from the types V to the VII. Thus the extreme outlier counteracts the influence over the error function that the *min* outliers have (only for the OLS).

The RANSAC algorithm is tuned for being similar to the LS in therms of asymptotic efficiency; but it can handle small quantities of outliers, even the extreme one. It is for that reason that in the type V, VI and VII of outliers its MSE is slightly higher than LS when the percentage of outliers are incremented. In other words, the capacity of the RANSAC to cope with the extreme outlier also excludes the positive influence that the extreme outlier does to the LS when the presence of *min* outliers is significant (higher percentages and types).

The S-Estimator seems to be the best algorithm when the percentage of outliers grows (and until its BDP); on the other hand, the performance of the S-Estimator is poor when the percentage of outliers is low or its distribution is similar to the Gaussian. This statement can be easily understood analyzing the graphs with 0% and 4% of outliers in the Figure 5.

The μ_{min} and μ_{max} are 15 times the standard deviation used to generate the white noise. When the proportion of *min-max* outliers is the same (type IV) and the outlier percentages are lower than the BDP, the noise effect is the same as the Gaussian.

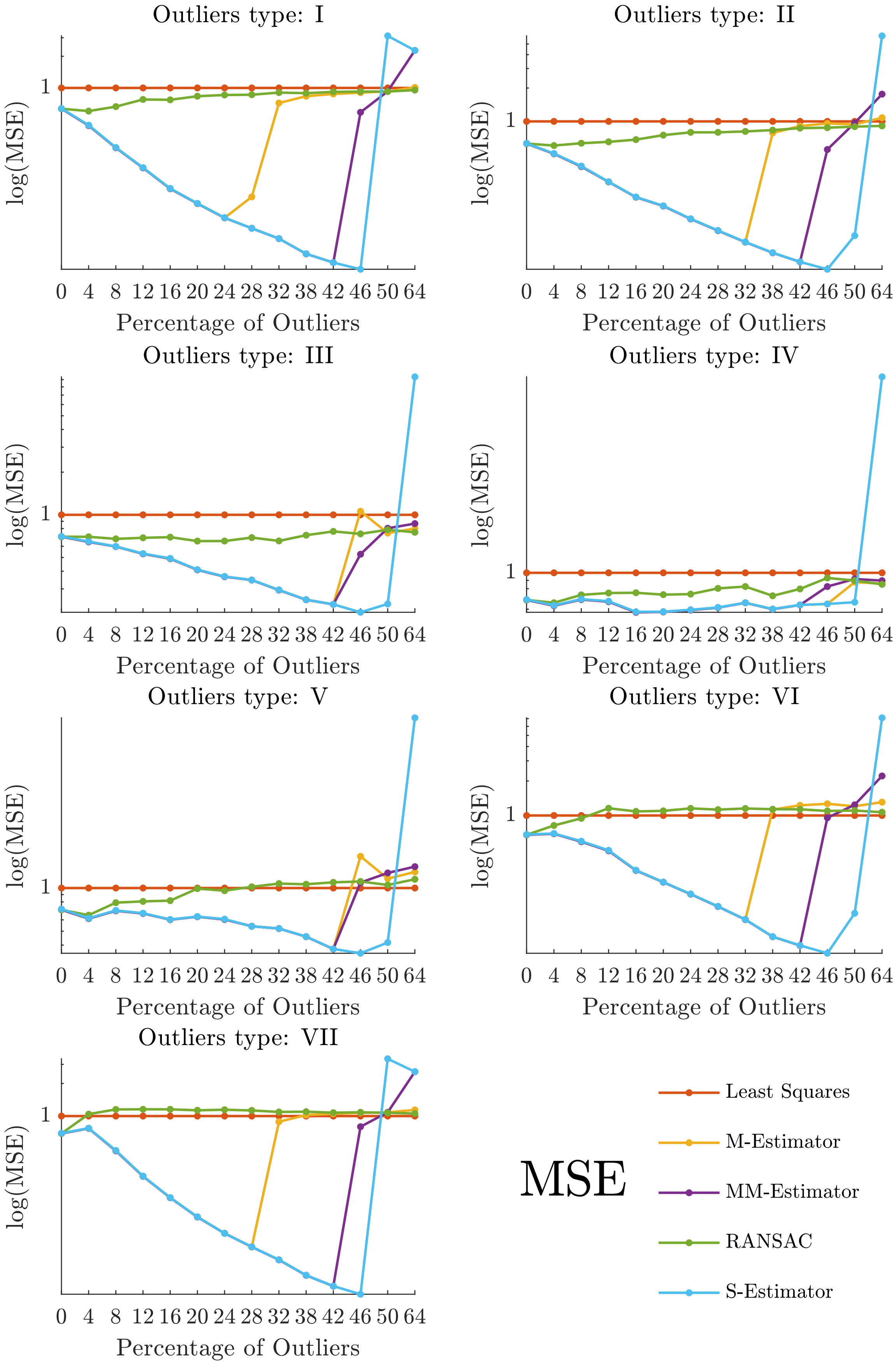


Figure 2 – Performance of the algorithms by type of Outliers over Dataset 2; each graphic shows the MSE in semilogarithmic scale (Normalized by the MSE of LS) of the estimations when varying the percentage of outliers.

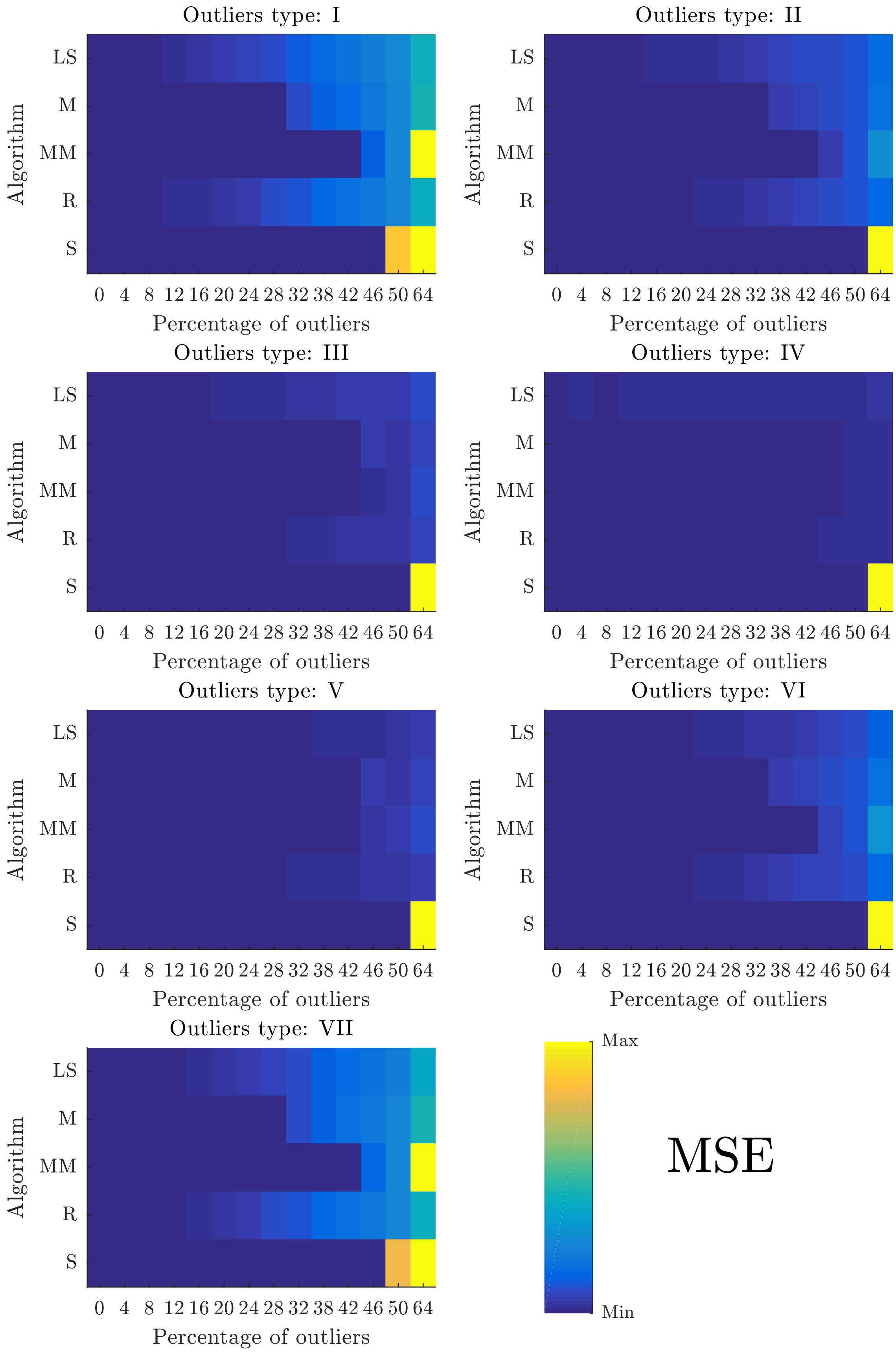


Figure 3 – Performance of the algorithms by type of Outliers over Dataset 2; aach graphic contains the MSE of the estimations by each percentage of outliers.

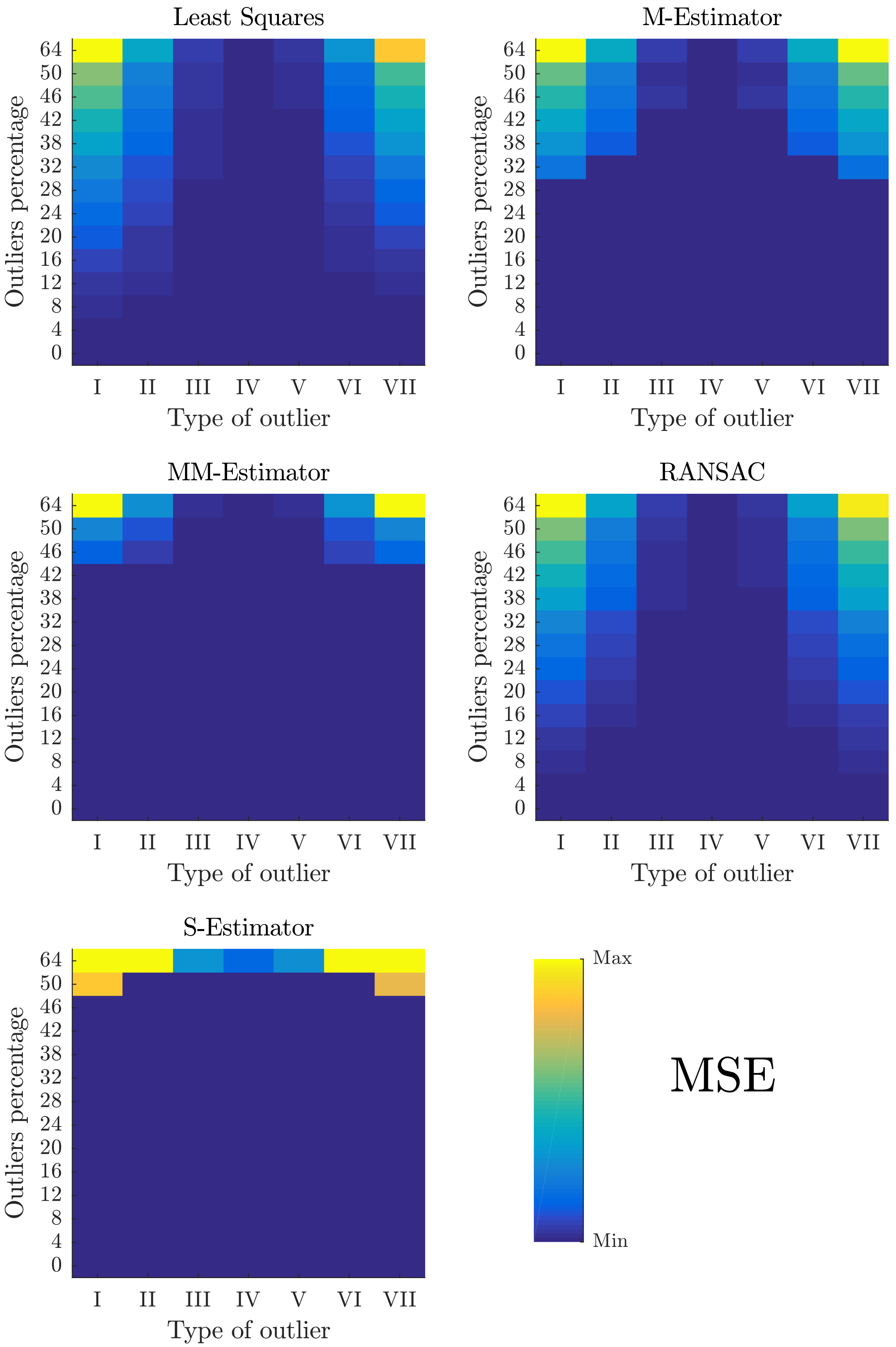


Figure 4 – Performance of each algorithm over Dataset 2; each graphic contains the MSE of one algorithm when varying type and percentage of outliers.

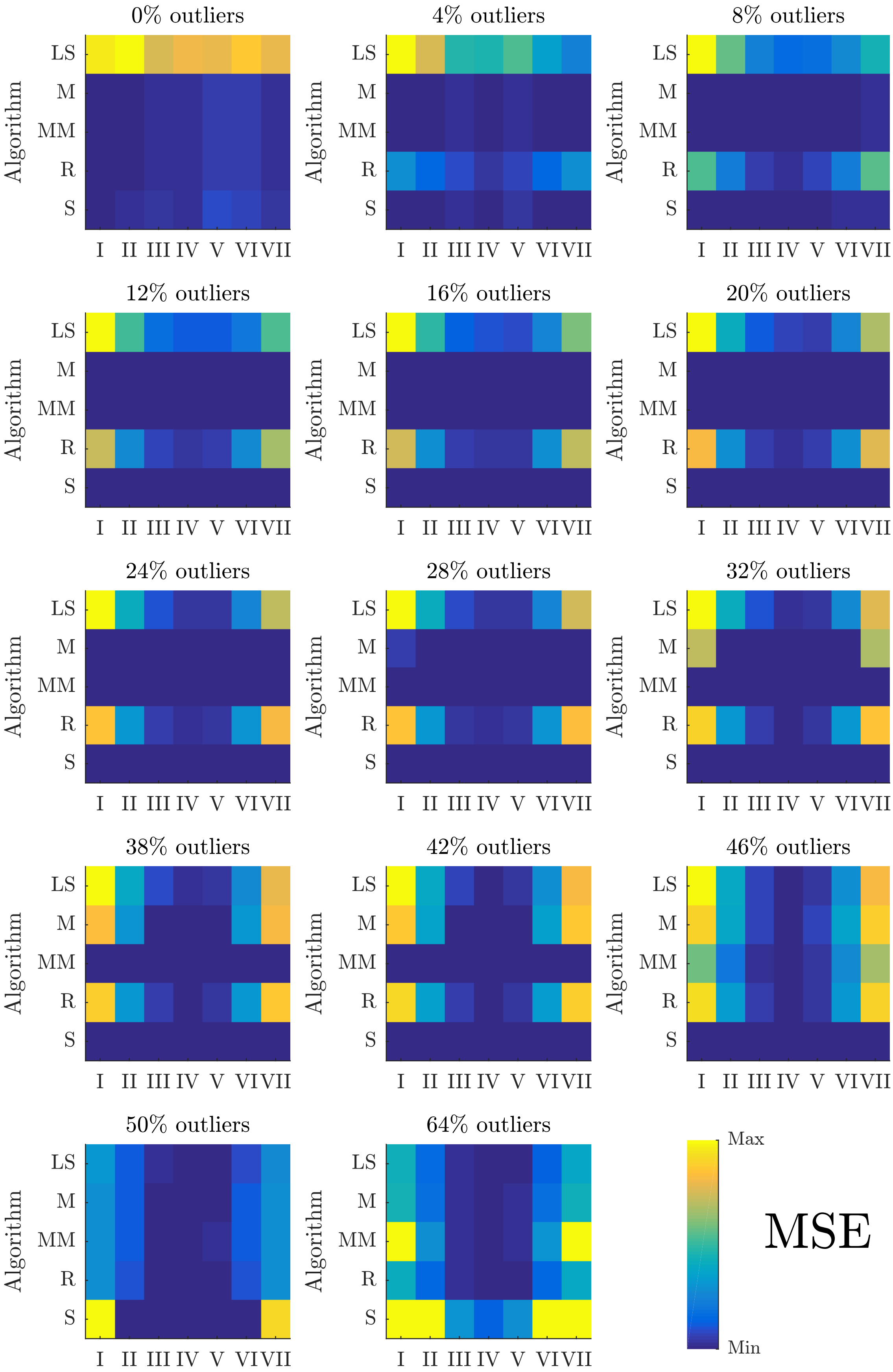


Figure 5 – Performance of the algorithms by percentage of Outliers over Dataset 2; each one of them contains the MSE of the estimations made by all the algorithms over each type of outliers.

4.2.2.2 Dataset 3

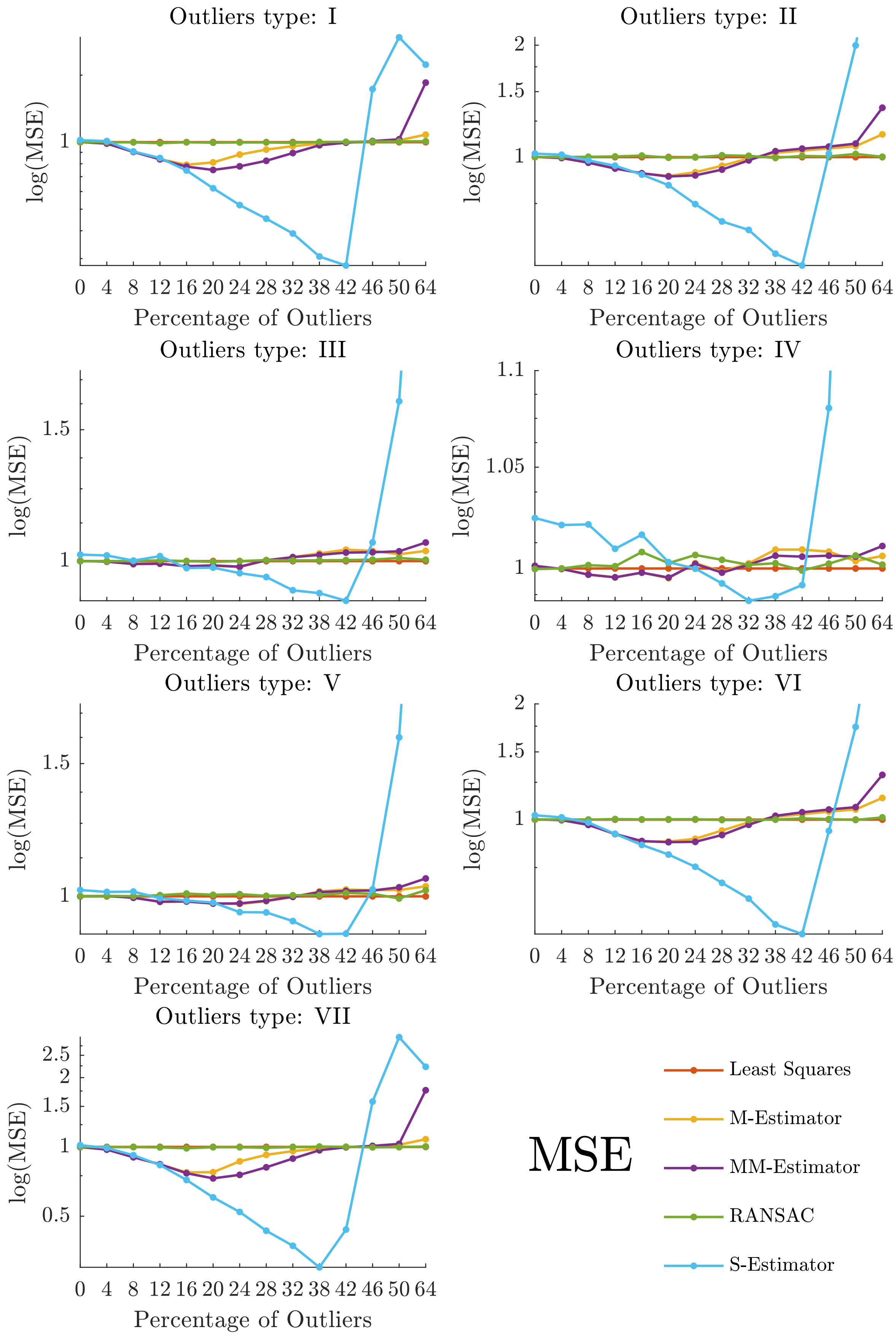
The dataset 3 have two main features: The absence of the extreme outlier and that the mean of the *min* and *max* outliers is nearer of the mean of the white noise than the Dataset 2 (Table 6 for details). Because of that and the standard deviation used in their generation, the tails of their probability density density functions overlap.

The first impression obtained by observing all the graphics was the symmetry related to the type of the outliers. The Type I corresponds with the Type VII; the Type II corresponds with the Type VI; finally, the Type III corresponds with the Type V. This is a good criterion for comparing the stability of the estimators. However, the S-Estimator demonstrates some short breaks in this symmetry aspect. The Figure 9 shows how the S-estimator does not maintain the symmetry in a good way; the same Figure also can be used to see other small asymmetries.

Inside the Figure 9, although the values are too close to each other in their scale, a little dissociation of the MSE value between the Types of outliers from the 4% graph to the 16% graph can be perceived. The explanation for that is found in the process of generation of outliers; the mean of the *min* is -0.3998 and the mean of the *max* is 3987. That makes the identification of the *min* outliers easier than the *max* outliers.

There is a particularity with the BDP of the S-Estimator. It looks like the S-Estimator cannot achieve the same BDP achieved over the dataset 2. The cause of that is the effect that the outliers and white noise overlapping can have and its general instability (with the possibility of local minimums). Besides that, the S-Estimator MSE is at least $47,5\%$ higher than any MSE from the other algorithms when the percentage of outliers reaches 50% (47.5% is in type IV). However, the S-Estimator looks like the best estimator for any type of outlier from 12% to 42% of outliers (see Figure 7).

The Figure 7 shows that the behaviors of the RANSAC and the least squares are almost the same in every context. Likewise, the M-Estimator and the MM-Estimator have very a similar performance; they follow the same pattern but the performance of the MM-Estimator remains between M-Estimator and S-Estimator performances.


Outliers type: I

Outliers type: II

Outliers type: III

Outliers type: IV

Outliers type: V

Outliers type: VI

Outliers type: VII

Figure 6 – Performance of the algorithms by type of Outliers over Dataset 3; each graphic shows the MSE in semilogarithmic scale (Normalized by the MSE of LS) of the estimations when varying the percentage of outliers.

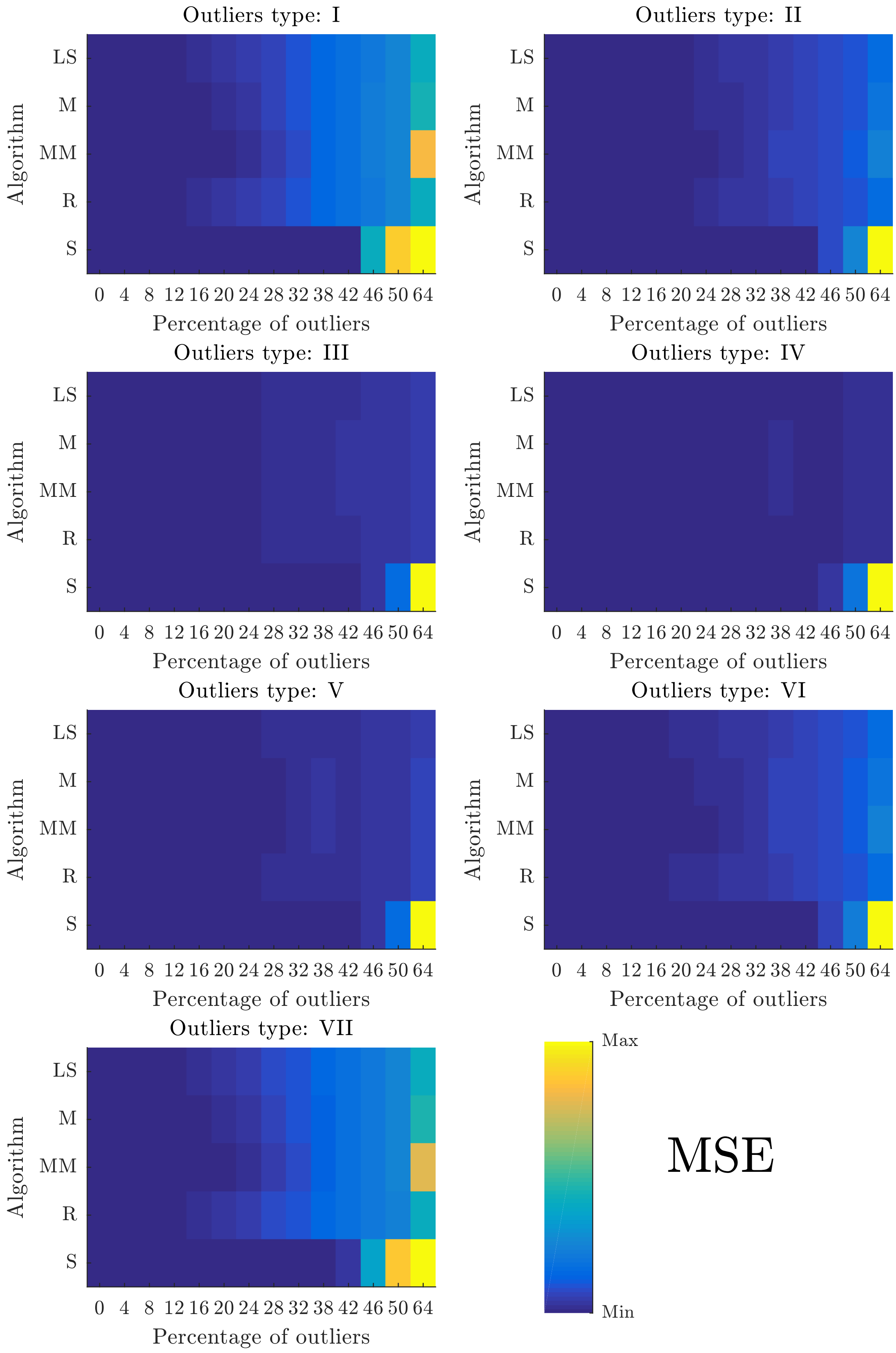
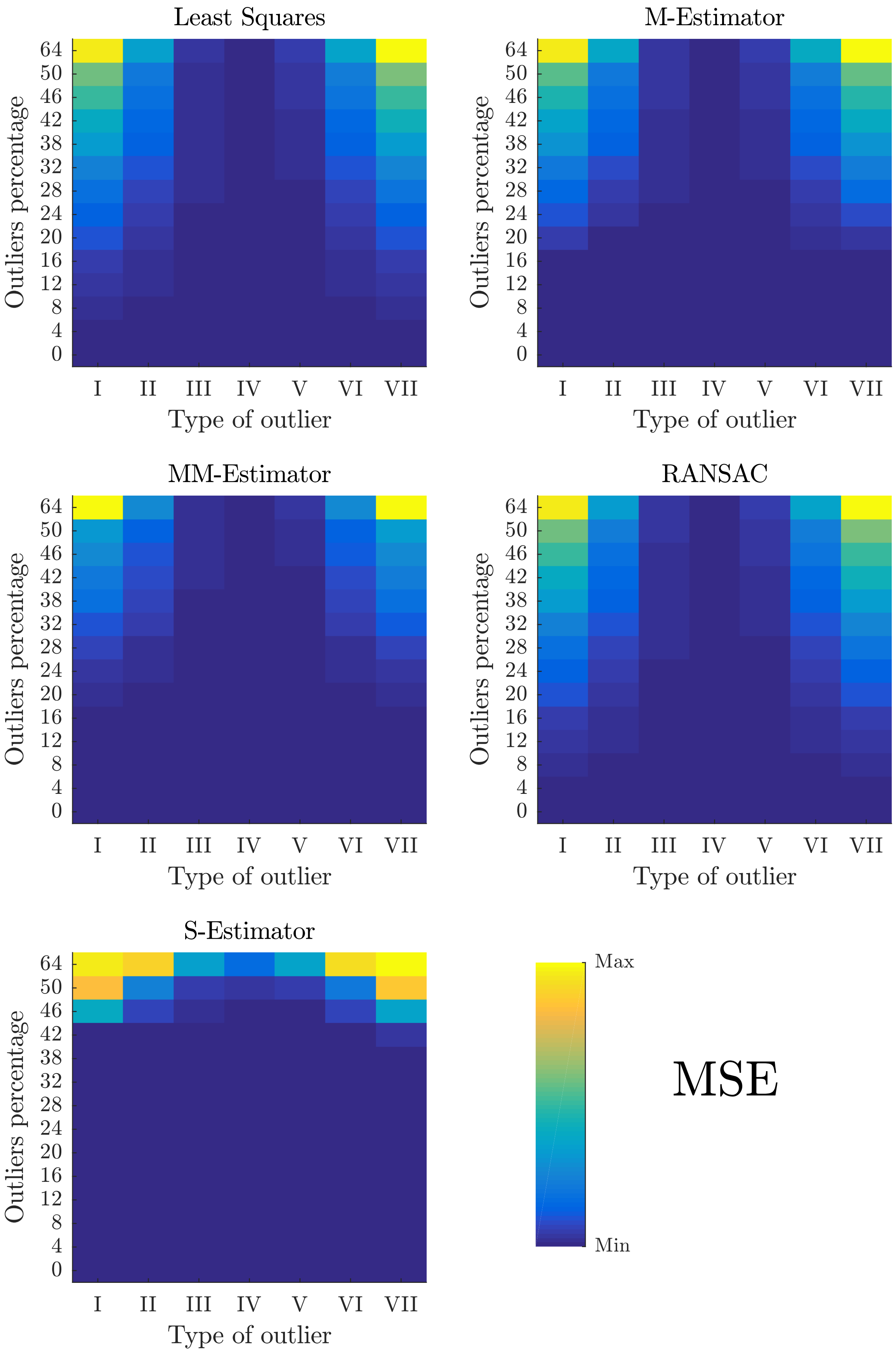


Figure 7 – Performance of the algorithms by type of Outliers over Dataset 3; each graphic contains the MSE of the estimations by each percentage of outliers.



Least Squares

M-Estimator

MM-Estimator

RANSAC

S-Estimator

Figure 8 – Performance of each algorithm over Dataset 3; each graphic contains the MSE of one algorithm when varying type and percentage of outliers.

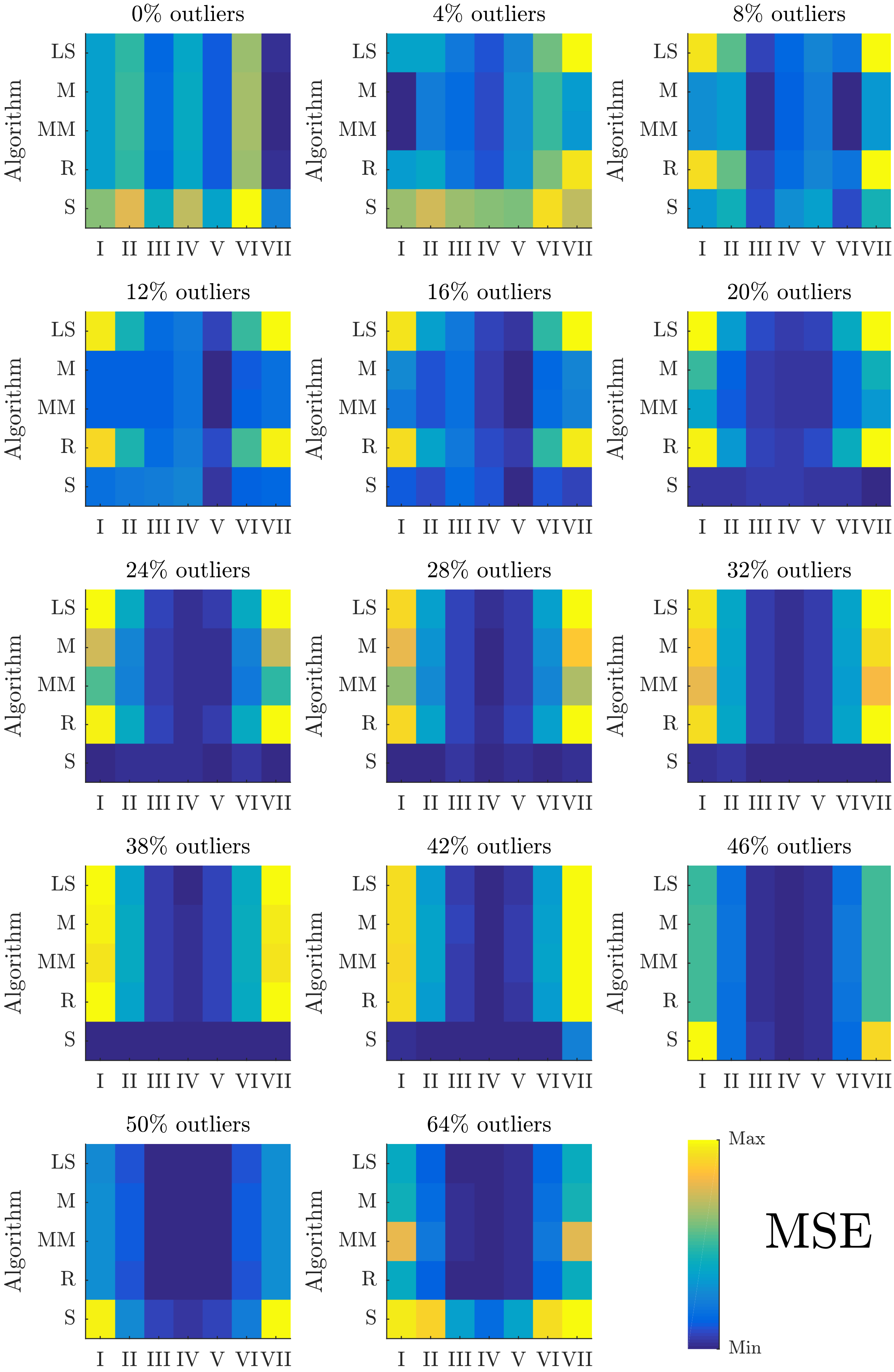


Figure 9 – Performance of the algorithms by percentage of Outliers over Dataset 3; each one of them contains the MSE of the estimations made by all the algorithms over each type of outliers.

4.2.2.3 Dataset 10

The dataset 10 differs from the previous two analyzed datasets because the means of its *max* and *min* outliers data are distinct. The mean of the *max* is 0.2391 and the mean of the *min* is -0.2853. Nevertheless, the standard deviation used for the generation process is the same for the two components.

The parameters used for the generation of the dataset were in part chosen to evaluate the influence that the deviations can made if they are placed in the same region as the normal errors; the overlapping area between the probability density density functions of the white noise, the *min* and the *max* outliers is greater than in the dataset 3. Because of the location where the outliers are placed, it is harder to recognize when a point is spurious or not.

Inside the Figures 10, 11 and 13, similar patterns are found but not a symmetry. Besides the least squares, it seems that the estimators detect more accurately the outliers when they are coming from the *max* outliers contamination (Types V, VI and VII). This is due to the mean difference between the *min* and the *max*.

The review of the S-Estimator performances in all the experiments of the dataset 10 leads to similar conclusions exposed from the other datasets. The S-Estimator demonstrate problems with white noise data or similar. The S-Estimator has the worst performance with 0% and 4% of outliers within the data. Besides that, when the percentage of outliers reaches 50%, the MSE of the S-Estimator is at least 39.9% higher than any other MSE.

Two patterns that were present inside the experiments of the dataset 2 and 3 are confirmed. The first pattern is that the least squares shows almost the same performance as the RANSAC estimator; the main reason is the parameters selection for the RANSAC execution and the method used to calculate the number of iterations (according to Tordoff e Murray (2005, p. 6), it is considerably overoptimistic). The second pattern is that the performance of MM-Estimator lies between the M-Estimator and S-Estimator; anyhow, this pattern is the expected behavior of the estimator. The MM is always closer to the M-Estimator.

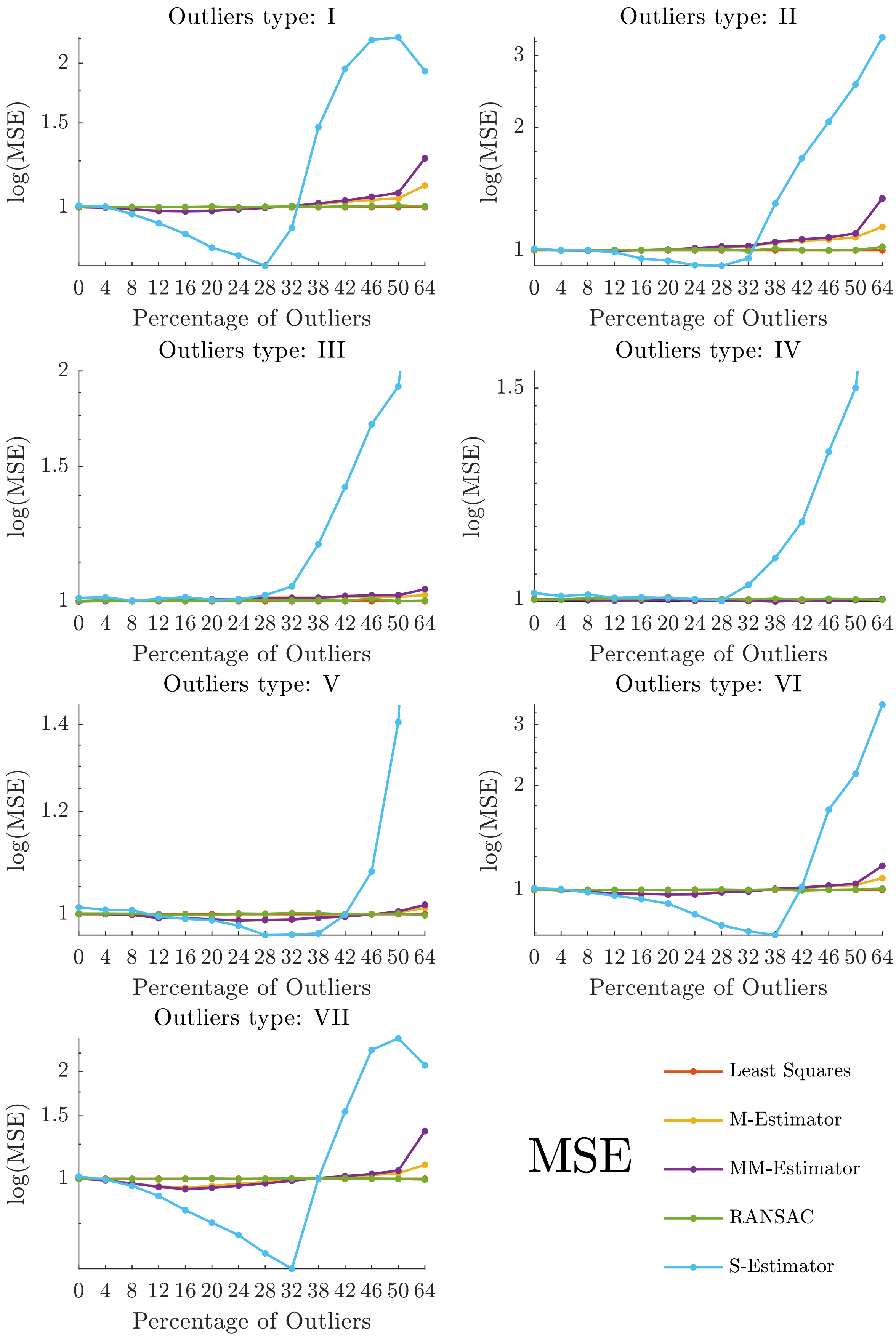


Figure 10 – Performance of the algorithms by type of Outliers over Dataset 10; each graphic shows the MSE in semilogarithmic scale (Normalized by the MSE of LS) of the estimations when varying the percentage of outliers.

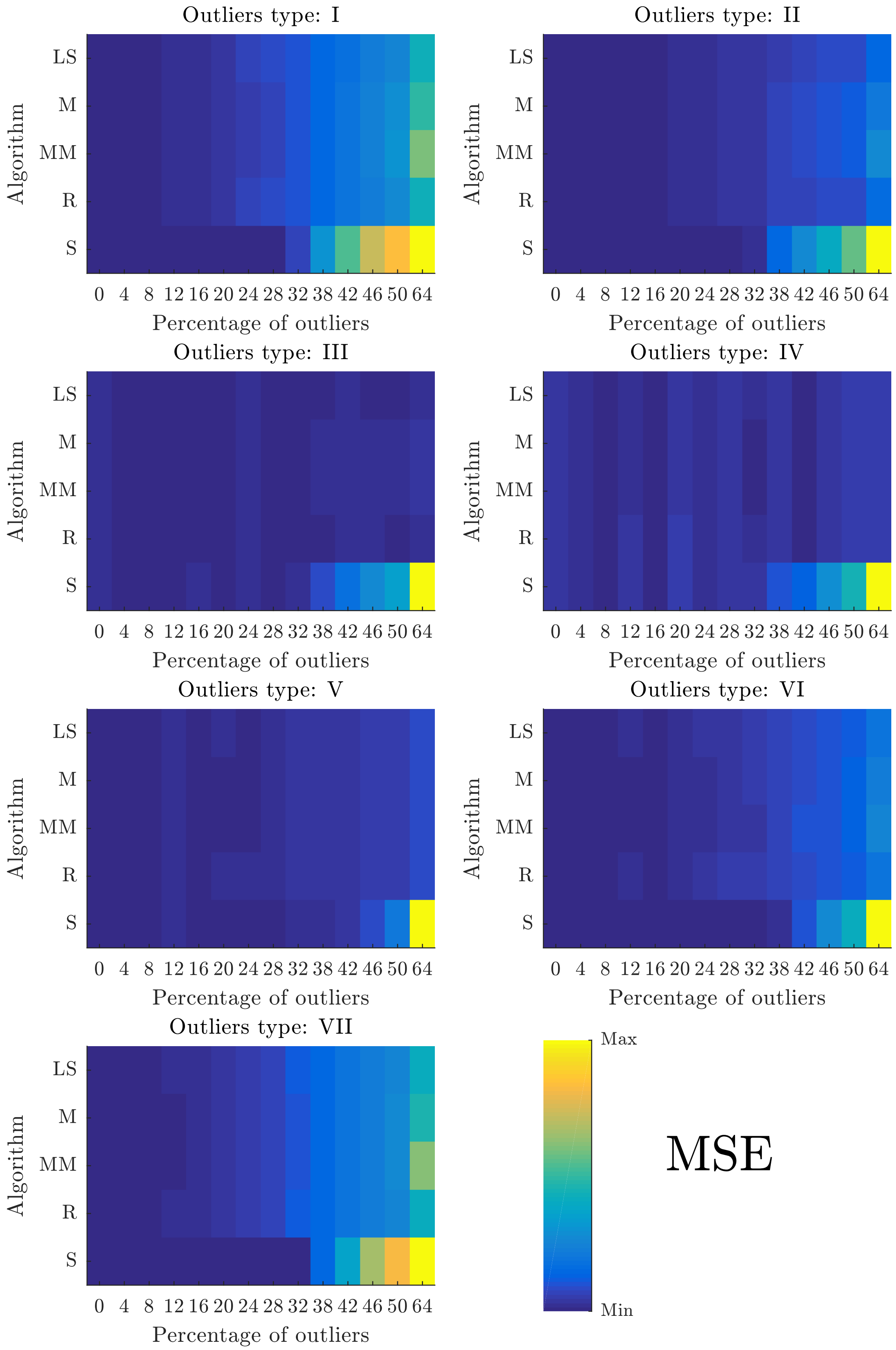
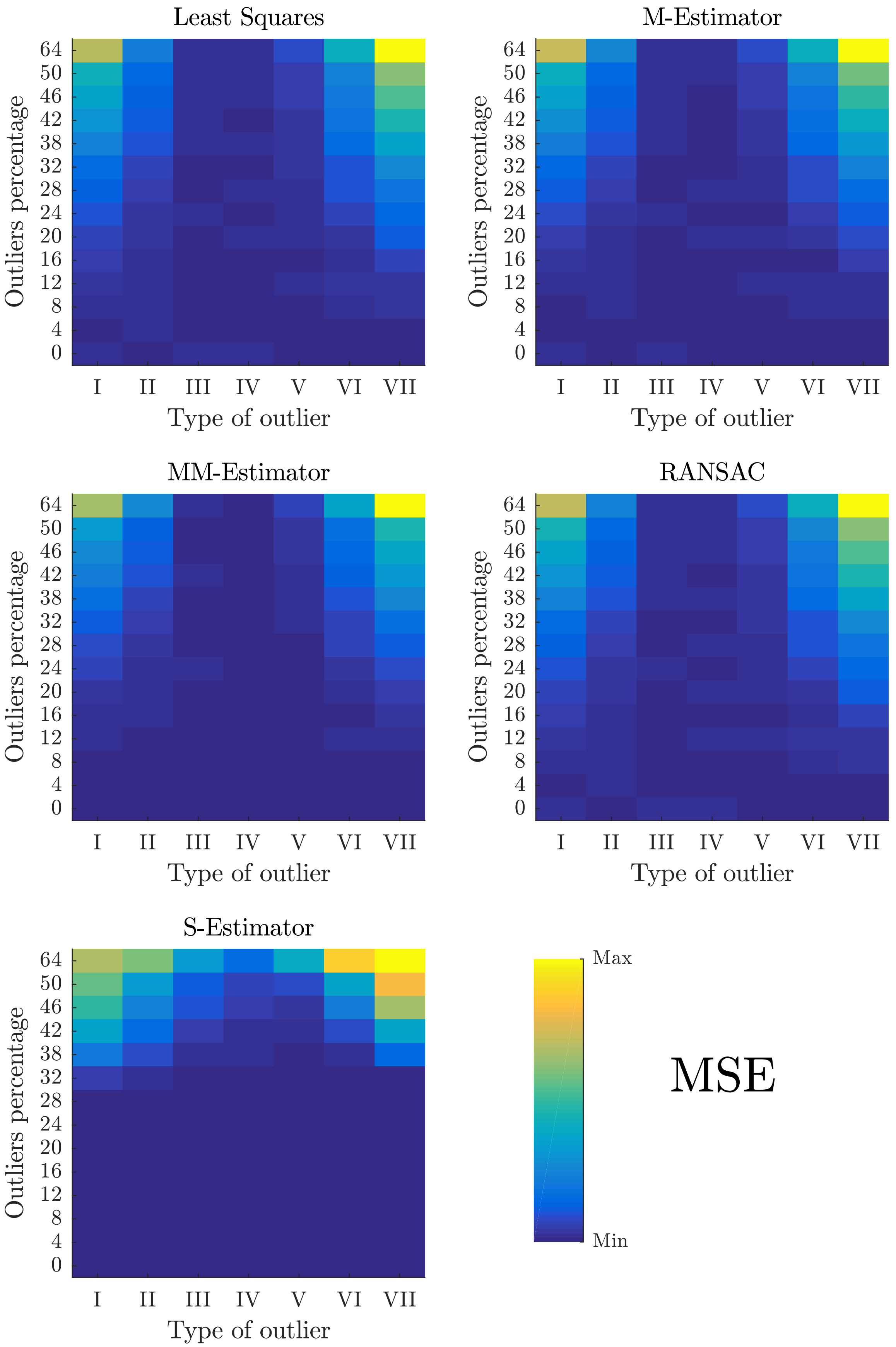


Figure 11 – Performance of the algorithms by type of Outliers over Dataset 10; each graphic contains the MSE of the estimations by each percentage of outliers.



Least Squares

M-Estimator

MM-Estimator

RANSAC

S-Estimator

Figure 12 – Performance of each algorithm over Dataset 10; each graphic contains the MSE of one algorithm when varying type and percentage of outliers.

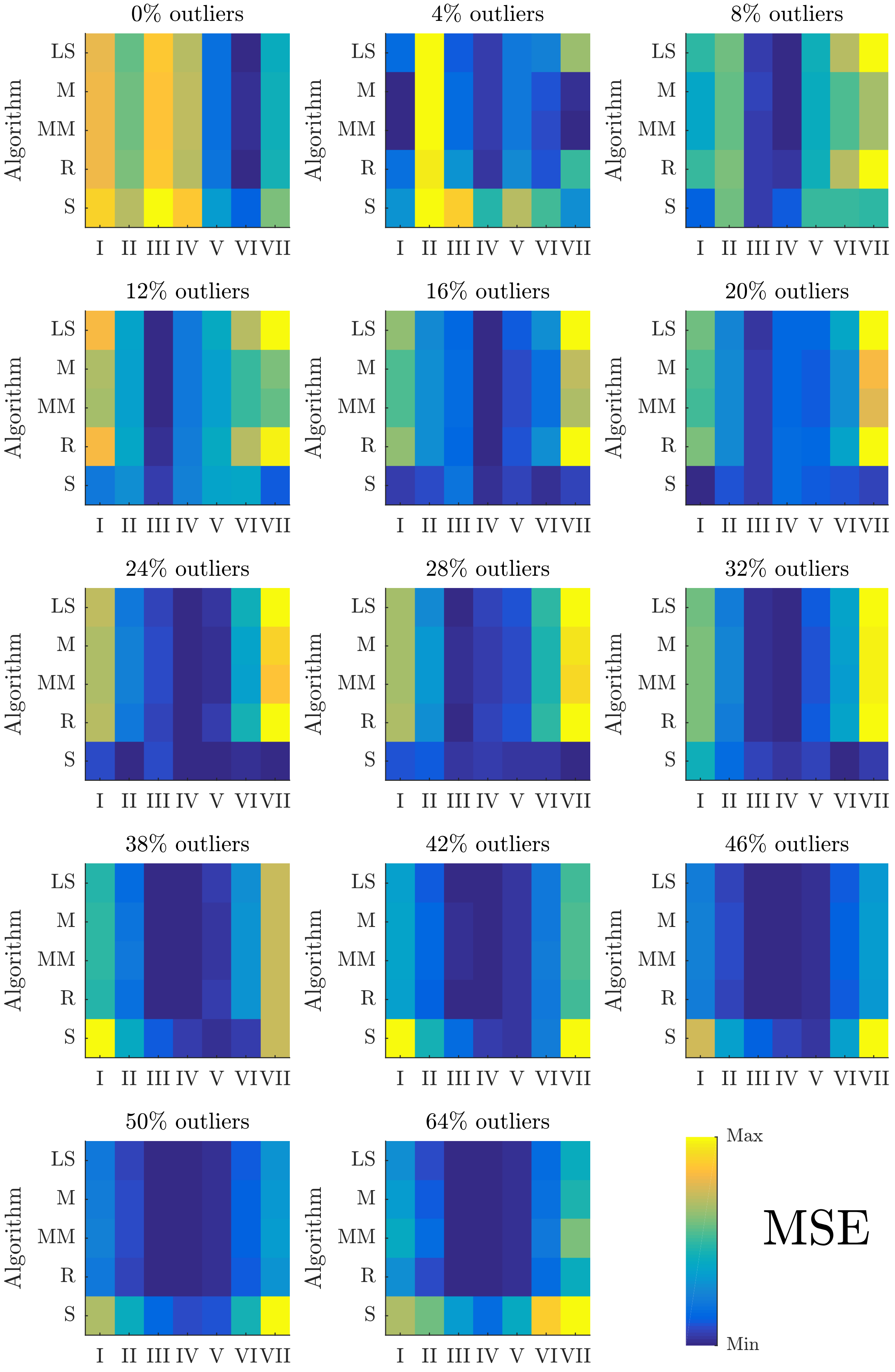


Figure 13 – Performance of the algorithms by percentage of Outliers over Dataset 10; each one of them contains the MSE of the estimations made by all the algorithms over each type of outliers.

4.2.2.4 Dataset 4B

This dataset follows a non-linear pattern, that is why are used Gaussian Basic Functions. The quantity of single experiments is high, as well as the number of graphics. The presentation of the results of the experiments executed with 31 centroids and $\sigma = 0.1$ for the Gaussian BF is made in order to discuss part of the findings and also because it was the configuration where the majority of the algorithms perform best. The Table 7 shows the configuration and scores when each of the algorithms performs best.

The Figure 15 shows how the least squares and the RANSAC algorithms are unstable under the set of conditions of outliers. As it was explained before, the RANSAC algorithm uses an overoptimistic technique to calculate the number of iterations to stop; this combined with the use of minimal sample sets makes the RANSAC unstable under this context of BF. The RANSAC stills perform similar to the LS , indeed, the lower MSE is similar to the MSE obtained by the LS; besides that the RANSAC seems to cope with the extreme outliers.

The behavior of the least squares, although unstable, shows some correspondence with the features exhibited in the other synthetics datasets. The 'positive' effect of the extreme outlier when the type of outliers is V, VI or VII is present.

The M-Estimator, MM-Estimator and the S-Estimators look to perform well. In Figure 15 and Figure 16 these estimators show symmetry and their performance until the breakdown point is better than LS and RANSAC; that means that the extreme outlier is well treated.

The M-Estimator and MM-Estimator achieve the best performances (mostly the M), even with high percentage of outliers and using the similar to Gaussian noise conditions (because of the poor asymptotic efficiency of the S-Estimator). That can be perceived in Figure 16 and in Figure 15 when the type is IV or the outliers percentage is 0.

Table 7 – Best MSE achieved by the algorithms in the 4B dataset. It contains the configuration values in where the algorithms performs best.

	Best MSE	Centroids Number	σ Gaussian BF	Outliers %	Outliers Type
Least Squares	0.0269	31	3	32	V
M-Estimator	0.0103		2	4	IV
MM-Estimator	0.0103				
RANSAC	0.0225			0	II
S-Estimator	0.0109			28	II

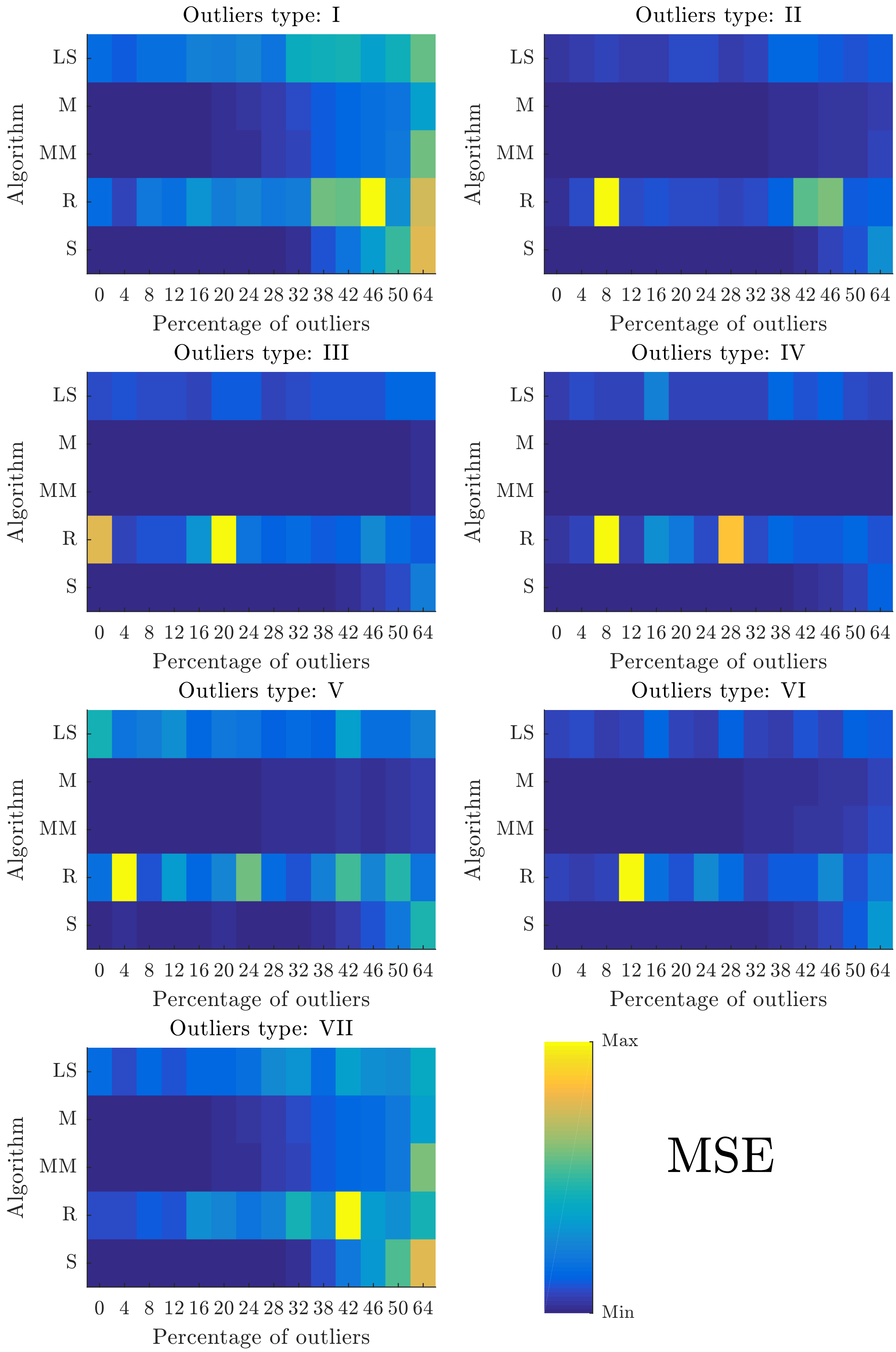


Figure 14 – Performance of the algorithms by σ of the GBF over the real dataset; each graphic shows the MSE in semilogarithmic scale of the estimations when varying the number of centroids.

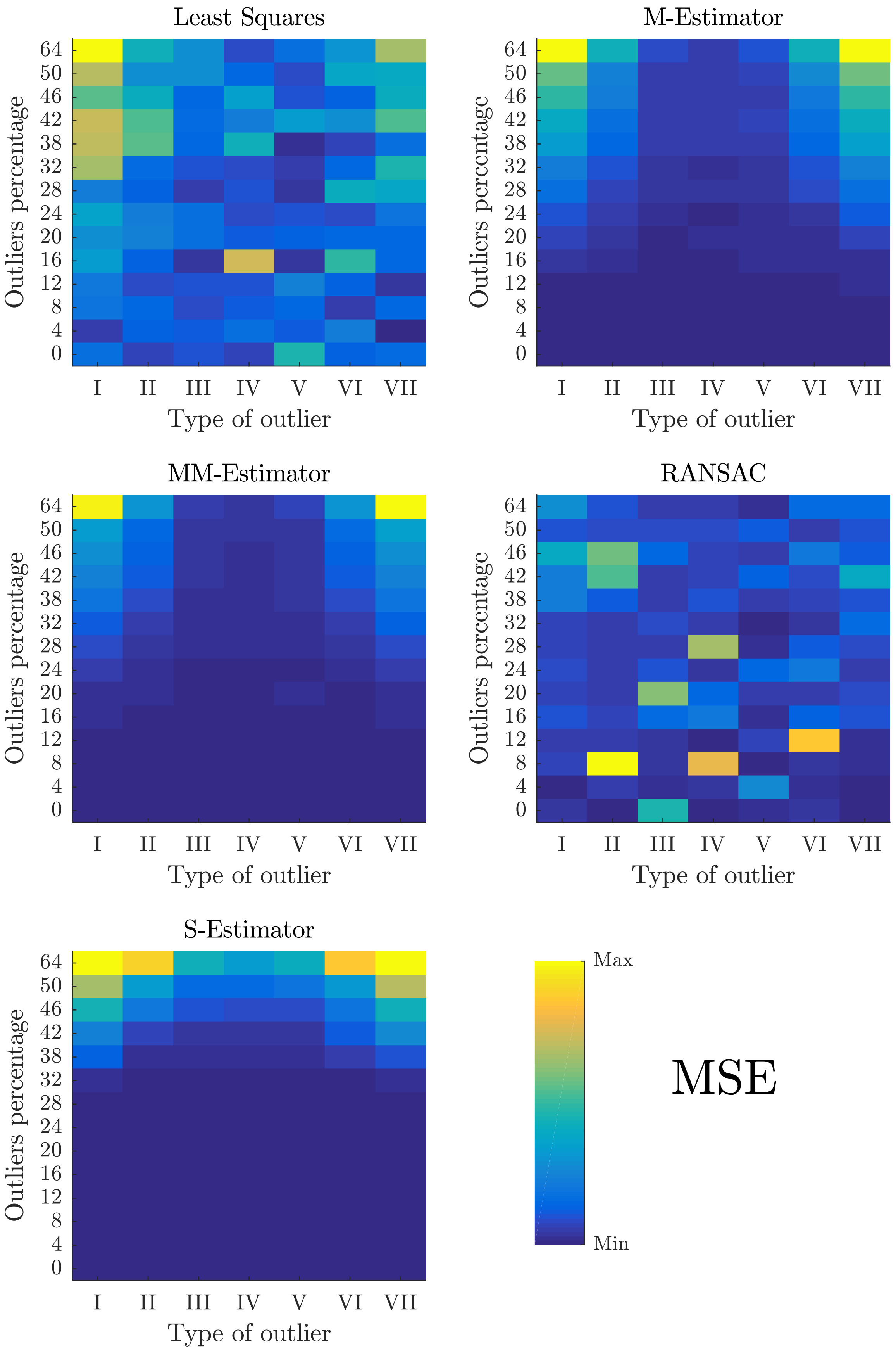


Figure 15 – Performance of the algorithms by number of centroids over the real dataset; each graphic shows the MSE in semilogarithmic scale of the estimations when varying standard deviation used on the GBF.

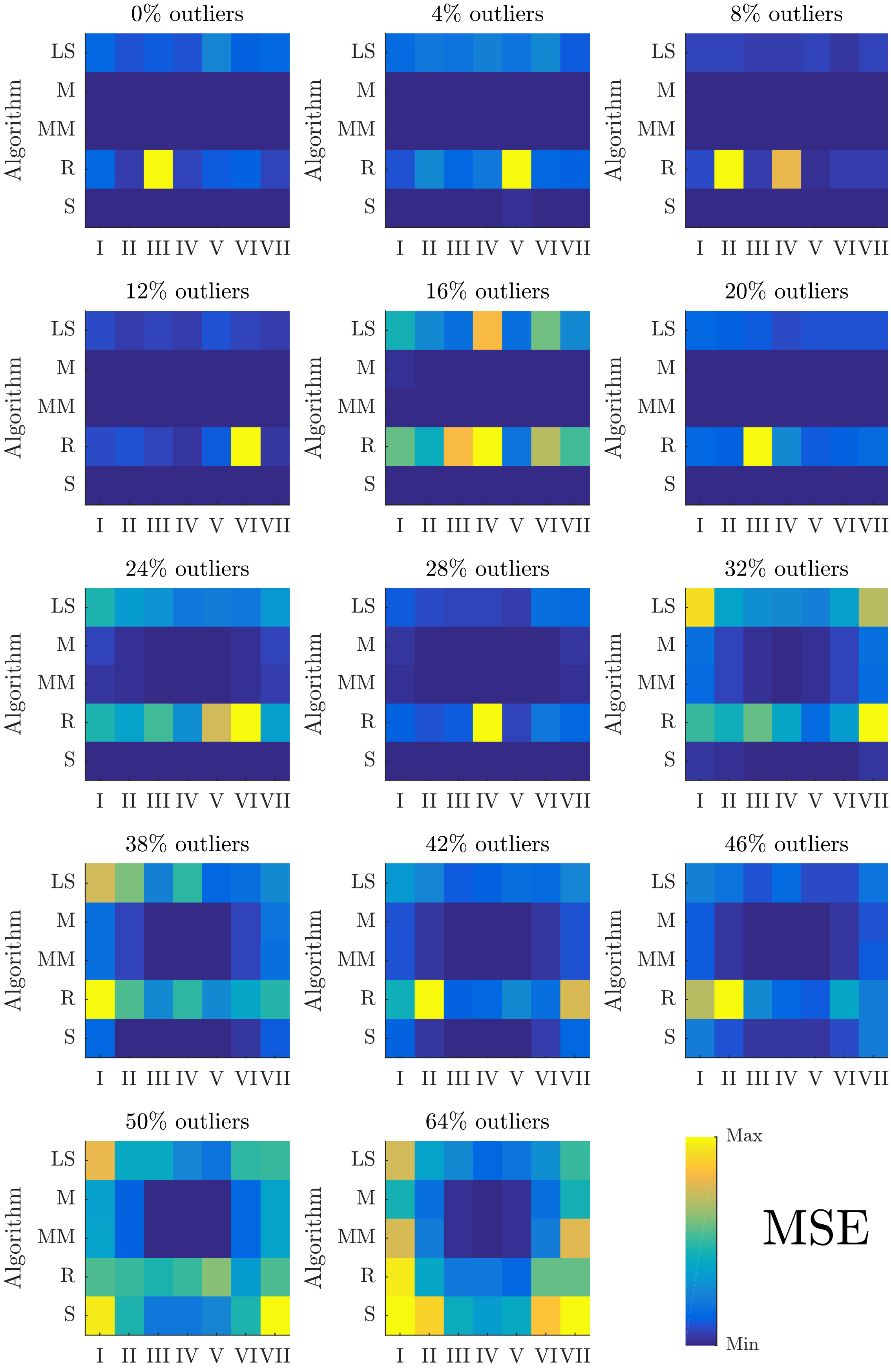


Figure 16 – Performance of the algorithms by percentage of Outliers over Dataset 10; each one of them contains the MSE of the estimations made by all the algorithms over each type of outliers.

4.3 Real Dataset

The real dataset used in the experiments of this work was the Yacht Hydrodynamics Data Set. It collects some characteristics of sailing yachts at its initial design stage. The objective is to predict the residuary resistance for “evaluating the performance of the ship and for estimating the required propulsive power” Lopez (1981).

4.3.1 Description

The Yacht dataset is composed of 308 experiments which were performed at the Delft Ship Hydromechanics Laboratory. The ships studied include 22 different hull forms. Variations concern hull geometry coefficients and the Froude number. The explanatory variables $\mathbf{x}_i \in \mathbb{R}^6$ are

1. Longitudinal position of the center of buoyancy.
2. Prismatic coefficient.
3. Length-displacement ratio.
4. Beam-draught ratio.
5. Length-beam ratio.
6. Froude number.

The measured (response) variable for every $\mathbf{t}_i$ is the residuary resistance per unit weight of displacement (LOPEZ, 1981). Let $\mathscr{A}$ be the entire dataset, then $\tilde{\mathscr{T}} = \mathscr{T}$ and $\tilde{\mathscr{M}} = \mathscr{M}$. All the dataset is normalized to have mean 0 and standard deviation 1.

4.3.2 Results

The results of the real dataset offer a different perspective from the results showed in the synthetic datasets experiments. The general performance of the algorithms is poor in all the experiments made, even when the Gaussian BFs are used. The regressions made using Gaussian basis functions are selected because the lowest MSE was reached using them. The best result achieved is with the LS using 63 centroids and 1 standard deviation; the MSE value was 0.2408.

High values of MSE are obtained by the S-Estimator when the number of centroids is increased. The underlying structure of the transformation made by the Gaussian basis function looks to have bad influence in the performance of the algorithms; nevertheless the causes of that influence is not in the scope of the work. The Figure 17 shows that the least squares and the

M-Estimator are more stable than the other algorithms when varying the number of centroids and the standard deviation.

The performance of the S-Estimator is the worst in the majority of the cases (see Figure 18). As expected, the performance of the MM-Estimator is related with the M performance and the S performance. The RANSAC algorithm has the second-best global performance (0.8125) and its performance follows the least squares in the majority of the cases.

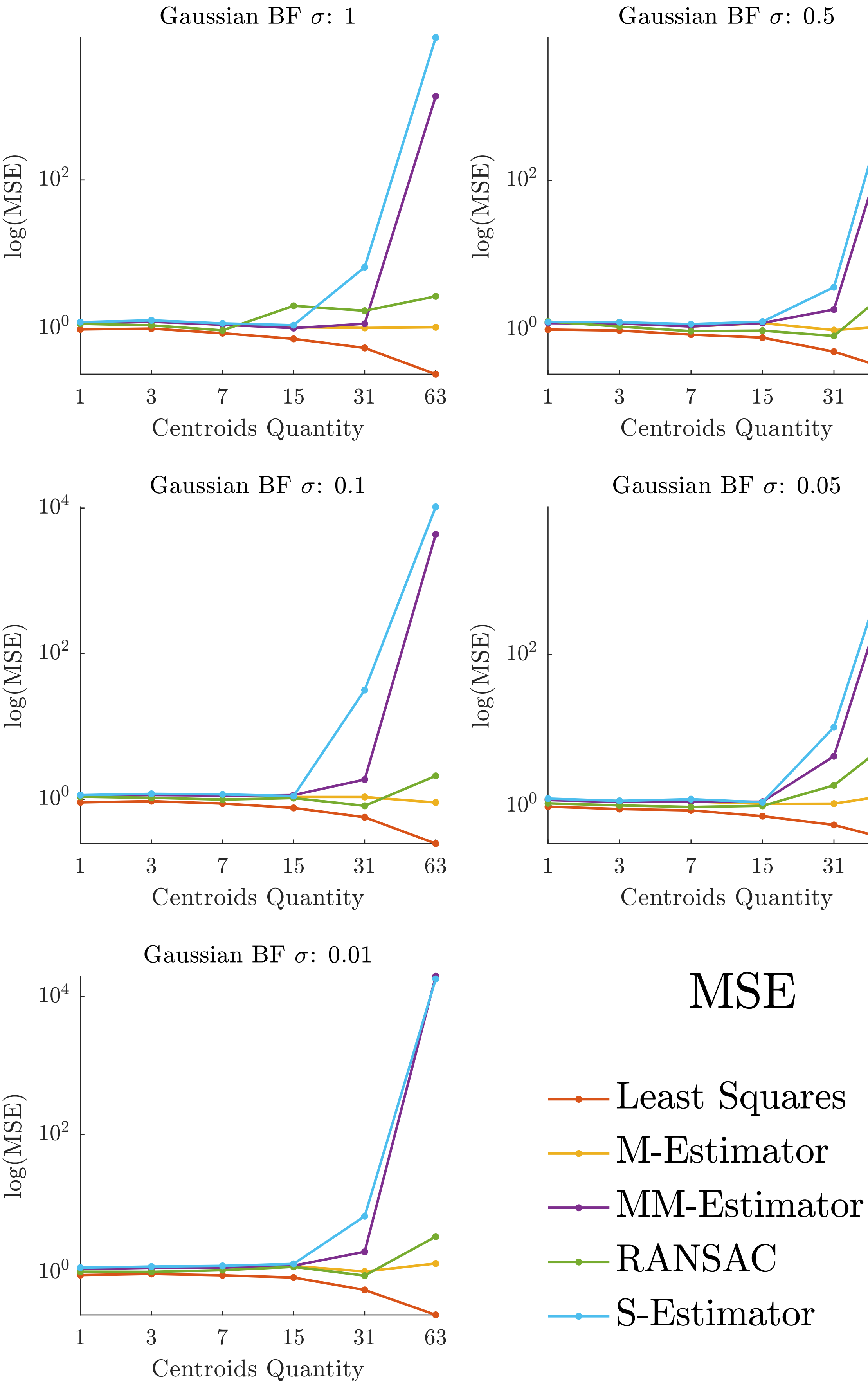


Figure 17 – Performance of the algorithms by standard deviation of the GBF over the yatch dataset; each graphic contains the MSE of the estimations when varying the number of centroids.

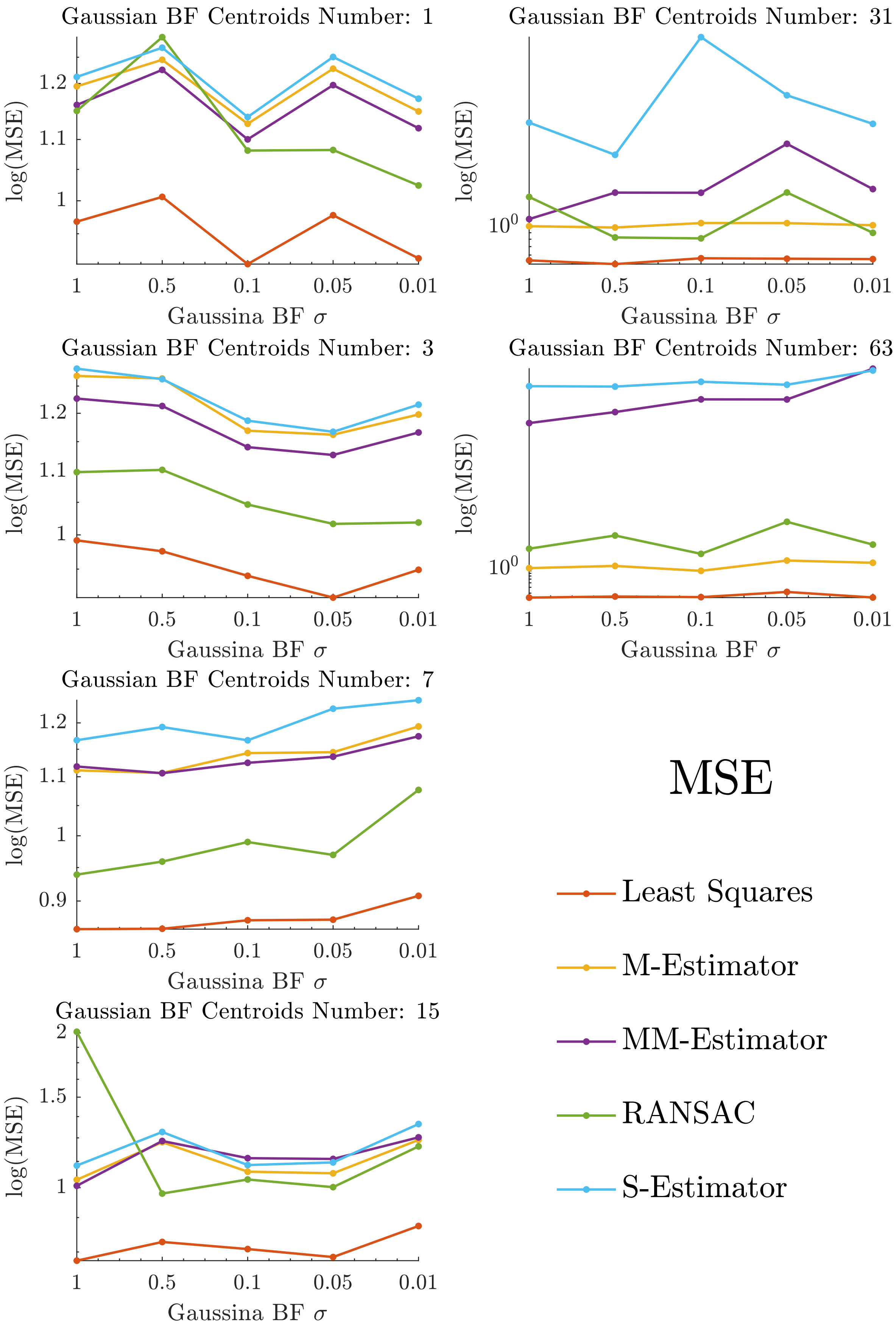


Figure 18 – Performance of the algorithms by number of centroids of the GBF over the yatch dataset; each graphic contains the MSE of the estimations when varying the standard deviation

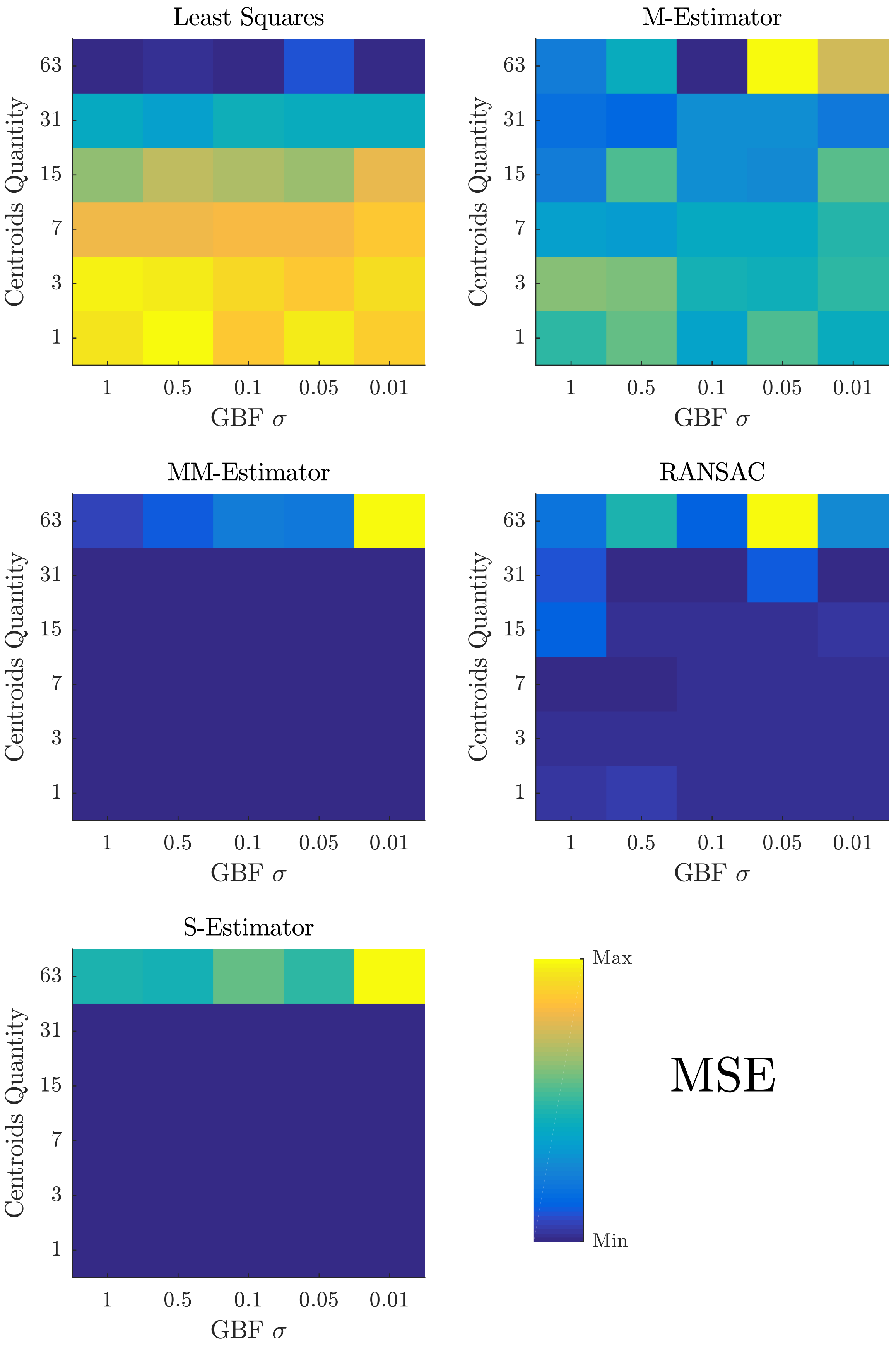


Figure 19 – Performance of each algorithm over the yatch dataset; each graphic contains the MSE of one algorithm when varying the standard deviation and the quantity of centroids used in the GBF.

5 EXPERIMENTS ROBUST LOCALLY LINEAR EMBEDDING

In order to compare the performance of the RALLE algorithm proposed in the Section 3.3.2, the realization of a set of experiments to reduce the dimensionality of some datasets is made. To accomplish this, the classic LLE, the Robust LLE proposed by Chang e Yeung (2006) and the RALLE are executed and their results evaluated. Three synthetic datasets with outliers and one dataset with real data are used.

5.1 Methodology of the experiments

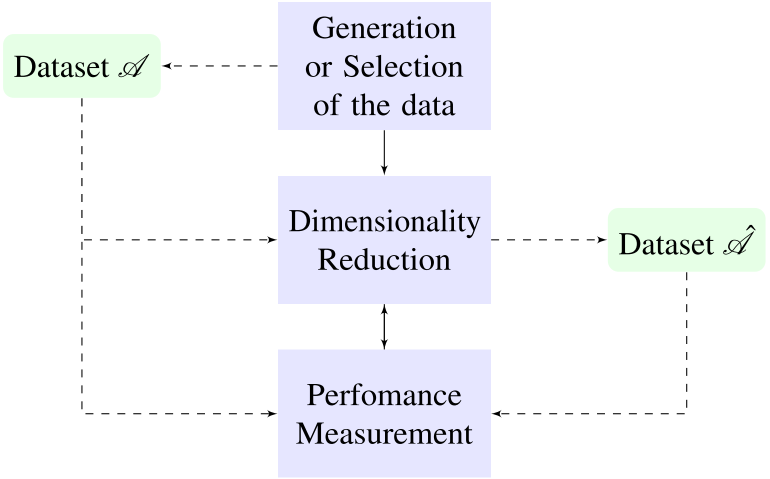


Figure 20 – Process of a single experiment

Each single experiment consists in the dimensionality reduction of some dataset. After the estimation, the resultant less-dimensional dataset is evaluated; the performances of the algorithms were measured with the Trustworthiness and Continuity. The following list of steps explains the experiment process:

1. A dataset is generated (synthetic dataset) or selected (real dataset); and a k-dimensional space is defined, where k is lower than the dimensionality of the original dataset.
2. The dimensionality reduction is executed using the classic LLE, the RLLE and the RALLE. The number of neighbors and the tolerance are the same for the three algorithms.
3. Performance measurement process is executed. The trustworthiness and continuity measures are used to evaluate the quality of the reduction. These are executed using all the neighborhood sizes as the k parameter. The highest mean of the trustworthiness score and continuity score is elected as the best result for that number of neighbors and tolerance.

Table 8 – Methodology and parameters of the dimensionality reduction experiments

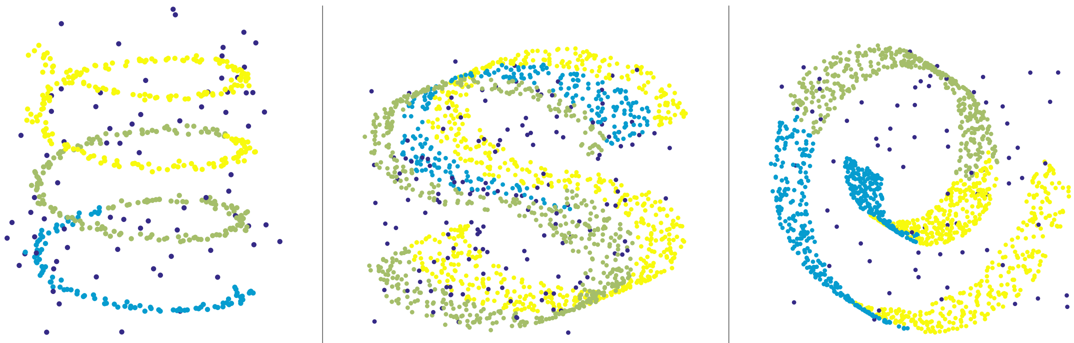
Dataset	New k Dimension	Neighbors Numbers	Tolerances α set	RLLE ε Threshold	RALLE T2 and Q
S-Curve	2	10, 11, ..., 30	1e-1, 1e-2, ..., 1e-7	0.5	90%, 95%, 99%
Helix	1				
Swiss Roll	2			0.75	90%, 91%, 91.5% 92.5%, 95%, 99%
Duck	2	4,5, ...,15	N/A	0.4, 0.5, 0.85, 1	85%, 90%, 95% 99%

4. Another set of parameters (neighbors-tolerance) is selected and the process starts again at step 2. If the whole combination of parameters was already selected, the experiment stops and the best result of all the executions is chosen.

The three algorithms used to reduce the dimension of the data require the definition of some parameters. The Table 8 shows the entire configuration defined for the three methods in each dataset.

5.2 Synthetic Datasets

With the main goal being to execute some tests and compare the algorithms, a synthetic data is generated. To accomplish the experimentation process, the swiss-roll, the scurve and the Helix figures with additional outliers are designed. The reason for choosing such datasets was because they are classic figures used in the related literature, even in the original proposals of the LLE and RLLE.



(a) Helix (b) S-Curve (c) Swiss Roll

Figure 21 – Figures of the generated datasets

The Table 9 indicates which are the settings used to generate the datasets. A clean design is made from every figure, then every point is polluted with some white noise. The white noise is Gaussian data with $\mu = 0$ and some specific σ. Additionally, a set of outliers are included in the dataset. The Outliers are generated using the continuous uniform distribution between $(max+\sigma)$ and $(min-\sigma)$, where *max* is the extreme positive of the data and *min* is the extreme negative.

Table 9 – Parameters used to generate the synthetic datasets

	Helix	S-Curve	Swiss Roll
Original Dimensionality	3	3	3
Quantity of points	497	1500	1500
σ of the white noise	0.05	0.01	0.01
Percentage of extra outliers	15%	10%	5%

5.2.1 Results

As it can be observed inside the Table 10, the Classic LLE did not achieve the highest performance in any of the transformed figures. It looks like the RALLE and, to a lesser extent, the RLLE algorithms obtain TC scores inside of a major range of values. That is commonly associated with some parametric ML algorithms (it makes the dependency on the assumptions stronger).

Generally the robust approaches of the LLE achieved better results than the classical proposal. Nevertheless, only one figure (helix) was transformed reaching the best possible score. The three methods perform worse in the Swiss Role figure than in any other figure, even though it was the dataset that included the lowest percentage of outliers. The parameters used in each algorithm that produce the embedding with the best scores are specified inside the Table 11.

Table 10 – It includes the best (max) TC score, the mean TC score and the lowest (min) TC score obtained by the three algorithms over the three figures

	Helix			S-Curve			Swiss Roll		
	max	mean	min	max	mean	min	max	mean	min
LLE	≈ 1	0.9863	0.8595	0.9971	0.9850	0.9172	0.9972	0.9279	0.8464
RALLE	≈ 1	0.9778	0.7444	0.9997	0.9781	0.8289	0.9967	0.9515	0.8041
RLLE	≈ 1	0.9852	0.8523	0.9997	0.9787	0.9074	0.9976	0.9469	0.7872

Table 11 – It contains the values of the parameters that correspond with the best scores of each algorithm over all the datasets. The RALLE algorithm offer the best result when using a number greater or equal to the number of the other algorithms.

	Tolerance			Neighbors Number			RLLE ε Threshold	RALLE T2 and Q
	LLE	RALLE	RLLE	LLE	RALLE	RLLE		
Helix	0.01			18	23	16	0.5	90%
S-Curve	0.01			10	27	18		
Swiss Roll	0.001	1e-4	0.001	15	20	20	0.75	91%

5.2.1.1 Helix

The unique figure with unidimensional embedding was also unique in that the algorithms perform best there. The 3 representations within the Figure 23 look almost ideal. RALLE obtains the widest range of TC scores. It can be noted analyzing the Figure 22 that the tolerance values with the least unstable TC scores were the higher ones (from 0.001 to 0.1).

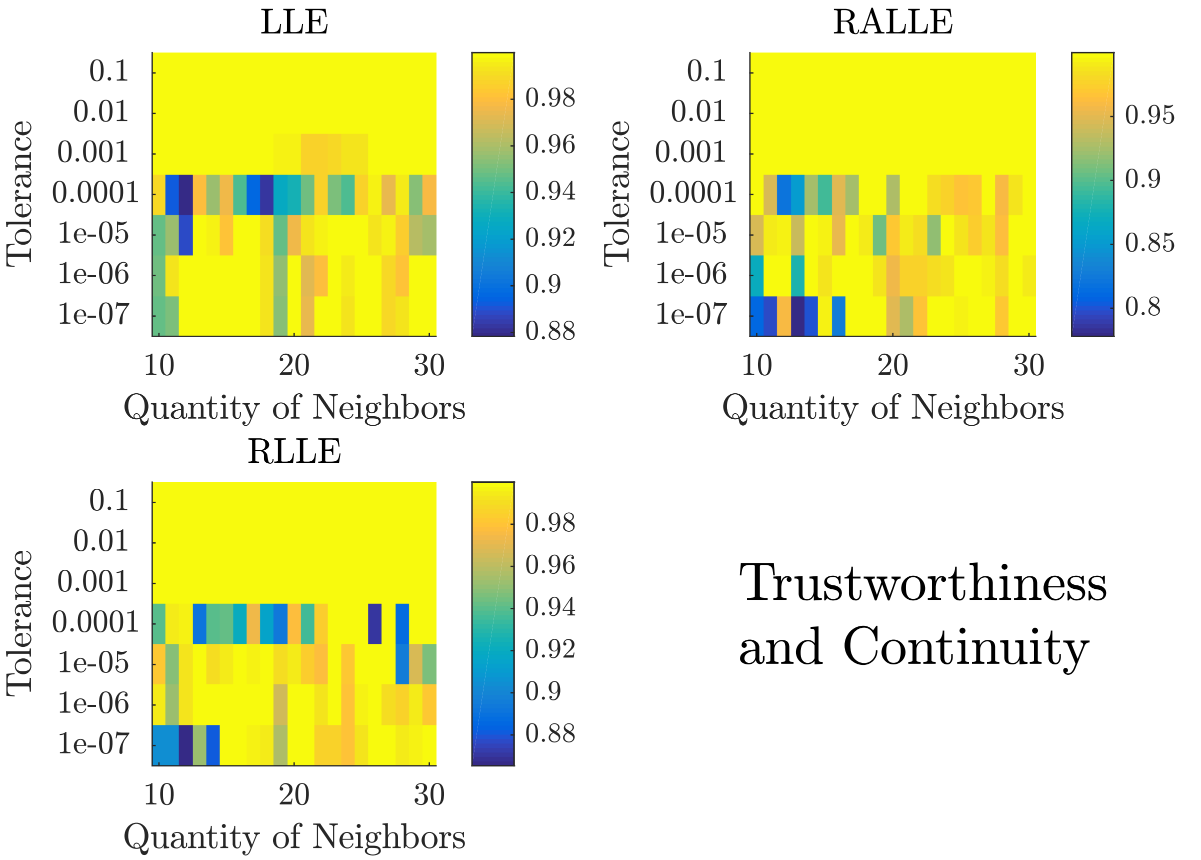
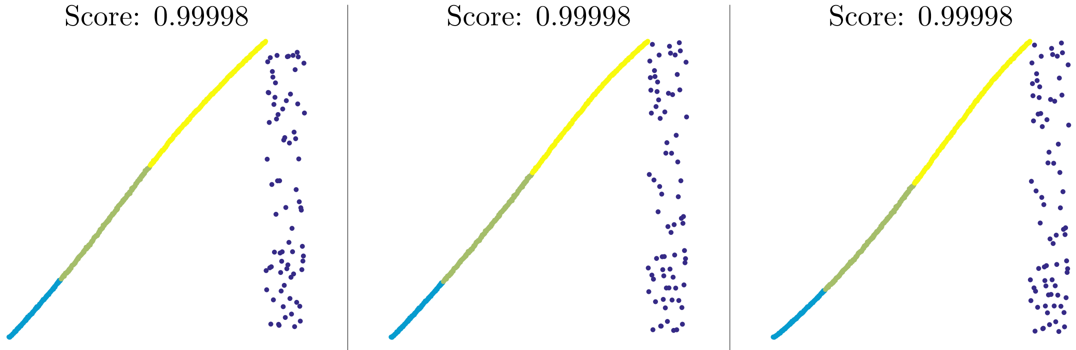
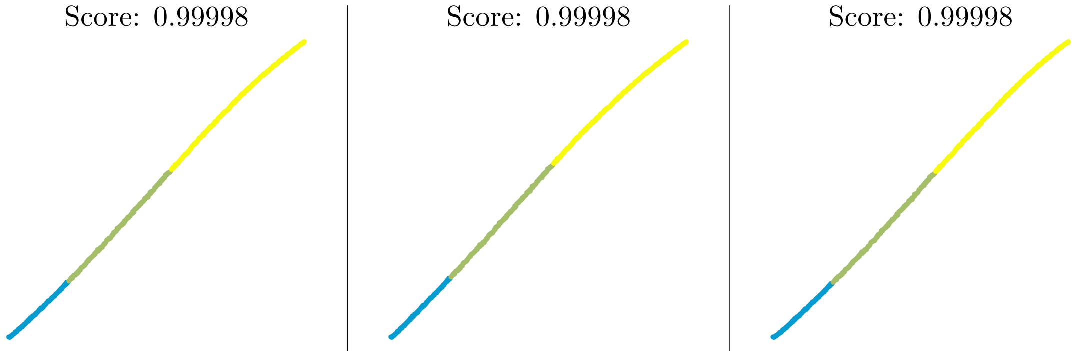


Figure 22 – Representation of the trustworthiness and continuity scores of the Helix embeddings. Each graphic contains the score values of one algorithm when varying the tolerance and the number of neighbors.



(a) LLE (b) RALLE (c) RLLE

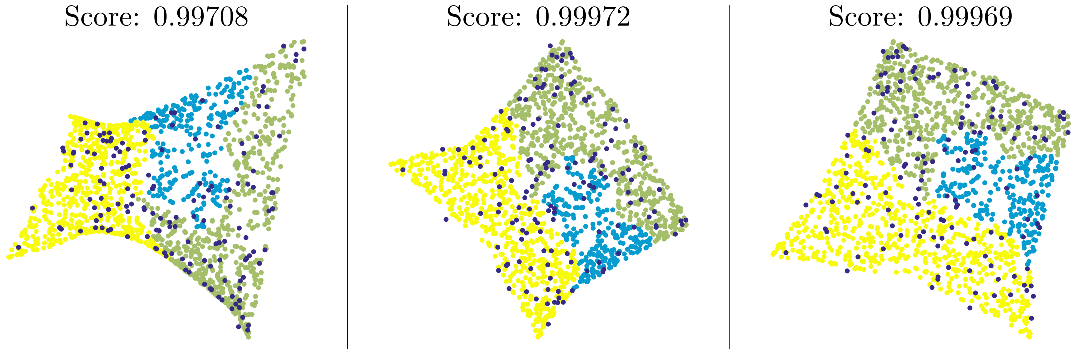
Figure 23 – Best 1-Dimensional embeddings of the algorithms. The x-dimension shows the indexes of all the points and the y-dimension shows its embedding values. The ideal embedding representation is the one in which the inliers form a straight diagonal line.



(a) LLE (b) RALLE (c) RLLE

Figure 24 – Best 1-Dimensional embeddings of the algorithms over the dataset without outliers. The x-dimension shows the indexes of all the points and the y-dimension shows its embedding values. The ideal embedding representation is the one in which the inliers form a straight diagonal line.

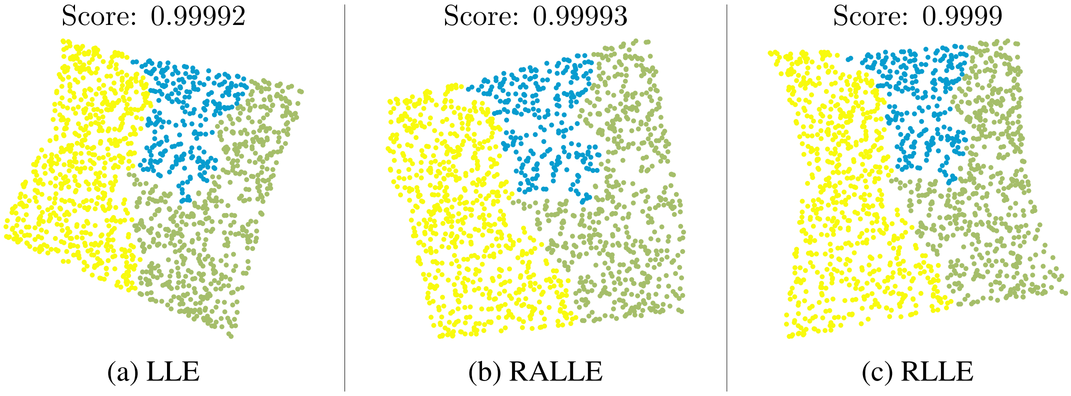
5.2.1.2 S-Curve



(a) LLE (b) RALLE (c) RLLE

Figure 25 – Best 2-Dimensional embeddings of the algorithms. The ideal embedding is a squared figure with three color clusters.

The best performance was obtained by the RALLE algorithm; it was closely followed by the RLLE embedding with minor visual differences. The LLE complete the list with a distorted figure (see 25 for details). Additionally, the Figure 27 shows that the higher values of tolerance used (from 0.001 to 0.1) seem to be favorable for all the algorithms.



(a) LLE

(b) RALLE

(c) RLLE

Figure 26 – Best 2-Dimensional embeddings of the algorithms over the dataset without outliers. The ideal embedding is a squared figure with three color clusters.

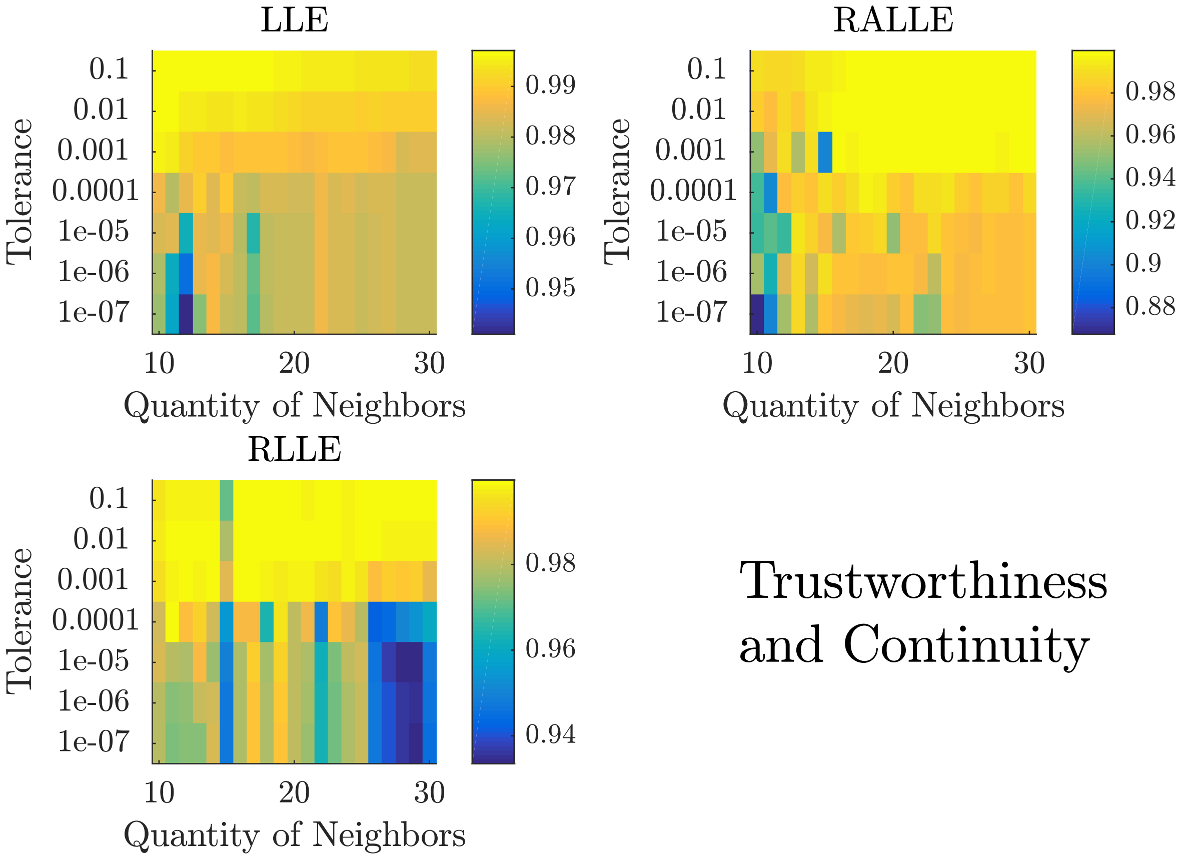
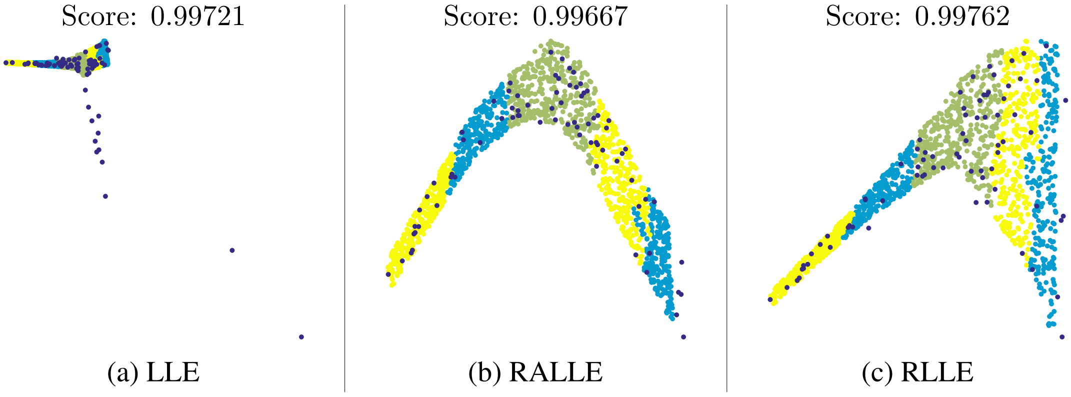


Figure 27 – Representation of the trustworthiness and continuity scores of the S-Curve embeddings. Each graphic contains the score values of one algorithm when varying the tolerance and the number of neighbors.

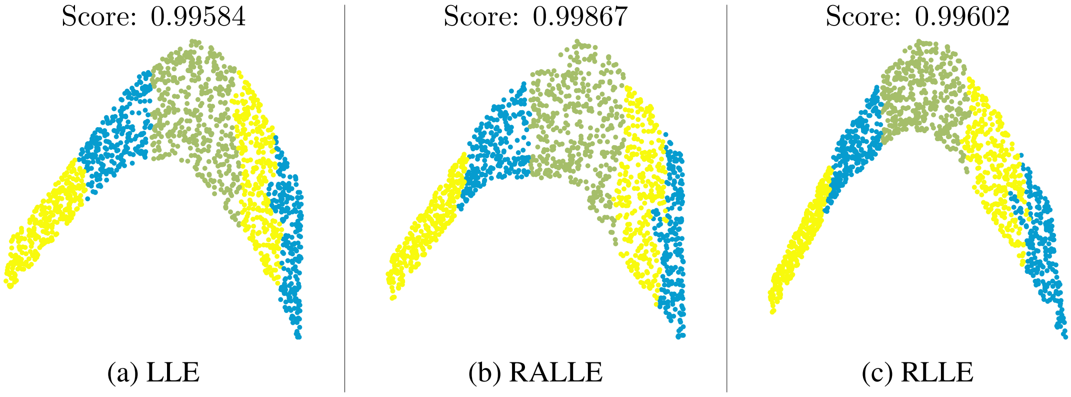
5.2.1.3 Swiss Roll



(a) LLE (b) RALLE (c) RLLE

Figure 28 – Best 2-Dimensional embeddings of the algorithms. The ideal embedding is a rectangular figure with well-defined color clusters.

The *best TC score* of the RALLE in the Swiss Roll embeddings was the lowest *best TC score* obtained by the RALLE algorithm over all the synthetic datasets. An interesting result from the LLE is taken; the influence of the outliers is clearly manifested inside Figure 28a. The TC classify the LLE as second because it only took the inlier points to calculate the score. To confirm this statement, the TC scores of the same figures were recalculated; the TC scores obtained by the same best figures of the LLE, RALLE and RLLE but also using the outliers to make the calculation were: 0.9883, 0.9959 and 0.9953 respectively.



(a) LLE (b) RALLE (c) RLLE

Figure 29 – Best 2-Dimensional embeddings of the algorithms over the datasets without outliers. The ideal embedding is a rectangular figure with well-defined color clusters.

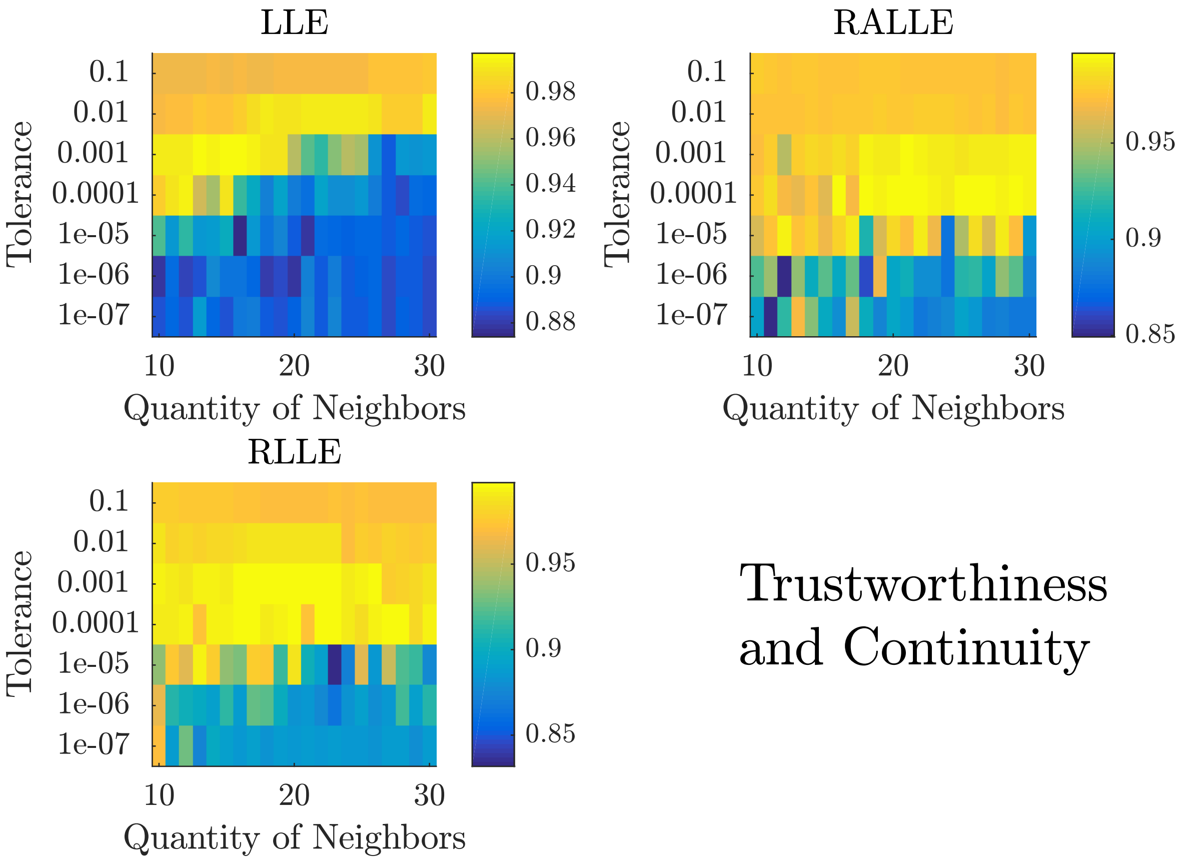


Figure 30 – Representation of the trustworthiness and continuity scores of the Swiss Roll embeddings. Each graphic contains the score values of one algorithm when varying the tolerance and the number of neighbors.

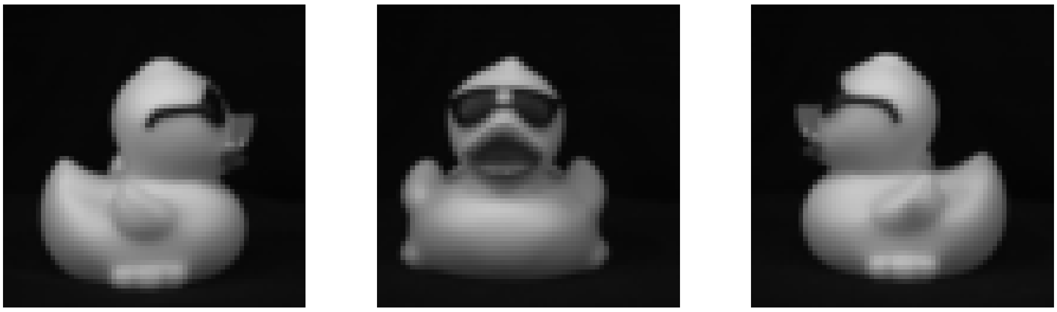
5.3 Real Dataset

The Amsterdam Library of Object Images (ALOI) is a database that contains images of 1000 distinct objects. Various imaging circumstances are captured for each element within the Library, including variations in the illumination angle, illumination color and viewing angle (GEUSEBROEK *et al.*, 2005). The last one, is the variation that is chosen to be part of the experiments. From all the objects that belong to the set, one was selected. It was the object number 62, the plastic yellow duck.

5.3.1 Description

The dataset is composed by images with 72 different viewing angles, starting on 0° and finishing on 355°. Every 5° a new picture was taken. Originally, the size of the images was 192 × 144 pixels, but for reducing the computational complexity, it was firstly cropped to

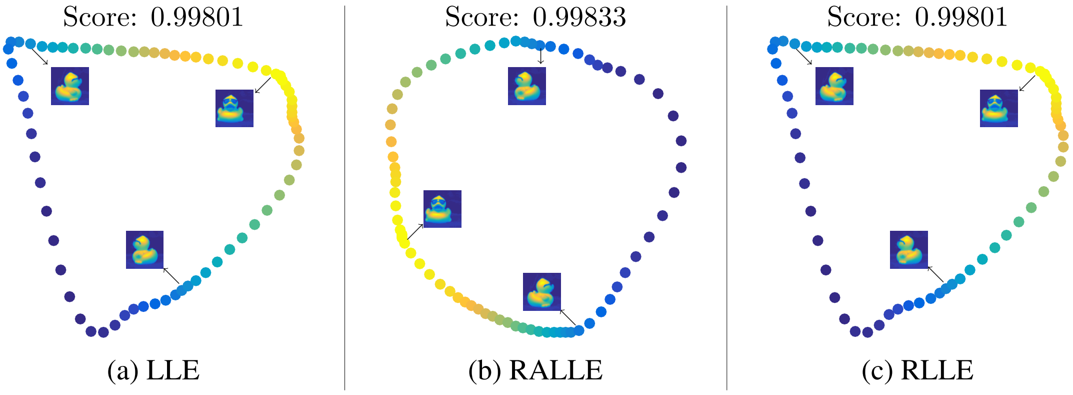
144×144 pixels and then rescaled to 64×64. Therefore, the dimensionality of the dataset is 4096.



(a) 90° Duck Image (b) 0° Duck Image (c) 270° Duck Image

Figure 31 – ALOI's Duck Dataset

5.3.2 *Results*



(a) LLE (b) RALLE (c) RLLE

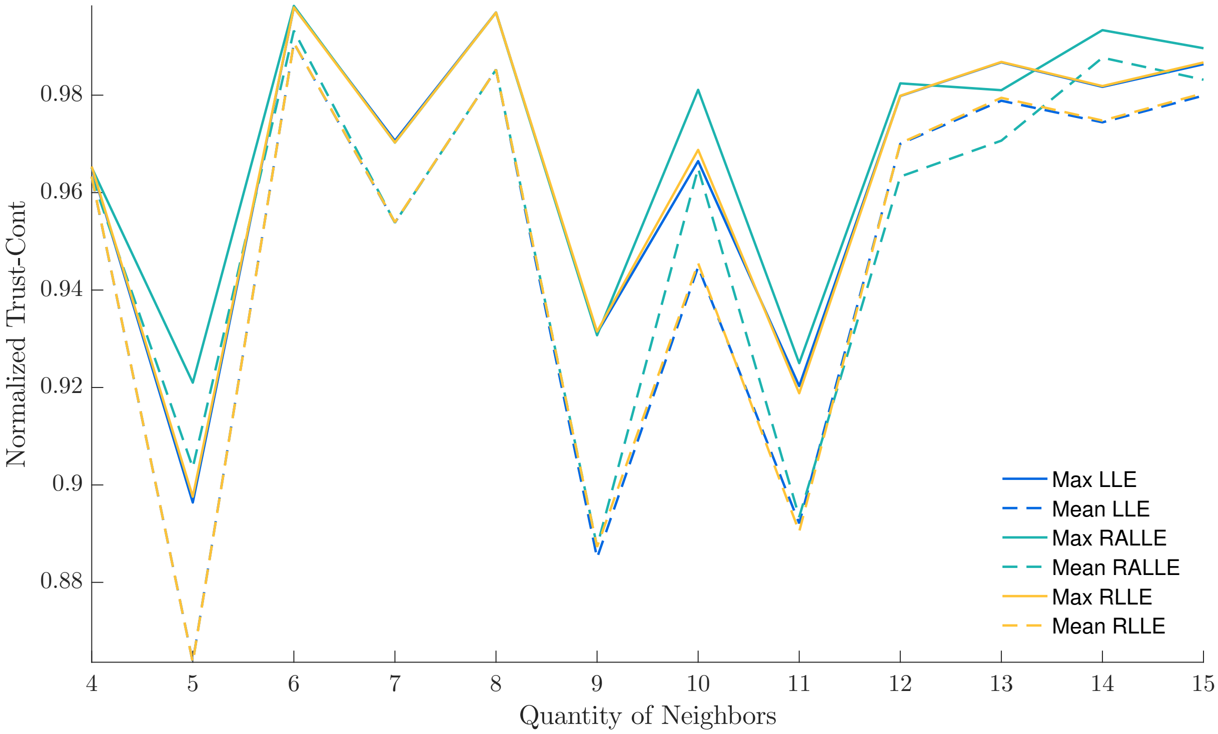
Figure 32 – Best Duck Embeddings of the algorithms.

The values of the parameters that correspond with the best scores of each algorithm over the duck database are the following

- LLE: Number of neighbors is equal to 9
- RALLE: Number of neighbors is equal to 5. The T2 and Q cut-off confidence is 99%.
- RLLE: Number of neighbors is equal to 9. The value of the δ threshold is 0.5.

Increasing the value of the δ threshold did not improve the performance of the RLLE algorithm. On the contrary, the resultant scores were lower than the classic LLE scores.

The dimensionality reduction of the Duck dataset was successfully achieved by the three algorithms. The analysis made over the general results shows that the RLLE and the LLE behavior is practically the same; the differences between all the images were imperceptive until detailed revision. It can be also confirmed by the Figure 33.



H

Figure 33 – Duck Trustworthyness and Continuity. It contains the maximum TC scores of the embeddings when varying the quantity of neighbors and its mean TC scores as well. All the scores are Normalized using the LLE mean TC score.

The LLE and the RLLE follow the same essential behavior patterns; the RALLE achieve the higher general mean (0.9541), followed by RLLE (0.9487) and LLE (0.9485). However the maximum scores behavior is similar among all the algorithms(see Figure 33); the RALLE reached the best embedding, not only by its score but also because of its visual presentation.

6 CONCLUSIONS

6.1 Linear Regression Conclusions

Every day much bigger datasets have to be processed and analyzed with different purposes but with some common issues. As it was explained in this thesis, the presence of atypical values in our datasets is almost certainly. Motivated by this and by the study of the robust statistics, the robust linear regression section was developed.

The main goal of that thesis section was study, analyze and test some robust algorithms for the generalized linear regression. The elemental concepts of robustness and some models of outliers were studied and discussed in the introduction sections. The adoption of linear models for studying robustness was a good decision; the analysis of the experiments made over the linear datasets were transparent.

Some important points to note are: the trade-off between the asymptotic efficiency and the breakdown point is strong over all the algorithms, then is important to make conscious selection of the algorithms knowing the weaknesses and strengths; the parameters of the algorithms have to be carefully chosen, some of them are designed to tune some features of the trade-off.

In this thesis, the default sets of parameters for each algorithm is used, detailed, and its effects in the resultant models are explained in the results and resumed here:

- The least squares perform best when the errors are truly with Gaussian distribution, but one simply error can break the estimation. It can perform best over some dataset with ambiguous linear relations or when the quantity of outliers is higher than the BDP of the other robust estimators.
- The RANSAC algorithm can handle with some atypical values with almost the same asymptotic efficiency of the LS. It has instability problems due to the use of minimal sets and its (overoptimistic) iteration limit process.
- The M-Estimator has high asymptotic efficiency (arround 95%) and can cope with enough percentage of outliers (28% at least) without break. Likewise the LS, it is stable. This estimator seems perform well under the majority of the tested circumstances.
- The performance of the MM-Estimator generally lies between the M-Estimator performance and the S-Estimator performance. It is asymptotic efficient and has high breakdown point; but it inherits the problems of the two algorithms that compound it.
- The S-Estimator has the highest BDP. It performs better than the other algorithms when

the percentage of outliers grows, but it performs worst in presence of Gaussian noise or similar conditions.

The use of the robust algorithms in combination with the classic algorithms (commonly with high asymptotic) is suggested. It is recommended to learn the concepts of the algorithms that can be employed over the specific problem and compare the results to note which of them are generalizing better.

6.2 Locally Linear Embedding Conclusions

In this thesis work, the principles of the Locally Linear Embedding were analyzed. Some robust approaches of it were also investigated, such as the RLLE. Besides the objective of understand how the outliers may influence the result of one dimensionality reduction process, the main goal of the dimensionality reduction research section was to propose a modification of the LLE algorithm to provide it with some robustness to outliers; that is why the RALLE was proposed.

The basic principle of the proposed algorithm was the notion of use different sizes of neighborhoods in each calculation of the reconstruction weights. This idea was based on the premise that is possible to have different sizes of the locally linear patches of the points and their neighbors; additionally to that was taken the implicit idea that some reconstruction weights can be zero or close to zero in the classic process used by the locally linear embedding. Thus, it is implemented a similar method used by the robust locally linear embedding; a classification of the neighbors of each point into inliers or outliers is made. The other algorithms use a fixed quantity of neighbors for each embedding process, while the RALLE uses variable sizes of neighborhoods between some minimum and some predefined parameter. In the embedding phase of the algorithm, and since an eigenvalue and eigenvector decomposition has to be made, a matrix of scores is used to do a weighted reduction.

The results obtained in the testing process revealed that the use of robust approaches of the LLE can improve the results in the presence of outliers. The experimental phase demonstrates the instability of the LLE and its variants; this means that one small change in the parameters used (number of neighbors and tolerance) can drastically change the result. These instability of the resultant embeddings is higher in the robust approaches. It can be explained by the inclusion of extra parameters to the algorithm that made stronger the dependency over the assumptions (locally linear patches).

In some cases the trustworthiness and continuity do not score properly the best visual representations; measuring embeddings without tacking into account the outliers can result in an erratic representations with high score (best LLE embedding for the swiss roll). In the other hand, the TC execution tacking into account the outliers can decrease the score of good embeddings in which the outliers are placed inside the set of inliers.

The data inside the duck database is obtained in a very controlled environment; this implicates that the duck dataset contains almost only inliers. The results can show how the notion of neighborhoods of variable size can be an effective tool, and also that the RLLE works identically to the LLE in the absence of outliers. For future develops of this idea, other techniques can be developed to precisely calculate the true size of the locally linear patches of the figures.

BIBLIOGRAPHY

AELST, S. V.; WILLEMS, G.; ZAMAR, R. H. Robust and efficient estimation of the residual scale in linear regression. **Journal of Multivariate Analysis**, Elsevier, v. 116, p. 278–296, 2013.

ANDERSEN, R. **Modern Methods for Robust Regression**. [S.l.]: SAGE Publications, 2008. (Modern Methods for Robust Regression, N^{o} 152). ISBN 9781412940726.

ANSCOMBE, F. J. Rejection of outliers. **Technometrics**, Taylor & Francis Group, v. 2, n. 2, p. 123–146, 1960.

BARNETT, V. The study of outliers: purpose and model. **Applied Statistics**, JSTOR, p. 242–250, 1978.

BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387310738.

CHANG, H.; YEUNG, D.-Y. Robust locally linear embedding. **Pattern recognition**, Elsevier, v. 39, n. 6, p. 1053–1065, 2006.

DAVIES, P. *et al.* Aspects of robust linear regression. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 21, n. 4, p. 1843–1899, 1993.

DIAKONIKOLAS, I.; KAMATH, G.; KANE, D. M.; LI, J.; MOITRA, A.; STEWART, A. Robust estimators in high dimensions without the computational intractability. In: IEEE. **Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on**. [S.l.], 2016. p. 655–664.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, ACM, v. 24, n. 6, p. 381–395, 1981.

FREIRE, A.; BARRETO, G. A robust and regularized extreme learning machine. 2014.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. **The elements of statistical learning**. [S.l.]: Springer series in statistics Springer, Berlin, 2001. v. 1.

GEUSEBROEK, J.-M.; BURGHOUTS, G. J.; SMEULDERS, A. W. The amsterdam library of object images. **International Journal of Computer Vision**, Springer, v. 61, n. 1, p. 103–112, 2005.

HAMPEL, F. R. Robust estimation: A condensed partial survey. **Probability Theory and Related Fields**, Springer, v. 27, n. 2, p. 87–104, 1973.

HARTLEY, R. I.; ZISSERMAN, A. **Multiple View Geometry in Computer Vision**. Second. [S.l.]: Cambridge University Press, 2004. ISBN 0521540518.

HORATA, P.; CHIEWCHANWATTANA, S.; SUNAT, K. Robust extreme learning machine. **Neurocomput.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 102, p. 31–44, fev. 2013. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2011.12.045>.

HUBER, P. J.; RONCHETTI, E. M. **Robust statistics**. 2. ed. [S.l.]: Wiley, 2009. (Wiley Series in Probability and Statistics). ISBN 9780470129906.

HUBERT, M.; DEBRUYNE, M. Minimum covariance determinant. **Wiley interdisciplinary reviews: Computational statistics**, Wiley Online Library, v. 2, n. 1, p. 36–43, 2010.

HUBERT, M.; ROUSSEEUW, P. J.; BRANDEN, K. V. Robpca: a new approach to robust principal component analysis. **Technometrics**, Taylor & Francis, v. 47, n. 1, p. 64–79, 2005.

HUBERT, M.; ROUSSEEUW, P. J.; VERBOVEN, S. A fast method for robust principal components with applications to chemometrics. **Chemometrics and Intelligent Laboratory Systems**, Elsevier, v. 60, n. 1, p. 101–111, 2002.

KOENKER, R.; JR, G. B. Regression quantiles. **Econometrica: journal of the Econometric Society**, JSTOR, p. 33–50, 1978.

KOHN, R.; SMITH, M.; CHAN, D. Nonparametric regression using linear combinations of basis functions. **Statistics and Computing**, Springer, v. 11, n. 4, p. 313–322, 2001.

LEE, J. A.; VERLEYSEN, M. **Nonlinear Dimensionality Reduction**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2007. ISBN 0387393501, 9780387393506.

LOPEZ, R. **Yacht Hydrodynamics Data Set**. 1981. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics#>.

MAATEN, L. V. D.; POSTMA, E.; HERIK, J. Van den. Dimensionality reduction: a comparative. **J Mach Learn Res**, v. 10, p. 66–71, 2009.

MITCHELL, T. **Machine Learning**. [S.l.]: McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673.

MÜLLER, C. Redescending m-estimators in regression analysis, cluster analysis and image analysis. **Discussiones Mathematicae-Probability and Statistics**, v. 24, p. 59–75, 2004.

MURPHY, K. P. **Machine learning: a probabilistic perspective**. [S.l.]: MIT press, 2012.

RATCLIFF, R. Methods for dealing with reaction time outliers. **Psychological bulletin**, American Psychological Association, v. 114, n. 3, p. 510, 1993.

ROUSSEEUW, P.; YOHAI, V. Robust regression by means of s-estimators. In: ____. **Robust and Nonlinear Time Series Analysis: Proceedings of a Workshop Organized by the Sonderforschungsbereich 123 "Stochastische Mathematische Modelle", Heidelberg 1983**. New York, NY: Springer US, 1984. p. 256–272. ISBN 978-1-4615-7821-5. Disponível em: <http://dx.doi.org/10.1007/978-1-4615-7821-5_15>.

ROUSSEEUW, P. J.; LEROY, A. M. **Robust regression and outlier detection**. [S.l.]: Wiley, 1987. (Wiley series in probability and mathematical statistics. Applied probability and statistics). ISBN 9780471725374.

ROUSSEEUW, P. J.; ZOMEREN, B. C. van. Unmasking multivariate outliers and leverage points. **Journal of the American Statistical Association**, [American Statistical Association, Taylor and Francis, Ltd.], v. 85, n. 411, p. 633–639, 1990. ISSN 01621459. Disponível em: <http://www.jstor.org/stable/2289995>.

ROWEIS, S. T.; SAUL, L. K. Nonlinear dimensionality reduction by locally linear embedding. **Science**, American Association for the Advancement of Science, v. 290, n. 5500, p. 2323–2326, 2000.

SAUL, L. K.; ROWEIS, S. T. An introduction to locally linear embedding. **unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html**, 2000.

SEN, P. K. Estimates of the regression coefficient based on kendallś tau. **Journal of the American Statistical Association**, Taylor and Francis Group, v. 63, n. 324, p. 1379–1389, 1968.

STUART, C. Robust regression. **Department of Mathematical Sciences. Durham University**, v. 169, 2011.

SUSANTI, Y.; PRATIWI, H. *et al.* M estimation, s estimation, and mm estimation in robust regression. **International Journal of Pure and Applied Mathematics**, Academic Publications, Ltd., v. 91, n. 3, p. 349–360, 2014.

THEIL, H. A rank-invariant method of linear and polynomial regression analysis, part 3. In: **Proceedings of Koninalijke Nederlandse Akademie van Weinenschatpen A**. [S.l.: s.n.], 1950. v. 53, p. 1397–1412.

TORDOFF, B. J.; MURRAY, D. W. Guided-mlesac: Faster image transform estimation by using matching priors. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 27, n. 10, p. 1523–1535, 2005.

TUKEY, J. W. A survey of sampling from contaminated distributions. **Contributions to probability and statistics**, v. 2, p. 448–485, 1960.

VENNA, J.; KASKI, S. Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In: CITESEER. **Proceedings of WSOM**. [S.l.], 2005. v. 5, p. 695–702.

VERARDI, V.; CROUX, C. Robust regression in stata. **Stata Journal**, StataCorp LP, v. 9, n. 3, p. 439–453, 2009.

WANG, C. K.; TING, Y.; LIU, Y. H. An approach for raising the accuracy of one-class classifiers. In: **Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on**. [S.l.: s.n.], 2010. p. 872–877.

XIN, Y.; XIAOGANG, S. **Linear regression analysis : theory and computing**. [S.l.]: World Scientific Pub. Co, 2009. ISBN 9789812834119.

YOHAI, V. High breakdown point and high efficiency robust estimates for regression. **The Annals of Statistics**, v. 15, p. 642–656, 1987.

ZHOU, W.; SERFLING, R. Multivariate spatial u-quantiles: A bahadur–kiefer representation, a theil–sen estimator for multiple regression, and a robust dispersion estimator. **Journal of Statistical Planning and Inference**, Elsevier, v. 138, n. 6, p. 1660–1678, 2008.