UNIVERSIDAD NACIONAL DE COLOMBIA

Robust Unsupervised Learning Using Kernels

Joseph Alejandro Gallego Mejía

Universidad Nacional de Colombia
Engineering School, Computing And Decision Science Department
Medellín, Colombia
2017

Robust Unsupervised Learning Using Kernels

Joseph Alejandro Gallego Mejía

In fulfillment of the requirements for the degree of:
Master on Systems Engineering

Advisor:
Fabio Augusto González Osorio, Ph.D.

Co-Advisor:
Demetrio Arturo Ovalle Carranza, Ph.D.

Research Field:
Machine Learning
Research Group:
MindLab

Universidad Nacional de Colombia
Engineering School, Computing And Decision Science Department
Medellín, Colombia
2017

This work is dedicated to my parents. For their endless love, support and encouragement.

Acknowledgements

I would like to express my gratitude to my supervisors, Professor Fabio Augusto González Osorio and Demetrio Arturo Ovalle Carranza for their great support and direction in the challenging field of machine learning in the last two years. Also to my family, who have enthusiastically supported and encouraged me throughout these two years. This work was partially funded by Colciencias grant 617 Jóvenes Investigadores 2013

Abstract

This thesis aims to study deep connections between statistical robustness and machine learning techniques, in particular, the relationship between some particular kernel (the Gaussian kernel) and the robustness of kernel-based learning methods that use it. This thesis also presented that estimating the mean in the feature space with the RBF kernel, is like doing robust estimation of the mean in the data space with the Welsch M-estimator. Based on these ideas, new robust kernel to machine learning algorithms are designed and implemented in the current thesis: Tukey's, Andrews' and Huber's robust kernels which each one corresponding to Tukey's, Andrews' and Huber's M-robust estimator, respectively. On the one hand, kernel-based algorithms are an important tool which is widely applied to different machine learning and information retrieval problems including: clustering, latent topic analysis, recommender systems, image annotation, and content-based image retrieval, amongst others. Robustness is the ability of a statistical estimation method or machine learning method to deal with noise and outliers. There is a strong theory of robustness in statistics; however, it receives little attention in machine learning. A systematic evaluation is performed in order to evaluate the robustness of kernel-based algorithms in clustering showing that some robust kernels including Tukey's and Andrews' robust kernels perform on par to state-of-the-art algorithms.

Keywords: Machine Learning, Dimensionality Reduction, Unsupervised Learning, Kernel Learning Approach, Robust Statistics, Welsch Estimator

Resumen

Esta tesis apunta a mostrar la profunda relación que existe entre robustez estadística y técnicas de aprendizaje de maquina, en particular, la relación entre algunos tipos de kernels (kernel Gausiano) y la robustez de los métodos basados en kernels. Esta tesis también presenta que la estimación de la media en el espacio de características con el kernel rbf, es como hacer estimación de la media en el espacio de los datos con el m-estimador de Welsch. Basado en las ideas anteriores, un conjunto de nuevos kernel robustos son propuestos y diseñdos: Tukey, Andrews, y Huber kernels robustos correspondientes a los m-estimadores de Tukey, Andrews y Huber respectivamente. Por un lado, los algoritmos basados en kernel es una importante herramienta aplicada en diferentes problemas de aprendizaje automático y recuperación de información, incluyendo: el agrupamiento, análisis de tema latente, sistemas de recomendación, anotación de imágenes, recuperación de informacion, entre otros. La robustez es la capacidad de un método o procedimiento de estimación aprendizaje estadístico automatico para lidiar con el ruido y los valores atípicos. Hay una fuerte teoría de robustez en estadística, sin embargo, han recibido poca atención en aprendizaje de máquina. Una evaluación sistemática se realiza con el fin de evaluar la robustez de los algoritmos basados en kernel en tareas de agrupación mostrando que algunos kernels robustos incluyendo los kernels de Tukey y de Andrews se desempeñan a la par de los algoritmos del estado del arte.

Palabras clave: Aprendizaje de máquina, Reducción de la dimensionalidad, aprendizaje no-supervisado, aprendizaje con métodos de Kernel, estadística robusta, estimador de Welsch.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Producing and storing a bunch of data is now easier with the recent development of computers. Often, when we get a new data set, we cannot be certain that every sample was saved with the same conditions. For example, some of the data points can be generated by different noise errors like rounding off, changing a measurement system, images where some pixels are occluded because of a defect in the lens of a camera, blurred pictures also given by a sudden movement before obstruction, images with different illuminations and so forth. Those kinds of errors bias the estimation of the parameter model. This means that when we try to estimate (for example) several clusters in a data set, if we contaminate it with a big point compared to the rest of the points, we will get a cluster that would try to describe the contaminated point instead of the whole data. Nowadays, almost every device produces and records several data because of this without proper algorithms that build meaningful information, we will probably not be able to acquire new information and find new patterns like clusters.

> *Robustness is a fundamental issue for all statistical analyses; in fact it might be argued that robustness is the subject of statistics.*
>
> J.B.Kadane (1984)

Robust statistics is a new branch of statistics that deals with atypical observation called outliers, these outliers are well separated from the vast majority of data. In classic estimation, estimation of the mean, standard deviation and correlation could be influenced by these outliers doing estimation fail to a provide good fit. Even a single point could bias the estimation. Different approaches are developed in robust statistics to deal with atypical observations, some of them rely on MLE (Maximum Likelihood Estimators), for instance, M-estimators, order statistics such as L-estimators, and, test ranks such as R-estimators. There are several methods in robust statistics that get better asymptotic results when compared to classic methods, i.e. they converge to a correct result when the sample grows, nevertheless, recent studies with computer simulation shows that not all

methods could be used when the sample is small. Several assumptions are made in classic statistics, including, for instance, the observations are randomly selected from a normal distribution; when comparing groups of data, classic methods assume that groups have the same variance even when the mean is different. Recent journal articles show that classic statistics methods have less power compared to robust statistics methods [2] [3].
A probability distribution that comes from real phenomenon are generally highly skewed, i.e. they have bigger tails and the samples often have outliers (unusually large measures among sample observations). Bigger tails and outliers are hard problems because they can grow the standard error of the parameter estimation, obtaining relative less power of the estimation when all the groups are compared. For instance, when mean estimator is compared to the median estimator, it can be observed that mean estimator is biased only with one infinite point, in contrast to the median estimator, where more than a half of the points are needed in order to bias the estimator. In this case, we can say that the median has a bounded influence curve. In general, robust statistics deals with estimators that have a bounded or asymptotically bounded influence curve.
The kernel methodology is a set of algorithms used in pattern recognition and machine learning and their main task is to find patterns like clusters, principal components analysis, correlations or classification in a data samples. Informally, in Machine Learning the kernel function is regarded as a similarity function, where after computing the inner product among data samples a number is obtained that measures how similar two different samples are, but with the constraint that the produces matrix need to be positive semi-definite, for further details see 2.3.1. Where a relationship that is a non-linear in first sight, is a linear relationship in the features or inner product space. This kind of kernels has been studied in different situations and was the bases that support vector machines to become one of the favorite techniques to find patterns and classify them. Additionally, the methods of kernels are one of the most important approaches to deal with unstructured data such as trees, graphs, and text [1].

1.1 Problem Definition

Data-driven unsupervised learning is a paradigm that has been gaining the biggest momentum and establishing itself as a critical component of scientific discovery in domains ranging from astrophysics to social sciences and computational journalism. It manifests itself, in many forms, through modeling, machine learning, data mining, pattern recognition, data analytics, anomaly detection, and visualization. Because this paradigm is fueled by real data as a source of discovery, and because real data is inherently noisy and imperfect, the robustness of the techniques employed within this paradigm is of paramount importance.
The general problem addressed by this thesis is to evaluate the robustness of kernel-based algorithms, and the design of a new robust kernel to non-supervised learning algorithms,

that can deal with noise and outliers. Robustness, in this context, means that the techniques should be affected by noise as little as possible, so that their results do not deteriorate with added contamination. Robustness has been studied mainly in statistical estimation (e.g. the median is more robust than the mean) and various robust extensions of the Maximum Likelihood Estimation (MLE) for Gaussian and other known distributions, such as the ϵ-contamination model, M-estimation, and robust clustering. Robust statistics has been used in the machine learning field almost for ten years. However, as far as we are concerned, none of these approaches showed connections among methods of kernels and robust statistics. Thus, a theoretical analysis of the robust behavior of different kernels is needed. A satisfactory solution to this general challenge requires answering some particular research questions:

- What does a kernel robust do?
- Is it possible to develop a new kernel approach that behaves in a robust way?

This thesis presents a unified view of distinct areas that have had an immense impact on data analysis and machine learning, and identify those methods that, although not conceived to be robust, can be shown to possess some naturally built-in robustness mechanisms. This work seeks to achieve a more profound understanding of the robustness of those methods that do have some robustness. Thus, our work results in a deeper understanding of the mechanisms of robustness in several crucial areas of machine learning. All critical applications with broad impact on society (such as Information Retrieval, Visualization, Personalization, Anomaly Detection, etc) that benefit from these techniques will naturally benefit from this deeper understanding. All the techniques and software developed are published and made available to the public.
This thesis present (1) a unified view between distinct areas (robust estimation and kernel-based clustering) that have had an immense impact on data analysis and machine learning and (2) methods which, although not conceived to be robust, can be shown to have some classical statistical robustness mechanisms naturally built-in. This thesis tries to fill this gap by asking the question as to whether these methods are robust, and what are the limits on their robustness. It follows rigorous standards and methodologies of a well-established area, Robust Statistics has pioneered the study of robust estimation, yet to date it has not intermingled much with the machine learning methods to be studied.

1.2 Objectives

The main objective of this thesis is to perform a systematic evaluation of the robustness of some unsupervised learning algorithms based on kernels. To achieve this objective, the work has been divided into the following specific objectives:

- To perform a theoretical analysis of robust estimation based on a feature space induced by a kernel.

- To propose new kernels based on robust estimators.

- To perform a systematic quantitative evaluation of the impact of using robust kernel estimation for unsupervised learning.

1.3 Results and Contributions

The results and contributions of this work can be summarized as follows:

- This thesis presents a formal proof that doing mean estimation in the feature space with some kinds of kernel is like doing robust mean estimation in the data space, which can be found in chapter 3. The minimum distance among data points and the mean in the feature space is approximate without explicitly calculate the mean in the feature space ($\mu = \frac{1}{n}\sum_{i=1}^{n}\Phi(d_i)$). This new approximate optimization function only depends on mapped data points in the feature space.

- This thesis presents a formal proof that when performing mean estimation with the Gaussian kernel in the feature space is like doing robust mean estimation with the robust Welsch's estimator in the data space, which can be found in chapter 3.

- The definition of three new robust kernels is proposed in chapter 5. The construction of these new kernels uses the definition of approximate pre-image to shows their relationship with robust estimators like Huber's, Tukey's and Andrews' estimators.

Most of the code is available in web repositories:

- Software for this thesis is available in:

  https://bitbucket.org/jdbermeol/semantic_methods_toolkit/src

1.4 Document Structure

The present thesis is divided into 7 chapters. Chapter 1 presents the problem statement, objectives, results and contributions, and document structure. Chapter 2 presents a brief review of the state-of-the-art divided into four parts; robust statistical estimator, clustering, kernels, and, robust statistics and robustness in unsupervised techniques. Chapter 3 presents a formal proof that doing mean estimation in the feature space with Gaussian kernel is like doing robust mean estimation with Welsch M-estimator in the data space. Chapter 4 presents two experiments where the Gaussian kernel is used in order to show the advantage of using a robust kernel in clustering. Chapter 5 define three new robust kernels with the ideas presented in chapter 3. Chapter 6 shows in an experiment a comparison among several state-of-the-art algorithms and the new kernels presented in chapter 3 and 5. Finally, Chapter 7 gives some concluding remarks and future work.

Chapter 2

Background and Related Work

2.1 Robust Statistical Estimators

Robust statistics has recently emerged as a family of theories and techniques for estimating the parameters of a parametric model while dealing with deviations from idealized assumptions [4, 5, 6, 7]. Examples of deviations include contamination of data by gross errors, rounding and grouping errors, and departure from an assumed sample distribution. Gross errors or outliers are data severely deviating from the pattern set by the majority of the data. This type of error usually occurs due to mistakes in copying or computation. They can also be due to part of the data not fitting the same model, as in the case of data with multiple clusters. Gross errors are often the most dangerous type of errors. In fact, a single outlier can completely spoil the Least Squares estimate, causing it to break down. Rounding and grouping errors result from the inherent inaccuracy in the collection and recording of data which is usually rounded, grouped, or even coarsely classified. Departure from an assumed model means that real data can deviate from the often assumed normal distribution. Departure from the normal distribution can manifest itself in many ways such as in the form of skewed (asymmetric) or longer tailed distributions. The most common and practical robust estimators are M and W-estimators. Other estimators include L, Least Trimmed Squares, and Reweighted Least Squares estimators. Below, we review only the first two estimators [6, 8].

2.1.1 M-estimators

The ordinary Least Squares (LS) method to estimate parameters is not robust because its objective function, $\sum_{j=1}^{N} d_j^2$, increases indefinitely with the residuals d_j between the j^{th} data point and the estimated fit, with N being the total number of data points in a data set. Hence, extreme outliers with arbitrarily large residuals can have an infinitely large influence on the resulting estimate. M-estimators [5] attempt to limit the influence of outliers by replacing the square of the residuals with a less rapidly increasing loss function

of the data value, x, and parameter estimate, t, $\rho(x;t)$. The M-estimate, $T(x_1, \cdots, x_N)$ for the function ρ and the sample $x_1, \cdots, x_N$, is the value that minimizes the following objective:

$$\min_t \quad \{J = \sum_{j=1}^{N} \rho(x_j;t)\}. \tag{2.1.1}$$

The optimal parameter, T, is determined by solving:

$$\frac{\partial J}{\partial t} = \sum_{j=1}^{N} \psi(x_j;t) = 0 \tag{2.1.2}$$

where, except for a multiplicative constant,

$$\psi(x_j;t) = \frac{\partial \rho(x_j;t)}{\partial t}. \tag{2.1.3}$$

When the M-estimator is equivariant, i. e., $T(x_1 + a, \cdots, x_N + a) = T(x_1, \cdots, x_N) + a$ for any real constant a, we can write ψ and ρ in terms of the residuals $x - t$. Also, in general, an auxiliary scale estimate, S is used to obtain the scaled residuals $r = \frac{x-t}{S}$. Hence, we can write:

$$\psi(r) = \psi\left(\frac{x-t}{S}\right) = \psi(x;t),$$

and

$$\rho(r) = \rho\left(\frac{x-t}{S}\right) = \rho(x;t).$$

The $\rho-$functions for some familiar M-estimators are listed in Table **2-1**. Note that LS can be considered an M-estimator, even though it is not a *robust* M-estimator. As seen in this table, M-estimators rely on both an accurate estimate of scale and a fixed tuning constant, c. Most M-estimators use a multiple of the Median of Absolute Deviations (MAD) as a scale estimate which implicitly assumes that the noise contamination rate is 50%. MAD is defined as follows:

$$MAD(x_i) = med_i\{|x_i - med_j(x_j)|\}$$

where med_i is the median value. The most common scale estimate used is $1.483 \times MAD$ where the 1.483 factor adjusts the scale for maximum efficiency when the data samples come from a Gaussian distribution.

2.1.2 W-estimators

W-estimators [3] represent an alternative form of M-estimators. Each W-estimator has a characteristic weight function, $w(.)$ that represents the importance of each sample in its

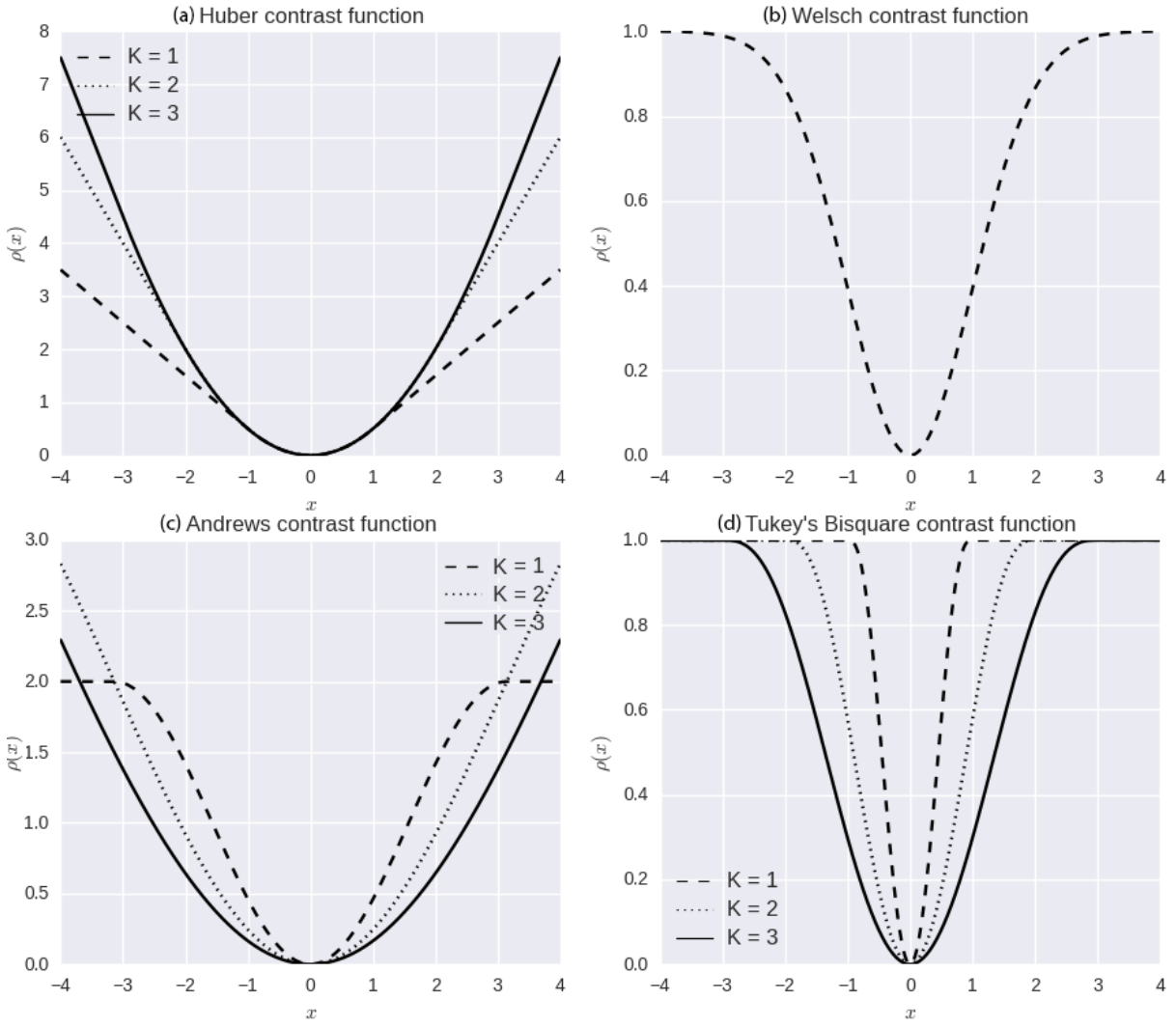

Figure **2-1**: Different examples of robust M-contrast functions using $c := 1$: (a) Huber contrast function for different k parameters, note that parameter k is chosen arbitrary and modifying the weight of the residuals 2.1.1 , (b) Welsch contrast function 2.1.1, (c) Andrews contrast function for different k parameters, note that a higher value of k decrease the power of the residual in the estimation of ρ 2.1.1, (d) Tukey's Bisqueare contrast function for different k parameters 2.1.1.

Table **2-1**: Different familiar M-estimators with ρ-contrast function, $\psi(r)$, $w(r)$, scale parameter k defines the breakpoint where the principal residual function is replaced with a less rapidly increasing loss function, *the range of* r defines the range for residual r where every residual function is defined, used scale is the most common value for the parameter k .

Type	$\rho(r)$	$\psi(r)$	w(r)	Range of r	used scale
L_2 (mean)	$\frac{1}{2}r^2$	r	1	$\mathbb{R}$	none
L_1 (median)	$\lvert r\rvert$	sgn(r)	$\frac{sgn(r)}{r}$	$\mathbb{R}$	none
Huber	$\frac{1}{2}r^2$ $k\lvert r\rvert - \frac{1}{2}k^2$	r k sgn(r)	1 $\frac{ksgn(r)}{r}$	$\lvert r\rvert \leq k$ $\lvert r\rvert > k$	MAD
Cauchy	$\frac{c^2}{2}log\left[1+(\frac{r}{c})^2\right]$	$\frac{r}{1+(\frac{r}{c})^2}$	$\frac{1}{1+(\frac{r}{c})^2}$	$\mathbb{R}$	MAD
Welsch	$\frac{c^2}{2}\left[1-exp(-(\frac{r}{c})^2)\right]$	$r\ exp(-(\frac{r}{c})^2)$	$exp(-(\frac{r}{c})^2)$	$\mathbb{R}$	MAD
Tukey's Biweight	$\frac{1}{6}\left[1-(1-\frac{r}{c})^2\right]^3$ $\frac{1}{6}$	$r(1-\left(\frac{r}{c}\right)^2)^2$ 0	$\left[1-\left(\frac{r}{c}\right)^2\right]^2$ 0	$\lvert r\rvert \leq k$ $\lvert r\rvert > k$	MAD
Andrews	$c\left[1-cos(\frac{r}{c})\right]$ $2c$	$sin(\frac{r}{c})$ 0		$\lvert r\rvert \leq k$ $\lvert r\rvert > k$	MAD

contribution to the estimation of T, which is related to the corresponding M-estimator as follows:

$$\psi(r) = w(r)\, r. \tag{2.1.4}$$

The optimal parameter is determined by solving:

$$\sum_{j=1}^{N} w(r_j)\, r_j = 0, \tag{2.1.5}$$

which is similar to the equations for a "weighted LS" regression problem. W-estimators offer a convenient and simple iterative computational procedure for M-estimators, where the W-estimator equations in the current iteration are solved by fixing the weight values, $w(r_j)$, to those of the previous iteration. The resulting procedure is called the Iterative Reweighted Least Squares (IRLS). The IRLS relies on an accurate and prefixed scale estimate for the definition of its weights. The most common scale estimate used is $1.483 \times MAD$. The ρ, ψ, and w functions for some familiar M- and W-estimators are listed in Table **2-1**.

2.1.3 Resistance Properties of M-estimators

An estimator is resistant if a small number of gross errors or any number of small rounding and grouping errors have only a limited effect on the resulting estimate [3]. As seen below,

most of the resistance properties of an M-estimator can be inferred from the shape of its Influence Curve.

The Influence Curve The Influence Curve (IC) tells us how an infinitesimal proportion of contamination affects the estimate in large samples. Formally [3], the IC gives a quantitative expression of the change in the estimate that results from perturbing the samples' underlying distribution, F, by a point mass at sample location x. For an estimator given by the functional $T(F)$ and defined by the ψ function $\psi(u)$, the IC at F_0 is:

$$IC\left(x; F_0, T\right) = \lim_{\epsilon \to 0} \frac{T\left[(1-\epsilon)F_0 + \epsilon\delta_x\right] - T\left(F_0\right)}{\epsilon},$$

where δ_x is a point mass perturbation at x and F_0 is the underlying or empirical distribution. The IC can be shown to reduce to:

$$IC\left(x; F_0, T\right) = \frac{S\left(F_0\right)\psi\left[\frac{x-T(F_0)}{S(F_0)}\right]}{\int \psi'\left[\frac{x-T(F_0)}{S(F_0)}\right] dF_0(x)}$$

where $S\left(F\right)$ is the auxiliary scale estimate. The IC can be further shown to be of the form:

$$IC = (\text{const})\psi(u). \tag{2.1.6}$$

Hence, the shape of the IC depends only on the shape of the ψ function, not the data distribution.
The Breakdown (BD) bound or point [5] is the largest possible fraction of observations for which there is a bound on the change of the estimate when that fraction of the sample is altered without restrictions. M-estimators of location with an odd $\psi()$ function have a BD bound close to 50%, provided that the auxiliary scale estimator has equal or better BD bound [9].

Gross Error Sensitivity The Gross Error Sensitivity (g.e.s.) expresses asymptotically the maximum effect a contaminated observation can have on the estimator. It is the maximum absolute value of the IC. The asymptotic bias of an estimator, defined as the maximum effect of the contamination of a given distribution with a proportion ϵ from an outlying distribution, is given by ϵ.(g.e.s.). Unfortunately, it was reported in [3] that - in general - poor g.e.s. corresponds to higher Gaussian efficiency, and vice versa.

Local Shift Sensitivity The local Shift Sensitivity (l.s.s.) measures the effect of the removal of a mass ϵ at y and its reintroduction at x. Therefore, it measures the effect of rounding and grouping errors on an estimator. For highest resistance, it is required that the l.s.s. be bounded. For a continuous and differentiable IC, l.s.s. is given by the

maximum absolute value of the slope of IC at any point. In [3], it was reported that - in general - a lower (hence better) l.s.s. corresponds to higher Gaussian efficiency.

Winsor's Principle Winsor's principle [3] states that all distributions are normal in the middle. Hence, the $\psi-$function of M-estimators should resemble the one that is optimal for Gaussian data in the middle. Since the Maximum Likelihood estimate for Gaussian data is the mean which has a linear $\psi-$ function, it is desired that $\psi(u) \approx ku$ for small $|u|$, where k is a nonzero constant. In general, a $\psi-$ function that is linear in the middle results in better efficiency at the Gaussian distribution.

2.2 Clustering

Clustering is the most important problem in non-supervised learning. In general, the goal of a clustering algorithm is to find groups in a set of data samples. The groups found are expected to be homogeneous, i.e. to contain similar samples. Depending on the type of clustering algorithm, this could be accomplished in different ways. In this study, we focus on a particular type of clustering algorithm which is based on the optimization of an objective function.
One of the most popular clustering algorithms is k-means. The algorithm represents the clusters by a set of centroids $M = \{m_1, \cdots, m_k\}$ that minimizes the following function:

$$SSE = \min_{M} \sum_{i=1}^{k} \sum_{x \in C_i} (x - m_i)^2 \tag{2.2.1}$$

where $\{C_1, \ldots, C_k\}$ is a disjunct partition of the input data set X, such that $X = \bigcup_{i=1}^{k} C_i$. The minimization is accomplished by an optimization process that iteratively reassigns data points to clusters refining the centroid estimations.

2.2.1 Non-negative Matrix Factorization

The general problem of matrix factorization is to decompose matrix X into two matrix factors A and B :

$$X_{n \times l} = A_{n \times k} B_{k \times l} \tag{2.2.2}$$

This could be accomplished by different methods including singular value decomposition (SVD), non-negative matrix factorization (NMF), and probabilistic latent semantic analysis, among others [10]. The factorization problem can be seen as a problem of optimization:

Algorithm 2.2 k-means Clustering

1: $D = \{d_1, \cdots, d_n \subseteq X$: input data set
2: $1 < m \in \mathbb{N}$: number of clusters
3: Initialize the set of centers $\Theta_0 = \{\mu_1^0, \cdots, \mu_m^0\}, \mu_i \in \mathcal{F}$
4: $l \leftarrow 0$
5: repeat
6: for all $j \in \{1, \cdots, m\} : C_j^{l+1} \leftarrow \emptyset$
7: for all $d_i \in D$:
8: $j \leftarrow \arg\min_j \left\|d_i - \mu_j^l\right\| \triangleright$ point-cluster assignment
9: $C_j^{l+1} \leftarrow C_j^{l+1} \cup \{d_i\}$
10: for all $j \in \{1, \cdots, m\}$:
11: $\mu_j^{l+1} \leftarrow \frac{1}{n} \sum_{d_i \in C_j^{l+1}} d_i \triangleright$ cluster centroid estimation
12: $l \leftarrow l + 1$
13: until $\Theta^l = \Theta^{l-1}$

$$\min_{A,B} d(X, AB) \tag{2.2.3}$$

where $d(,)$ is a distance or divergence function and the problem could have different types of restrictions. For instance, if $d(,)$ is the Euclidean Distance and there are no restrictions, the problem is solved by finding the SVD; if X, A and B are restricted to be positive, then the problem is solved by NMF.

This type of factorization may be used to perform clustering. The input data points are the columns of X (l n-dimensional data points). The columns of A correspond to the coordinates of the centroids. The columns of B indicate to which cluster each sample belongs, specifically if x_j belongs to C_i, then $B_{i,j} = 1$, otherwise $B_{i,j} = 0$. With this interpretation, the function in (2.2.1) is equivalent to the function in (2.2.3) with Euclidean distance. An important advantage of this approach is that values in the matrix B are not required to be binary; in fact, they can be continuous values. These values can be interpreted as soft membership values of data samples to clusters, i.e. NMF can produce a soft clustering of the input data [10].

NMF has been shown to be equivalent to other non-supervised learning methods such as probabilistic latent semantic analysis and kernel k-means. Also, there are different versions of NMF which impose new restrictions or weaken some of its restrictions. For instance, semi-NMF allows negative values in matrices X and A in (2.2.3), convex-NMF impose that A must be a combination of the data input, $A = XW$ [11].

Let X=$(\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n})$ be the input positive data matrix containing n samples as column vectors, NMF attempts to factorize X as:

$$X_+ \approx F_+ G_+^T$$

Algorithm 2.3 NMF Clustering

$D = \{d_1, \cdots, d_n \subseteq X$: input data set
$1 < m \in \mathbb{N}$: number of clusters
ϵ : reconstruction error
1: Initialize W and G.
2: $l \leftarrow 0$
repeat
update $G_{kn} \leftarrow G_{kn} \frac{(F^T X)_{kn}}{(F^T FG)_{kn}}$
update$F_{pk} \leftarrow F_{pk} \frac{(XG^T)_{pk}}{(FGG^T)_{pk}}$
set $X_{calculate} = FG^T$
until $\|X - X_{calculate}\|^2 > \epsilon$

where positive symbols means $X, F, G > 0$ and $X \in \mathbb{R}^{p \times n}$, $F \in \mathbb{R}^{p \times k}$ and $G \in \mathbb{R}^{n \times k}$. Having said this, NMF solves the following optimization problem:

$$\arg\min_{W,G} ||X - FG^T||_F^2 \qquad F, G \geq 0$$

where $||\cdot||_F$ is the Frobenius norm and it is defined as $||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$. This problem is non-convex in both matrices F and G. If we set one then optimize the other, the problem becomes convex. Lee and Seung propose the following multiplicative update rules:

$$G'_{kn} \leftarrow G_{kn} \frac{(F^T X)_{kn}}{(F^T FG)_{kn}} \qquad F'_{pk} \leftarrow F_{pk} \frac{(XG^T)_{pk}}{(FGG^T)_{pk}}$$

2.2.2 Convex NMF

Convex nonnegative matrix factorization (Convex-NMF) method [11] was developed by Ding and Li in *Convex and Semi-Nonnegative Matrix Factorizations*. The CNMF is also a nonnegative matrix factorization method that attempts to find a subspace where the data lies, but this time the basis vector F is restricted to the space spanned by X space. This occurs because matrix F can be anything in space. There are no restrictions in the factorization of X that we obtain, i.e. the elements of A that do not have any restrictions and can be any value.

$$X_{n \times l} \approx A_{n \times k} G^T_{k \times l} \tag{2.2.4}$$

The objective underlies the use of convex NMF is to restrict the values of A in such a way as for the values of A to be a linear combination of points of X, that is to say:

$$A = w_1x_1 + w_2x_2 + ... + w_nx_n$$

Thus, we can define A as a weighted sum of certain data points. Therefore, it can be observed that we can define CNMF as:

$$X_{n\times l} \approx X_{n\times l}W_{l\times k}G^T_{k\times l} \tag{2.2.5}$$

It is worth noting that one of the interesting properties when doing this type of factorization is that vectors W and G are sparse, i.e. the majority of the matrices have zeros. Upon a revision of Chris Ding's paper, it is found that the matrix G also captures information with regard to the centroids wherein these data fall [11]. Ding performs optimization in 2.2.5, based on an iterative rule algorithm, i.e. Ding leaves one of the variables as a constant whilst optimizing the other. This process is repeated until Frobernius' difference between variable X and its approximation $X_{n\times l}W_{l\times k}G^T_{k\times l}$ is less than a threshold.

$$||X_{n\times l} - X_{n\times l}W_{l\times k}G^T_{k\times l}||^2_F < \epsilon$$

Let X=$(\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n})$ be the input positive data matrix containing n samples as column vectors, CNMF attempts to factorize X as:

$$X_{\pm} \approx X_{\pm}W_{+}G^T_{+}$$

Where $X \in \mathbb{R}^{p\times n}$, $F \in \mathbb{R}^{p\times k}$ and $G \in \mathbb{R}^{n\times k}$. Having said that, CNMF solve the following optimization problem:

$$\arg\min_{W,G} ||X - XWG^T||^2_F$$

We can solve this problem with the following update rules:

$$G'_{kn} \leftarrow G_{kn}\sqrt{\frac{[(X^TX)^+W]_{ik}+[GW^T(X^TX)^-W]_{ik}}{[(X^TX)^-W]_{ik}+[GW^T(X^TX)^+W]_{ik}}}$$

$$W'_{pk} \leftarrow W_{pk}\sqrt{\frac{[(X^TX)^+G]_{ik}+[(X^TX)^-WG^TG]_{ik}}{[(X^TX)^-G]_{ik}+[(X^TX)^+WG^TG]_{ik}}}$$

2.2.3 Projective NMF

Projective NMF [12] minimizes the reconstruction error of the original matrix according to two possible measures of dissimilarity; Frobenius matrix norm and the modified Kullback-Leibler distance. The Kullback-Leibler distance is defined as: $D_I(X||\hat{X}) = \sum_{ij}(X_{ij}log\frac{X_{ij}}{\hat{X}_{ij}} - X_{ij} + \hat{X}_{ij})$.When the reconstruction error is minimized, two constraints on the optimization are added; the matrix needs to be positive and at a low rank. In this case, both measures are minimized using multiplicative rules [12].

Algorithm 2.4 CNMF Clustering

1: $D = \{d_1, \cdots, d_n \subseteq X$: input data set
2: $1 < m \in \mathbb{N}$: number of clusters
3: ϵ : reconstruction error
4: 1: Initialize W and G.
5: $l \leftarrow 0$
6: repeat
7: update $G_{kn} \leftarrow G_{kn} \sqrt{\frac{[(X^TX)^+W]_{ik} + [GW^T(X^TX)^-W]_{ik}}{[(X^TX)^-W]_{ik} + [GW^T(X^TX)^+W]_{ik}}}$
8: update $W_{pk} \leftarrow W_{pk} \sqrt{\frac{[(X^TX)^+G]_{ik} + [(X^TX)^-WG^TG]_{ik}}{[(X^TX)^-G]_{ik} + [(X^TX)^+WG^TG]_{ik}}}$
9: set $X_{calculate} = FG^T$
10: until $\|X - X_{calculate}\|^2 > \epsilon$

Let X=($\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}$) be the input positive data matrix containing n samples as column vectors, PNMF attempts to factor X as:

$$X_+ \approx W_+ W_+^T X^T$$

where positive symbols means $W > 0$ and $X \in \mathbb{R}^{p \times n}$ and $W \in \mathbb{R}^{p \times k}$. Having said this, PNMF solves the following optimization problem:

$$\arg\min_{W,G} ||X - WH||_F^2$$

$$s.t. W \geq 0, H = W^T X$$

Where $|| \cdot ||_F$ is the Frobenius norm. Zhirong Yang proposed the following multiplicative update rules:

$$W'_{pn} \leftarrow W_{pn} \frac{(XX^TW)_{pn}}{(WW^TXX^TW)_{pn}}$$

2.2.4 Orthogonal NMF (ONMF)

Orthogonal Nonnegative matrix factorization (ONMF) method [13] was developed by Ding and Li in *Orthogonal Nonnegative Matrix Tri-factorizations for Clustering*. ONMF is also a matrix factorization method that attempts to find a subspace where the data lies, but this time the basis vector F and G are restricted to be orthogonal, i.e. $F^TF = I$ and $G^TG = I$. This occurs because the matrix F and G can be anything in the space in general. With this restriction, we achieve (1) uniqueness from the solution and (2) the interpretation of clustering. The problem is that doing the reconstruction matrix as FG^T makes the optimization very restricted and gets a poor matrix low-rank approximation [13]. In order to solve this problem, a new parameter S is added to the optimization

problem: FSG^T, where this new parameter S absorbs the different scales of the matrices F and G. The optimization problem is as follows:

$$\arg\min_{W,G} ||X - FSG^T||_F^2 \qquad F, G \geq 0$$

$$s.t. F^T F = I, G^T G = I$$

where $X \in \mathbb{R}^{p \times n}$, $F \in \mathbb{R}^{p \times k}$,$S \in \mathbb{R}^{k \times l}$ and $G \in \mathbb{R}^{n \times l}$. Ding and Li proposed the following multiplicative update rules:

$$G'_{jk} \leftarrow G_{jk} \sqrt{\frac{(X^T FS)_{jk}}{(GG^T X^T FS)_{jk}}}$$

$$F'_{ik} \leftarrow F_{ik} \sqrt{\frac{(XGS^T)_{jk}}{(FF^T XGS^T)_{jk}}}$$

$$G'_{ik} \leftarrow G_{ik} \sqrt{\frac{(F^T XG)_{ik}}{(F^T FSG^T G)_{ik}}}$$

2.2.5 Normalized Cuts (NCUT)

Normalized cuts (NCUT) *[14]* try to take an arbitrary set of data and represent them as a weighted undirected graph G = (V, E), where each node represents a data point in the feature space, and the edge is a function of similarity between data. With this graph, we can partition the nodes into a disjoint set $v = v_1, v_2, \cdots, v_n$, where the measure of similarity between each pair of nodes of one of this sets v_i is high and the other is low between nodes of two different sets v_i, v_j.
The degree of similarity between two sets can be calculated as the total weight of vertices that have been removed. In graph theory, this is called the cut and it is defined as follows:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

Although it is an np-complete problem, an approximate discrete solution can be found efficiently. When using the cut for clustering tasks, k-subgroups can be found that have some dissimilarities between them. A problem with using the cut is that the measure favors the creation of isolated groups of a single element, to counteract this unnatural behavior, the normalized cut is used and it is defined as follows:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

where $assoc(A, V) = \sum_{u \in A, t \in v} w(u, t)$ is the total connection from the nodes in A to all the nodes in the graph and $assoc(B, V)$ is similarly defined.

Algorithm 2.5 Normalized cuts

1: *Given an image or image sequence, set up a weighted graph $G(V, E)$ and set the weight on the edge connecting two nodes to be a measure of the similarity between the two nodes.*
2: *Solve $(D - W)x = \lambda Dx$ for eigenvectors with the smallest eigenvalues.*
3: *Use the eigenvector with the second smallest eigenvalue to bipartition the graph.*
4: *Decide if the current partition should be subdivided and recursively repartition the segmented parts if necessary.*

Given a partition of nodes of a graph, V, into two sets A and let x be an $N = |V|$ dimensional indicator vector, $x_i = 1$ if node i is in A and 1, otherwise. Let $d(i) = \sum_j w(i, j)$ be the total connection from node i to all other nodes. Let D be an $N \times N$ diagonal matrix with d on its diagonal called the adjacency matrix, W be an $N \times N$ symmetrical matrix with $W(i, j) = w_{ij}$ called the degree matrix and after some mathematical operations, the optimization problems becomes as follows:

$$min_x Ncut(x) = min_y \frac{y^T(D - W)y}{y^T Dy}$$

with the condition $y(i) \in \{1, -b\}$ and $y^T D1 = 0$. This optimization problem is minimized by solving the generalized eigenvalue system [14],

$$(D - W)x = \lambda Dx$$

2.2.6 Nonnegative Spectral Cut (NSC)

The principal idea in a Nonnegative Spectral Cut (NSC) [15] is to impose a nonnegative restriction in the optimization of a Normalized Cut:

$$max_{H^T DH=I, H \geq 0} Tr(H^T WH)$$

where the following multiplicative update rules are used, in order to impose the nonnegative restriction to the eigenvector:

$$H_{ij} \leftarrow H_{ij} \sqrt{\frac{(WH)_{ij}}{(DH\alpha)_{ij}}}$$

The initialization is as follows: *An approximate solution to the normalized cut problem can be obtained by K-means clustering in an eigenspace embedding. It use such a clustering to initialize H. Specifically, it is compute the K principal eigenvectors of $D^{-1/2}WD^{-1/2}$, (2) normalize each embedded point to the unit sphere, and (3) perform K-means on the normalized points. This gives H0. then initialize the update rule Eq. (2.2.6) with H0 + 0.2 and iterate until convergence* [15].

2.2.7 Left-Stochastic Matrix Factorization (LSD)

The principal idea in the left-stochastic matrix factorization (LSD) [16] is to produce a probability matrix $P \in \mathbb{R}^{n \times k}$ where each column i indicates a probability that each row sample j belongs to the i cluster. This method deals with the similarity matrix that is a more general matrix compared to the kernel matrix where the similarity matrix does not need to be positive semi-definite (PSD) and it can be an indefinite kernel matrix. For further details, see 2.3.1 and 2.3.5.
The optimization problem is given a similarity matrix K and the algorithm finds a probability matrix P^* that solves:

$$argmin_{c \in \mathbb{R}_+, P \geq 0, P^T 1_k = 1_n} ||cK - P^T P||_F^2$$

where c is given by $c^* = \frac{||(MM^T)^{-1} M 1_n||_2^2}{k}$ and $K = M^T M$ is any decomposition of K [16]. The algorithms in order to solve the last equation is as follows:

Algorithm 2.6 Left-Stochastic Matrix Factorization (LSD)

1: Scale the similarity matrix K by c^* given to produce $K_0 = c^* K$.
2: Factor $K_0 = M^T M$, for some $M \in \mathbb{R}^{k \times n}$.
3: Rotate M to form $Q \in \mathbb{R}^{k \times n}$, whose column vectors all lie in the hyperplane that contains the probability simplex.
4: Rotate Q about the vector $\hat{u} = 1_k[1, ..., 1]^T$ to form the left-stochastic solution P^* , whose column vectors all lie in the probability simplex.

2.3 Learning with Kernels

Learning with kernels is a strategy where it is desired to map the data to a feature space, so that, in the new space, the patterns can be discovered as a linear relationship. The process for mapping and finding new patterns can be done in two steps. The first step is the kernel function, which defines the mapping implicitly. This mapping is defined according to the type of data and the knowledge domain. The second step is the algorithm, which work with any type of kernels (general purpose), and must be robust, statistically stable and efficient.
In [1] the author proposes four key aspects as the main ingredients for finding patterns with a kernel methodology, which are as follows:

1. *Data items are embedded into a vector space called the feature space.*
2. *Linear relations are sought among the images of the data items in the feature space.*
3. *The algorithms are implemented in such a way that the coordinates of the embedded points are not needed, only their pairwise inner product.*

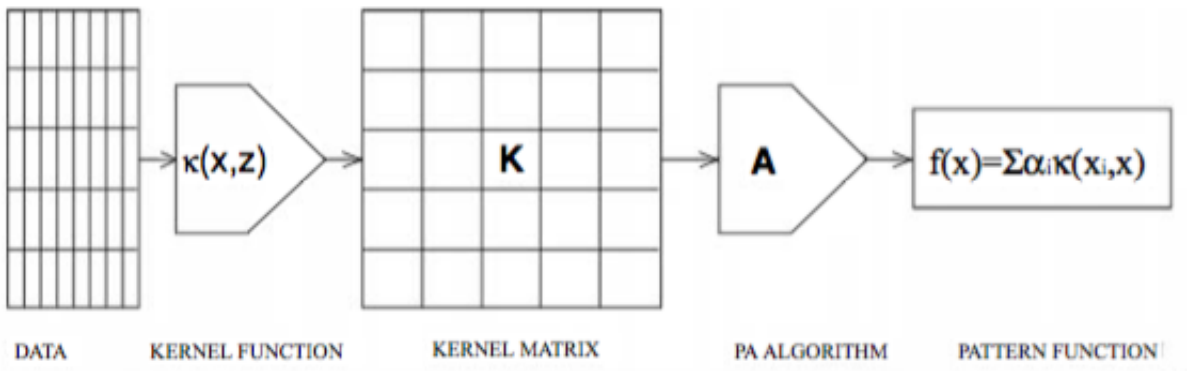

Figure **2-2**: . The stages involved in the application of kernel methods. First, we have a data in a matrix form where each rows represents a sample. Second, we have the kernel function that is a inner product among samples. Third, we have a kernel matrix that must to be semi definitive positive in order to satisfies Mercer theorem's2.3.1. Fourth, a pattern algorithm use the kernel matrix as the input data in order to generate knowledge. This figure was taken from [1].

4. *The pairwise inner product products can be computed efficiently from the original data items using a kernel function.*

2.3.1 Kernels

Definition. A kernel is a function k that for all $\mathbf{x}$, $\mathbf{z} \in$ X satisfies

$$k(x, z) =< \Phi(x), \Phi(z) >$$

where Φ is a mapping from X to an inner product or feature space F

$$\Phi : x \to \Phi(x) \in F$$

Kernel methodology emerges like an algorithmic procedure adapted to use only pairwise inner products. The main idea behind this procedure is to use a learning algorithm that work with any kernel and hence in any domain. The kernel component is data specific, but can combined with different algorithms to solve the full range of tasks that we will consider. In fig.**2-2** shows the stages involved in the application of kernel methods. The data is processed using a kernel function to create a kernel matrix. This kernel matrix is used by a pattern analysis algorithm to produce a pattern function [1].

Theorem 2.3.1. *Characterization of kernels. A function*

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

which is either continuous or has a finite domain, can be decomposed

$$k(x,z) =< \Phi(x), \Phi(z) >$$

into a feature map Φ *into a Hilbert space* F *applied to both its arguments followed by the evaluation of the inner product in* F *if and only if it satisfies the finitely positive semi-definite property*

Definition 2.3.2. A gram matrix K is a real symmetric NxN matrix, and it is defined as a positive semi-definite matrix if for any nonzero coefficient vector c = $[c_1, c_2, \ldots, c_n]$, its associated quadratic form is positive, i.e.

$$< f, f >= \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_k k(x_i, x_j) = \alpha' K \alpha \geq 0$$

The semi-positive definite matrix guarantees the existence of a mapped feature space. Semi-Positive definite kernels (sPDK) have some interesting properties, which make it desirable for a variety of Machine Learning tasks. sPDK are symmetric given by the inner product symmetric property:

$$k(i,j) =< \Phi(x), \Phi(y) >=< \Phi(y), \Phi(x) >= k(j,i)$$

Definition. The reproducing property of the kernel is defined as:

$$< f, k(x,\cdot) >= \sum_{i=1}^{N} \alpha_i k(x_i, x) = f(x)$$

Theorem. *(Mercer) Let* X *be a compact subset of* $\mathbb{R}^n$*. Suppose* k *is a continuous symmetric function such that the integral operator* $T_k : L_2(X) \to L_2(X)$

$$(T_k f)(\cdot) = \int_X k(\cdot, x) f(x) dx$$

is positive, that is

$$\int_X k(x,z) f(x) f(z) dx dz \geq 0$$

for all $f \in L_2(X)$*. Then we can expand* $k(x,z)$ *in a uniformly convergent series in terms of* function Φ, satisfying $< \Phi_i, \Phi_j >= \delta_{ij}$

$$k(x,z) = \sum_{j=1}^{\infty} \Phi_j(x) \Phi_j(z)$$

New semi positive definite kernels (sPDK) can be constructed because they have some interesting algebra properties. For instance, if $k_1$and $k_2$are two PDK, and $c_1$and c_2 are two positive real numbers, then

$$c_1 k_1(x_i, x_j) + c_2 k_2(x_i, x_j) = k(x_i, x_j)$$

is a kernel. When we multiply two sPDK, the resulting kernel is a sPDK [17]:

$$k_1(x_i, x_j) k_2(x_i, x_j) = k(x_i, x_j)$$

Definition. An isotropic kernel is a kernel which depends only on the pairwise distance of the data samples

$$k(x_i, x_j) = f(||x_i - x_j||)$$

where $||\cdot||$ is any suitable norm, usually the L_2-norm. Intuitively, this means that the direction of the deviation is not important. For instance, in a two dimensions spaces, a change in variable x_1 is equally important to a change in variable x_2, which is of course often a too strong assumption. Isotropic kernel are rotationally (and also translationally) invariant, i.e. their level curves are cycles and they can be expressed as unitary variable functions [18].

Definition. A radial basis function kernel (RBF Kernel) is defined as follows:

$$k(x, y) =< \Phi(x), \Phi(y) >= e^{\frac{||x-y||}{2\sigma^2}}$$

where σ is a parameter [17]. This kernel is an isotropic kernel.
Many authors have modified existing algorithms and added the version with kernels. One of such examples is the kernel version of k-mean, that is kernel-k-means. This new approach is defined only on terms of inner products, using the kernel trick, i.e. the full high dimensionality of the embedded map is not computed explicitly. Instead, we compute only the inner product of pairs for samples [19, 20].

2.3.2 Kernel K-means

Kernel k-means (KKM) is a generalization of the k-means clustering algorithm. Basically, KKM performs k-means in a feature space implicitly defined by a kernel. Let $\Phi : X \to F$ be mapping from an input space X to a Hilbert space, called the feature space, F. The dot product of X in F is calculated by a kernel function $k : X \times X \to \mathbb{R}$, i.e. $k(x, y =< \Phi(x), \Phi(y) >$ for all $x, y \in X$. The KKM algorithms is as follows:
The most important characteristic of KKM is that it is possible to carry it out without explicitly use the mapping Φ. This is done using the *kernel trick*:

Algorithm 2.7 Kernel k-means Clustering

1: $D = \{d_1, \cdots, d_n \subseteq X$: input data set
2: $1 < m \in \mathbb{N}$: number of clusters
3: Initialize the set of centers $\Theta_0 = \{\mu_1^0, \cdots, \mu_m^0\}, \mu_i \in \mathcal{F}$
4: $l \leftarrow 0$
5: repeat
6: for all $j \in \{1, \cdots, m\} : C_j^{l+1} \leftarrow \emptyset$
7: for all $d_i \in D$:
8: $j \leftarrow \arg\min_j \left\|\Phi(d_i) - \mu_j^l\right\|$ ▷ point-cluster assignment
9: $C_j^{l+1} \leftarrow C_j^{l+1} \cup \{d_i\}$
10: for all $j \in \{1, \cdots, m\}$:
11: $\mu_j^{l+1} \leftarrow \frac{1}{n} \sum_{d_i \in C_j^{l+1}} \Phi(d_i)$▷ cluster centroid estimation
12: $l \leftarrow l + 1$
13: until $\Theta^l = \Theta^{l-1}$

$$\begin{aligned}\left\|\Phi(d_i) - \mu_j^l\right\|^2 &=< \Phi(d_i) - \mu_j^l, \Phi(d_i) - \mu_j^l > \\ &=< \Phi(d_i), \Phi(d_i) > + < \mu_j^l, \mu_j^l > -2 < \Phi(d_i), \mu_j^l > \\ &= k(d_i, d_i) + \frac{1}{|C_j^l|^2} \sum_{d_p \in C_j^l} \sum_{d_q \in C_j^l} k(d_p, d_q) - \frac{2}{|C_j^l|} \sum_{d_q \in C_j^l} k(d_i, d_q)\end{aligned}$$

The KKM algorithms attempts to solve the following optimization problem:

$$\arg\min_c \sum_{C_j \in C} \sum_{d_i \in C_j} ||\Phi(d_i) - \mu_j||^2$$

where $\mu_j = \frac{1}{n} \sum_{d_i \in C_j^{l+1}} \Phi(d_i)$

2.3.3 Kernel-Convex NMF

Convex NMF can be generalized to utilize a function of Kernels, instead of just the data, thereby gaining the possibility of conducting optimization within an area of great dimensionality [11]. The first thing to be done is mapping the values of $x_i \rightarrow \phi(x_i)$, where $\phi(x_i)$ is a non – linear function of x_i. In this case, the factorization of 2.2.5 becomes:

$$\phi(X_{n\times l}) \approx \phi(X_{n\times l}) W_{l\times k} B_{k\times l} \tag{2.3.1}$$

The interesting thing that happens as we conduct this non - linear transformation, is that by performing all necessary changes it is found that target function minimization optimization only depends on the kernel function $< \phi(X), \phi(x) >$, i.e.

$$||\phi(x) - \phi(x)WB||^2 = Tr(I - B^T W^T) < \phi(X), \phi(x) > (I - B^T W^T)$$

Where Tr is the trace of the corresponding matrix, i.e.

$$Tr(A) = a_{11} + a_{22} + ... + a_n \tag{2.3.2}$$

2.3.4 Pre-image of RBF Kernel

The cluster centroids estimated by the kernel versions are in the feature space and correspond to the points $C_j = \frac{1}{n} \sum_{x_i \in C_j} \Phi(x_i)$. However, we are interested in the pre-image in the original space of these centroids, i.e. points $\hat{C}_j$ such that $\Phi(\hat{C}_j) = C_j$. However, it is possible that a exact pre-image may not even exist, so we look for the $\hat{C}_j$ that minimizes the following objective function: $\min_{\hat{C}_j} \left\| \hat{C}_j - C_j \right\|^2$. According to [21], the optimum C_j can be found by iterating the following fixed-point formula:

$$\hat{C}_j^{t+1} = \frac{\sum_{i=1}^{N} \exp\left(\frac{-||\hat{C}_j^t - x_i||)}{s}\right) x_i}{\sum_{i=1}^{N} \exp\left(\frac{-||\hat{C}_j^t - x_i||}{s}\right)} \tag{2.3.3}$$

2.3.5 Indefinite Kernels and Reproducing Krein Spaces

Indefinite kernels are those kernels that do not satisfy Mercer's theorem. Such a kernels induce an associated functional spaces called Reproducing Kernel Krein Spaces (RKKS). RKKS are a generalization of Reproducing Kernel Hilbert spaces. The key difference between indefinite kernels and positive definite kernels is that the inner product among samples could be either positive or negative such that the eigenvalues associated with the kernel matrix M are a mix of positive and negative values, i.e. $zMz^T \geq 0$ for every non zero column vector z [22].
First, we may define a positive inner product that is positive, if $\forall f \in k$ we have $< f, f >_k \geq 0$ and a negative, if $\forall f \in k$ we have $< f, f >_k \leq 0$. An indefinite kernel is defined as a kernel that is neither positive nor negative. Second, a Krein Spaces ($\mathcal{K}$) is a spaces where exist two Hilbert spaces $\mathcal{H}_+$ and $\mathcal{H}_-$spanning $\mathcal{K}$ such that:

- f can be decomposed into $f = f_+ - f_-$ where $f_+ \in H_+$and $f_- \in H_-$
- $\forall f, g \in \mathcal{K}$ the inner product of $< f, g >_{\mathcal{K}}$ can be decomposed in $< f, g >_{\mathcal{K}} = < f_+, g_+ >_{\mathcal{H}_+} - < f_-, g_- >_{\mathcal{H}_-}$

Third, the decomposition of a Krein Spaces into the Hilbert spaces defines an associated Hilbert Space:

$$\bar{\mathcal{K}} = \mathcal{H}_+ \oplus \mathcal{H}_- =< f_+, g_+ >_{\mathcal{H}_+} + < f_-, g_- >_{\mathcal{H}_-}$$

Now, we can define the Reproducing Kernel Krein spaces. Let $\mathcal{K}$ be a RKKS with $\mathcal{K} =< f_+, g_+ >_{\mathcal{H}_+} - < f_-, g_- >_{\mathcal{H}_-}$ then

1. H_+ and H_- are Reproducing Kernel Hilbert Spaces with kernels k_+ and k_-.
2. $k = k_+ - k_-$

2.4 Robust Statistics and Robustness in Unsupervised Techniques

Many methods have been developed in order to robustify machine learning algorithms, these methods are meant to improve learning when data is contaminated. Some of these techniques have a deep formal proof to show its robustness whereas other techniques rely in experiments. The main idea of this section is to show how some of the most updated methods are dealing with contaminated data.

2.4.1 K-means

k-means has been one of the main methods used to partition the data into groups or clusters that have a similarity in terms of the Euclidean distance. In this case, it can be seen that the Euclidean distance as it looked in the section 2.2 is not robust, since a single outlier will bias the centroid estimation. In this sense, several methods have been proposed in order to robustify the k-means algorithm: create a new cluster where all the outliers are added so that every point in the data set will be equidistant from the cluster [23]; finding a new optimization function using expectation-maximization(EM) algorithm in which a parameter $o_i, \forall i \in 1, \cdots, n$, will have a value of one if the point is considered an outlier and zero otherwise [24]; robust fuzzy c-means contains a fuzzy membership parameter and the typical value matrix, in this case we have a parameter u_{ij} that indicates the degree of fuzzy membership matrix, and the matrix T is the typical value matrix or the possibilistic membership matrix [25]; and the α-cut fuzzy k-means which redefines the optimization function in such a way that when μ_j is greather than threadshold the sample will belong with probability 1 to the j-cluster [26].

2.4.2 PCA

The principal component analysis is a method of projecting the data into an orthogonal space from the original coordinates system in order to reduce the dimensionality of the input vector space. This new system of coordinates given by the orthogonal space is called

principal components. Like k-means, you have a dataset $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d \times n}$ where d is the dimensionality of the points and n is the number of samples. The idea is to find a low-rank matrix that fits the original matrix X. Thus, we want to find the solution of the following optimization problem:

$$\arg\min_{Z} ||X - Z||_F^2$$

After having the full rank decomposition, it can be found that the problem becomes an eigenvalue decomposition in which $\lambda \mathbf{v} = C\boldsymbol{v}$, where C is defined as the covariance matrix of the matrix X, λ is defined as the eigenvalues of the matrix C and $\mathbf{v}$ is defined as the eigenvectors. Since the optimization was made from the euclidean distance, it makes the traditional PCA not robust against outliers. **[27].**
In one of the first efforts of robustify the PCA algorithm was finding a robust estimate of the covariance matrix. However, this approach where the dimensionality of the data set is the main concern becomes the knowledge extraction intractable **[28].**
Another problem addressed in the robustness of PCA is to center the data because the data are assumed typically centered, i.e. $\sum_{i=1}^{n} x = 0$. One idea to address this problem is to use the standard $l_{2,1}$ defined as $\sum_{i=1}^{n} ||\cdot||_2$. This approach will give equal weight to the outlier and normal values **[29].**
Reformulating the function of reconstruction has been one of the ways to robustify PCA, (Fernando de la Torre), in his paper shows that the lost function can be changed by a related m-estimator (Geman-MCLure). The idea is to define the reconstruction error as:

$$\arg\min_{Z,\mu,\sigma} \rho(X - \mu - Z, \sigma)$$

where σ is scale parameter of the Geman-MClure function, defined as $\rho(x;\sigma) = \frac{x^2}{x^2+\sigma^2}$. In this case σ is responsible of the outlier percentage [30].

2.4.3 Kernel PCA

Kernel principal component analysis (Kernel PCA) is an extension of the well-studied method of PCA where data is projected into a lower dimensionality. In this case, data are first transformed from a Euclidean space into a feature space by a ϕ-function where the normal PCA method is performed. Interestingly, explicit vectors in the feature space are not necessary because with the kernel trick only the actual data is needed, where the dot product can be exchanged for any kernel function.
Several methods to robustify Kernel PCA have been proposed: One of them is developed by Michiel Debruyne, he defines a new way of center the data in the feature space. He uses the norm L_1 in order to center the data in a robust way **[31];** Instead of removing the outliers in the input space, the outliers can be deleted in the feature space. This is achieved through the reconstruction error which can be found in the feature space and

by doing so, see the points that have a large deviation with respect to the normal values. The authors claims that in the case of Kernel PCA, the kernel chosen does not have any significance [32].

2.4.4 Kernel Methods

Working with kernels has been well studied by Vapnik, Scholkopf, Cristianini and others. Despite this, little work is found that links m-estimators with kernels. Two representative works can be found below. Chen proposes a new kernel for support vector machines. In this new process, the euclidean distance between the samples is exchanged for a more robust distance given by the m-estimators defined as $SRD_{\rho,\gamma}(x,x') = \sum_{j=1}^{d} \rho(|x_j - x'_j|, \gamma)$, so the new kernel is constructed as shown below:

$$K(x,x)' = exp\left(\frac{SRD_{\rho,\gamma}(x,x')}{2\sigma^2}\right)$$

In the paper is shown that by using the tri-media as the robust distance, the best performing is achieved. [33]
CT Liao propose a robust kernel inspired in a robust m-estimator. In this paper states that the robustness is one of the critical points in machine learning, and the author achieves the robustness by building a kernel inspired from the m-estimators, so that the new kernel is less corrupted than a normal kernel. One of the key ideas in the paper is that a-priori knowledge from the distribution of the data robustifies the algorithm to various types of pollution, but it is very difficult to predict in reality. To make the new kernel, a Geman-McLure m-estimator defined as $\rho(x;\sigma) = \frac{x^2}{x^2+\sigma^2}$, where σ^2 is a parameter of scale. Thus, the new kernel is built as follows:

$$K(x_i,x_j) = \sum_{l=1}^{n}\left(\frac{\sigma_l^2}{|x_l^{(i)} - \mu_l|^2 + \sigma_l^2}\right)\left(\frac{\sigma_l^2}{|x_l^{(j)} - \mu_l|^2 + \sigma_l^2}\right)$$

where μ is a measure of central tendency and σ is a measure of spread. In the paper, μ is calculated as the mean and σ is calculated based on the median absolute deviation (MAD). Although the authors claim that this is inspired in m-estimators a formal proof is not given [34].

2.4.5 Robust Manifold Nonnegative Matrix Factorization (RMNMF)

Robust manifold nonnegative matrix factorization (RMNMF) [35] uses the norm $l_{2,1}$ instead of Frobernius' rule. Based on the following data set $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{p,n}$ where p defines the number of dimensions, and n the number of points. The rule $l_{2,1}$ is defined as follows:

$$||X||_{2,1} = \sum_{i=1}^{n} ||x_i||_2 \tag{2.4.1}$$

The objective function of algorithm RMNMF between matrix X and factorization FG^T need to be robust. This is achieved by using a robust distance at the values with noise in matrix X. The target function is defined as follows:

$$\min_{G \geq 0} ||X - FG^T||_{2,1} \tag{2.4.2}$$

In addition to having a function which depends on the distance between the points, the author proposes a regularization with the laplacian graph, in order to build spectral clustering. The target function proposed by the author is as follows:

$$\min_{G \geq 0} ||X - FG^T||_{2,1} + \lambda Tr(G^T LG) \tag{2.4.3}$$

where lambda is a parameter of regularization. Since this is the greatest value, factorization will be more sparse, since the trace will have more weight in minimizing the function. To solve this minimization - related problem, the author proposes an iterative, 4 - step method. For further reference, see [35].
RMNMF solves the following optimization problem:

$$min_{F,E,H,G} ||E||_{2,1} + \lambda Tr(G^T LH) + \tag{2.4.4}$$

$$\frac{\mu}{2}||X - FG^T - E + \frac{1}{\mu}\Lambda|_F^2 + \frac{\mu}{2}||G - H + \frac{1}{\mu}\Sigma||_F^2 \tag{2.4.5}$$

$$s.t. H \geq 0, G^T G = I$$

where $H = G$ and μ, Λ and Σ are parameters for the Augmented Lagrangian Multiplier framework. A scheme of 4 steps was proposed by Huang herein to optimize the formula 2.4.4. For further details, see [35].

2.4.6 Nonnegative Matrix Factorization Using Graph Random Walks (NMFR)

Nonnegative matrix factorization using graph random walks (NMFR) [36] uses the random walks notion in order to improve clustering results in Spectral Clustering. To do so, a W regularization parameter is added, thus:

$$\min_{W \geq 0} -Tr(W^T AW) + \lambda \sum_i (\sum_k W_{ik}^2)^2 \tag{2.4.6}$$

$$s.t. W^T W = I$$

In so doing, the author claims that the trace is minimized since by augmenting parameter λ optimization will tend to give lesser values to its diagonal. This is desirable since the self-similarity usually does not give us valuable information. In order to perform optimization, the author uses Lagrangian multipliers to impose the restriction of $W^T W$. Interestingly, a random walk is used in the optimization process. This means a random step in each point of the optimization process, in order to randomize optimization and run a smooth optimization process [36]:

$$\text{update } A \leftarrow \alpha QF + (1 - \alpha)W \tag{2.4.7}$$

Iterative rules are utilized to optimize the function. The following update is conducted in each iteration:

$$W_{ik}^{new} \leftarrow W_{ik} \sqrt{\frac{(AW + 2\lambda WW^T VW)_{ik}}{(2\lambda VW + WW^T AW)_{ik}}}$$

Chapter 3

The Robustness of Kernel Estimation

In [37] the author proposes that the uses of non-linear kernel make the unsupervised techniques more robust to outliers compared to a linear kernel. The results show that when the contamination is increased, the linear kernel has a higher bias measure between estimators and real parameters compared to the Guassian kernel. Taking this into account, in this chapter we want to show the intrinsic relationship that exists between kernels and some robust m-estimators. This chapter proposes a formal proof that doing mean estimation in the features space with some kinds of kernel is like doing robust mean estimation in the data space. Figure **3-1** shows in graphic form the main idea of the chapter: first, data is located in the problem space where a non-robust centroid is calculated with the mean estimator; second, a kernel function is used to map every data point in the data space to an induced feature space; third, the parameters are estimated in the feature space where a centroid is calculated (note that in the feature space the centroid is not necessary robust); and, finally, with the help of an approximate inverse function 4.2.2 ($P(\Phi)$), every point in the feature space is mapped to the data space. Furthermore, parameter estimates in the data space with nonrobust estimate are biased toward contaminated data. One of the most interesting findings of this chapter is the relationship between kernels like Gaussian kernel and m-estimators like Welsch-estimator. This relationship is shown in proposition 3.2.2

3.1 Approximate Pre-Image Definition

Lets $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel (not necessarily positive semidefinite) and lets $\Phi : \mathcal{X} \to \mathcal{F}$ a mapping from the original space to the corresponding reproducing kernel Hilbert space (RKHS), or reproducing kernel Krein space (RKKS) in case k being indefinite. Since Φ is not onto, in general, there are elements $\phi \in \mathcal{F}$ which do not have a pre-image, i.e. $\nexists x \in \mathcal{X}, \Phi(x) = \phi$. The next definition is motivated by this fact.

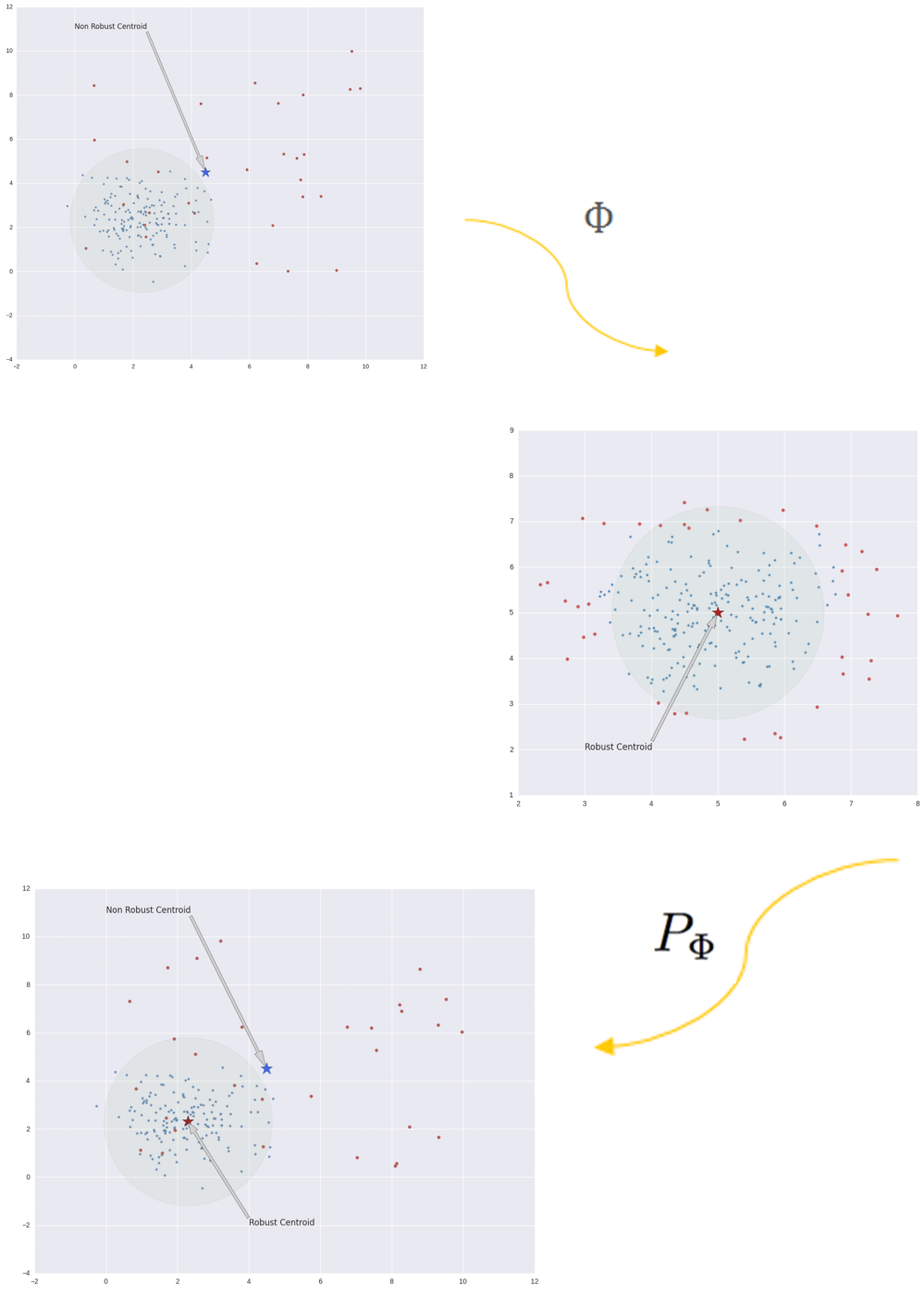

Figure **3-1**: Graphical Explanation of the intrinsic relationship that exist between kernels and some robust m-estimators: first, data is located in the problem space where a non-robust centroid is calculated with the mean estimator; second, a kernel function is used to map every data point in the data space to an induced feature space; third, the parameters are estimated in the feature space where a centroid is calculate (note that in the feature space the centroid is not necessary robust); and, finally, with the help of an approximate inverse function 4.2.2 ($P(\Phi)$), every point in the feature space is mapped to the data space.

Proposition 3.1.1. *Approximate pre-image.*

$$\begin{array}{rccl} P_\Phi: & \mathcal{F} & \to & \mathcal{P}(\mathcal{X}) \\ & \phi & \mapsto & \arg\min\limits_{x\in X} \|\Phi(x)-\phi\|_{\mathcal{F}}^2 \end{array}$$

where $\|f\|_{\mathcal{F}}^2 = \langle f, f\rangle_{\mathcal{F}}$ *is the squared norm in the corresponding Hilbert (or Krein space).*

3.2 Kernel Robust Estimation

In the following propositions, we want to show that if we find the centroid's pre-image-from a set of points projected in a feature space, they correspond to robust mean estimators, if the appropriate kernel is used [21]. In this case, appropriate means that the kernel is isotropic. An isotropic kernel is a kernel that only depends on the distance, in the input space, between its arguments, i.e $k(x,y) = g(\|x-y\|)$ [17]. We will indistinctly use the notation $k(\|x-y\|)$ and $k(x,y)$ to refer to the application of the kernel hoping the meaning will be clear from the context.

Proposition 3.2.1. *Let* $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ *be an isotropic kernel with* $\Phi : \mathcal{X} \to \mathcal{F}$ *the associated mapping to the induced feature space and* $\{d_1, \cdots, d_n\} \subseteq \mathcal{X}$ *a set of samples, then*

$$P_\Phi(\mu) = \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \|\Phi(d_i)-\Phi(y)\|_{\mathcal{F}}^2, \text{ with } \mu = \frac{1}{n}\sum_{i=1}^{n}\Phi(d_i)$$

Proof.

$$\begin{aligned} P_\Phi(\mu) &= \arg\min_{y\in\mathcal{X}} \|\Phi(y)-\mu\|_{\mathcal{F}}^2 = \arg\min_{y\in\mathcal{X}} \left\|\Phi(y) - \frac{1}{n}\sum_{i=1}^{n}\Phi(d_i)\right\|_{\mathcal{F}}^2 \\ &= \arg\min_{y\in\mathcal{X}} k(\|y-y\|) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k(\|d_i-d_j\|) - \frac{2}{n}\sum_{i=1}^{n} k(\|y-d_i\|) \\ &= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} k(\|y-y\|) + \sum_{i=1}^{n} k(\|d_i-d_i\|) - 2\sum_{i=1}^{n} k(\|y-d_i\|) \\ &= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \big(k(\|y-y\|) + k(\|d_i-d_i\|) - 2k(\|y-d_i\|\,\big) \\ &= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} <\Phi(y),\Phi(y)>_{\mathcal{F}} + <\Phi(d_i),\Phi(d_i)>_{\mathcal{F}} -2<\Phi(y),\Phi(d_i)>_{\mathcal{F}} \\ &= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \|\Phi(d_i)-\Phi(y)\|_{\mathcal{F}}^2 \end{aligned}$$

The equality in the third line follows by the fact that the first and second terms in the right side of the second equality do not depend on y, since $k(\|y - y\|) = k(0)$, so they can be substituted by an arbitrary constant (an expression that does not depend on y). □

Proposition 3.2.2. *Given a set of points $S = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$, the approximate pre-image of its centroid in a feature space, $\mathcal{F}$, induced by a Gaussian kernel, k, corresponds to the Welsch location M-estimator. In other words:*

$$P_\Phi(\mu) = P_\Phi\left(\frac{1}{n}\sum_{i=1}^{n}\Phi(d_i)\right) = \arg\min_{y\in\mathcal{X}}\sum_{x_i\in S}\rho_{\text{welsch}}(\|x_i - y\|)$$

Proof. Let $P_\Phi(\mu)$ be the approximate pre-image of μ, defined as in proposition 3.2.1,

$$\begin{aligned}
P_\Phi(\mu) &= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \|\Phi(x_i) - \Phi(y)\|_{\mathcal{F}}^2 \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} < \Phi(x_i), \Phi(x_i) >_{\mathcal{F}} + < \Phi(y), \Phi(y) >_{\mathcal{F}} -2 < \Phi(x_i), \Phi(y) >_{\mathcal{F}} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} k(x_i, x_i) + k(y, y) - 2k(x_i, y) \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} 2 - 2e^{-\left(\frac{\|x_i - y\|^2}{2\sigma^2}\right)} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \rho_{welsch}(\|x_i - y\|)
\end{aligned}$$

with c = $\sqrt{2}\sigma$. Where the first equality follows by the proposition 3.2.1. □

3.3 Conclusions

In this chapter, it is proposed that when an estimation is doing with the Gaussian kernel is like doing robust estimation in the data space using a robust Welsch estimator. Also, it is defined the Approximate pre-image, because depending on the kernel used, it is not always possible to find the exact pre-image as in the case of the Gaussian kernel. With the definition of the Approximate Pre-Image, in section 3.2 it is proved that the estimation of the mean in the feature space is not explicitly calculated and only needs the mapped data points. It is also proved that when an estimation is doing with the Gaussian kernel is like doing estimation with the Welsch estimator in the data space. In chapter 5 it is defined a three new kernels; Huber's, Andrews', and Tukey's kernels with the proofs exposed in this chapter.

Chapter 4

Contamination Experiment

The main goal of the chapter is to conduct two experiments to prove that when a robust kernel is used in unsupervised learning like in the chapter 3, the estimation of the parameters is less biased compared to non robust approaches. There are different types of noise which could exist in a database, for the current chapter two experiments are performed; the first experiment use a synthetic uniform contamination, and, the second experiment use an occlusion pixels contamination. See section 2.1 for further information. In the chapter 3 it is proved that doing an mean estimation in the features space with some kinds of kernel is like doing robust mean estimation in the data space. For example, when using the Gaussian kernel to do parameter estimates in the feature space is like doing parameter estimate with the robust Welsch estimator in the data space. In light of what has been put forward up to this point, different algorithms are used; first experiment use k-means and Kernel k-means with the Gaussian kernel, and, second experiment use k-means, Kernel k-means, Kernel Convex NMF and RMNMF with the Gaussian kernel.

4.1 Datasets

Two experiment are conducted; the first experiment use a synthetic dataset generated by three multi-normal Gaussian distribution, see figure **4-1**. To perform the contamination experiment a uniform contamination is used in the range of the data points, see figure **4-2**.

The second experiment uses Jaffe dataset. Jaffe is a dataset of images with 10 different Japanese females, where each set of images has 7 facial expressions. To manage the dimensionality of the images that have 400 pixels x 400 pixels, we use an image resize of 1 over 25, obtaining image of 32 pixels x 32 pixels, see figure **4-3**. Every image in the dataset was contaminated with occlusion pixels contamination **4-3**, i.e. a random half of the picture is chosen, afterwards, the pixels of the selected half are shutdown (number zero for gray images).

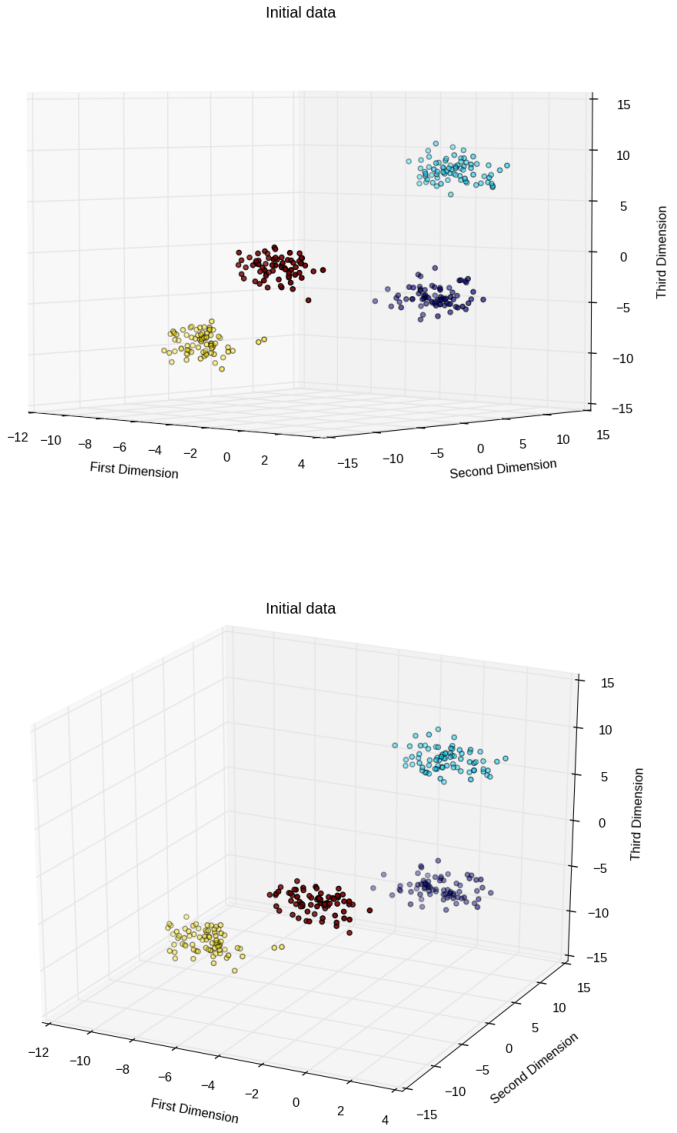

Figure **4-1**: Synthetic data set generated by a multivariate normal distribution and uniform contamination in a three-dimensional space.

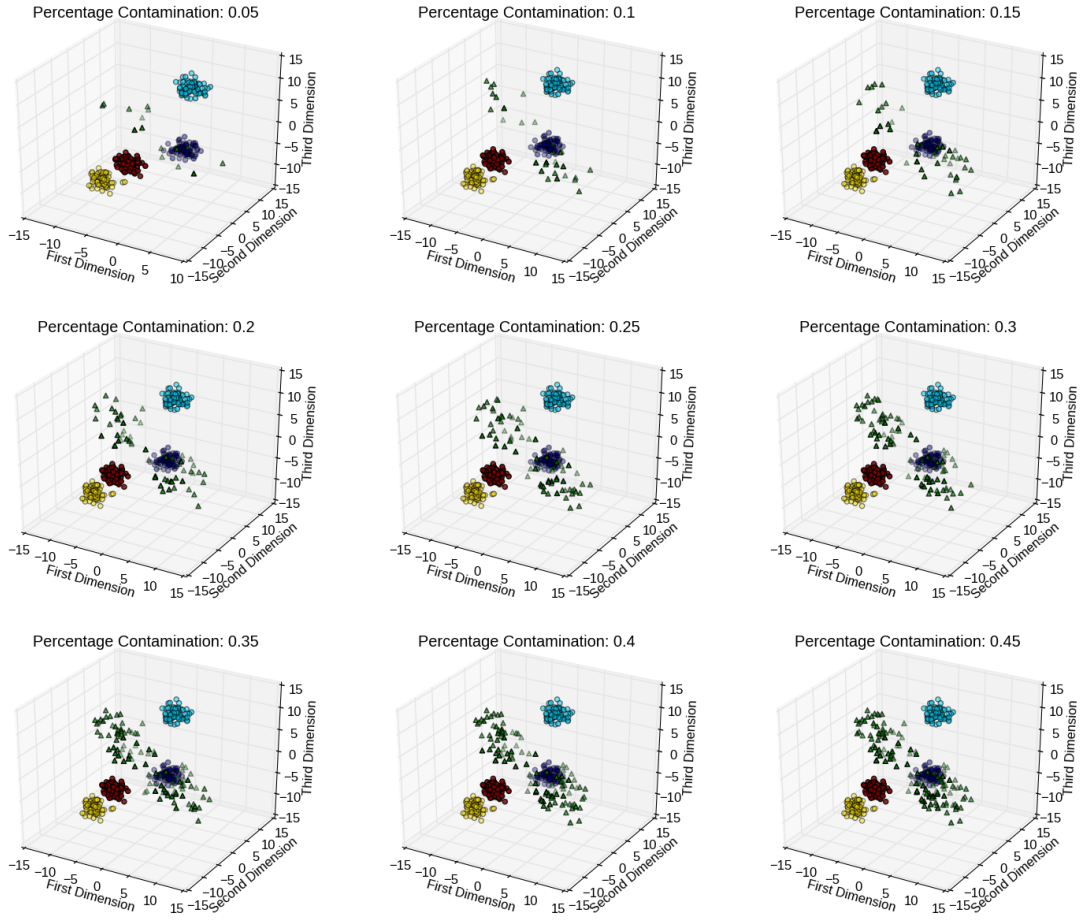

Figure **4-2**: Synthetic data set generated with different degrees of Guassian Contamination

Figure **4-3**: Synthetic data set generated by Jaffe image dataset with Occlusion Contamination: Every image in the dataset was contaminated with occlusion pixels contamination **4-3**, i.e. a random half of the picture is chosen, afterwards the pixels of the selected half are shutdown (number zero for gray images).

4.2 Experimental Setup

The goal of the experimental evaluation in this chapter is to prove that when a robust kernel is used in unsupervised learning, the estimation of the parameter is less bias compared to nonrobust approaches. As shown in chapter 3, Gaussian Kernel is robust because of its close relationship with Welsch M-Estimator. In order to prove the main hypothesis, the bias between the real mean parameter and the pollute mean estimation will be used to show the robustness of Gaussian Kernel.

Bias In statistics, it is very often assumed that we have a model parameterized by θ that gives us the possibility of building a probability model $P_\theta(X) = P(X|\theta)$ and a statistical parameter $\hat{\theta}$ that is an estimate of the unknown θ parameter. With this in mind, we can obtain the bias as the difference between the estimated parameter and the real parameter as follows:

$$Bias_\theta[\hat{\theta}] = E[\hat{\theta}] - \theta \tag{4.2.1}$$

where E_θ denotes expected value over the distribution $P_\theta(x) = P(x \mid \theta)$, i.e. averaging over all possible observations x. So, for our experiment, we get $\mathrm{E}[\hat{\theta}]$ as the centroids estimation given by k-means and pre-image of Kernel k-means. The real parameter θ is known beforehand when synthetic data are generated.

Bias k-means We obtain the clusters of a data set with k-means algorithm. After that, we can calculate the bias of the estimation as subtractions between the real centroids and the estimated centroids of the cluster.

Bias Kernel k-means In the case of Kernel k-means we need to calculate the pre-image in order to obtain the estimated centroids. Set x a point in the input space. Given $\phi \in F$, it's pre-image $P\phi$ is defined as:

$$P\phi = \arg\min_{x \in X} \|\Phi(x) - \phi\| \tag{4.2.2}$$

Several different methods have developed in the last decades to resolve this problem, one of the most widely used methods to retrieve the pre-image with the RBF kernel, is the method developed by Kwok when the kernel-k-means algorithm is used [21] (see section 2.3.3).

Percentage of Contamination For this experiment, we will be adding one percentage of uniform contamination to the real data set, so that the new contamination sample will be at the end of the data set.

Sigma Parameter In the case of Kernel-k-means we need a sigma parameter that can be understood as a spread parameter. For this experiment we search the sigma parameter in a logarithm vector [2^-5,2^-4,...,2^5].

Noise factor

Noise factors for our experiments depend on the initialization of k-means, or Kernel-k-means. This initialization for our experiment is random and it depends on the clock of the computer, so we can't handle or measure the noise factor. We can avoid this noise factor with a different initialization like heuristics or performing a preliminary cluster; however, in order to compare the technique rather than the initialization itself, we choose random initialization. For every 1 percentage of contamination adding, we run each algorithm 30 times and compute bias for every time. After that, we compute the mean of the values. We need to define the number of iterations to calculate the pre-images of the kernel (3.1.1), in this case we use 200 iterations to get the pre-images of the clusters centroids.

Algorithms

- First experiment

To perform the Gaussian experiment the algorithms KMeans and Kernel-Kmeans are used.

- Second experiment

To perform the occlusion pixels contamination experiment the algorithms k-means, Kernel k-means, Convex NMF, NMF y RNMF are used.

4.3 Results and Discussion

In the first experiment, for every percentage, we plot a graph **4-4** with the mean of the bias one for each algorithm. In addition, we can see the dataset is contaminated with each run. In figure **4-4**, it can be observe that when a property parameter σ is chosen, Kernel K-Mean does not change the centroids' estimation. In the case of k-means, it tries to explain every sample with what it suffices only one extreme point to pollute the centroid estimation.
Choosing the sigma parameter is sometimes not so easy, given the restriction of time by the increasing complexity of the algorithms using Kernels. Calculating the kernel matrix is $O(n^2)$, so if we were to make an exhaustive parameter search with a large n-samples, the process would turn out to be highly time consuming. In our case, a logarithmic vector was used in order to find the optimal σ parameter. With this σ-parameter, the Kernel

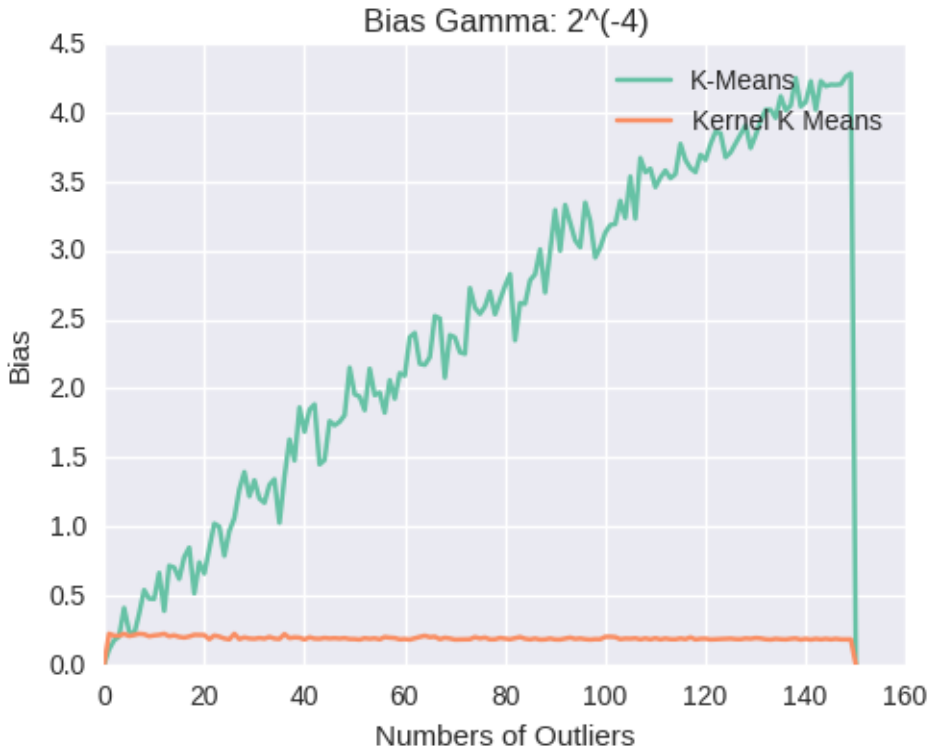

Figure **4-4**: First experiment: results comparing KMeans vs Kernel k-means clustering accuracy in Gaussian Contamination. The c-parameter for Gaussian kernel is 2^{-4}. The x-axis indicates the number of outliers added to the original dataset. The y-axis indicates the bias measure and is obtained using equations given in 4.2.

k-means algorithms with Gaussian kernel is less affected in contrast of k-means algorithm where the increase in pollution shows a linear bias, moving the current centroid away from the current centroid of the data set.

In the second experiment, for every percentage, we plot the mean of the bias in a graph - one per algorithm. Figure **4-5** shows an increasing pollution when images with occlusion are added to the training data. The estimation of the clusters in the case of k-means is biased to a greater extent than in the case of Kernel KMeans and Kernel Convex NMF. It can be explained given the existing relationship between Gaussian kernel and the Welsch M-estimator also shows an insignificant difference among the estimations of the centroids for the three algorithms in the beginning, but when the contamination increases, the difference between the bias of Kernel Convex-NMF and the Robust Manifold NMF begins to increase. This goes to show that the use of a robust kernel improves the robustness of the estimation. It can be observe that Kernel CNMF with a Guassian kernel is less corrupted with the occlusion contamination. When less than thirty(30) percent of the dataset was contaminated with occlusion pixels the differences between the reals clusters and the approximates one's are not so different compare to the dataset with

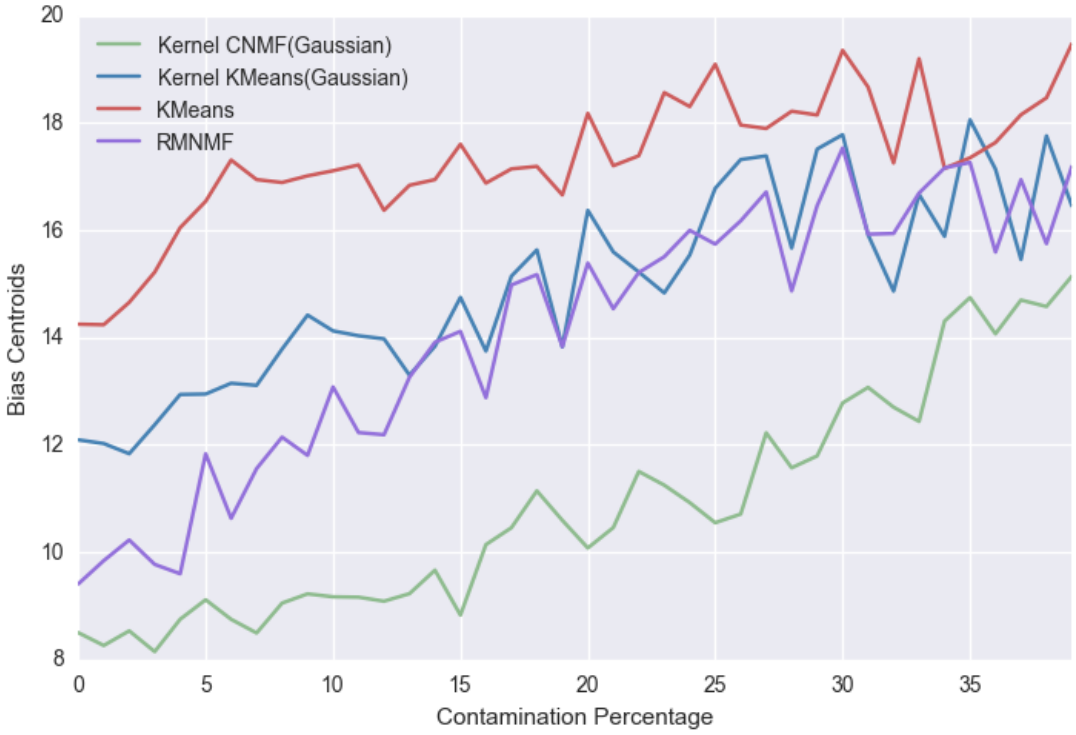

Figure **4-5**: Second experiment: bias of Kernel Convex NMF, Kernel KMeans and KMeans in occlusion contamination in Jaffe Database. The x-axis indicates the percentage of contaminated data. The y-axis indicates the bias in squared-distance from estimated centroids to the real centroids.

no contamination.

4.4 Conclusions

The robustness of using Gaussian kernel in clustering is discussed in chapter 3. This chapter shows a comparison between non-kernel algorithms like k-means and RNMNF, and, kernel algorithms like Kernel k-means and Kernel CNMF. The kernel-based techniques shows better result when the data is contaminated compared to non kernel-based techniques like RMNMF. In the next chapter 5 is proposed three new kernels that have similar properties to the Gaussian kernel and in chapter 6 these new kernels are evaluated in a systematic experiment.

Chapter 5

Robust Kernels

Chapter 3 proposed that when an estimation is being made with the Gaussian kernel. This is like doing a robust estimation in the data space using a robust Welsch estimator. Despite this example, the proposition 3.2.1 is a general result, and it can be used as a framework to build new robust kernels. Given the number of useful ideas in Chapter 3 a new set of three robust kernels has been building; Tukey's, Andrew's, and, Huber's kernel are motivated by their corresponding robust estimators. This set of kernels has the characteristic of doing a mean estimation of the parameters in the feature space which is like doing robust mean estimation in the data space. These three new kernels defined in this chapter are indefinite kernels, i.e. these new kernels are embedded in a Reproducing Kernel Krein Spaces, see sec. 2.3.1. The issue of optimizing the loss function in Reproducing Kernel Krein Spaces to obtain a local minimum or maximum value is no longer possible, instead, we need to stabilize the loss function [22]. One of the most remarkable findings of these new kernels has been the relationship found between Tukey's robust kernel and the Epanechnikov defined by Ong Cheng.

5.1 Tukey's Robust Kernel

Definition. Tukey's robust kernel is defined as follows:

$$k(\|y - x_i\|) = \left\{ \begin{array}{cc} \frac{1}{12}(1 - (\frac{||x_i - y||}{c})^2)^3, & \text{if } \frac{||x_i - y||}{c} <= 1 \\ 0, & \text{if } \frac{||x_i - y||}{c} > 1 \end{array} \right\} \tag{5.1.1}$$

Proposition. *Given a set of points $S = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$, the preimage of its centroid in a feature space, $\mathcal{F}$, induced by a Tukey's kernel, k, corresponds to the Tukey Bisquare location M-estimator. In other words:*

$$P_\Phi(\mu) = P_\Phi\left(\frac{1}{n}\sum_{i=1}^{n}\Phi(d_i)\right) = \arg\min_{y\in\mathcal{X}} \sum_{x_i \in S} \rho_{\text{tukey}}(\|x_i - y\|)$$

Proof. Let $P_\Phi(\mu)$ be the approximate pre-image of μ, defined as in proposition 3.2.1,

$$
\begin{aligned}
P_\Phi(\mu) &= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \|\Phi(x_i) - \Phi(y)\|_{\mathcal{F}}^2 \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} < \Phi(x_i), \Phi(x_i) >_{\mathcal{F}} + < \Phi(y), \Phi(y) >_{\mathcal{F}} -2 < \Phi(x_i), \Phi(y) >_{\mathcal{F}} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} k(x_i, x_i) + k(y, y) - 2k(x_i, y) \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \left\{ \begin{array}{cc} \frac{1}{12} + \frac{1}{12} + \frac{2}{12}(1 - \left(\frac{||x_i - y||}{c}\right)^2)^3, & \text{if } \frac{||x_i-y||}{c} <= 1 \\ \frac{1}{12} + \frac{1}{12} + 0, & \text{if } \frac{||x_i-y||}{c} > 1 \end{array} \right\} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \left\{ \begin{array}{cc} \frac{1}{6}(1 - (1 - \left(\frac{||x_i - y||}{c}\right)^2)^3 & \text{if } \frac{||x_i-y||}{c} <= 1 \\ \frac{1}{6}, & \text{if } \frac{||x_i-y||}{c} > 1 \end{array} \right\} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \rho_{tukeys}(\|x_i - y\|)
\end{aligned}
$$

□

In [22] Ong Cheng defines the Epanechnikov kernel as follows:

$$
k(\|y - x_i\|) = \left\{ \begin{array}{cc} (1 - \left(\frac{||x_i - y||^2}{\sigma}\right))^p, & \text{if } \frac{||x_i-y||}{\sigma} <= 1 \\ 0, & \text{if } \frac{||x_i-y||}{\sigma} > 1 \end{array} \right\}
$$

He shows that if principal eigenvalues are calculated, some of them are negative. For this reason, this kernel is an indefinite kernel. Some modifications were made to use this kernel as a similarity measure, noting that a similarity measure should be between zero and one, using one when the two samples are very similar and zero when the two samples are dissimilar. Given this, the fraction $\frac{1}{12}$ is removed in order to have a $[0, 1]$ co-domain of the kernel function, and it is defined as follows:

$$
k(\|y - x_i\|) = \left\{ \begin{array}{cc} (1 - \left(\frac{||x_i - y||}{c}\right)^2)^3, & \text{if } \frac{||x_i-y||}{c} <= 1 \\ 0, & \text{if } \frac{||x_i-y||}{c} > 1 \end{array} \right\} \tag{5.1.2}
$$

This kernel is sparse since after a threshold given by $\frac{||di-y||}{c} > 1$ every value goes to zero. It can be observed that when the difference is close to zero the function is not smooth. However, the sparse condition of the kernel brings us the possibility of designing new kernel algorithms that scale with the increasing amount of data given in big data **5-1**.
This new Tukey's kernel does not satisfy Mercer's theorem, i.e. the kernel matrices generated by the Tukey's kernel have negative eigenvalues [1]. Figure **5-2** presents an

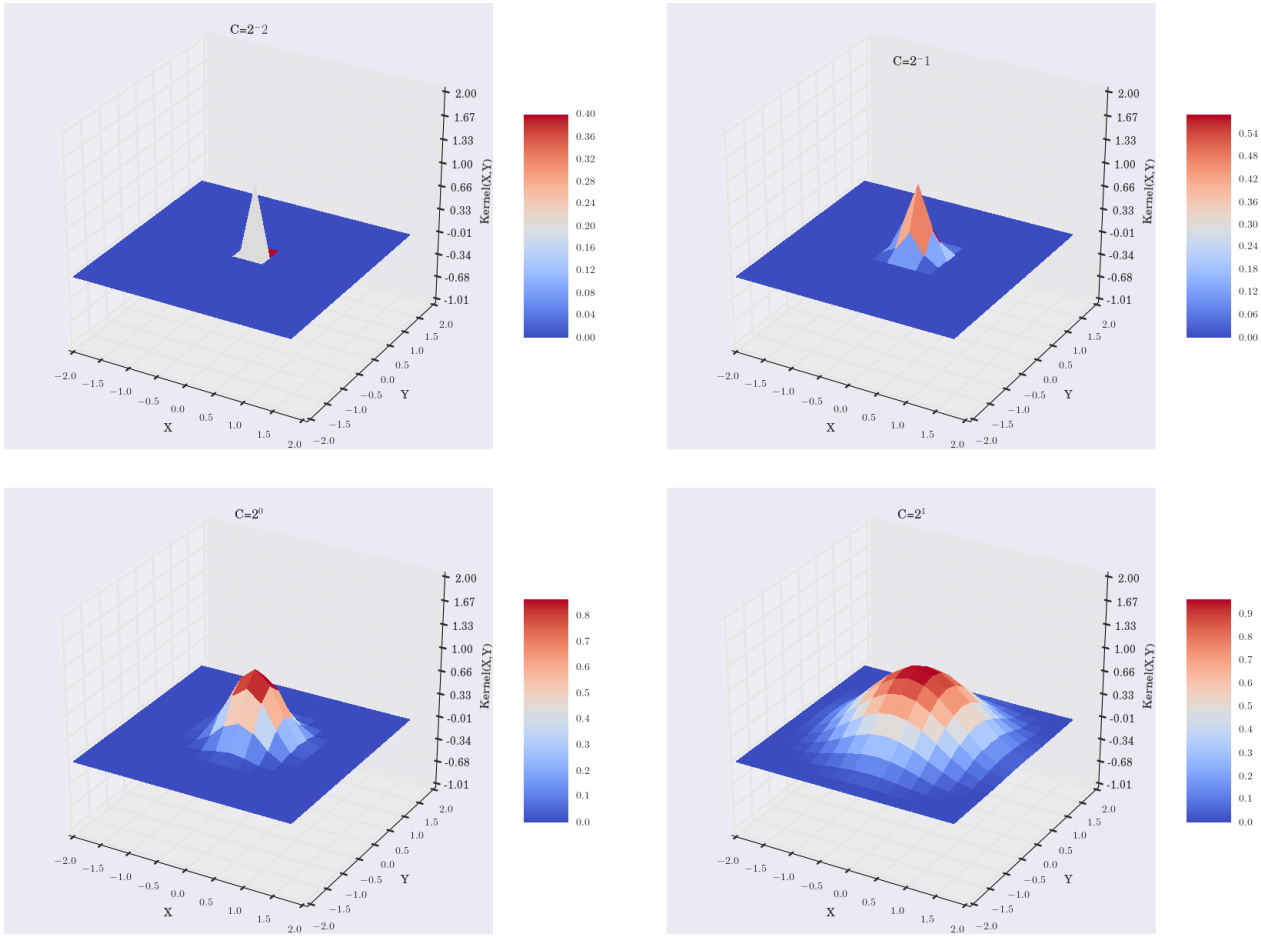

Figure **5-1**: 3D Surface of Tukey's robust kernels in respect to the origin. Each surface is computed with a different C-value. It can be observed that when a tiny C-value is used the vast majority of the kernel data goes to zero.

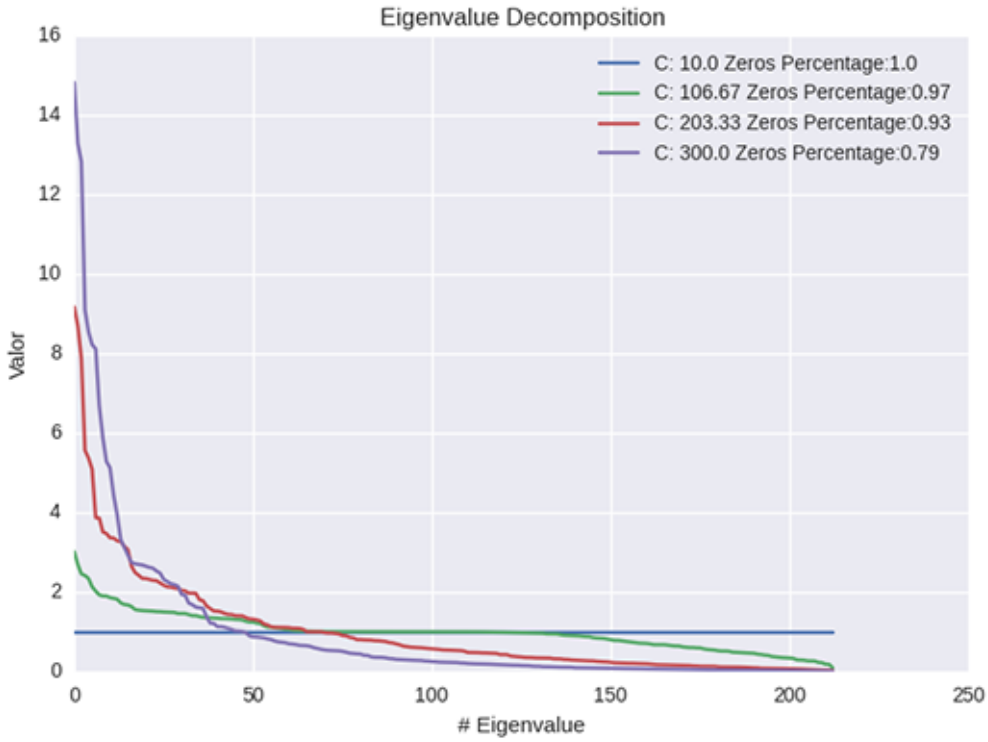

Figure **5-2**: Eigenvalues of Tukey's robust kernel in Jaffe dataset. Each line on the graph is computed with a different C-value. The **zeros percentage** indicates the percentage of zeros that exist in the computed kernel matrix.

overview of the eigenvalues, obtained from kernel matrices in the Jaffe dataset. In this figure, various parameters that were used have shown negative eigenvalues. For this reason, it was necessary to extend the proof in 3.1.1 for Krein space, where the distance between two vectors can be negative.

5.2 Andrews' Robust Kernel

Definition 5.2.1. Andrews' robust kernel is defined as follows:

$$k(\|y - x_i\|) = \left\{ \begin{array}{cc} \frac{1}{2\pi^2} cos\left(\pi \frac{||x_i - y||}{c}\right), & \text{if } \frac{||x_i - y||}{c} <= 1 \\ -\frac{1}{2\pi^2}, & \text{if } \frac{||x_i - y||}{c} > 1 \end{array} \right\} \tag{5.2.1}$$

Proposition. *Given a set of points $S = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$, the preimage of its centroid in a feature space, $\mathcal{F}$, induced by Andrews' kernel, k, corresponds to the Andrews' location M-estimator. In other words:*

$$P_\Phi(\mu) = P_\Phi\left(\frac{1}{n}\sum_{i=1}^{n}\Phi(d_i)\right) = \arg\min_{y\in\mathcal{X}} \sum_{x_i \in S} \rho_{\text{andrews}}(\|x_i - y\|)$$

Proof. Let y be the Φ-preimage of μ, defined as in proposition 3.2.1,

$$\begin{aligned}
P_\Phi(\mu) &= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \|\Phi(x_i) - \Phi(y)\|_{\mathcal{F}}^2 \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} < \Phi(x_i), \Phi(x_i) >_{\mathcal{F}} + < \Phi(y), \Phi(y) >_{\mathcal{F}} -2 < \Phi(x_i), \Phi(y) >_{\mathcal{F}} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} k(x_i, x_i) + k(y, y) - 2k(x_i, y) \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \left\{ \begin{array}{cc} \frac{1}{2\pi^2} + \frac{1}{2\pi^2} - \frac{2}{2\pi^2}(cos\left(\pi \frac{||x_i - y||}{c}|\right)), & \text{if } \frac{||x_i - y||}{c} <= 1 \\ \frac{1}{2\pi^2} + \frac{1}{2\pi^2} - 2\left(-\frac{1}{2\pi^2}\right), & \text{if } \frac{||x_i - y||}{c} > 1 \end{array} \right\} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \left\{ \begin{array}{cc} \frac{1}{\pi^2}(1 - cos\left(\pi \frac{||x_i - y||}{c}\right) & \text{if } \frac{||x_i - y||}{c} <= 1 \\ \frac{2}{\pi^2}, & \text{if } \frac{||x_i - y||}{c} > 1 \end{array} \right\} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \rho_{andrews}(\|x_i - y\|)
\end{aligned}$$

□

By making experiments with different datasets and different parameters for Andrews' kernel, it was found that some of the eigenvalues are negative. Thus, this kernel is an

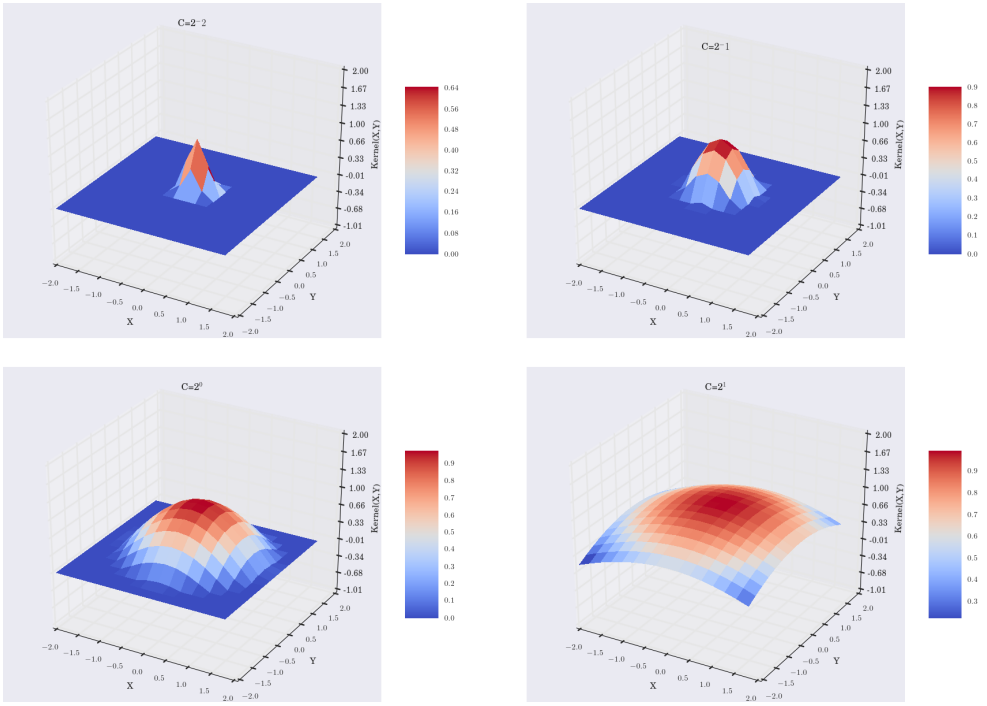

Figure **5-3**: 3D Surface of Andrews' robust kernels in respect to the origin. Each surface is computed with a different C-value. It can be observed that when a tiny C-value is used the vast majority of the kernel data goes to zero.

indefinite kernel. Like Tukey's kernel, some modifications are needed to use the kernel as a similarity measure. Given this, the fraction $\frac{1}{\pi^2}$ is removed and also the fraction $\frac{2}{\pi^2}$ exchanging it for zero is removed. Now, Andrews' kernel is defined as follows:

$$k(\|y - x_i\|) = \left\{ \begin{array}{cc} cos\left(\pi \frac{||x_i - y||}{c}\right), & \text{if } \frac{||x_i - y||}{c} <= 1 \\ 0, & \text{if } \frac{||x_i - y||}{c} > 1 \end{array} \right\} \tag{5.2.2}$$

Just as the Tukey's robust kernel, figure **5-3** presents a sparse matrix when several C-values were used, given the new definition of Andrew's kernel 5.2.2. It can be observed that when a huge number of C is chosen, all the values of the kernel matrices go to zero. That makes this new kernel more useful to machine learning approaches, where new functions can be constructed using this sparse matrix.

Figure **5-4** presents the eigenvalues decomposition of the kernel matrix using the Andrews' robust kernel in the ATT dataset. It can be observed that some of the eigenvalues are negative consequently Andrews' robust kernel is an indefinite kernel. Andrews' robust kernel is indefinite as Tukey's robust kernel. For this reason, Andrews' robust kernel does not satisfy Mercer's theorem.

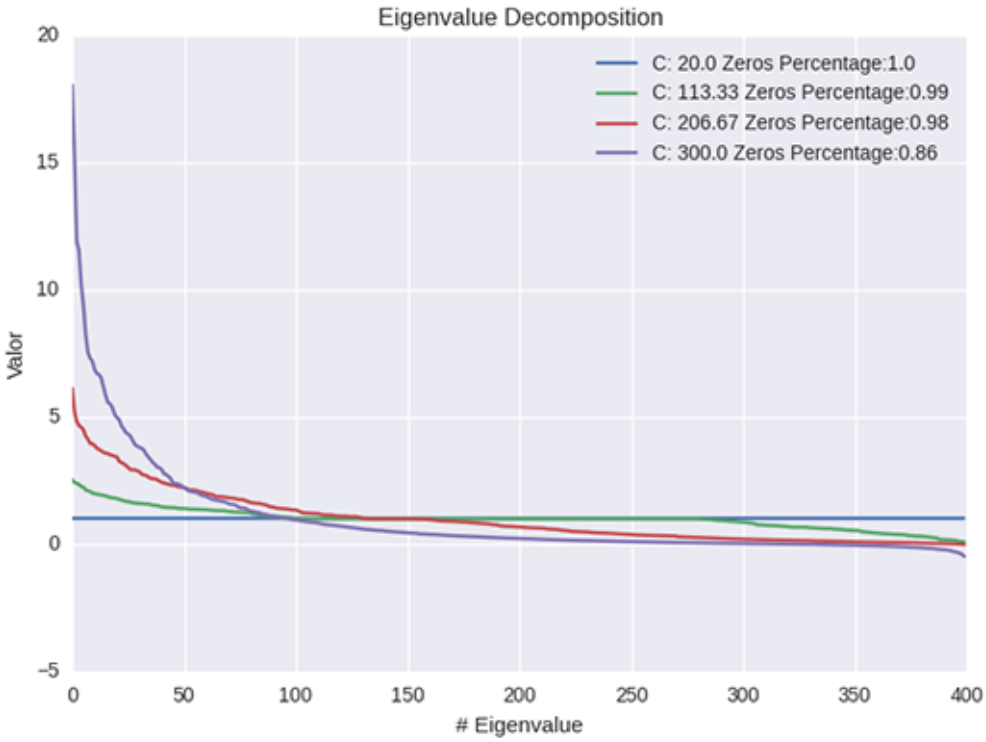

Figure **5-4**: Eigenvalues of Andrews' robust kernel in ATT. Each line on the graph is computed with a different C-value. The **zeros percentage** indicates the percentage of zeros that exist in the computed kernel matrix.

5.3 Huber's Robust Kernel

Definition 5.3.1. The Huber's robust kernel is defined as follows:

$$k(\|y - d_i\|) = \left\{ \begin{array}{cc} -\frac{1}{4}||x-y||^2, & \text{if } \frac{||di-y||}{c} <= 1 \\ -\frac{c}{2}||x-y|| + \frac{c^2}{4}, & \text{if } \frac{||di-y||}{c} > 1 \end{array} \right\} \tag{5.3.1}$$

Proposition. *Given a set of points $S = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$, the preimage of its centroid in a feature space, $\mathcal{F}$, induced by a Huber's kernel, k, corresponds to the Huber's location M-estimator. In other words:*

$$P_\Phi(\mu) = P_\Phi\left(\frac{1}{n}\sum_{i=1}^{n}\Phi(d_i)\right) = \arg\min_{y\in\mathcal{X}} \sum_{x_i \in S} \rho_{\text{huber}}(\|x_i - y\|)$$

Proof. Let y be the Φ-preimage of μ, defined as in proposition 3.2.1,

$$\begin{aligned}
P_\Phi(\mu) &= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \|\Phi(x_i) - \Phi(y)\|_{\mathcal{F}}^2 \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} < \Phi(x_i), \Phi(x_i) >_{\mathcal{F}} + < \Phi(y), \Phi(y) >_{\mathcal{F}} -2 < \Phi(x_i), \Phi(y) >_{\mathcal{F}} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} k(x_i, x_i) + k(y, y) - 2k(x_i, y) \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \left\{ \begin{array}{cc} 0 + 0 + \frac{1}{2}||x_i, y||, & \text{if } \frac{||x_i-y||}{c} <= 1 \\ 0 + 0 - 2(-\frac{c}{2}||x_i, y|| + \frac{c}{4}), & \text{if } \frac{||x_i-y||}{c} > 1 \end{array} \right\} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \left\{ \begin{array}{cc} \frac{||x_i,y||^2}{2} & , \text{if } \frac{||x_i-y||}{c} <= 1 \\ c||x_i, y|| - \frac{c}{2}, & \text{if } \frac{||x_i-y||}{c} > 1 \end{array} \right\} \\
&= \arg\min_{y\in\mathcal{X}} \sum_{i=1}^{n} \rho_{huber}(\|x_i - y\|)
\end{aligned}$$

□

5.4 Conclusions

In this chapter, three new kernels are proposed; Tukey's, Andrews', and, Huber's kernel. These three new kernels are motivated by their corresponding robust estimators. The intrinsic relationship between those kernels and robust statistics will be validated in an experimental setup in the next chapter. In order to use those indefinite kernels in real problems, it is appropriate to amend the co-domain of the gram-matrix, i.e. for all x, y

pairs of samples, the kernel value is $k(x, y) \in [0, 1]$. This ad-hoc amendment is validated through experiments in the next section.

Chapter 6

Evaluation of Robust Kernels

The use of kernels in unsupervised learning has been well studied [23,33,38,39]. In Chapter 3 a general result is presented in which it is possible to calculate the approximate pre-image of the mean in the feature space without explicitly calculating the mean in the feature space. Later in the same chapter, the intrinsic relationship between the Gaussian kernel and the robust Welsch m-estimator was shown. In Chapter 5 using the general result of Chapter 3, three new kernels were defined: Tukey's, Andrews' and Huber's kernel. This chapter is intended to show how the use of these new robust kernels outperforms or equals the results given by state-of-the-art algorithms. As already stated in Chapter 5, the issue of optimizing the loss function in Reproducing Kernel Krein Spaces to obtain a local minimum or maximum value is no longer possible, instead, we need to stabilize the loss function [22]. Nevertheless, in this section, we will show in a systematic approach that Tukey's robust kernel outperforms the results given by linear and Gaussian kernels. This section evaluates different clustering algorithms; some of them use the kernel trick, thereby allowing the use of a linear, Gaussian, Tukey's, and Andrews' kernel. We use a clustering accuracy metric to evaluate and compare the results of the different algorithms.

6.1 Datasets

Thirteen different data set were used, five of them were image datasets AR, AT&T, Jaffe, Orl and Yale; the remaining eight sets were UCI datasets: abalone, balance scale, coil, fault, movement libras, image segmentation, Umist and wine quality [40, 41]

- Abalone is a dataset that predicts the age of an abalone by physical measurement. Predicting the age of an abalone is a time-consuming task. Because of this, eight different features are extracted such as sex, length, diameter, and height, to predict age in three different classes with an automatic classifier [40].

- AR is a dataset of cropped images having 100 different faces of people, 50 men and 50 women. Those images are frontal view faces that were taken in two different

sessions with different facial expressions, lighting conditions, and occlusions which use of glasses and scarves. To manage the dimensionality of the images, we use an image resize of one over five [41].

- AT&T is a dataset of 40 distinct subjects, where each subject has 10 different images. Those images were taken at different times, varying the lighting, facial expressions, and facial details. To manage the dimensionality of the images, we use an image resize of one over eight [42].

- Balance Scale is a dataset that models psychological and experimental results where each example is classified as having the balance scale tip to the right, tip to the left, or balanced [40].

- Coil is a dataset that consists of images of twenty objects that contain both the object and the background [43].

- Steel Plate Faults are a dataset of steel plates, classified into seven different types [40].

- Jaffe is a dataset of images with 10 different Japanese females, where each set of images has 7 facial expressions. To manage the dimensionality of the images, we use an image resize of one over twenty-five [44].

- Movement Língua Brasileira de Sinais (LIBRAS) is a dataset that contains references to a hand gesture in LIBRAS which is the name of the official Brazilian sign language [40].

- Orl database of faces is a dataset of images. It contains a set of facial images of 40 distinct subjects. The images were taken at different times, varying light, different facial expressions, among others. This database is the same as AT&T but with a non-reduction of the dimensionality [40].

- Image Segmentation (Segment) is a dataset of high-level numeric-valued attributes. The images were drawn randomly from a database of 7 outdoor images [40].

- The Sheffield (Umist) is an images dataset of 20 individuals (mixed races/genders/appearances). The photograph of the individuals shows a range of poses from profile to frontal views [45].

- Wineq is a dataset of chemical analysis to determine the origin of different wines [40].

- Yale is a grayscale image dataset of 15 individuals. There are 11 images per individual. Each image has a different configuration or expression [40].

Table **6-1**: Data set description

Data set	Number of features	Number of samples	Number of class
Abalone	8	4177	3
AR	792	2600	100
AT&T	168	400	40
Balance Scale	4	625	3
Coil	484	1440	20
Fault	27	1941	7
Jaffe	100	213	10
Movement Libras	90	360	15
Orl	1024	400	40
Segment	19	2310	7
Umist	644	575	20
Wineq	11	4898	3
Yale	1024	2414	38

6.2 Experimental Setup

The goal of the experimental evaluation in this chapter is to show the robustness of kernel algorithms in unsupervised learning. In the case of kernel algorithms, four different kernels will be used; linear kernel, Gaussian kernel, Tukey's robust kernel and Andrews' robust kernel. In order to use the kernels defined in Chapter 5, it is necessary to modify the original definition of the kernels. Without this modification, it will be necessary to modify the current definition of the algorithms.

To evaluate our goal, we compared eleven different algorithms, some of them using the kernel trick. The algorithms are k-means, kernel k-means, Convex NMF, kernel Convex NMF [11], Robust Manifold NMF [35], NCUT, PNMF, NSC, ONMF, LSD and NMFR. We used one extrinsic metric extensively in data mining to evaluate our algorithms. This metric is clustering accuracy which is a one to one map between the cluster and the class. With this metric we want to get the precision of the clustering algorithm. First, construct the contingency matrix, and second, we change the rows and columns of the contingency matrix, so the trace of the matrix is maximized.

$$Acc = \frac{\max Trace(contigency\_matrix())}{N}$$

.

6.3 Results and Discussions

Table **6-2** presents the findings after running each algorithm thirty (30) times for each algorithm in every dataset. Each cell of the table presents the mean accuracy and the standard deviation for the column algorithm with the row dataset. It can be observed that none of the algorithms is the best in every dataset, also that the Gaussian, Tukey's and Andrews' kernel together are the best in five of the thirteen datasets. NMFR state-of-the-art algorithm performs quite well, being the best in five of the thirteen datasets but with poor performance in datasets like abalone, AR and WineQ. Given that none of the algorithms is the best in every dataset, it is necessary to do a non-parametric test to compare if a significant difference exists among algorithms.

Non-parametric Test for Significant Difference Between Algorithms

A variance analysis in a non-parametric test was used in order to test the behavior of different algorithms. It was sought to determine whether or not there was a significant difference between the different algorithms in table **6-2**. The test was implemented as follows: based on $\{x_{ij}\}_{m\times n}$ - a data table where m was the data base number and n was the number of algorithms used. The range was defined for each of the databases, organizing them from highest to lowest in accordance with the clustering accuracy presented in table **6-2**, it must be taken into account that, because there were two equal data in different algorithms; the range would be the average of their corresponding ranges. Thus, we obtained the following table:
Let r_i^j be the range for the $j-th$ of k algorithms over the $i-th$ database. Friedman's test compared the range $R_j = \frac{1}{N}\sum_i r_i^j$. The null hypothesis was that all algorithms were equivalent; therefore, Friedman's statistical data must be compared:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right)$$

it was distributed according to a χ_F^2 with $k-1$ degrees of freedom. A better statistic was found using:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}$$

This statistic was distributed in accordance with an F-distribution containing $k-1$ and $(k-1)(N-1)$ degrees of freedom:

$$\sum_j R_j^2 = 1532.93$$

Where N = 13 and K= 15

Table **6-2**: Clustering Accuracy Results for different robust kernel and non-kernel methods.

Algorithm	RBF Kernel KMeans	KMeans	Andrews Kernel KMeans	Tukey Kernel KMeans	RBF KConvexNMF
Abalone	0.5088±0.0001	0.4991± 0.0002	0.5051± 0.0019	0.5075± 0.0	0.5133±0.0004
AR	0.1461±0.0270	0.1061±0.0064	0.1049±0.0049	0.1108±0.0092	**0.2234±0.0017**
AT&T	0.6684±0.0049	0.6505±0.0049	0.5918±0.0210	0.5965±0.0443	0.7353±0.0074
Balance Scale	0.5313±0.0253	0.5170±0.0249	**0.5965±0.0169**	0.5158±0.0159	0.5386±0.0083
Coil	0.6176±0.0568	0.5613±0.0531	0.5149±0.0297	0.6496±0.0464	0.6204±0.0398
Fault	0.1677±0.0055	0.3337±0.0228	0.1686±0.0036	0.3460±0.0017	0.3459±0.0.0016
Jaffe	0.7763±0.0596	0.7077±0.0852	0.7629±0.0746	0.8187±0.0271	0.8488±0.0445
Movement Libras	0.4333±0.0250	0.4478±0.0188	0.4422±0.0206	0.4350±0.0336	0.4750±0.0074
ORL	0.5140±0.0372	0.4925±0.0742	0.5120±0.0366	0.5055±0.0129	0.6230±0.0163
Segment	0.5201±0.0719	0.2248±0.0734	0.5452±0.0477	0.5037±0.0675	0.6155±0.0408
Umist	0.4456±0.0350	0.3802±0.0378	0.4362±0.0229	0.4748±0.0155	0.4320±0.0277
Wineq	0.4806±0.0292	0.4379±0.0235	0.4730±0.0005	0.4605±0.0166	0.4574±0.0039
Yale	0.1234±0.0102	0.0957±0.0067	0.0871±0.0101	0.0852±0.0033	0.2426±0.0111

Algorithm	ConvexNMF	Andrews KConvexNMF	Tukey KConvexNMF	RMNMF	NCUT
Abalone	0.4079±0.0071	**0.5297±0.011**	0.5006±0.0024	0.5202±0.0377	0.387±0.0014
AR	0.1606±0.0030	0.1813±0.010	0.2033±0.045	0.212±0.005	0.1818±0.0070
AT&T	0.6386±0.0264	0.6310±0.0394	0.7380±0.0159	0.6937±0.0156	0.8017±0.0059
Balance Scale	0.5438±0.0654	0.5904±0.0875	0.5331±0.0312	0.5853±0.0719	0.5596±0.0155
Coil	0.5733±0.0363	0.6074±0.0302	0.6688±0.0183	0.5543±0.0122	0.8049±0.0028
Fault	0±0	0.2754±0.0007	**0.3464±0.0002**	0.3379±0.0204	0.2624±0.0005
Jaffe	0.8089±0.0612	0.8667±0.0671	0.8751±0.0475	0.8091±0.0998	**0.9108±0**
Movement Libras	0.4328±0.0200	0.4794±0.0220	0.4867±0.0093	0.4292±0.0119	0.4679±0.0015
ORL	0.5515±0.0302	0.5645±0.0405	0.6000±0.0352	0.3280±0.0266	0.6711±0.0063
Segment	0±0	0.5448±0.0217	0.5688±0.0070	0.4915±0.0314	0.6105±0.0084
Umist	0.4223±0.0278	0.4616±0.0196	0.4456±0.0257	0.4334±0.0292	0.5886±0.0013
Wineq	0.4501±0.0272	0.4661±0.0042	**0.4823±0.0122**	0.4436±0.0001	0.4565±0.000
Yale	0.0998±0.0079	0.0728±0.0000	0.2805±0.0032	0.2155±0.0070	0.3086±0.0003

Algorithm	PNMF	NSC	ONMF	LSD	NMFR
Abalone	0.404±0.0014	0.3886±0.0009	0.3838±0.0003	0.3889±0.0011	0.3726±0.003
AR	0.1776±0.0070	0.1837±0.0020	0.1383±0.0077	0.1718±0.002	0.162±0.0022
AT&T	0.8±0.0048	0.7994±0.0058	0.7981±0.0039	0.8106±0.0074	**0.8100±0.000**
Balance Scale	0.557±0.0252	0.5621±0.0251	0.5509±0.0392	0.5463±0.0515	0.5691±0.0376
Coil	0.735±0.016	0.8063±0.000	0.6698±0.0144	0.7542±0.0058	**0.8505±0.0018**
Fault	0.2468±0.000	0.2793±0.0001	0.2463±0.0144	0.2524±0.000	0.2341±0.0008
Jaffe	**0.9108±0**	**0.9108±0.000**	**0.9108±0.000**	**0.9108±0.000**	**0.9108±0.000**
Movement Libras	0.500±0.0019	0.4701±0.0019	**0.5000±0.000**	0.4966±0.0065	0.4611±0.000
ORL	0.6625±0.000	0.6617±0.0066	0.6653±0.0092	**0.6764±0.0022**	0.6719±0.0017
Segment	0.5066±0.0367	0.6114±0.0034	0.5042±0.0069	0.6416±0.0052	**0.7333±0.000**
Umist	0.5130±0.000	0.6155±0.0005	0.5757±0.000	0.6139±0.000	**0.6806±0.033**
Wineq	0.4134±0.000	0.4459±0.0000	0.4175±0.000	0.4002±0.000	0.3938±0.000
Yale	0.2688±0.000	**0.3140±0.000**	0.2585±0.000	0.3007±0.0014	0.2962±0.0016

Table **6-3**: The average range for clustering accuracy experiment according to table **6-2**.

	Abalone	AR	AT&T	Balance Scale	Coil	Fault	Jaffe	Movement Libras	ORL	Segment	Umist	Wineq	Yale	**Average**
NSC	12	4	5	5	2	6	1	7	6	4	2	9	1	**4.923**
NCUT	13	5	3	6	3	8	1	8	3	5	4	7	2	**5.230**
LSD	11	8	1	9	4	9	1	3	1	2	3	14	3	**5.307**
NMFR	15	9	2	4	1	12	1	9	2	1	1	15	4	**5.846**
Tukey KCNMF	7	3	7	12	7	1	7	4	8	6	9	1	5	**5.923**
PNMF	10	7	4	7	5	10	1	1	5	10	6	13	6	**6.538**
RBF KCNMF	3	1	8	11	9	3	9	6	7	3	13	6	8	**6.692**
Andrews KCNMF	1	6	13	2	11	7	8	5	9	8	8	4	15	**7.461**
ONMF	14	12	6	8	6	11	1	1	4	11	5	12	7	**7.538**
RMNMF	2	2	9	3	14	4	11	15	15	13	12	10	9	**9.153**
RBF KKMeans	4	11	10	13	10	14	13	13	11	9	9	2	10	**9.923**
Tukey KKMeans	5	13	14	15	8	2	10	12	13	12	7	5	14	**10**
Andrews KKMeans	6	15	15	1	15	13	14	11	12	7	11	3	13	**10.461**
ConvexNMF	9	10	12	10	12	15	12	14	10	15	14	8	11	**11.692**
KMeans	8	14	11	14	13	5	15	10	14	14	15	11	12	**12**

$$\chi_F^2 = \frac{12 * (13)}{15(15+1)} \left(1021.49 - \frac{15(15+1)^2}{4} \right)$$

$$\chi_F^2 = 39.9653$$

$$F_F = \frac{(13-1)39.9653}{13(15-1) - 39.9653}$$

$$F_F = 3.3765$$

With 15 algorithms and 14 databases, we found that F-Fisher will be $n = 15-1 = 14$ and $(15-1)*(13-1) = 204$ degrees of freedom. In the table, it yielded 1.7510. With this, we ruled out the null hypothesis that stated that there is no significant statistical difference in the accuracy of the methods. In light of the above results, a Nemenyi post-hoc test was performed in order to attempt to determine whether there was a significant difference between two algorithms. To this effect, two algorithms were significantly different if the range average differed by at least one critical difference:

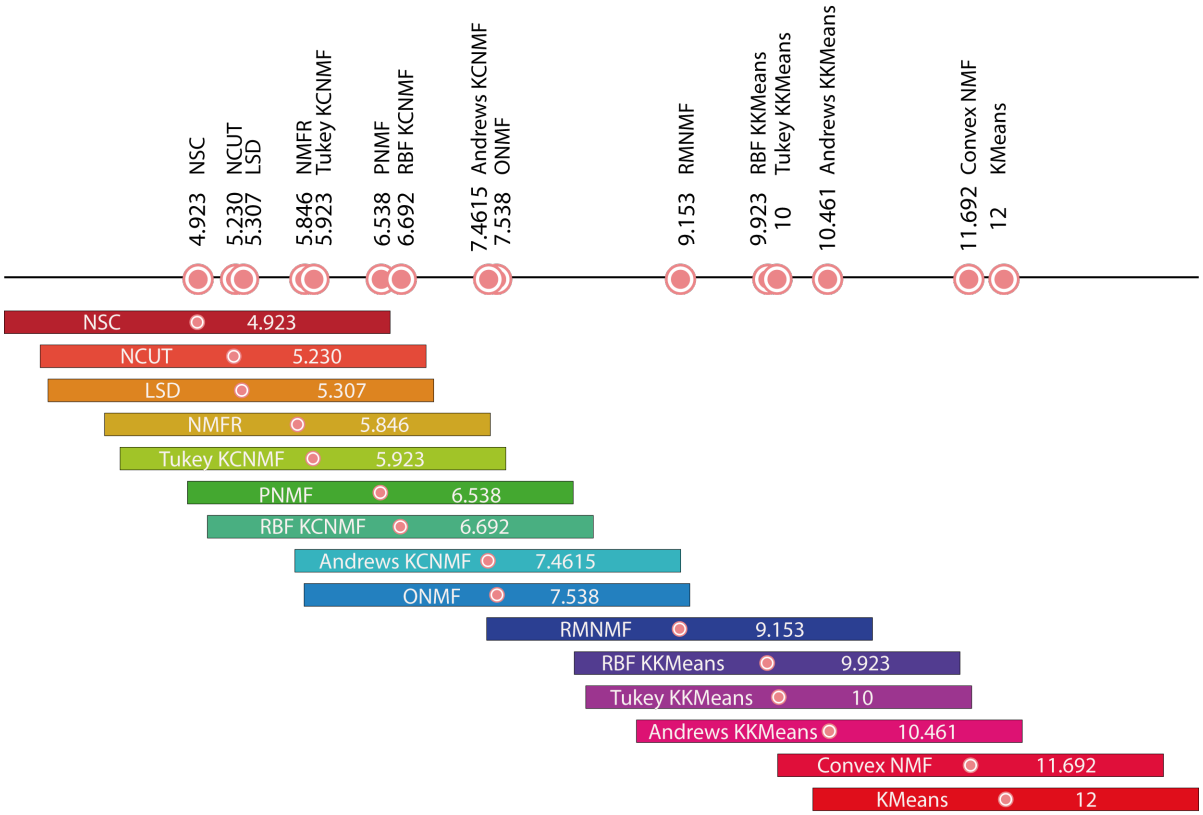

Figure **6-1**: indicated Pair-wise Nemenyi-test with a critical difference of 3.4 in results of clustering accuracy in table **6-2**. NSC was the algorithm with a less average range (4.923) followed by NCUT with 5.230 of average range. The worst method according to the average range is k-means with 12.

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

Where $k = 18$ and $N = 13$. q_α is obtained as the t standardized distribution, divided by $\sqrt{2}$.

Table **6-4**: Critical Distance

#Classifiers	2	3	4	5	$\cdots$	12	13	14	15
$q_{0.05}$	1.96	2.34	2.56	2.72	$\cdots$	3.273	3.32	3.36	3.40

Upon conducting a pair-wise Nemenyi-test, it was concluded that no significant difference could be found amongst the 15 algorithms compared.

6.4 Conclusions

In this chapter, a systematic comparison between state-of-the-art algorithms and robust kernels built under proposition 3.2.1 was presented. It was found at first glance that the kernel algorithms; kernel k-means and kernel Convex NMF, performed better when a robust kernel was used like Gaussian or Tukey's robust kernel compared with the use of a linear kernel. Furthermore, it was found in the experiment that Tukey's robust kernel improved the results obtained by the Gaussian kernel. This would imply that this kernel could be used in other domains where the Gaussian kernel had been successful such as support vector machines. Finally, when comparing the state-of-the-art algorithms with Convex NMF and Tukey's kernel there was not a significant difference found. Something worth noting was that the two algorithms RMNMF and Random Walks NMF used the framework of augmented Lagrangian optimization, which could be used to improve the results of kernel Convex NMF. Another possibility was to kernelized the two algorithms RNMF and RMNMF in order to use the three new robust kernels.

Chapter 7

Conclusions and Future Works

Unsupervised methods in data analysis are used to find patterns in data samples. Due to the high dimensionality and increase in the number of data samples, these patterns are difficult to obtain in a robust way. Robustness in the statistical sense is treated as shown in Chapter 2 in the fact of learning patterns, despite that data do not come from a normal distribution, or contain atypical values due to possible contamination. In machine learning, it is common to find images containing occluded pixels, deviations in light and camera movements. A typical example of occluded pixels in an image dataset is showed in Chapter 4. To work in these databases containing atypical values, in this thesis we based our work on the Huber, Hampel and Cauchy work, where robust M-estimators are constructed. These estimators allow making the residual of the extreme atypical values grow not too fast as in the case of least square.

Clustering tries to find subgroups within the dataset that contain some similarity between each sample. In Chapter 2 we have discussed several methods to perform clustering; among them is k-means which tries to find k centroids and minimizes the quadratic distance to each point within the subgroup k. This construction method of the objective function is not robust since by taking only one point to infinity will bias the cluster's median estimation to infinity. The nonnegative matrix factorization splits matrix X into two matrices F and G, so that the matrices are nonnegative. As for the NMF, and the Frobenius norm is used to optimize, the residuals become more significant as they move away from the average, with the corresponding optimization bias.

Different modifications have been made to these two algorithms. For example: kernel k-means uses the kernel matrix to perform the learning and Convex NMF restricts the F matrix in NMF in such a way that it relapses in space spanned by the X data. There is also the kernelized version of Convex NMF which was used in this work with Gaussian, Tukey's, Andrews', and Huber's kernel showing good performance in clustering tasks. Generally, the databases come with atypical values so new methods were developed to fight against pollution. Some of them are the RMNMF that uses the Laplacian graph framework to make the optimization smooth and the use of the $L_{2,1}$-norm to decrease the

weight of the atypical values and the NMFR that uses spectral clustering for optimization and adds a parameter of regularization lambda in order to minimize the trace of the optimization. Both RMNMF and NMFR use a similarity matrix. This matrix is a kernel as long as it satisfies the positive semi-definite property explained in 2.3.1. In the case that matrix A is not semi-definite positive, the matrix A becomes indefinite where the inner product can be either positive or negative as seen in section 2.3.5. Although RMNMF is presented as a robust algorithm with the use of $L_{2,1}$-norm, in the experiments performed in Chapter 4, it has found that the accuracy of the algorithm was not robust.

In Chapter 3, it has been shown that estimating the mean in the feature space with the RBF kernel is like to estimating the mean in the data space with the Welsch M-estimator. Consequently, the pre-image was defined in the same way as the rbf kernel case in which a centroid in the feature space may not have an exact pre-image. With the help of a proposition 3.2.1, a framework was obtained whereby new kernels were built. M-estimators have been used recently in different areas of machine learning, pattern recognition, and data mining. This new connection between the M-estimators and the kernels provides the possibility of working with contaminated data in a non-linear space, opening new possibilities in the area of machine learning. As seen in chapter 5, three new kernels are built; Andrews', Tukey's and Huber's robust kernel. Although only the rbf kernel satisfies the positive semi-definite property, it was possible to discover, in a practical way, that the kernels of Tukey's and Andrews' work quite well. Moreover, without the help of Convex NMF as well as NSC and NCUT it could not have been statistically proven that significant difference between their methods existed, see ec.

During the experimentation, it was found that when generating uniform noise in three Gaussian distributions, k-means moved away from the real centroids whenever we increased the contamination. On the contrary, when using a robust kernel like the rbf kernel, the kernel k-means kernel version of k-means the estimation of clusters is no bias despite the contamination. RMNMF, which uses the $l_{2,1}$ standard to increase the contamination (increase occlusion of the images) increases the bias in greater proportion to the Convex NMF Kernel with the Gaussian kernel. Tukey's and Andrews' robust kernel show good results in a machine learning task called clustering. In the experimentation of Chapter 6, it was found that the NSC algorithm was the best one to behave among the databases, although when performing the Nemenyi test on paired data, no significant difference between the algorithms could be found.

For future research, it is necessary to build new algorithms that escalate with new robust kernels. These kernels have the particularity of being sparse in accordance with the regularization parameter. Finally, other suggestions include optimization conducted with sparse matrices instead of full ones. Additional testing in other areas such as dimensionality reduction with PCA or classification with support vector machines is needed to show the superiority of using this type of robust kernels for basic machine learning tasks. These kernels have the property of being sparse according to the regularization parameter. With

this, you could think of a work in which the optimization is using sparse matrices instead of full ones. On the other hand, one could think of using some of the four techniques indicated in *Similarity-based Classification: Concepts and Algorithms [46]* to make the kernels become Mercer's kernels. In the case of wanting Andrews, Tukey's and Huber's robust kernels to show properties of the Mercer's theorem, one could think of four possible solutions that were not tested in this work. These possibilities are: spectrum clip, spectrum flip, spectrum shift and spectrum square. It should be noted that the LSD, NMFR, and PNMF algorithms use a similarity matrix for learning and this could be a kernel. For experimentation, the matrix of k-nearest-neighborhood was used. More could be thought of using the algorithms of LSD, NMFR and PNMF, with Andrews', Tukey's, RBF and Huber's robust kernels.

Bibliography

[1] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. 2000.

[2] J. D. Jr and R. Welsch, "Techniques for nonlinear least squares and robust regression," *Communications in Statistics-Simulation ...*, 1978.

[3] C. Goodall, *M-estimators of location: An outline of the theory*, vol. 5. New York: Wiley, 1983.

[4] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*, vol. 114. John Wiley & Sons, 2011.

[5] P. Huber, *Robust statistics*. 2011.

[6] F. R.Hampel, *Robust statistics*. 1985.

[7] H. Rieder and P. Huber, *Robust statistics, data analysis and computer intensive methods*. 1996.

[8] D. E. Tyler, "Robust statistics: Theory and methods," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 888–889, 2008.

[9] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*, vol. 589. John Wiley & Sons, 2005.

[10] C. Ding, T. Li, and W. Peng, "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing," *Computational Statistics & Data Analysis*, vol. 52, no. 8, pp. 3913–3927, 2008.

[11] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 45–55, 2010.

[12] Z. Yang and E. Oja, "Linear and nonlinear projective nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 21, no. 5, pp. 734–749, 2010.

[13] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 126–135, ACM, 2006.

[14] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[15] C. Ding, T. Li, and M. I. Jordan, "Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 183–192, IEEE, 2008.

[16] R. Arora, M. Gupta, A. Kapila, and M. Fazel, "Clustering by left-stochastic matrix factorization," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 761–768, 2011.

[17] M. G. Genton, "Classes of Kernels for Machine Learning : A Statistics Perspective," vol. 2, pp. 299–312, 2001.

[18] G. E. Fasshauer, "Positive definite kernels: past, present and future," *Dolomite Research Notes on Approximation*, vol. 4, pp. 21–63, 2011.

[19] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," pp. 551–556, 2004.

[20] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel pca and de-noising in feature spaces.," in *NIPS*, vol. 4, p. 7, Citeseer, 1998.

[21] J. T.-y. Kwok and I. W.-h. Tsang, "The pre-image problem in kernel methods.," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 15, pp. 1517–25, Nov. 2004.

[22] C. S. Ong, X. Mary, S. Canu, and A. J. Smola, "Learning with non-positive kernels," in *Proceedings of the twenty-first international conference on Machine learning*, p. 81, ACM, 2004.

[23] R. N. Davé and R. Krishnapuram, "Robust clustering methods: a unified view," *Fuzzy Systems, IEEE Transactions on*, vol. 5, no. 2, pp. 270–293, 1997.

[24] P. A. Forero, V. Kekatos, and G. B. Giannakis, "Outlier-aware robust clustering," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 2244–2247, IEEE, 2011.

[25] Z. Chen, X. Shixiong, and L. Bing, "A robust fuzzy kernel clustering algorithm," *Appl. Math*, vol. 7, no. 3, pp. 1005–1012, 2013.

[26] M.-S. Yang, K.-L. Wu, J.-N. Hsieh, and J. Yu, "Alpha-cut implemented fuzzy clustering algorithms and switching regressions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 3, pp. 588–603, 2008.

[27] B. Scholkopf, S. Mika, C. J. Burges, P. Knirsch, K. Muller, G. Ratsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1000–1017, 1999.

[28] F. H. Ruymgaart, "A robust principal component analysis," *Journal of Multivariate Analysis*, vol. 11, no. 4, pp. 485–497, 1981.

[29] S.-A. Berrani and C. Garcia, "Robust detection of outliers for projection-based face recognition methods," *Multimedia Tools and Applications*, vol. 38, no. 2, pp. 271–291, 2008.

[30] F. De la Torre and M. J. Black, "Robust principal component analysis for computer vision," vol. 1, pp. 362–369, 2001.

[31] M. Debruyne, M. Hubert, and J. Van Horebeek, "Detecting influential observations in kernel pca," *Computational Statistics & Data Analysis*, vol. 54, no. 12, pp. 3007–3019, 2010.

[32] C. Lu, T. Zhang, R. Zhang, and C. Zhang, "Adaptive robust kernel pca algorithm," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 6, pp. VI–621, IEEE, 2003.

[33] J.-h. Chen, "M-estimator based robust kernels for support vector machines," *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 1, no. 1, pp. 168–171 Vol.1, 2004.

[34] C.-T. Liao and S.-H. Lai, "Robust kernel-based learning for image-related problems," *IET image processing*, vol. 6, no. 6, pp. 795–803, 2012.

[35] L. Zhang, Z. Chen, M. Zheng, and X. He, "Robust non-negative matrix factorization," *Frontiers of Electrical and Electronic Engineering in China*, vol. 6, no. 2, pp. 192–200, 2011.

[36] Z. Yang, T. Hao, O. Dikmen, X. Chen, and E. Oja, "Clustering by nonnegative matrix factorization using graph random walk," in *Advances in Neural Information Processing Systems*, pp. 1079–1087, 2012.

[37] F. A. González, D. Bermeo, L. Ramos, and O. Nasraoui, "On the robustness of kernel-based clustering," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 122–129, Springer, 2012.

[38] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description - ODD '13*, pp. 8–15, 2013.

[39] A. Ben-Tal and S. Bhadra, "Efficient methods for robust classification under uncertainty in kernel matrices," *The Journal of Machine ...*, 2012.

[40] K. Bache and M. Lichman, "UCI machine learning repository," 2013.

[41] A. M. Martinez, "The ar face database," *CVC Technical Report*, vol. 24, 1998.

[42] F. Samaria, "The orl database of faces," *AT&T Laboratories Cambridge*, vol. 1, 1994.

[43] S. K. N. S. A. Nene and H. Murase., "Columbia university image library (coil-20)," *Technical Report CUCS-005-96*, vol. 1, 1996.

[44] M. J. Lyons, "Coding facial expressions with gabor wavelets," *3rd IEEE International Conference on Automatic Face and Gesture Recognition*, vol. 1, 1998.

[45] A. N. Graham Daniel, "Face recognition: From theory to applications," *Face Recognition: From Theory to Applications*, vol. 163, 1998.

[46] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti, "Similarity-based classification: Concepts and algorithms," *Journal of Machine Learning Research*, vol. 10, no. Mar, pp. 747–776, 2009.