Méthodes à noyau pour l'analyse de données de vols appliquées aux opérations aériennes

Nicolas Chrysanthos

▶ **To cite this version:**

Nicolas Chrysanthos. Méthodes à noyau pour l'analyse de données de vols appliquées aux opérations aériennes. Operations Research [math.OC]. Université de Technologie de Troyes, 2014. English. NNT : 2014TROY0030 . tel-03358396

HAL Id: tel-03358396
https://theses.hal.science/tel-03358396

Submitted on 29 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse
de doctorat
de l'UTT

Nicolas CHRYSANTHOS

Kernel Methods for Flight Data Monitoring

Spécialité :
Optimisation et Sûreté des Systèmes

2014TROY0030

Année 2014

utt
université de technologie
Troyes

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Nicolas CHRYSANTHOS

le 24 octobre 2014

Kernel Methods for Flight Data Monitoring

JURY

M. S. CANU	PROFESSEUR DES UNIVERSITES	Président
M. P. BEAUSEROY	PROFESSEUR DES UNIVERSITES	Directeur de thèse
Mme É. GRALL-MAËS	MAITRE DE CONFERNCES	Examinateur
M. R. JENSSEN	ASSOCIATE PROFESSOR	Rapporteur
M. H. SNOUSSI	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. J.-P. VERT	PROFESSEUR MINES PARISTECH	Rapporteur

Personnalités invitées

M. F. FERRAND	INGENIEUR
M. L. KOKANOSKY	INGENIEUR

Abstract

Flight Data Monitoring (FDM), is the process by which an airline routinely collects, processes, and analyses the data recorded in aircrafts with the goal of improving the overall safety or operational efficiency.

The goal of this thesis is to investigate machine learning methods, and in particular kernel methods, for the detection of atypical flights that may present problems that cannot be found using traditional methods. Atypical flights may present safety of operational issues and thus need to be studied by an FDM expert.

In the first part we propose a novel method for anomaly detection that is suited to the constraints of the field of FDM. We rely on a novel dimensionality reduction technique called kernel entropy component analysis to design a method which is both unsupervised and robust.

In the second part we solve the most salient issue regarding the field of FDM, which is how the data is structured. Firstly, we extend the method to take into account parameters of diverse types such as continuous, discrete or angular. Secondly, we explore techniques to take into account the temporal aspect of flights and propose a new kernel in the family of dynamic time warping techniques, and demonstrate that it is faster to compute than competing techniques and is positive definite.

We illustrate our approach with promising results on real world datasets from two partner airlines comprising hundreds of flights.

Résumé

L'analyse de données de vols appliquée aux opérations aériennes ou "Flight Data Monitoring" (FDM), est le processus par lequel une compagnie aérienne recueille, analyse et traite de façon régulière les données enregistrées dans les avions, dans le but d'améliorer de façon globale la sécurité.

L'objectif de cette thèse est d'élaborer dans le cadre des méthodes à noyau, des techniques pour la détection des vols atypiques qui présentent potentiellement des problèmes qui ne peuvent être trouvés en utilisant les méthodes classiques.

Dans la première partie, nous proposons une nouvelle méthode pour la détection d'anomalies. Nous utilisons une nouvelle technique de réduction de dimension appelée analyse en entropie principale par noyau afin de concevoir une méthode qui est à la fois non supervisée et robuste.

Dans la deuxième partie, nous résolvons le problème de la structure des données dans le domaine FDM. Tout d'abord, nous étendons la méthode pour prendre en compte les paramètres de différents types tels que continus, discrets ou angulaires. Ensuite, nous explorons des techniques permettant de prendre en compte l'aspect temporel des vols et proposons un nouveau noyau dans la famille des techniques de déformation de temps dynamique, et démontrons qu'il est plus rapide à calculer que les techniques concurrentes et est de plus défini positif.

Nous illustrons notre approche avec des résultats prometteurs sur des données réelles de deux compagnies aériennes partenaires comprenant plusieurs centaines de vols.

Je dédie cette thèse à mes parents.

Remerciements

Les travaux présentés dans cette thèse ont été conduits au sein de l'équipe Airline Services de la division avionique (DAV) de Sagem Défense Sécurité, en partenariat avec le Laboratoire de Modélisation et de Sûreté des Systèmes (LM2S) de l'Université de Technologie de Troyes (UTT).

Tout d'abord je tiens à remercier monsieur Jean-Philippe Vert et monsieur Robert Jenssen d'avoir accepté de relire cette thèse et d'en être les rapporteurs. Je tiens à remercier également monsieur Stéphane Canu d'avoir accepté d'être président du jury. Je remercie également tous les membres du jury d'avoir accepté d'assister à la présentation de ce travail.

Je tiens à exprimer ma profonde gratitude envers monsieur Pierre Beauseroy et monsieur Hichem Snoussi, mes directeurs de thèse ainsi qu'à madame Edith Grall; qui ont suivi et supervisé mes travaux durant cette thèse. Durant ces trois années ils ont su à la fois me conseiller et me guider tout en me laissant suffisamment d'indépendance pour expérimenter des idées plus personnelles. Je tiens à souligner leurs qualités tant scientifiques qu'humaines, leurs précieux conseils tout au long de ces trois années m'ont permis de progresser et d'avoir une meilleure compréhension du métier de chercheur et du monde académique en général.

Je tiens aussi à exprimer ma gratitude envers monsieur Fabrice Ferrand, qui a été l'instigateur de cette thèse au sein de Sagem, ainsi que mon superviseur technique.

Je tiens à remercier monsieur Bruno Larois, alors chef de l'unité, pour m'avoir accueilli dans son équipe, et surtout pour avoir tout mis en

œuvre pour que ma thèse se passe dans de bonnes conditions.

Je souhaiterais aussi exprimer ma gratitude à monsieur Daniel Duclos, qui a suivi avec un œil bienveillant ma thèse durant ces trois années et qui m'a fait profiter de son expérience dans le domaine de la recherche industrielle.

Je voudrais exprimer ma profonde gratitude envers Laurent Kokanosky, à la fois pour m'avoir supervisé d'un point de vue métier, pour avoir mis à ma disposition l'ensemble des moyens matériels et humains pour le bon déroulement de cette thèse ; et surtout pour avoir cru en le bien-fondé de cette approche.

Je dirige mes remerciements vers toute l'équipe Airline Services, la "Team Cassiopée": Jean-Philippe et ses nombreux conseils sur la vie au quotidien chez Sagem, Victor et Ivan les analystes FDM qui m'ont introduit à ce domaine au cours de très nombreuses discussions. Yannick, ingénieur en opérations aériennes, qui s'est intéressé de près à mes travaux et qui m'a aussi beaucoup éclairé sur ces problématiques. Valérie, Elsa et Damien pour leur bonne humeur contagieuse et les nombreuses pauses café sur la terrasse. Edouard bien sûr, qui très vite est devenu un ami cher, qui a su m'apporter conseils et support dans les moments difficiles, et dont la droiture et la finesse d'esprit forcent le respect. Pierrick l'apprenti, humble artisan et "petite main" de Sagem, à la culture geek sans borne, et qui m'a beaucoup aidé lorsqu'il a fallu utiliser les techniques les plus avancées de Python, ou déployer une VM Linux en environnement hostile.

Je voudrais aussi remercier toute l'équipe AGS, pour leur aide et leur disponibilité en toutes circonstances: Michel, Jean-Philippe et Sandrine. Sans oublier aussi l'équipe des commerciaux et en particulier Kevin et Xavier, pour leur aide précieuse en beaucoup de circonstances et en particulier quand il a fallu obtenir des NDA ....

Je tiens tout particulièrement à remercier chaleureusement Hélder Mendes de la compagnie aérienne TAP, son enthousiasme pour tout ce qui touche à l'application des mathématiques au domaine aérien

et pour sa bonne humeur en toutes circonstances. Je voudrais aussi remercier Georges Michaud de la compagnie aérienne Transavia, pour avoir cru en mes travaux et pour m'avoir encouragé dans cette voie.

Je voudrais terminer sur une note plus personnelle en commençant par remercier de tout mon cœur Camille, qui pendant trois ans m'a soutenu, encouragé et réconforté dans les moments les plus difficiles. Je n'aurais pas pu terminer ces travaux sans elle. Enfin, mes pensées vont bien sûr à mes amis et à ma famille. Leur confiance, leur soutien et leurs encouragements en toutes circonstances m'ont permis d'arriver où je suis maintenant. Je ne saurai jamais leur être suffisamment reconnaissant.

Je voudrais tout particulièrement remercier ma mère, qui a mis toute l'énergie et fait tous les sacrifices nécessaires pour l'éducation de ses enfants.

Important Note

For confidentiality reasons we could not disclose the name of the two airline companies which have provided data as well as feedback for these studies. These airlines will thus be called respectively `Airline1` and `Airline2` in this work. Whenever necessary we will change the name of cities or maps if these could be used to identify the companies.

Contents

List of Figures

Nomenclature

Flight Data Monitoring

AGS Analysis Ground Station

CAA Civil Aviation Authority

EUROCAE European Organization for Civil Aviation Electronics

FDM Flight Data Monitoring

ICAO International Civil Aviation Organization

QAR Quick Access Recorder

SFIM Société de Fabrication d'Instruments de Mesure

SO Safety Officer

SOP Standard Operating Procedures

Machine Learning and Kernel Methods

KECA Kernel entropy component analysis

KPCA Kernel principal component analysis

PD Positive definite

RKHS Reproducing kernel Hilbert space

SVM Support vector machine

Chapter 1

Introduction

1.1 Introduction to the Industrial Context

1.1.1 Flight Safety

It is widely agreed that aviation is one of the safest means of transport, at least in terms of fatalities per kilometers. However, the aviation community is under constant pressure to achieve safety improvement. As seen in Figure 1.1, the accident rate has been relatively stable since the early 80s. However, the volume of air transportation traffic has surged in the last two decades, going from about twenty-five million flight hours per year to more than fifty million in 2012 [Boeing, 2013] and will very likely continue to grow. This will result in an overall increase in the number of accidents and fatalities. This increase in the number of accidents is unacceptable for neither aircraft manufacturers nor airlines. Aside from the tragic human losses, each accident comes with huge financial and economical cost, from the replacement and loss of revenue of the aircraft to the media exposure following any accident.

Flight failures are often due to a combination of factors, either technical, such as for example an engine or a structural failure; or natural events such as lightning, ice, bird strikes; or human factors such as crew errors, organizational failure, improper communications etc.

The flight industry as a whole has been working on these issues since the very beginning of commercial airborne transportation. Aircraft and component

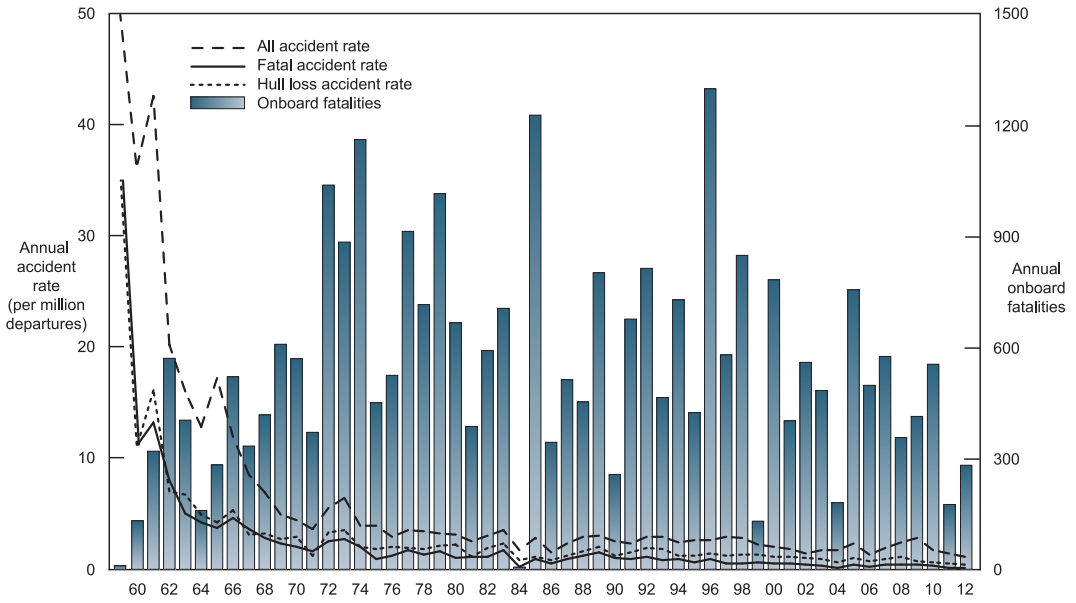


Figure 1.1: Evolution of accident rates and fatalities from 1959 to 2012. Source: [Boeing, 2013]

manufacturers have designed ever more reliable and safe aircrafts, the regulation has evolved, and a number of other innovations were developed such as navigation aids, instrument flight etc.

Among these innovations one of the most important is the invention of flight data recorders (commonly called “black boxes”) that are embedded into aircrafts.

1.1.2 Flight Data Recorders

The first flight data recorder was invented by François Hussenot and Paul Beaudoin in 1939. It was a photograph-based flight recorder; the image on the photographic film was made by a thin ray of light deviated by a tilted mirror according to the magnitude of the data to record. The photographic foil had to be revealed, much in the same way as analog photographies. Another technique was the metal foil engraver, where the continuous movement of a metallic stylus is used to record parameters. Contrary to the photographic technique the metal foil could be read almost directly. Around 1954 in Australia Dr David Warren produced the first recorder combining voice and data, using an innovative magnetic

recording medium. At this time, flight data recorders were used to investigate crashes, and thus these devices were put in fire and shock proof cases. Modern recorders must comply with the specifications from the European Organization for Civil Aviation Electronics (EUROCAE), such as for example [EUROCAE, 2003] resistance to impact shock, penetration, crush, high temperature, deep sea pressure and fluid immersion. One of the manufacturers of these types of recorder was the *Société de Fabrication d'Instruments de Mesure* [1] (SFIM) More information regarding flight data recorders can be found in [Mendes, 2012].

1.1.3 Flight Data Monitoring

1.1.3.1 Presentation

From the 1960's some airlines found it beneficial to analyze crash recorder data to assist for maintenance. Multiple replays tended to reduce the lifespan of crash recorder and consequently Quick Access Recorder (QAR) were introduced to record data in parallel with the crash recorder, and the data were used for the prevention of accidents. As storage capacity increased over the years, it became possible to store data from several flights. These technological advancements lead to the first Flight Data Monitoring (FDM) programs.

As defined by the Civil Aviation Authority (CAA) FDM is the "systematic, pro-active and non-punitive use of digital flight data from routine operations to improve aviation safety". The idea is to make use of the data recorded on flight, not only after a crash for investigation, but also for the prevention of accidents.

Since 2005, by legislation of the International Civil Aviation Organization (ICAO) it is now mandatory to carry a FDM program for any commercial aircraft of more than 27 tonnes.

1.1.3.2 The Four Stages of FDM

A typical FDM program can be divided into four phases. First, the transfer part, which consists in recovering data from the aircraft. This can be done in many ways, for example by downloading data from the aircraft QAR into a portable

[1] Now known as SAGEM Défense Sécurité of the SAFRAN Group.

solid state drive or by wirelessly uploading data to the operator office via radio or satellite.

The second phase is the processing phase, where raw data frames are first decoded into engineering values. Then, in a classical *event-driven* analysis technique, a specific software such as SAGEM's *Analysis Ground Station* (AGS) sifts through the data in search of specific events, which most of the time are defined as threshold exceedances on certain parameters. Events can be defined either by the manufacturer, the airline or the legislation. For example an engine manufacturer may define thresholds on engine parameters to ensure that the engine stays within the physical limits it was designed to. The airline on another hand will be more interested for example in parameters related to the descent profile in order to identify deviations from the Standard Operating Procedures (SOP).

The third phase is the validation phase. Each event that was detected in the previous phase is examined by an analyst, an expert in flight procedures. Its role is to assess if the events are of real safety concern. The software provides a number of tools in order to assist the analyst, such as search facilities, graphical representations, charts, key statistics etc. If an event is deemed to be important, it is reported to the airline, and in particular to the Safety Officer (SO), a pilot in the airline company who stands between the airline, the FDM provider and other pilots.

The final phase is the remedial action phase. Informed of a critical event, the SO is responsible for taking action in his company. The FDM culture is a *just* and *non-punitive* culture. Pilots and crews are in general quite reluctant to see their data thoroughly studied as is done in a FDM program, this is because of a fear of punishment, even though in the vast majority of cases an accident is due to a systemic or organizational failure and not to a single individual. As a consequence, most of the time, data are anonymized so that it is not possible to recover the pilot or crew identity of a flight. Thus, the actions consist in modifying the training procedures or SOPs in order to reduce the identified risk.

Note that FDM is not restricted to safety aspect, operational efficiency such as fuel consumption is also an important subject. As outlined by these four phases, the goal of FDM is a continuous improvement of both safety and efficiency through preventive and corrective actions.

1.1.3.3 Limits of the Current Approach

There are however some limits to the current event-driven approach, and these have pushed operators to search for more advanced techniques.

The first limitation is that a strictly event-based FDM program can only by design detect problems which were foreseen during the creation of the event table. This is unfortunate since from a safety perspective it would be very valuable on the contrary to detect *unexpected* problems.

The second limitation is that as the instruments and flight recorders gain in sophistication there are ever more parameters that are recorded in modern aircraft. As a comparison, the first flight data recorders that were used in a FDM program could store a handful of parameters, whereas in a modern aircraft such as the A380 more than 2000 parameters are recorded, some at a frequency of more than 32 Hz. There is thus a massive increase in the volume of data that is recorded and can be studied in a FDM program, however much of this data is stored but not used in most FDM program. This is because operators rely on tried-and-true fundamental parameters that are well-known by actors in the industry. This still leaves however a huge volume of data that is available but not used, and which represents, *a priori*, a great value for airline companies.

1.2 Goal of Thesis

The goal of the work described in this thesis is to create a method to detect *atypical flights* among a dataset of hundreds or thousands of flights.

An atypical flight is a flight which is in a sense different from most other flights in a dataset, consequently such a flight may present operational or safety issues and thus needs to be studied by an expert, in the same way that flights which exhibit classical events are studied by experts. Such method should be complementary with classical event-based techniques, with the hope that it detects flights that are overlooked by traditional analysis, and with a sufficiently low false-positive ratio.

From a practical point of view, the method could be implemented in a software program and used with the following steps by an operator:

1. Choose a set of related flights (for example all flights that landed on a certain runway last year),
2. Choose a set of parameters (for example all parameters related to the descent profile such as altitude, vertical speed, flaps and slats etc.),
3. Launch the analysis,
4. After some computation time the program would produce a complete report with all detected atypical flights.

The report could provide scores measuring how likely a given flight is atypical, as well as a number of indicators, statistics and most importantly graphical representations. This report would be used by a flight operation analyst to study the atypical flights and report to the airline if needed.

We have a number of requirements for this method:

- The method should be *unsupervised*. As will be detailed, developing an unsupervised method for novelty detection is of great value for FDM. "Unsupervised" means that we will be able to detect atypical flights without a training set of normal and abnormal flights. Producing such a set is a long, costly and tedious process; since field experts would have to review dozens of flights and assign them a label. This would have severely undermined our ability to work in collaboration with airline companies: we only have to ask them for data. Asking them to additionally put expert resources to the task of labeling flights would probably never have been met with a positive response. Besides, we *could* have fed our algorithm with the labels recovered from classical event-based analysis; but this would have had the effect of implicitly calibrating our algorithms to detect the very same problems that are *already* detected by classical analysis, whereas the method we wish to design should be able to detect *unforeseen* problems.
- The method should be able to study any combination of parameters. The parameters can be of any type (continuous, angular, binary/discrete) and

of any frequency. The method should be able to detect abnormalities that stem from "synchronization" problems between parameters: a flight may be normal with respect to parameter A and normal with respect to parameter B, but could be abnormal when studying parameters A and B in combination.

- The method should be able to take into account the temporal aspect of a flight, which means that preferably it should not rely solely on *features* associated to flights. This is an issue that we will detail in Section 1.5 and which will be the subject of Chapter 6.
- The method should be fast enough to run a study with a moderate number of parameters (~ 20), around a thousand flights, and covering a whole flight phase such as the landing phase in less than a couple of hours with a standard recent computer.

1.3 Mathematical Approach

These limits of the current event-driven approach have fostered research into more statistical approaches of FDM; the idea being that instead of measuring deviations from the SOP by a set of fixed events, the flights themselves would define what is normal or abnormal.

1.3.1 One Flight as One Sample

When carrying a statistical study, the first step is usually to define a dataset, or more exactly a set of *samples*, and the associated factors (sometimes also called features, or dimensions, depending on the context). Most often statistical methods rely on an assumption such as:

$$X_1, \ldots, X_n \quad \text{i.i.d.} \tag{1.1}$$

Where $X_1, \ldots, X_n$ are the samples, and i.i.d. stands for *independent, identically distributed*; which means that each sample is supposed to be generated from the

same probability distribution, and that samples are generated independently from each others.

The approach we have chosen for the work in this thesis is to carry statistical methods where *one flight is one sample*.

Note that it is possible to carry statistical algorithms on flight data using other approaches. For example in the field of structural health monitoring, usually one sample is the data recorded at one instant (what we shall later define as a *time-sample*).

It is our belief that for the field of FDM, and contrary to health monitoring, it is better to consider that one flight constitute one sample.

Using this approach, one can see that the dataset is very much in accordance with the assumption defined in Equation 1.1. The first reason is that pilots and crew are supposed to follow a certain flight procedure, so the "identically distributed" part of the assumption is sensible. Of course there will be many variations among flights, but this only means from a statistical point of view that the distribution will be diffuse to some degree. The second reason is that each new flight is supposed to be conducted after a significant number of check-ups, concerning both the aircraft and the crew, which means that each new flight can reasonably be considered independent from other flights.

Using this approach, and as a consequence of the law of large numbers, a flight which is considered "central" from a statistical point of view should be "normal" from a domain perspective, and conversely a flight which is a statistical outlier is likely to be "abnormal" from a domain perspective. Of course this is only true as far as the mathematical and statistical methods that we design are suited to the type of problems that we want to discover.

1.4 Introduction to Kernel Methods

Among all statistics and machine learning methods, we have chosen the class of kernel methods, for reasons that will hopefully be clear at the end of this introduction. In this section we present the most important concepts of kernel

	Sepal Length	Sepal Width	Petal Length	Petal Width
Flower 1	5.1	3.5	1.4	0.2
Flower 2	7.0	3.2	4.7	1.4
Flower 3	6.3	3.3	6.0	2.5

Table 1.1: Example of a tabular dataset

methods. This is intended for readers who may be domain expert, have some mathematical background but never had to deal with any kind of machine learning or pattern recognition method.

1.4.1 Feature-based Representation of a Dataset

Most pattern recognition algorithms work with a *feature-based* representation of the dataset. This means that each individual entity in the dataset, which we call *samples* in statistical terms is represented by a *fixed* number of *factors*, that are also called features or dimensions. Suppose for example that one is studying a group of fifty flowers of the *Iris* genus, and the dataset contains for each flower 4 features: the length and the width of the sepals and petals.

The advantage of having the same number of features for every sample is that the dataset can be described in a table, like in Table 1.1 for example. If in addition factors can be represented by real numbers (and we shall see that is not always the case, especially for the field of FDM, as will be the subject of Chapter 5) this means that each sample can be represented as *a point in vector space*. This vector space representation is powerful because it is very convenient from a mathematical point of view, and because many algorithms can be defined in a geometric fashion. In this representation for example, a dataset with twelve features would be represented as points (or vectors) in a twelve-dimensional vector space. In the remainder of this thesis we shall refer to this type of data as *vector data*.

So long as the number of features is *finite*, the theory of Euclidian spaces let us carry geometrical methods in spaces with any number of dimensions much in the same way as in three dimensional spaces. Even though a dataset never practically contains an infinite number of features, we shall see later on that it may be useful

to project data into a feature space of higher or even infinite dimensions. In this case the mathematical tools that we shall use are not Euclidian spaces anymore, but rather Hilbert spaces. Dealing with samples in a Hilbert space is a topic that is treated partly in the first part of this thesis.

1.4.2 Pattern Recognition Illustration: Two-Classes SVM

In this subsection we present the quintessential pattern recognition algorithm, the support vector machine (SVM) [Boser et al., 1992; Cortes and Vapnik, 1995]. We present the SVM in its simplest form, non-kernelized and with linearly separable data. The intent of this subsection is not to go into technical details but rather to present the ideas behind a typical pattern recognition algorithm. This shall be sufficient for the reader to develop a good intuition of such mechanisms.

Like many pattern recognition algorithms, the SVM in its "standard" form works with vector data. The SVM is a supervised machine learning algorithm. It is first trained with a labeled training data set. To each sample $\mathbf{x}_i$ is assigned a label y_i in the set $-1, 1$; which is why this kind of SVM is called a two-class SVM. For example, the dataset could contain the length and width of the sepals and petals of fifty flowers of the *Iris* genus; and -1 might stand for `iris versicolor` while 1 might stand for `iris virginica`. Once trained, the goal of the SVM is to determine the label of unseen data. For example determine if a flower is in the class `iris versicolor` or `iris virginica` just by looking at the length and width of its sepals and petals.

The training phase of the SVM is the process of searching for a *hyperplane*[1] in the feature space that separates the data such that all samples from one class are at one side of the hyperplane and the samples from the other class at the other side. Of course it is only possible to find such a hyperplane when the dataset is indeed separable. Otherwise one has to recourse to using a variation of the SVM called *soft-margin*. Even in the separable case, there is an infinite number of hyperplanes that separate the data. The SVM algorithm consists in finding the hyperplane with maximizes the *margin*, which is the distance from

[1] A hyperplane in a space of dimension d is a linear subspace of dimension $d - 1$; so for example in the two dimensional plane, hyperplanes are merely lines.

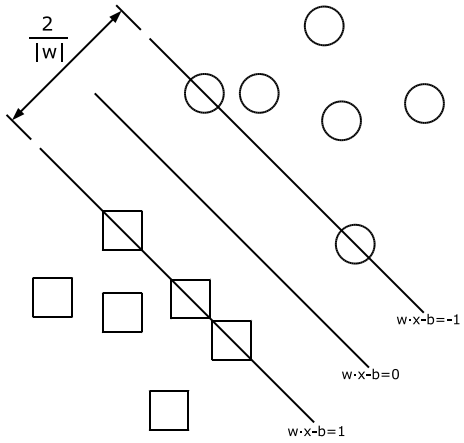


Figure 1.2: Two-class SVM illustration in two dimensions

the hyperplane to the nearest data point of each class. This is illustrated in the two dimensional case in Figure 1.2, where the two classes are the squares and the circles. Mathematically the hyperplane is defined by the following equation:

$$\mathbf{w}^\top \cdot \mathbf{x} + b = 0$$

Let us consider a dataset $\mathbf{x}_i, y_i,\ i = 1, \ldots, n$. Finding this optimal hyperplane is equivalent to solving the following optimization program:

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & L\mathbf{w}, b = \prod^{\mathbf{w}} \prod \\ \text{subject to} \quad & y_i \mathbf{w}^\top \cdot \mathbf{x}_i + b \geq 1,\ i = 1, \ldots, n. \end{aligned}$$

By introducing Lagrangian multipliers α it is possible to express this optimization program in the following dual form:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \tilde{L}\alpha = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{subject to} \quad & \alpha_i \geq 0,\ i = 1, \ldots, n \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned}$$

This second optimization program is computationally easier to solve because the constraints are simpler. After the optimal solution α^* has been found it is possible

to recover the optimal hyperplane:

$$\mathbf{w}^* = \sum_i \alpha_i^* y_i \mathbf{x}_i$$

Only a handful of α_i will be non-zero, the corresponding vectors are called *support vectors*. It is then possible to compute the optimal coefficient b^*, which verifies for any non-zero α_i:

$$b^* = \mathbf{w}^\top \cdot \mathbf{x}_i + y_i$$

1.4.3 The Kernel Trick

In the previous subsection we have seen the SVM in its simplest form, as a linear classifier. However one of the reasons the SVMs have been so popular is because they can be used in the context of a *kernel machine*. Using the famous *kernel trick* [Aizerman et al., 1964], it is possible to implicitly and non-linearly project the data vectors into a space with a higher (and possibly infinite) number of dimensions. The hyperplane is searched in this *transformed feature space*, which is sometimes just called the *feature space* or *kernel space*, while the original space is usually called the *input space*.

Formally the optimization program is the same, except that dot products are replaced by the evaluation of a possibly non-linear kernel.

$$\mathbf{x}_i^\top \mathbf{x}_j \;\rightarrow\; k\mathbf{x}_i, \mathbf{x}_j$$

Generally speaking, kernels are real functions of pairs of samples from the input space, usually denoted $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. Examples of kernels that can be used with vector data are:

$$\begin{aligned}
\text{Gaussian kernel:} \quad & k\mathbf{x}, \mathbf{y} = \exp - \left\| \mathbf{x} - \mathbf{y} \right\|^2 \\
\text{Polynomial kernel:} \quad & k\mathbf{x}, \mathbf{y} = \mathbf{x}^\top \mathbf{y} + b^d \\
\text{Exponential dot product:} \quad & k\mathbf{x}, \mathbf{y} = \exp \mathbf{x}^\top \mathbf{y} \\
\text{Hyperbolic tangent:} \quad & k\mathbf{x}, \mathbf{y} = \tanh \kappa \mathbf{x}^\top \mathbf{y} + c
\end{aligned}$$

The condition required for using a kernel in the framework of the kernel trick is that the kernel must be *positive definite*, which is sometimes abbreviated p.d.:

Definition 1. *A kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite if for any $n \in \mathbb{N}$, and $\forall X_1, \ldots, X_n \in \mathcal{X}^n,\ \forall a_1, \ldots, a_n \neq 0 \in \mathbb{R}^n$:*

$$\sum_{i,j}^{n} a_i a_j k X_i, X_j > 0$$

Although the SVM is the classic example of a kernel algorithm, any algorithm that makes only use of dot products can be "kernelized" using a positive definite kernel. This results from the Moore-Aronszajn theorem [Aronszajn, 1951], which states that for every positive definite kernel on $\mathcal{X}$ there exists a unique *reproducing kernel Hilbert space* (RKHS) $\mathcal{H}$ and a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that:

$$\forall x, y \in \mathcal{X}, \quad kx, y = \coprod \Phi x, \Phi y \tilde{}_{\mathcal{H}} \tag{1.2}$$

Although there is an infinite number of spaces $\mathcal{H}$ that verify Equation 1.2, only one is a RKHS; consequently from now on when we mention the feature space, we implicitly refer to the unique RKHS that is associated to the kernel.

With the kernel trick it becomes possible to extend a linear estimation procedure into a non-linear one, thanks to the mapping Φ. It is not even necessary to compute explicitly the coordinates of the mapping because dot products can be computed from coordinates in the input space thanks to Equation 1.2.

Another widely used algorithm that has been extended using the kernel trick is the principal component analysis method, which thus becomes the kernel principal component analysis (KPCA), as proposed by Schölkopf et al. [1998b]. It is a very powerful method as it can extract non-linear features from a dataset. This algorithm will be studied in details in the first part of this thesis, and is even more important when dealing with non-vector structured data as detailed later.

In the case of vector data, kernel methods can be very useful because sometimes in the input space the data is not separable, as illustrated in Figure 1.3.

The advantage of using a higher dimensional space as the feature space then becomes clear: datasets become more separable as the number of dimensions increases. In the case of the widely used Gaussian kernel for example, it has been demonstrated that the associated RKHS is infinite-dimensional [Steinwart et al.,

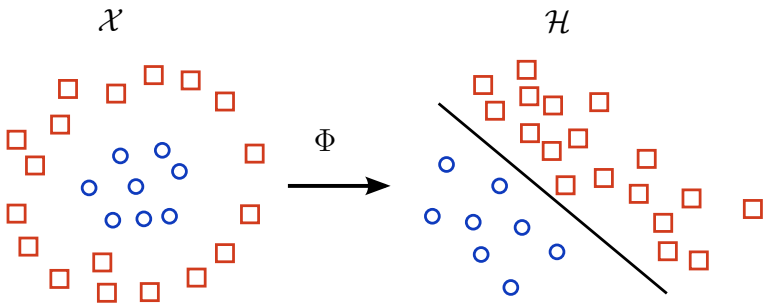


Figure 1.3: Illustration of a mapping Φ from an input space $\mathcal{X}$ to a feature space $\mathcal{H}$.

2006]. An infinite-dimensional space is "rich" enough such that any dataset of any (finite) number of samples can always be separated with a hyperplane [Bousquet et al., 2004], which is not the case in a Euclidian space.

1.4.4 Dealing With Structured Data

There is however another very useful property concerning kernel machines that will be of paramount importance for this work, and that we will develop in the second part of this thesis. In addition to the possibility of non-linearly mapping vector data to higher dimensional spaces, the kernel trick can also be used to map non-vector, structured data to a Hilbert space.

We call "structured data"[1] everything that cannot be described as a mathematical vector, as explained in Section 1.4.1. This may be due to several reasons: the number of features may be different across samples; or a specific feature cannot be described as a number on the real line for example. There may not even be "features" anymore.

In reality vector data are anything but the norm in real-world datasets. One has to deal with such diverse structures as strings, histograms, graphs, time series, images etc. It is clear that one cannot "sum" two graphs, or multiply them by a scalar as is done in vector spaces. One of the reasons so much of the literature in classical statistics deals with tabular/vector datasets is that they are much more

[1] In the rest of this work we shall use indifferently the terms "non-vector" or "structured", they refer to the same concept.

convenient to deal with from a mathematical point of view; the whole topic of linear algebra is dedicated to the study of such spaces, as explained in Section 1.4.1.

Hence the power of kernel methods is that in most cases, the input space $\mathcal{X}$ can be any *arbitrary* space. So long as the kernel k is positive definite, then one can use any kernel method on this dataset and be sure that there exists a RKHS $\mathcal{H}$ to which data are projected, and that dot products of these projected points in the RKHS are simply evaluations of the kernel k.

Although a Hilbert space is not as convenient as a Euclidian space, especially in the case where the number of dimensions is infinite (and with structured data this is most often the case), it is noteworthy that it is possible to retain only the most important dimensions of a dataset by carrying a kernel principal component analysis for example. It is thus possible to recover for each sample a finite[1] number of features, which allows us to consider a structured dataset as a tabular one, thanks to the power of the kernel trick and the kernel principal component analysis.

1.5 Structure of Flight Data

The domain we study in this work is especially suited to kernel methods because the datasets we encounter are indeed highly structured.

1.5.1 Structure of Time-Samples

Firstly, at each instant (frequencies vary from 0.5Hz to 32Hz or more), more than one parameter are recorded. We call the data that is recorded at each instant a *time-sample*. We thus model each time-sample as a *composite structure*, which contains elements of potentially different types.

For example, we might consider a study with 4 parameters: ALT, AIRSPEED, AUTO and PITCH. In this case, ALT and AIRSPEED are *continuous* values and thus can be modeled by real numbers. On the contrary, AUTO is a discrete

[1] By retaining only a finite number of dimensions we loose information, but the quantity of information that is lost can be estimated, as will be described in Chapter 2.

ALT	8500
AIRSPEED	254
AUTO	ON
PITCH	5°

Table 1.2: Structure of a time-sample

parameter whereas PITCH is an angular parameter. An example of time-sample is illustrated in Table 1.2.

Although the distinction between continuous and discrete parameters is quite clear, many times angular values are treated as continuous ones, even if their topology is totally different (real values lie on a line, and angular values on a circle). This may be because their numerical representation in a computer is the same. However this confusion leads to incorrect results: even a simple statistic such as the mean cannot be computed in the same way for continuous or angular values [Jammalamadaka and Sengupta, 2001]. The issue of dealing with parameters of different types will be the subject of Chapter 5.

1.5.2 Structure of Flights as Sequences

Secondly, we shall model each flight as a *sequence of time-samples*, which could be described as a generalized time series.

This is quite obviously a good model as flight data are recorded continuously during a flight. As explained before, our approach is that from a statistical point of view, one flight is one sample. We have introduced the term "time-sample" in order to avoid any misunderstanding.

It is possible to consider whole flights, but generally we shall see that it is better to cut flights into phases and to compare only the same phase of different flights.

This structure for flights is illustrated in Table 1.3. It is clear that such a dataset is very different from a tabular one such as for example Table 1.1. Even if there were only continuous parameters such as ALT, it would not have been possible to use a vector space model, because sequences may not have the same length just as flights may not have the same duration.

Flight 1				
ALT	8500	8400	...	2000
AIRSPEED	254	256	...	100
AUTO	ON	ON	...	OFF
PITCH	5°	5°	...	0°
time	0s	1s	...	612s

Flight 2					
ALT	6300	6200	6100	...	1000
AIRSPEED	120	122	110	...	100
AUTO	ON	ON	ON	...	OFF
PITCH	3°	4°	2°	...	2°
time	0s	1s	2s	...	598s

Flight 3				
ALT	7300	7200	...	1500
AIRSPEED	254	256	...	200
AUTO	OFF	OFF	...	OFF
PITCH	4°	4°	...	2°
time	0s	1s	...	703s

Table 1.3: Structure of an example flight dataset with 3 samples.

Another layer of complexity is added as parameters are not recorded at the same frequency. Depending on aircraft and flight recorder types, parameters can be recorded at frequencies ranging from 0.5Hz to 32Hz or even more.

1.5.3 Flight Data in Practice

Flight data in general is nowhere near as "clean" as in other domain applications of machine learning. Sensors are imprecise, or flight recorders do not record with as much precision as is provided by the sensors. Parameters have glitches, limited ranges of validity. Flights or flight phases are sometimes badly cut, parameters may be missing. Parameters sometimes even get swapped for a few seconds in the flight recorder. Values may be invalid for extended periods of time.

FDM operators are used to coping with these shortcomings, and they have developed over the years a great number of workarounds and solutions. In SAGEM, where parts of the work described in this thesis have been carried, there is a substantial amount of resources dedicated to such preprocessing of flight data. Algorithms have been developed for the filtering, correction, validation of data Garnier de Labareyre and Donadey [2013]; and also for the cutting of flights into phases or the detection of events. These algorithms rely not only on mathematics and signal processing but are also and above all the implementation of heuristics carefully tuned by field experts.

The methods and algorithms developed in this work stand on top of these crucial preprocessing steps. It assumed in this work that the data analyzed are "clean" even though the dataset may contain flights that are atypical in any of a number of ways. To put it another way, the goal of this work is not to detect glitches, invalid parameters or recording problems but to detect flights which are atypical from a safety or operational perspective.

1.6 Outline of Thesis

This work is divided into two parts. The first part deals with novelty detection (also called anomaly detection). A novelty detection algorithm is capable of identifying among a dataset of samples which are not normal. Commonly novelty

detection methods are supervised, which means that they rely on a training phase with examples of normal samples, such that the algorithm is able to infer the characteristics of normal samples. However in this thesis we develop a method for unsupervised novelty detection. Although very important for our work, this part is arguably the most technical and mathematical and maybe of more interest for machine learning researchers than FDM practitioners. This part of the thesis ends with experiments on FDM data from airline `Airline1` [1]. This experiment is done by extracting features from each flight which yields a tabular dataset. We detail how we extract these features, the selection of parameters and of the samples. We compare the results with state-of-art algorithm MKAD from NASA and from classical analysis from AGS.

The second part of the thesis is dedicated to the creation of a kernel that is suited to the structure of flight data as explained in Section 1.5. This kernel can be used in conjunction with the novelty detection method we propose in the first part. We start this part of the thesis by introducing in Chapter 4 important results from kernel methods theory that are relevant for structured data: we introduce concepts such as distances, similarities, infinitely divisible kernels etc.

Afterwards, we design this kernel using a bottom-up approach: Chapter 5 presents an approach for designing a kernel that can be used to compare time-samples, in other words this kernel can be used to compare data composed of continuous, discrete and angular values.

In Chapter 6 is presented the final step in the construction of our kernel for flight data. We present a new kernel on sequences, that we call the one-sided mean kernel, that we will use for comparing *sequences of time-samples*, which is how we model flights. We discuss how to efficiently implement this kernel using dynamic programming, and we also illustrate the consistent behavior of this kernel in the case of time series sub-sampling. Finally in Chapter 7 we experiment the one-sided mean kernel on flight data from airline `Airline2`.

[1] For confidentiality reasons we could not disclose the name of the two airline companies which have provided data as well as feedback for these studies. These airlines will thus be called respectively `Airline1` and `Airline2` in this work.

Part I

Novelty Detection

Chapter 2

Novelty Detection with Vector Data

2.1 Introduction

In many applications such as fault detection [Clifton et al., 2008], biomedical engineering [Tarassenko et al., 1995], visual object recognition and robotics [Sofman et al., 2009], it is useful to distinguish normal from abnormal data. For example in fault detection one wants a machine to be able to raise an alert when reaching very unusual conditions, which could potentially lead to hazards. Classical two-class detection algorithms are not suited to this type of problem, because they would by design only detect one type of abnormality. Instead, it is more convenient to learn only the normal behavior and flag as abnormal every sample that deviates from it. Such is the approach of novelty detection, also called one-class classification or outlier detection.

A popular quote from Hawkins [1980] could define the notion of novelty (also called an outlier): "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism". There are many types of novelty detection techniques, ranging from purely statistical [Clifton et al., 2010], inspired by immune systems [Hofmeyr and Forrest, 2000], to kernel methods, the most popular techniques being the one-class support vector machine (OC-SVM) designed by Schölkopf et al. [2001], and the support vector

data description (SVDD) proposed by Tax and Duin [2004]. For a review of classical methods one can refer to the work by Markou and Singh [2003a,b].

Framework

This chapter deals with the particular case of unsupervised novelty detection. Most common novelty detection techniques such as the one-class SVM (OC-SVM) work in a supervised setting: they begin by a training phase using only normal data, after which they can be used to evaluate unseen test data. There has been some attempts to take into account the presence of abnormal data in the training phase, such as the small sphere and large margin (SSLM) method proposed by Wu and Ye [2009], in a framework called *novelty detection with few outliers*, or NDFO. However, most of these approaches are still in a supervised framework; in the SSLM method for example abnormal samples are labeled as such in the training phase, and they are used to estimate a better boundary between normal and abnormal data.

On the contrary, in the unsupervised framework the machine has only access to unlabeled data, which may contain both normal or abnormal data. This framework has attracted less attention in the literature. In this work, we will focus on an even more challenging task: the dataset may contain a *significant* amount of abnormal data, including the case where there is as much abnormal data as normal data. We call this approach *unsupervised novelty detection with many outliers*, or UNDMO. It may seem surprising that a machine may be able to detect normal data in these conditions, especially since there is no training phase; but it will be shown that this can be achieved when some prior information about the distribution of normal data is available.

2.1.1 Importance for the field of FDM

There are real benefits of developing such techniques for the field of FDM. Firstly, the unsupervised framework has a strong practical advantage over the supervised framework: one does not need a training dataset. This is a real boon for the field of FDM, as the task of labeling flights is a very tedious one, in addition to being prone to errors. Flight analysts deal with hundreds of flights on a daily

basis, so it would be very difficult and costly to prepare such a labeled dataset. While we could have labeled the flights with the information provided by the classical event-based approach to FDM, this would have had the pernicious effect of implicitly calibrating our algorithms to find the very same problems that we already can detect using the classical approach, thus reducing the added benefit of developing a statistical approach.

Methodology

Most one-class detection algorithms rely either explicitly or implicitly on one or both of the two following informal assumptions:

1. "Most" samples are normal,

2. Normal samples are more "concentrated".

The first assumption is very reasonable in most, but not all cases, since for example a machine would be of poor use if it were too often out of order. The second assumption also makes sense: since normal samples are generated by the same mechanism it is only natural that in the long run they tend to be more and more concentrated.

In the second assumption, the term "concentrated" can translate into statistical concept, such as in the minimum volume set approach [Scott and Nowak, 2006], or geometrical concepts, such as in the support vector data description method [Tax and Duin, 2004] which finds a sphere of minimum volume containing the data.

In the UNDMO approach, one can no longer rely upon the first assumption, thus the algorithms presented here will have a strong focus on the second assumption. Our approach is innovative in that we define "concentrated" using an information theoretic concept, namely the Rényi entropy [Rényi, 1961], in the spirit of the *information theoretic learning* (ITL) [Principe, 2010] framework.

First we use a recent dimensionality reduction technique called kernel entropy component analysis, proposed by Jenssen [2009] to decompose the data distribution into orthogonal components. Then, we select some of these components according to their contribution to the Rényi entropy and our a priori information

on the distribution of normal data. This is the step when parts of the distribution resulting from abnormal behavior are discarded, and what is left is an estimation of the "true", noiseless distribution of normal data. Finally, samples are ranked according to how much they fit this distribution, and outliers can then be detected.

Contributions

The main contribution of this chapter is the design of an unsupervised novelty detection method that works even with a significant amount of outliers.

To this end, we propose a theoretical justification for the choices of the KECA components, based on Rényi entropy. To the best of the author's knowledge it is the first time that a priori knowledge about the complexity of the distribution has been used to improve the results of novelty detection. In addition to this theoretical justification, we propose practical procedures for selecting these components, in particular cases for example when one knows in advance that the data distribution is unimodal.

Most importantly, we demonstrate an upper bound on the probability density in input space based on the reconstruction error in feature space. This inequality has not only been used for the purpose of this work but it also gives a probabilistic justification to other studies, such as the KPCA for novelty detection by Hoffmann [2007] or the denoising by projection on the kernel principal subspace [Honeine and Richard, 2010], which were until now purely geometrical.

Organization of the Chapter

The remainder of the work is organized as follows. Section 2.2 builds the mathematical framework necessary to expose results. Section 2.3 presents the KECA method, and Section 2.4 its application to novelty detection. Finally in Section 2.5 we apply the proposed method to synthetic and real world datasets to demonstrate its superior performance compared to other state-of-the-art techniques.

2.2 Mathematical Framework

In this section we start by briefly recalling the main properties concerning kernel methods and the corresponding notations. We then state results that establish a link between principal directions in the kernel feature space and the kernel integral operator, as well as formulas that apply to the empirical case.

2.2.1 Kernel Methods and Notations

In this work we make use of a recent kernel method, the kernel entropy component analysis method [Jenssen, 2009]. Thus we start by defining some notations and stating some important results concerning kernel methods in general. The input space is denoted by $\mathcal{X}$, and k is a positive definite kernel.

The Moore-Aronszajn theorem [Aronszajn, 1951] ensures that there exists one unique reproducing kernel Hilbert space (RKHS) denoted $\mathcal{H}$, as well as a *feature map* $\Phi : \mathcal{X} \mapsto \mathcal{H}$ such that:

$$\forall x, y \in \mathcal{X}, \qquad \amalg \Phi x, \Phi y \tilde{}_{\mathcal{H}} = kx, y \tag{2.1}$$

Although the reproducing property [Aronszajn, 1951] will not be used, most results in this work rely on the fact that the feature space $\mathcal{H}$ is a Hilbert space.

2.2.2 Random Variable in a Hilbert Space

We look upon this problem from a statistical perspective by making the classical assumption that every sample in the dataset is an independent realization of a random variable $X \in \mathcal{X}$ with probability distribution P:

$$X_1, \ldots, X_n \sim P, \quad \text{i.i.d.} \tag{2.2}$$

The mapping Φ itself is deterministic, however as X is a random variable then the object of interest, ΦX, is itself a random variable, which lies in a Hilbert space, as stated in Section 2.2.1.

The advantage of using a statistical framework is threefold: first, this will ensure the consistency of our approach, which means that the detection will improve

as more data become available. Secondly, we will expose a probabilistic interpretation of the detection of outliers, akin to the high density region approach; and thirdly, we will be able to choose which dimensions to retain according to information theoretic concepts.

We assume that the feature space $\mathcal{H}$ is a separable Hilbert space of real functions on $\mathcal{X}$, that ΦX is a well-defined random variable of $\mathcal{H}$ and that its expectation $\mathbb{E}\left(\Phi X\right]$ is also well-defined.

2.2.2.1 Empirical and Asymptotic Measures

Most of the results exposed in the following will hold with respect to some measure of probability. In the general case we denote this measure by μ, in order to stress the fact that the results holds for both the asymptotic case where $\mu = P$, and the empirical case where $\mu = P_n = \frac{1}{n}\mathcal{P}_{i=1}^{n}\delta_{X_i}$, with δ being the Dirac distribution.

In practice the PCA procedure consists in finding the linear subspace which maximizes the empirical variance, which corresponds to the empirical measure P_n, while the asymptotic case corresponds to the probability measure P.

2.2.3 Principal Directions in a Hilbert Space

This section builds upon work established by Blanchard et al. [2007]. We start by recalling the definition of the non-centered covariance operator of a random variable in a Hilbert space.

Definition 2. *Let Z be a random variable in $\mathcal{H}$. Provided $\mathbb{E}\left(\prod Z \prod^2\right] < \infty$, then there exists one unique operator $C_Z : \mathcal{H} \to \mathcal{H}$ such that:*

$$\forall f, g \in \mathcal{H}, \quad \amalg f, C_Z \cdot g^{\sim} = \mathbb{E}\left(\amalg f, Z^{\sim} \amalg g, Z^{\sim}\right].$$

We now define an integral operator which is related to the kernel function k.

Definition 3. *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel function, $\mathcal{H}$ and Φ its associated feature space and feature map, respectively. Let μ be a probability measure on $\mathcal{X}$. Then, the kernel integral operator associated with the kernel k*

and the measure μ is defined as $K_\Phi : L_2\mu \to L_2\mu$ such that

$$\forall f \in L_2\mu, \forall t \in \mathcal{X}, \quad K_\Phi ft = \mathcal{R}\, fxkx, t\, d\mu x. \tag{2.3}$$

We now state two theorems that establish a link between principal directions in the feature space $\mathcal{H}$ and eigenfunctions of the kernel integral operator K_Φ.

Theorem 1. *Let $X \in \mathcal{X}$ a random variable distributed according to a probability measure μ, k a kernel function with its associated feature space $\mathcal{H}$ and feature map $\Phi : \mathcal{X} \to \mathcal{H}$, C_Φ the non-centered covariance operator associated to ΦX and K_Φ the kernel integral operator associated to k and μ. Then C_Φ and K_Φ share the same eigenspectrum:*

$$\lambda C_\Phi = \lambda K_\Phi.$$

It is also possible to establish a relation between principal directions in feature space and eigenfunctions of the kernel integral operator, while also providing a very useful property concerning the dot product between feature vectors and principal directions:

Theorem 2. *Let $\alpha \in L_2 P$ be an eigenfunction of operator K_Φ and λ its associated eigenvalue. Then,*

$$\phi = \frac{1}{\lambda}\mathbb{E}\big(\Phi X \alpha X\big]$$

is the associated principal direction (eigenvector) of C_Φ with eigenvalue λ, and of unit norm. Moreover, the projection of the feature vector Φx with $x \in \mathcal{X}$ on a principal direction ϕ verifies

$$\Pi^{\Phi x, \phi^\sim} = \lambda \cdot \alpha x. \tag{2.4}$$

Both theorems were proved (in a more general form) by Blanchard et al. [2007]. Now denote by Π the projection operator, and by $\mathcal{V}_m$ the principal subspace of dimension m spanned by $\phi_1, \ldots, \phi_m$. The squared distance of a point Φx to $\mathcal{V}_m$

is called the reconstruction error ρ_m and verifies the following equation:

$$\forall x \in \mathcal{X}, \quad \rho_m x = \left\| \Phi x - \Pi_{\mathcal{V}_m} \Phi x \right\|^2.$$

The following theorem gives an expression of the reconstruction error as a function of the eigenvalues and eigenfunctions of the kernel integral operator. This expression can be trivially proved using Equation (2.4) and the fact that $\left\| \Phi x \right\|^2 = kx, x$.

Theorem 3. *Let $\rho_m x$ be the reconstruction error of Φx with respect to the principal subspace $\mathcal{V}_m$ spanned by $\phi_1, \ldots, \phi_m$, with $\alpha_1, \ldots, \alpha_m$ and $\lambda_1, \ldots, \lambda_m$ the associated eigenfunctions and eigenvalues of the kernel integral operator. We have*

$$\rho_m x = kx, x - \sum_{k=1}^{m} \lambda_k \cdot \alpha_k x^2. \tag{2.5}$$

2.2.4 Finding Principal Directions in Practice

This section shows the practical importance of Theorem 1. Indeed, in practice one has to deal with a dataset $X_1, \ldots, X_n$, which is the empirical case, where $\mu = P_n$. Let us denote by C_Φ^n the empirical covariance operator of ΦX and by K_Φ^n the kernel integral operator associated to k and P_n. According to Theorem 1, $\lambda C_\Phi^n = \lambda K_\Phi^n$. Thus, the problem of finding the eigenvalues of C_Φ^n, which is a rather abstract covariance operator in a space that is potentially infinite-dimensional (in the Gaussian case for example), consists in finding the eigenvalues of K_Φ^n, which can be cast as a matrix eigenvalue problem.

In fact, according to Equation (2.3), in the case of $\mu = P_n$ the eigenvalue equation $K_\Phi^n \cdot \alpha = \lambda \cdot \alpha$ becomes

$$\forall x \in \mathcal{X}, \frac{1}{n} \sum_{i=1}^{n} kx, X_i \alpha X_i = \lambda \cdot \alpha x. \tag{2.6}$$

By applying Equation (2.6) to $x = X_1, \ldots, X_n$, we obtain the following matrix equation:

$$\boldsymbol{K_n} \cdot \boldsymbol{u} = \lambda \cdot \boldsymbol{u}, \tag{2.7}$$

where $\boldsymbol{K_n} = kX_i, X_j{}_{\uparrow\uparrow} n_{i,j=1,\ldots,n}$ and $\boldsymbol{u} \in \mathbb{R}^n$. $\boldsymbol{K_n}$ is the normalized Gram matrix, which is central to most kernel learning algorithms. Note that in practice, as $\prod^{\boldsymbol{u}} \prod^{\mathbb{R}^n} = 1$, one normalizes the eigenvectors $\boldsymbol{a} = n \cdot \boldsymbol{u}$ so that $\prod^{\boldsymbol{a}} \prod^{L_2 P_n} = 1$.

2.2.4.1 Out-of-sample Extension

Note that as an eigenfunction of $L_2 P_n$, α is defined on the whole space $\mathcal{X}$, and thus outside of $X_1, \ldots, X_n$. Once one has solved Equation (2.7) for a particular eigenvalue and obtained the values $\alpha X_i = a_i$, $i = 1 \ldots n$, one can use once again Equation (2.6) to compute values of α on the whole space $\mathcal{X}$, as in Equation (2.8), also known as the Nyström formula.

$$\forall x \in \mathcal{X}, \quad \alpha x = \frac{1}{n\lambda} \sum_{i=1}^{n} a_i k x, X_i. \tag{2.8}$$

2.2.5 Convergence of Principal Subspaces

We have exposed a common framework which encompasses both the asymptotic case $\mu = P$ and in the empirical case where one is given a finite dataset $X_1, \ldots, X_n \sim P$, i.i.d. and $\mu = P_n$.

It is of interest to study whether some kind of convergence exists from the empirical principal directions to the asymptotic principal directions. Fortunately some authors have carried studies regarding this matter, for example Shawe-Taylor et al. [2005] and Blanchard et al. [2007] have proven the convergence in terms of mean reconstruction error, while Braun et al. [2008] is interested in the convergence of dot products between eigenvectors and some functions in the feature space.

This shows that the estimation of principal directions is consistent: when one has more data to analyze, one can be confident that the empirical principal subspaces become closer to the "true" ones and consequently outliers which are not a realization of the same process are more likely to be detected.

2.3 Kernel Entropy Component Analysis

In this section will be described the kernel entropy component analysis using the mathematical framework established in the previous section.

2.3.1 KECA-compliant Kernels

In order to simplify our exposition we will assume $\mathcal{X} = \mathbb{R}^d$, however it should not be difficult to extend our results to other cases. As stated by Jenssen [2009] we will assume two important properties regarding the kernel.

Property 1. *The kernel is* positive definite*: For any* $n \in \mathbb{N}$*, and* $\forall X_1, \ldots, X_n \in \mathcal{X}^n, \forall a_1, \ldots, a_n \in \mathbb{R}^n$*:*

$$\sum_{i,j}^{n} a_i a_j k X_i, X_j \geq 0$$

Property 2. *Any partial evaluation of the kernel can be used as a Parzen window, for any* $y \in \mathcal{X}$*:*

- $\mathcal{R}_{x \in \mathcal{X}} kx, y dx = 1$
- $\mathcal{R}_{x \in \mathcal{X}} \bigcup kx, y \bigcup dx < \infty$
- $\lim_{\bigcup x \bigcup \to \infty} \bigcup xky, x \bigcup = \lim_{\bigcup x \bigcup \to \infty} \bigcup xk^2 y, x \bigcup = 0$

The first property, already stated in previous sections, guarantees the existence of the corresponding reproducing kernel Hilbert space. The second property guarantees that the kernel can be used for nonparametric density estimation [Parzen, 1962]. Together these two properties are used for kernel entropy component analysis, that is why we will call them *KECA-compliant kernels*. Note however that recently the KECA method has been extended to be used with non positive definite kernels [Jenssen, 2011]; however in this case the kernel induce a feature space which is reproducing kernel Krein spaces [Ong et al., 2004] whereas in our work we make explicit use of the property that the feature space is a Hilbert space. That is why in this chapter we shall require both properties, and in the remainder k will denote a KECA-compliant kernel unless otherwise stated.

2.3.2 Gaussian Kernel

In the case where $\mathcal{X}$ is a subset of $\mathbb{R}^d$ one of the most used kernels is the Gaussian kernel, defined by the following equation:

$$kx, y = 2\pi \cdot \sigma^{2^{-d}} \Uparrow^2 \exp -\frac{\prod x - y \prod^2}{2\sigma^2} . \tag{2.9}$$

The Gaussian kernel is positive definite and hence it induces a reproducing kernel Hilbert space, which is of infinite dimension [Steinwart et al., 2006]. In addition it can be trivially proved that it satisfies Property 2 of Section 2.3.1, which is why in Equation (2.9) we have left a normalization factor that is not mandatory for kernel methods in general. Originally much of this work was conceived for the Gaussian kernel but remains valid for any KECA-compliant kernels.

2.3.3 Orthogonal Series Density Decomposition

We assume that the probability distribution P admits a density p with respect to the Lebesgue measure. In this section, a relation between eigenfunctions of the kernel integral operator and components of the probability density will be established. This approach was pioneered by Girolami [2002]; as eigenfunctions are respectively orthogonal in $L_2\mu$ it will be possible to obtain an estimation of the density as an orthogonal series. We will present a slightly different approach than Girolami so as to insist on the geometrical aspect of the Hilbert space framework established in Section 2.2.2; this will allow us to easily derive the results in the empirical and asymptotic cases.

Sections 2.3.4 and 2.3.5 will present a well-founded tool for choosing which terms of the decomposition to retain, based on concepts from Information Theoretic Learning.

2.3.3.1 Density Estimation and Mean Vector in Feature space

Suppose one is given data $X_1, \ldots, X_N$ and wants to estimate the density associated with the process X. By using the Parzen window estimation and Equation

(2.1) an estimation px of the density is given as:

$$px = \frac{1}{n} \overset{n}{\underset{i=1}{\mathcal{P}}} kx, X_i \tag{2.10}$$

$$= \frac{1}{n} \overset{n}{\underset{i=1}{\mathcal{P}}} \coprod \Phi x, \Phi X_i \tilde{}$$

$$= \coprod \Phi x, \frac{1}{n} \overset{n}{\underset{i=1}{\mathcal{P}}} \Phi X_i \tilde{}$$

$$px = \coprod \Phi x, \mathbb{E}_{P_n} \left(\Phi X \right] \tilde{}, \tag{2.11}$$

where $\mathbb{E}_{P_n} \left(\Phi X \right]$ being the mean of $\Phi X_1, \ldots, \Phi X_n$. Equation (2.11) is a relation between a geometrical concept, the mean of vectors, and a probabilistic concept, the density estimation at a point $x \in \mathcal{X}$.

2.3.3.2 Convergence of the Kernel Density Estimator

It shall be noted that to be precise one should use a notation k_n instead of k in Equation (2.10): in order for the density estimation to be consistent the width of the kernel has to decrease to 0 at an appropriate rate when the number of samples n increases, see for example Silverman [1978]. In the case of the Gaussian kernel this can be achieved through the scale parameter σ. We have not included the subscript n for the sake of clarity in our notations.

By assuming some properties regarding the smoothness of the kernel, as well as a sufficiently fast decay of the width of the kernel to 0 it has been demonstrated [Nadaraya, 1965] that p uniformly converges almost surely to p:

$$\mathbb{P} \lim_{n \to \infty} \coprod p - p \coprod \overset{=}{\Uparrow} 0 = 0$$

In practice the width of the kernel is often chosen in a data-dependent way so that it effectively converges to 0; one must bear in mind though that in high dimensions the kernel density estimator is known to be problematic, for the purpose of this work we make the assumption that the number of samples is sufficient with respect to the dimensionality of the data.

2.3.3.3 Asymptotic Density with Fixed Kernel Width

For the purpose of our demonstration we must also define the asymptotic density estimation with a *fixed* kernel width. Recall Equation (2.10) that gives the Parzen estimate of the probability density: $px = \frac{1}{n}\mathcal{P}_{i=1}^{n} kx, X_i$. According to the law of large numbers, when $n \to \infty$,

$$\lim_{n\to\infty} px = \bar{p}x = \mathcal{R}_{y\epsilon\mathcal{X}}\, kx, ypydy.$$

We denote by $\bar{p}x$ this asymptotic density, which is the convolution product of p with the Gaussian kernel k. By virtue of the continuity of the dot product we also have:

$$\bar{p}x = \coprod^{\Phi x, \mathbb{E}_P\left(\Phi X\right)^{\sim}}.$$

2.3.3.4 Orthogonal Decomposition in the Asymptotic Case

As in Section 2.2.2, the results hold with respect to a probability measure μ which can be either P or P_n; thus results are given in the asymptotic case since the corresponding results for the empirical case can be easily obtained by using the probability measure P_n. For the sake of clarity, in the remainder of this chapter we use notations $\mathbb{E}\left(\Phi\right)$ and $\mathbb{E}\left(\alpha_k\right)$ instead of respectively $\mathbb{E}\left(\Phi X\right)$ and $\mathbb{E}\left(\alpha_k X\right)$.

The approach taken by Girolami could be described as follows: instead of projecting Φx onto $\mathbb{E}\left(\Phi\right)$, we replace $\mathbb{E}\left(\Phi\right)$ by its projection on a principal subspace: $\Pi_{\mathcal{V}_m}\mathbb{E}\left(\Phi\right)$. As orthogonal principal directions in the feature space correspond to orthogonal functions of $L_2 P$ (see Theorem 2), an orthogonal series density estimation is given, with as many terms as the dimension of the principal subspace $\mathcal{V}_m$. Now let us denote λ_k the eigenvalues of the kernel integral operator, α_k and ϕ_k the associated eigenfunctions and principal directions respectively. Using

Theorem 2 with the continuity of the dot product gives the following equations:

$$\begin{aligned}
\bar{p}x &= \amalg^{\Phi x, \mathbb{E}\left(\Phi\right]^{\sim}} \\
&= \overset{\infty}{\underset{k=1}{\mathcal{P}}} \amalg^{\Phi x, \phi_k{}^{\sim}} \cdot \amalg^{\mathbb{E}\left(\Phi\right], \phi_k{}^{\sim}} \\
&= \overset{\infty}{\underset{k=1}{\mathcal{P}}} \lambda_k \alpha_k x \cdot \lambda_k \mathbb{E}\left(\alpha_k\right] \\
\bar{p}x &= \overset{\infty}{\underset{k=1}{\mathcal{P}}} \left(\lambda_k \mathbb{E}\left(\alpha_k\right] \right| \cdot \alpha_k x. \qquad (2.12)
\end{aligned}$$

Equation (2.12) is indeed an orthogonal series density estimation, with the α_k being the series terms (they are orthogonal in $L_2 P$) and $\left(\lambda_k \mathbb{E}\left(\alpha_k\right] \right|$ the weights associated to each term. In the remainder, we denote by $\bar{p}_m$ a truncated estimation of $\bar{p}$ with m terms, though as shown later, the order of the terms may not follow the order of λ_k.

2.3.3.5 Application to the Empirical Case

One now simply applies Equation (2.12), uses P_n instead of P, and replaces asymptotic variables with their empirical counterparts obtained in Section 2.2.4:

$$px = \overset{n}{\underset{k=1}{\mathcal{P}}} \left(\lambda_k \mathbb{E}_{P_n}\left(\alpha_k\right] \right| \cdot \alpha_k x.$$

This equation can also be written :

$$px = \overset{n}{\underset{k=1}{\mathcal{P}}} \left(\lambda_k \cdot \mathbf{1_n}^{\mathrm{T}} \boldsymbol{a_k} \right] \cdot \alpha_k x \qquad (2.13)$$

In Equation (2.13), λ_k and $\boldsymbol{a_k}$ are respectively the normalized eigenvalues and eigenvectors of Equation (2.7), $\alpha_k x$ is given by the Nyström formula (2.8), and $\mathbf{1_n}^{\mathrm{T}}$ is the row vector $\left(\overset{1}{\Uparrow} n, \ldots, \overset{1}{\Uparrow} n \right]$.

2.3.4 Rényi Entropy

Equation (2.12) is an orthogonal series decomposition of the density that corresponds to orthogonal principal directions in the feature space. The question is

what terms of this expansion to retain? The classical solution consists in keeping the terms with the highest eigenvalue λ, but the following proves that this is not the best solution.

The quadratic Rényi entropy is a measure of uncertainty that generalizes the role of variance in Gaussian distribution. In his seminal work, Rényi [1961], states four properties that entropies should verify, and he then defines a new family of entropies, among them the quadratic Rényi entropy. The quadratic Rényi entropy (simply referred to as Rényi entropy in the remainder) has been used in ITL because it is differentiable and is easily estimated from data using non-parametric techniques.

The Rényi entropy for a variable X is defined as:

$$H_2X = -\log \mathcal{R}\, p^2 x \mathrm{d}x. \tag{2.14}$$

The argument of the logarithm, noted VX is called the information potential,

$$VX = \mathcal{R}\, p^2 x \mathrm{d}x.$$

In this work, the Rényi entropy will be used as a means to estimate the complexity of the distribution. Recall that the entropy is a measure for uncertainty or disorder. Then, complex distributions, such as multimodal distribution have *low* entropy while simple distributions have *high* entropy. For example, in the discrete case, the maximum entropy is achieved when all outcomes have the same probability, which corresponds to the *simplest* distribution. The remainder of this chapter will show that it is possible to significantly improve the performance of the novelty detection algorithm by incorporating *a priori* knowledge of the complexity of the distribution (expressed in terms of Rényi entropy).

2.3.5 Orthogonal Series Estimation of the Rényi Entropy

In this section will be presented expressions of an estimation of the Rényi entropy of the distribution with respect to its orthogonal series decomposition; in both the asymptotic and empirical cases.

2.3.5.1 Asymptotic Case

As discussed in Section 2.3.3.2, we make the assumption that $\bar{p} \approx p$, and thus the information potential could be defined as:

$$VX = \mathbb{E}\left(\bar{p}X\right). \tag{2.15}$$

By replacing in Equation (2.15) the expression of $\bar{p}x$ given by Equation (2.12), one has $VX = \mathbb{E}\left(\mathcal{P}_{k=1}^{\infty}\left(\lambda_k \mathbb{E}\left(\alpha_k\right)\right) \cdot \alpha_k x\right)$. Finally by virtue of the continuity of the expectation, one has an expression of the Rényi information potential with respect to the eigendecomposition:

$$VX = \overset{\infty}{\underset{k=1}{\mathcal{P}}} \gamma_k, \tag{2.16}$$

where the *entropyvalues* are defined as:

$$\gamma_k = \lambda_k \cdot \left(\mathbb{E}\left(\alpha_k\right)\right)^2.$$

2.3.5.2 Empirical Case

As in Section 2.3.3.5, by using the particular probability measure P_n the following estimate for the Rényi entropy in the empirical case can be found:

$$VX = \overset{n}{\underset{k=1}{\mathcal{P}}} \gamma_k, \tag{2.17}$$

with the estimated entropyvalues:

$$\gamma_k = \lambda_k \cdot \mathbf{1_n}^{\mathrm{T}} \cdot \boldsymbol{a_k}^2.$$

2.3.6 Choice of Dimensions in KECA

In the classical PCA procedure and its kernel counterpart, one retains the dimensions with the largest variance (eigenvalues). The idea pioneered by Jenssen [2009] is to focus instead on the entropyvalues γ_k.

2.3.6.1 Rényi Entropy as an a priori on the Distribution

As was explained Section 2.3.4, as one keeps more entropy components, the information potential *increases*, and then according to Equation (2.14) the Rényi entropy *decreases* which means that the distribution becomes more and more *structured*.

In other words, each dimension that one keeps adds a layer of structure to the distribution. Consequently if one had *a priori* a precise measure of the entropy of the distribution, one would know precisely the number of dimensions to retain, and one would be sure that the other dimensions are due to spurious data, such as noise or outliers.

In practice of course, one would not know in advance the entropy of the distribution, but the idea is to keep the simplest distribution, and this is not unlike the Occam's Razor principle that underlies many statistical learning algorithms. We will use this principle to increase the accuracy of our detection in the following section.

Thus choosing the dimensions with the largest entropyvalues amounts to keeping the closest distribution in terms of information potential (and thus in terms of Rényi entropy). In this respect the KECA departs significantly from the classical KPCA procedure.

2.3.6.2 Procedure

In a first step, one would first classify the entropyvalues in decreasing order $\gamma_1 > \ldots > \gamma_n$. In the sequel $\mathcal{V}_m$ and ρ_m will now refer to respectively the *entropy-principal* subspace and the reconstruction error to this subspace instead of the classical subspace.

There are three possibilities for the choice of components.

- First, one can choose the m largest entropyvalues so as to retain a predetermined proportion, for instance 90%, of the total information potential:

$$\sum_{k=1}^{m} \gamma_k > 0.9 \cdot \sum_{k=1}^{n} \gamma_k$$

- Secondly, one may retain enough dimensions so as to reach a certain amount

of information potential, corresponding to an *a priori* knowledge:

$$\sum_{k=1}^{m} \gamma_k > V_{min},$$

or equivalently in terms of Rényi entropy:

$$-\log \sum_{k=1}^{m} \gamma_k < H_{max}.$$

As stated previously, while in general one may not know in advance the entropy of the distribution we do not exclude that there may be special cases where the entropy may be estimated *a priori*, for example by the study of the mechanism itself generating the data.

- Thirdly, in the case where clusters in the data set are sufficiently separated with respect to the scale parameter σ^2, it has been shown [Jenssen, 2011; Shi et al., 2009] that dimensions with high entropyvalue correspond to clusters of the dataset (modes of the distribution). Thus one may have *a priori* knowledge of the number of modes and thus only retain a predetermined number $m_{apriori}$ of dimensions, knowing that the other modes are the consequences of abnormal behavior:

$$m = m_{apriori}.$$

2.3.7 Application to Toy Dataset

2.3.7.1 Presentation of the Dataset

In the following we illustrate these ideas on a toy dataset, bivariate and bimodal, and with a high concentration of outliers, as in Figure 2.1. Both clusters are from a Gaussian distribution, the first cluster (pluses "+") contains 300 samples while the second cluster (circles "∘") contains 100 samples. A number of 100 outliers (diamonds) were uniformly sampled around both clusters. Components vary from 0 to 70, and the scale parameter σ^2 for the Gaussian kernel was chosen to be in the order of magnitude of 6-nearest neighbors: $\sigma^2 = 20$. This example, though simple, can be used to illustrate many interesting concepts, and will be

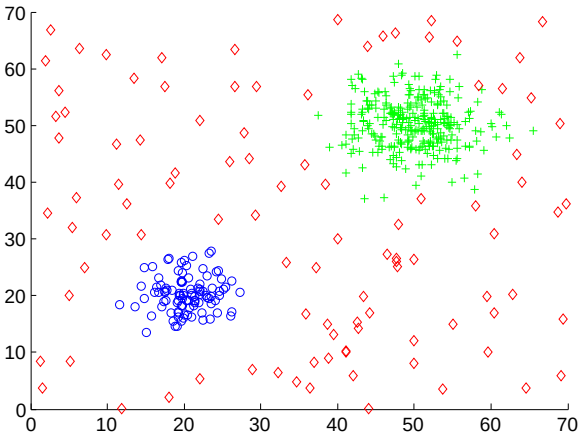


Figure 2.1: Toy dataset consisting of two clusters

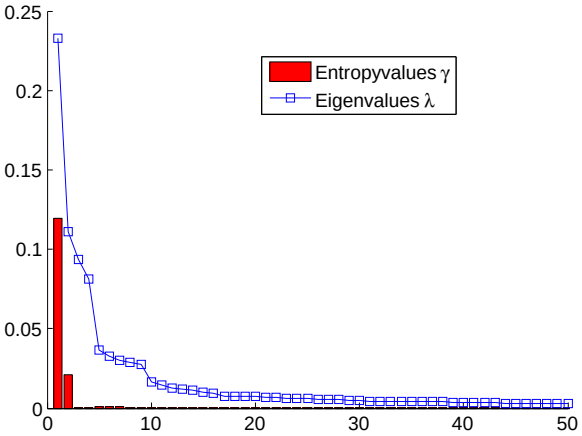


Figure 2.2: Comparison of the 50 first eigenvalues and entropyvalues.

studied throughout this chapter.

2.3.7.2 Entropyvalues

We have computed the entropyvalues γ_k and compared them to the eigenvalues λ_k considered in the KPCA procedure in Figure 2.2. First, as noted by Jenssen [2009] the entropy components do not follow the same order as the eigenvalues. Secondly there are much fewer significant entropy components than eigenvalues : only two entropy components have non-zero values. For instance, in the KPCA procedure one would select the first 43 components to account for 90.1% of the variance, while in the KECA procedure one only needs to keep the first two components to

retain 96.74% of the information potential. This sparser representation is another significant advantage of KECA over KPCA.

2.3.7.3 Components

Using the Nyström formula (2.8) it is possible to estimate the components of the distribution corresponding to principal directions in the feature space. We estimate the first two components, which correspond to the only non-negligible entropyvalues, and display the results in Figure 2.3. We see that each principal direction corresponds to a cluster in the input space. This property was used to design a clustering algorithm named data spectroscopy, by Shi et al. [2009], and which is related to the KECA algorithm, as explained for example by Jenssen [2011]. In this case it can be seen that for each cluster, the estimated component of the distribution nicely corresponds to the density that would have been estimated using parametric techniques.

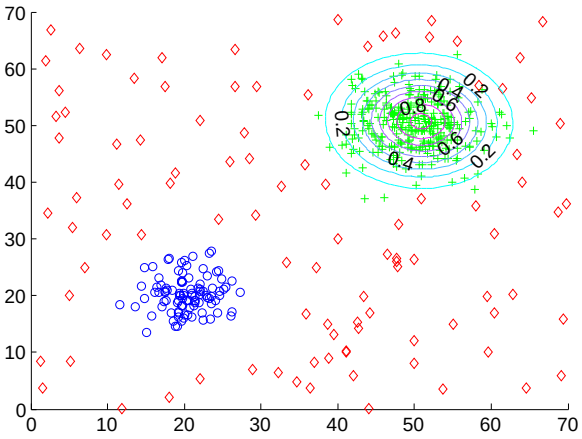
2.4 Novelty Detection with the Reconstruction Error

In this section the KECA technique is used for novelty detection, and a probabilistic interpretation of the reconstruction error is exposed that bridges the gap with the geometric approach proposed by Hoffmann [2007].

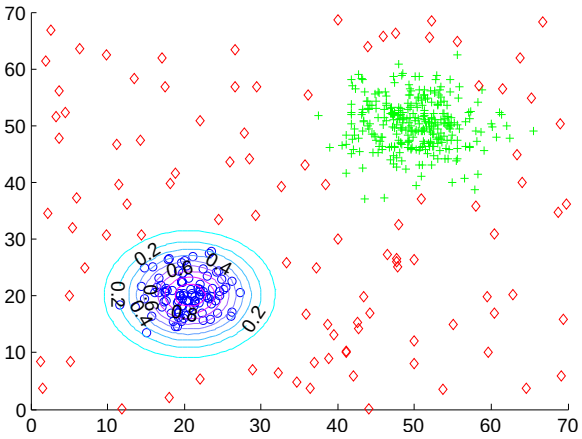
2.4.1 Probabilistic Interpretation of the Reconstruction Error

In this section a proposed theorem establishes a relation between the truncated density and the reconstruction error to the corresponding principal subspace of the feature space. In order to simplify equations we use the notation $R^2 = 2\pi \cdot \sigma^{2^{-d}}{}^{\Uparrow 2} = kx, x, \quad \forall x \in \mathcal{X}$.

Theorem 4. *Suppose one has selected m principal entropy components according to the KECA procedure. Let $x \in \mathcal{X}$ be a point of the input space, $\rho_m x$ the*



(a) Contour plot of first component



(b) Contour plot of second component

Figure 2.3: Densities of the first two principal components

reconstruction error to the m-dimensional entropy-principal subspace and $\bar{p}_m$ *the corresponding truncated density. Then*

$$\bar{p}_m x \leq \sum_{k=1}^{m} \gamma_k \cdot R^2 - \rho_m x \tag{2.18}$$

Proof. As explained in Section 2.3.3.4 the truncated density can be expressed as

a dot product with the projection of the mean vector on the principal subspace:

$$\begin{aligned}\bar{p}_m x &= \coprod^{\Phi x, \Pi_{\mathcal{V}_m} \mathbb{E}\left(\Phi x\right)}, \\ &= \coprod^{\Pi_{\mathcal{V}_m} \Phi x, \Pi_{\mathcal{V}_m} \mathbb{E}\left(\Phi x\right)},\end{aligned}$$

By using the Cauchy-Schwartz inequality we obtain:

$$\begin{aligned}\bar{p}_m x &\leq \prod^{\Pi_{\mathcal{V}_m} \Phi x} \prod \cdot \prod^{\Pi_{\mathcal{V}_m} \mathbb{E}\left(\Phi x\right)} \prod \\ &\leq R^2 - \rho_m x \cdot \mathop{\mathcal{P}}_{k=1}^{m} \coprod^{\mathbb{E}\left(\Phi x\right), \phi_k}{}^{2} \\ &\leq R^2 - \rho_m x \cdot \mathop{\mathcal{P}}_{k=1}^{m} \gamma_k.\end{aligned}$$

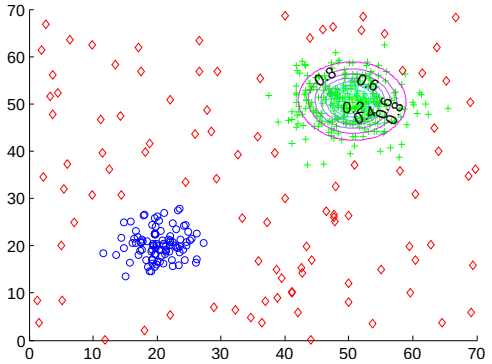
□

Theorem 4 can of course be trivially extended to the empirical case. According to the discussion in Section 2.3.6.1, $\bar{p}_m$ is now the best estimate of the true density taking into account the *a priori* one has about the distribution. The bound we obtain in Equation (2.18) guarantees that when the reconstruction error tends to R^2, the associated density tends to 0. This leads to the following corollary:

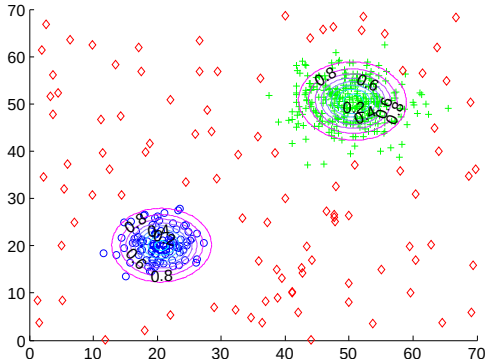
Corollary 1. *Points with high reconstruction error in feature space lie in low density regions in input space.*

Equation (2.18) thus bridges the gap between the geometrical approach proposed by Hoffmann [2007] and the probabilistic approach: using the reconstruction error as a measure for novelty is a valid approach, from both perspectives. Moreover, this corollary has consequences in other studies, such as the denoising approach which consists in projecting a noisy point onto the principal subspace in feature space, such as in studies by Schölkopf et al. [1998a] or more recently by Honeine and Richard [2010]: we now have proof that in the input space this amounts to displacing the point to high density regions where it should belong.

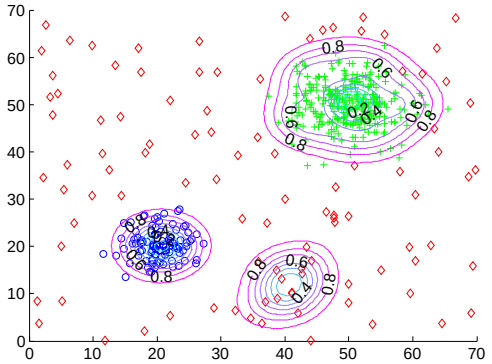
It should be noted though that our probabilistic interpretation of KECA is very different from the one presented by Zhang et al. [2004]: they bring a probabilistic interpretation by modeling the kernel as a Wishart process whereas we



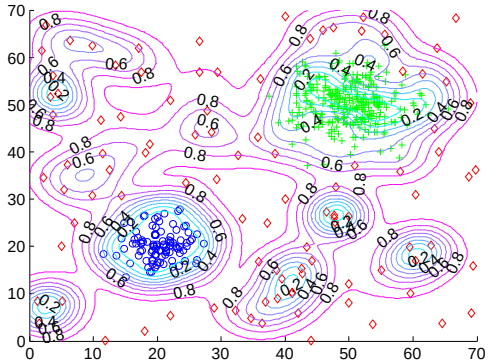
(a) m = 1, proportion of entropy retained : 0.8245



(b) m = 2, proportion of entropy retained : 0.9674



(c) m = 6, proportion of entropy retained : 0.9908



(d) m = 23, proportion of entropy retained : 0.999

Figure 2.4: Reconstruction errors with different dimensions of the principal subspace

are interested in a non parametric density estimation in the input space.

2.4.2 Application to Toy Dataset

We have computed the reconstruction error of the whole input space for principal subspaces of different dimensions. The results are displayed in Figure 2.4. It is now clear that the entropy of the principal subspace can be considered as an *a priori* for the complexity of the distribution.

Indeed, in Figure 2.4(a) we have retained only the largest entropy component, which accounts for more than 82% of the total entropy. This amounts to supposing that *a priori* the distribution shall be as simple as possible. In this case only

the most significant part of the dataset is retained: the distribution is assumed to be unimodal, we see that all outliers and even all point from the second clusters are considered abnormal. This approach is very powerful when one knows in advance that the distribution is unimodal but may be highly polluted, as we shall see in a real world example later.

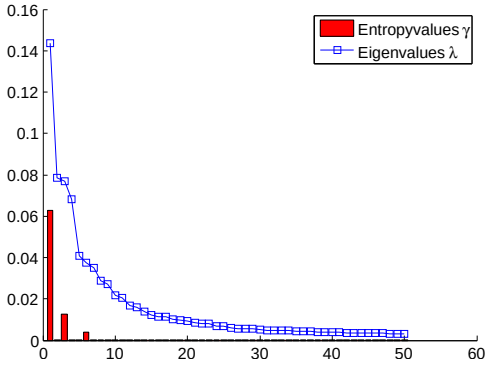
In Figure 2.4(b) one allows the distribution to be more complicated and selects the first two entropy component that account for more than 96% of the total entropy and we see that in this case the two clusters are considered normal and all outliers are considered abnormal. In Figure 2.4(c) and Figure 2.4(d), as one adds more components to the principal subspace, little clusters that were previously considered abnormal are now deemed to be part of the distribution, and only the most isolated points are now considered outliers.

2.4.3 Comparison of the Reconstruction Errors Between KPCA and KECA

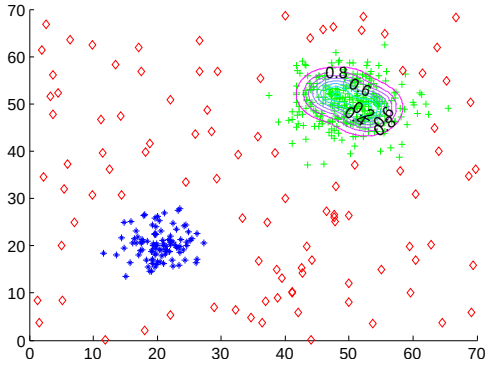
The probabilistic interpretation presented in the previous section is indeed also valid for the classical non-centered KPCA. However there is a very fundamental difference between the two methods: the KECA procedure selects in priority principal directions with high entropyvalues instead of principal directions with high eigenvalues. From an information theoretic perspective, the KECA procedure makes the best choice of dimensions. Indeed as explained before variances and entropyvalues do not necessarily follow the same order.

In order to illustrate this principle we have computed the principal directions for different kernel width, $\sigma^2 = 9$, we notice in Figure 2.5(a) that the principal direction with second higher entropyvalue corresponds to the third higher variance. We have plotted the reconstruction errors in the input space using the two most important principal directions using the KPCA procedure in Figure 2.5(b) and KECA procedure in Figure 2.5(c).

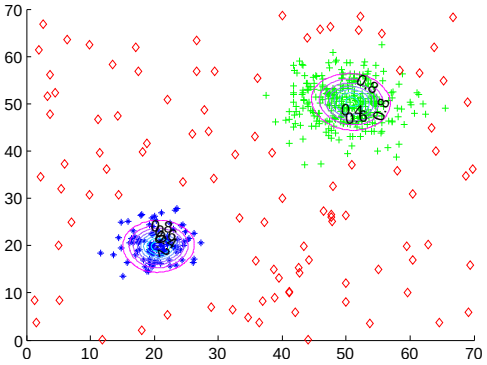
It can be noticed that with only two components the KECA procedure captures the essential parts of the distribution while the KPCA procedure only focuses on the first cluster. In Section 2.5 we will also present the differences in detection results that stem from using either one of these procedures.



(a) Eigenvalues and entropyvalues



(b) Reconstruction error in input space using KPCA and two dimensions



(c) Reconstruction error in input space using KECA and two dimensions

Figure 2.5: Reconstruction error comparison

2.4.4 Comparison with OC-SVM

One of the most used novelty detection is the OC-SVM, which consists in separating the data from the origin with maximum margin. The following quadratic program is solved:

$$\min_{w\in\mathcal{H},\,\xi\in\mathbb{R}^n,\,\rho\in\mathbb{R}} \frac{1}{2}\left\| w \right\|^2 + \frac{1}{\nu n}\sum_i \xi_i - \rho \qquad (2.19)$$
$$\text{subject to} \quad \left\langle w, \Phi X_i \right\rangle \geq \rho - \xi_i, \quad \xi_i \geq 0$$

Here ρ is the offset parameter, ξ_i are slack variables and it can be shown [Schölkopf et al., 2000] that ν is an upper bound on the number of outliers. Thus the one-

class SVM can also be used as an unsupervised outlier detection method. However one can see that in such a case the algorithm is not robust to the presence of outliers: the sum of slack variables is minimized, this has the consequence that the boundary tends to be attracted to the outliers, such that the slack variables themselves remain low. This effect is particularly strong in very high dimensional feature space such as the Gaussian RKHS. One could overcome this problem by for instance replacing in Equation (2.19) ξ_i by $\mathbf{1}_{\xi_i>0}$, or a smoother function such as a sigmoid. However in this case the optimization problem is no longer convex, and thus one looses one of the main advantages of using kernel methods.

2.5 Experimental Results

In this section we will apply the methods defined previously and demonstrate its superior performance compared to state of the art algorithms. As stated before, experiments are carried in an unsupervised setting: there is no training phase, we only assume that normal samples are issued from a more defined or simple mechanism than abnormal samples, so that outliers are likely to be abnormal samples.

We would like to emphasize that the unsupervised novelty detection is a very particular problem: one cannot take a two class data set and use it for novelty detection by arbitrarily selecting one of the classes as the normal class. Indeed, in this case most often the two classes are both well defined, and thus the dataset is not suited to the problem presented here.

The goal of this section is to demonstrate the effectiveness of the presented method on commonly used data sets, as well as to illustrate some interesting properties.

2.5.1 Datasets and Experimental Settings

We have considered five medical diagnosis data sets Biomed, Breast cancer, Hepatitis, Thyroid [1]. Each data set has different dimensionality and a different number of normal (positive) and abnormal, malignant or diseased (negative) samples.

[1] These data sets are available online from http://homepage.tudelft.nl/n9d04/occ/index.html

Dataset	#pos	#neg	d
Biomed	127	67	5
Breast cancer	458	241	9
Hepatitis	123	32	19
Thyroid	93	3679	21

Table 2.1: Datasets used in the experiments

These figures are summed up in Table 2.1.

We have compared four different techniques, the one-class SVM, the Parzen window density estimator with a Gaussian kernel, the KPCA and the KECA. Note that it is not necessary to compare with the support vector data description method [Tax and Duin, 2004] since it has been demonstrated by Schölkopf et al. [2001] that in the case of the Gaussian kernel this method produces the same results as OC-SVM. Each method produces what can be considered a score which is used to decide whether a sample is abnormal or not. In the case of one-class SVM the score is the distance to the separating hyperplane, for the Parzen window the score is the estimated density, and finally for the KPCA and KECA the distance to the principal subspace is used.

As each technique produces a score, we have compute receiving operator curve (ROC), which gives the proportion of true positives as a function of false positives, as the threshold on the score varies. The ROC curve is a common method to compare detection techniques, and the area under the curve (AUC) a good estimate of the global performance of the technique.

To evaluate the algorithms in the UNDMO setting, we have constructed several datasets with a growing number of abnormal samples, from five percent to half of the data set. This is useful to assess the robustness of the presented algorithms. Moreover, in cases where the number of abnormal was sufficient we have randomly divided the set of abnormal samples into multiple sets, and carried many studies (called “trials”), each with the same normal samples but with different abnormal samples, and we have retained the mean and standard deviation of all areas under the curve. This guarantees that the performance measured is not the result of the choice of a particular set of abnormal samples.

2.5.2 Preprocessing and Parameter Selection

As the data sets all contain a significant number of outliers we have chosen to normalize the data using a median and a median absolute deviation instead of the commonly used mean and variance. Certain data sets contained parameters with nominal values, we have chosen to discard these parameters as we have used a Gaussian kernel suited only for continuous values.

Each method tested rely on the width of the kernel, σ^2. For each study we have computed a reference width σ_0, which is the median of all pairwise distances in the dataset. Then we have tested for each method all following kernel widths $\sigma_0^2 \Uparrow 4, \sigma_0^2 \Uparrow 2, \sigma_0^2, 2\sigma_0^2, 4\sigma_0^2, 8\sigma_0^2, 16\sigma_0^2$, and we have kept only the best performance for each method. Note that for the same dataset different method may have different optimal width.

Regarding KPCA, we have chosen to keep 80% of the variance, and for KECA we have made two studies, one which keeps 80% of the information potential, referred to as "KECA"; and one which *only retains the most informative principal direction*. This second study, referred to as "KECA-1", corresponds to the strongest "denoising" of the data distribution, and as stated before may be useful when one knows in advance that the data distribution is unimodal. It has proved very interesting to compare these two approaches. This framework is useful to compare the reconstruction errors of KECA and KPCA in terms of their impact for novelty detection.

Concerning the one-class SVM, we have set the ν parameter to the actual proportion of outliers.

2.5.3 Interpretation of the Results

The numerical results are presented from Table 2.2 to Table 2.5. "Trials" refers to the number of different sets of abnormal samples that were considered. The numerical values are $100 \cdot AUC$, where AUC refers to the mean of the area under curve for all trials, the standard deviation is also given. Note that with a high proportion of outliers certain datasets do not contain enough abnormal samples to carry more than one trial, in that case no standard deviation is given and the value given is merely the area under curve of the only trial.

For each dataset and proportion of outliers the best performing method is put in bold.

We see that our method performs really well, especially considering that there is no training phase and a significant proportion of outliers. In most of the cases our method surpasses the other methods. We note that in general the performance tends to decrease when the proportion of outliers increases, which makes sense; however we can see the KECA method is in this sense more robust, for example in the Hepatitis the performance of KECA-1 (which keeps only the most informative component of the data distribution) remains consistent even when the proportion of outliers reaches 20% of the dataset.

We can also see that in addition to having in general the best performance the KECA also has in general very stable results, the standard deviation of the results are most often smaller than competing methods. The least stable method is the one-class SVM.

The Breast Cancer is a very simple data set, and corresponds perfectly to the framework of this work as the normal samples are very concentrated and abnormal samples are very diffuse. Indeed, on this dataset the KECA method achieves an area under the curve of 1, which means that normal and abnormal samples are perfectly separated, even when the proportion of outliers reaches 35% of the dataset.

On the contrary, the Thyroid dataset is much more challenging, and we note an interesting phenomenon: when the proportion of outliers becomes high the performance of all methods decreases steadily, especially for KECA-1, which has an area under the curve lower than 0.5 and even close to 0. In fact in this particular case it happens that the abnormal samples are more concentrated that the normal samples, and thus there is an inversion in the detection. This inversion is particularly visible for the KECA-1 method which focuses only on the most informative component of the distribution. The exception is KPCA; after investigation we have found that this score was achieved with the lowest σ^2. In this case the total variance in the feature space is much more spread out across all principal dimensions. As we have chosen a fixed proportion 80% of variance to be kept this has the consequence that more principal directions are kept and thus a more comprehensive part of the data set is considered normal, which explains

Biomed					
Trials	11	4	2	1	1
Outliers	5 %	10 %	20 %	30 %	35 %
OC-SVM	85.42± 9.8	85.76± 5.5	84.18± 5.7	84.56	82.41
Parzen	91.04± 7.7	91.21± 6.6	90.06± 3.1	89.82	89.22
KPCA	91.45± 7.7	90.37± 6.3	88.74± 3.3	87.17	86.34
KECA	92.34± 6.6	92.86± 5.1	91.06± 2.6	91.13	90.66
KECA-1	**92.47**± 7.4	**92.90**± 5.5	**92.05**± 2.0	**91.83**	**91.46**

Table 2.2: Results on dataset Biomed

Breast Cancer					
Trials	10	4	2	1	1
Outliers	5 %	10 %	20 %	30 %	35 %
OC-SVM	77.34± 2.0	79.40± 2.4	83.96± 0.9	95.00	95.07
Parzen	99.25± 0.4	98.54± 0.7	97.69± 0.4	98.01	97.82
KPCA	98.35± 0.6	91.67± 2.0	82.99± 12.9	96.53	95.74
KECA	**100.00**± 0.0	**100.00**± 0.0	**100.00**± 0.0	**100.00**	**100.00**
KECA-1	**100.00**± 0.0	**100.00**± 0.0	**100.00**± 0.0	**100.00**	**100.00**

Table 2.3: Results on dataset Breast Cancer

that the area under curve is above 0.5.

It should be noted that this effect is a consequence of the purely unsupervised framework: it may be interesting to investigate a semi-supervised extension as only a few examples of labeled normal samples would be sufficient so that the detection is not inverted.

Conclusion

First, a mathematical framework of Hilbert space was used to define KECA in a unified manner suitable for both asymptotic and empirical cases. Using the dual nature of KECA-compliant kernels which are both positive definite kernels and probability distributions, a relation between the reconstruction error in feature space and the truncated, "denoised" density in input space has been established. This relation justifies the use of the reconstruction error as a measure for novelty.

Hepatitis			
Trials	5	2	1
Outliers	5 %	10 %	20 %
OC-SVM	77.24± 10.8	78.46± 4.6	74.12
Parzen	84.96± 5.8	83.71± 6.0	84.09
KPCA	85.28± 5.1	83.99± 5.4	81.79
KECA	85.20± 5.9	85.30± 8.0	86.26
KECA-1	**86.83**± 4.5	**86.34**± 5.1	**86.50**

Table 2.4: Results on dataset Hepatitis

Thyroid (1)			
Trials	100	100	100
Outliers	5 %	10 %	20 %
OC-SVM	80.35± 6.9	70.61± 5.8	60.53± 4.9
Parzen	91.62± 4.4	88.63± 2.7	81.76± 3.3
KPCA	**96.28**± 4.3	75.92± 6.9	64.84± 7.1
KECA	90.30± 3.2	**88.87**± 2.7	**84.90**± 3.5
KECA-1	90.30± 3.2	88.87± 2.7	84.90± 3.5

Thyroid (2)			
Trials	94	59	39
Outliers	30 %	40 %	50 %
OC-SVM	55.32± 3.1	47.52± 5.5	45.67± 3.9
Parzen	60.12± 4.2	36.97± 3.6	16.73± 3.2
KPCA	51.90± 5.4	**59.03**± 4.9	**70.05**± 2.3
KECA	**60.26**± 5.1	50.37± 5.7	42.12± 6.3
KECA-1	60.26± 5.1	32.29± 4.3	8.42± 2.8

Table 2.5: Results on dataset Thyroid

Most importantly, it was shown that the entropy associated with the principal subspace can be used as a means to express the complexity of the distribution, and using it as *a priori* knowledge has proven to be an efficient method for novelty detection in the unsupervised case, where there is no training phase. This ability to control the complexity of the distribution is a significant asset over current state of the art methods, it allows our method to have consistent detection results even when the proportion of outliers in the data set is significant. Our method is able to discard significant parts of the data set if it does not fit the a priori on the distribution complexity.

There remain however several challenges with respect to future research. Firstly in this work data are not centered in feature space, it shall be interesting to investigate whether there exists an interpretation of centered feature space principal directions in terms of information theory. Secondly it has been shown that in an unsupervised case even with some a priori the detection can sometimes be inverted between normal and abnormal samples. One solution to this issue would be to work on a semi-supervised extension of this method, where very few labeled normal samples would be sufficient to induce good detection results.

Chapter 3

Results on the `Airline1` dataset

3.1 Introduction

In this chapter we present the results obtained from the application of the algorithms of Chapter 2 on data provided by partner airline company `Airline1`.

The approach we have taken for this campaign is a feature-based one. It is possible to achieve very satisfying results with this approach provided that, as explained in the introduction of this thesis, the feature extraction is guided by a good understanding of the domain. Furthermore the feature extraction step comes after many steps of preprocessing of the flight data.

Once again we would like to emphasize the fact that the approach we have chosen is the one pioneered by NASA [Amidan and Ferryman, 2005] which is that of *"one flight is one sample"*. Consequently we shall consider a fixed number of features per flights. In the context of FDM we have seen that this is the right approach, as other ways, such as for example one sample per instant recorded did not yield any sensible results for FDM (contrary to structural health monitoring).

We compare the results of our approach with the results obtained form classical analysis with AGS (recall that AGS is the software made by SAGEM which is used for classical FDM analysis) as well as with state of the art NASA MKAD. More generally, we try to assess the added value of a statistical approach for the field of FDM.

3.2 Presentation of the dataset

3.2.1 Samples

We consider a set of 721 flights of airline `Airline1` from Porto, Portugal to runway 26 of Paris-Orly, France in the 2011 to 2012 period. Note that this restriction is central to our approach, since we are interested in the flight procedures. In previous approaches we had tried not restricting on the origin of flight but still restricting on the arrival runway, this resulted in flights coming from unusual origins (such as a flight from Berlin when most other flights are from Porto) being flagged as outliers, which is of course not very interesting for a FDM practitioner.

Flight phase Typically a flight is divided in several phases, which comprise the taxi, takeoff, climb, cruise, descent and landing. In this study we cut the flights in the descent and landing phase. More precisely, the flights are cut from 10000 feet until the touchdown.

3.2.2 Features

We consider 13 of the most important parameters related to the trajectory and descent profile of flights. Of these parameters we take features, most importantly the mean but sometimes also extrema or first or last values. When dealing with angular parameters the mean is computed as described in Chapter 5. The set of features is described in Table 3.1. We also added the duration of the considered flight phase as a feature. Consequently, the dataset we consider can be represented as 721 vectors in a 22-dimensions Euclidian space.

3.3 Procedure for KECA

In the sequel we refer to our method as "KECA", in opposition to the state-of-the-art MKAD from NASA.

Code	Parameter	Features
ALTQNH	Barometric Altitude	Mean
RALTC	Radio Altitude	Mean
LONG	Longitudinal Acceleration	Mean, Absolute Max
HEAD	Heading	Mean
IVV	Vertical Speed	Mean, Absolute Max
PITCH	Pitch	Mean, Absolute Max
ROLL	Roll	Mean, Absolute Max
LATPC	Corrected Latitude	Mean, First
LONPC	Corrected Longitude	Mean, First
CASC	Corrected Air Speed	Mean, Max
GSC	Corrected Ground Speed	Mean, Max
N11C	Primary Thrust	Mean, Max
DURATION	Duration of flight phase	-

Table 3.1: Features retained

3.3.1 Detection of atypical flights

The first step is the extraction of the features. Then we have normalized these features using the median value as a centroid and the median absolute deviation as a measure of dispersion. These values have the benefit of being more robust to outliers than the traditional mean and variance couple. This robustness is very important in an unsupervised setting as the dataset is supposed to contain a significant number of outliers.

Next we computed all pairwise distances between samples, and calculated a bandwidth parameter for the Gaussian kernel as the median of all these distances. We computed the principal directions in the feature space, and the associated *entropyvalues* as described in Section 2.2.4. Retaining the dimensions that cover more than 80% of the entropyvalues resulted in keeping only the first and the 16^{th} dimensions as the entropy-principal subspace. We then proceeded to compute the distances of all samples from this entropy-principal subspace in the feature space. After fitting a Gamma distribution on these distances, we kept as the detected atypical flights the sample whose pvalues were under 0.05.

3.3.2 Report for analysts

We have put a considerable amount of effort in developing a program that not only detects atypical flights but also provides as much information as needed to the flight analysts in order to understand the results. Once the atypical flights have been detected a report is produced, which presents for each atypical flight the estimated pvalue, a set of parameters which are likely responsible, as well as a set of graphical representations that highlight the atypical flight by comparing it to the rest of the fleet.

In order to identify responsible parameters we compute the gradient of the reconstruction error evaluated at the detected atypical flight. In this case the gradient is thus a 22-dimensional vector; in practice we have observed that in the vast majority of cases this vector is sparse, and the sign of the non-null values gives the direction to which the feature should be modified for the flight to be normal. This makes sense as the principal subspace is supposed to represent the normal distribution of flights as explained in Section 2.4. Furthermore this approach is all the more practical as the gradient can be obtained as a closed-form equation in the case of the Gaussian kernel.

3.4 Procedure for NASA MKAD

The MKAD source code is available freely on the Dashlink website [NASA, 2014]. The MKAD is based on a symbolic representation of time series named *SAX*, as described by Lin et al. [2003] and in Chapter 5. Consequently, it does not rely on features; but it is still interesting to compare the results of the two methods. We study the same parameters listed in Table 3.1, but ignore the features. Flights are still cut from 10000 feet till touchdown. We keep the default parameters for the symbolic representation of time series [Lin et al., 2003], which are a window width of 30 seconds and an alphabet size of 10. We set 0.05 as the pvalue threshold for detection, the same that we have set for our method.

Class	Total Number	Detected by KECA	Detected by MKAD
Class 1	23 flights	11 flights	19 flights
Class 2	5 flights	3 flights	4 flights
Class 3	1 flight	0 flight	1 flight

Table 3.2: Comparison with classical analysis

3.5 Results

Of the 721 flights, 43 were detected as outliers by MKAD, 35 were detected as outliers by our method. 14 flights were detected by both methods. In total there are thus 64 flights that were detected by either method. Note that the expectation of the number of detected flights is $0.05 \cdot 721 = 36$.

We also ran classical event detection on all 721 flights using the AGS. Recall that in classical analysis events are divided into classes depending on their severity (most often this is related to how far from the threshold the value is). Events of class 3 are the most critical class of events and are systematically reported to the airline. We have summed up the results from classical analysis in Table 3.2. With the help of flight data analysts we made an exhaustive study of these 64 flights and almost all of them were deemed to be of interest. Among the flights detected we encountered the following patterns:

Go-arounds The go-around is the decision to re-initiate the approach if it is deemed it cannot be continued (for example in the case of an unstabilized approach, runway obstructed etc.). The go-around is considered to be a normal flight phase, but its good execution is critical for safety. More precisely, it is the lack of go-around decision that is the leading risk factor in approach and landing accidents and is the primary cause of runway excursions during landing. In a report by Flight Safety Foundation et al. [2013] it is even stated that no other single decision could have a higher impact on the overall aviation accident rate. Both KECA and MKAD methods detected 4 flights with go-arounds in the dataset, whereas the classical heuristic-based method only detected 2 of them. Flights with go-arounds are usually detected as outliers with very low ($< 10^{-3}$) pvalue.

Wind shear Wind shear is a sudden change in wind velocity or direction that can affect the dynamics of the aircraft. It can result in a loss of airspeed or power, and can make an approach unstable at a point where a go-around is no longer possible, with disastrous consequences. The KECA method has found one fight ($pvalue = 0.0229$) which is very likely due to a wind shear.

Among the other interesting patterns that were found we had hard landings, incorrect flares which is the transition phase from the final approach and the touchdown, late flap settings which are symptomatic of a rushed approach.

More generally many flights detected took a too sharp turn, some conversely took a too large turn. Some flights had a very high altitude profile, and some others had too low altitude profile.

Less interestingly we came across flights that simply had recording glitches, which resulted in incorrect values and explain that these flights were detected as atypical. Although not very interesting it is still quite reassuring that these are detected by our method.

Most interestingly we came across flights that had very atypical profiles that could not be easily classified and that baffled our flight analysts. Most often these flights had no event severe event detected so were totally overlooked by classical systems. We argue that there lies the value of such a system, we present some examples of these flights in the next section.

3.5.1 Examples of atypical flights

In this section we present a set of four interesting flights, their associated pvalues and some graphical representations that compare the atypical flight with the rest of the studied fleet. Note that the graphical representations themselves are novel and generally spark much interest among the FDM practitioners. The atypical flight is plotted in red whereas the other flights in the dataset are plotted in transparent green, such that parts of the domain that a large number of flights cross are darker, and conversely unusual parts of the domain are left white. Most graphics display the time remaining until touchdown (in seconds) as the horizontal axis, except for the trajectory graphs where the horizontal axis is the longitude and the vertical axis is the latitude.

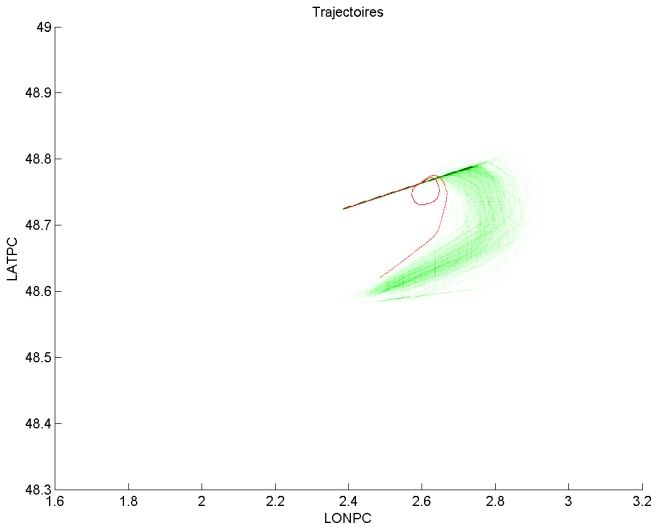


Figure 3.1: Trajectory of Flight 1

In the following we name the flights Flight 1, Flight 2 etc. for easy referencing but these numbers are unrelated to the original identification numbers in the database or even the order of detection.

Flight 1 This flight was detected as an outlier with an extremely low pvalue ($< 10^{-14}$). Interestingly, FDM analysts all agreed that this flight was the perfect example of an "atypical flight". One can see in the trajectory graph of Figure 3.1 that this flight made a very tight loop just before landing. This flight was not detected by MKAD, and only raised a classical event of class 1, and even after investigating with the company it remains unclear why the pilot proceeded that way. It may be because the aircraft had too much speed, or maybe the runway was found at the last minute to be obstructed.

Note that the other parameters from this flight also display some kind of atypicality.

Flight 2 This flight was detected as an outlier with pvalue under 10^{-4}. It is an example of a go-around. In this case the decision to re-initiate the landing was done very close to the ground, as one can see in Figure 3.8 and 3.10. One can

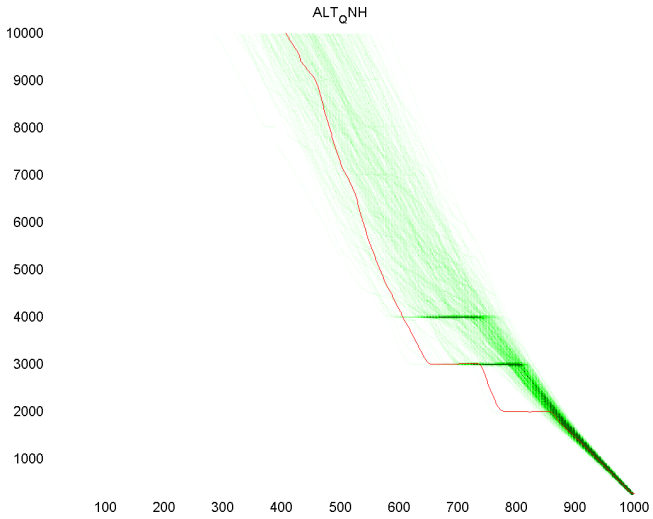


Figure 3.2: Altitude of Flight 1

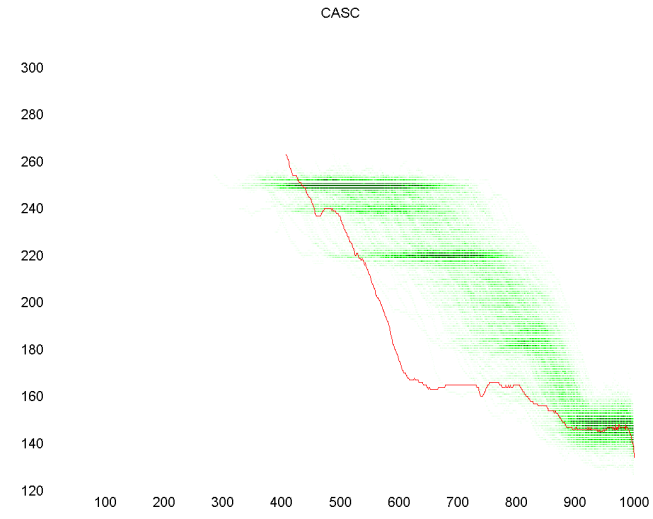


Figure 3.3: Airspeed of Flight 1

Figure 3.4: Ground speed of Flight 1

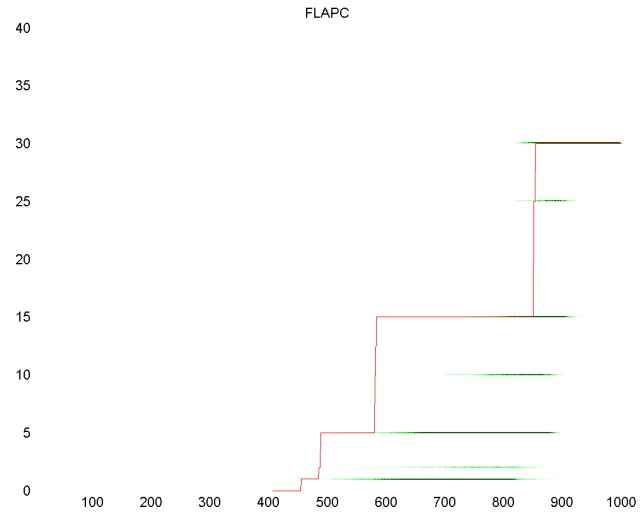


Figure 3.5: Flaps of Flight 1

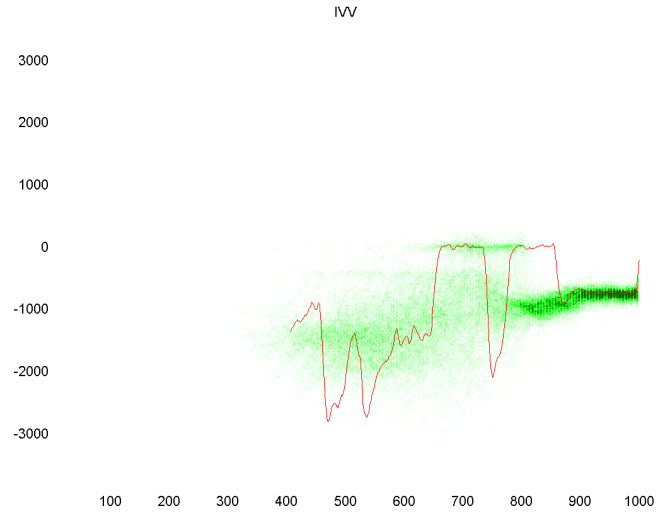


Figure 3.6: Vertical speed of Flight 1

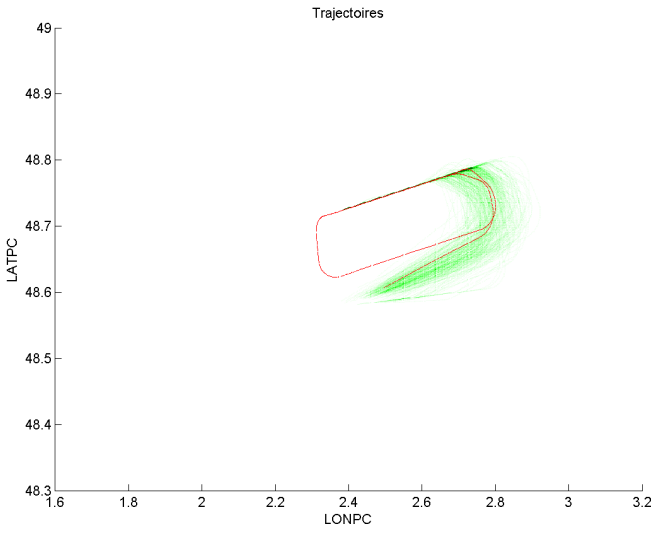


Figure 3.7: Trajectory of Flight 2

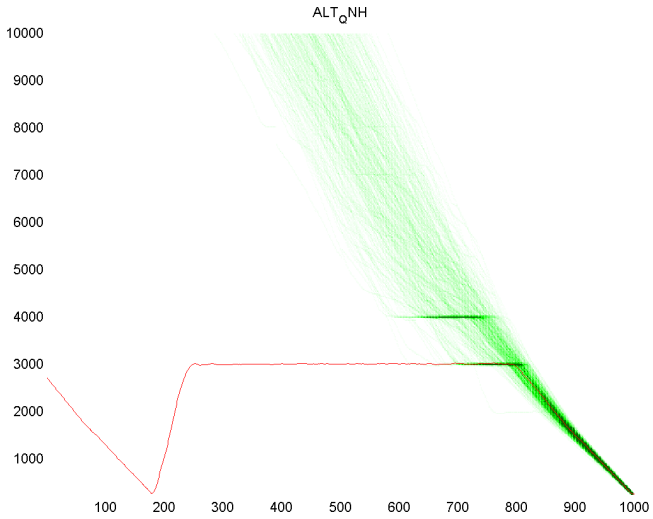


Figure 3.8: Altitude of Flight 2

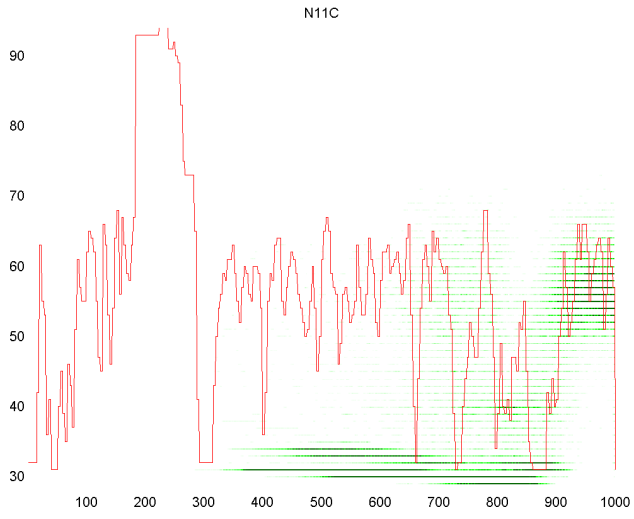


Figure 3.9: Primary thrust of Flight 2

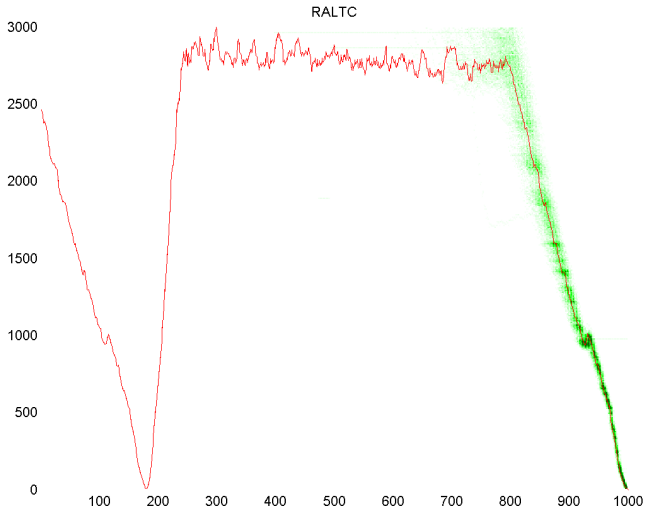


Figure 3.10: Radio altitude of Flight 2

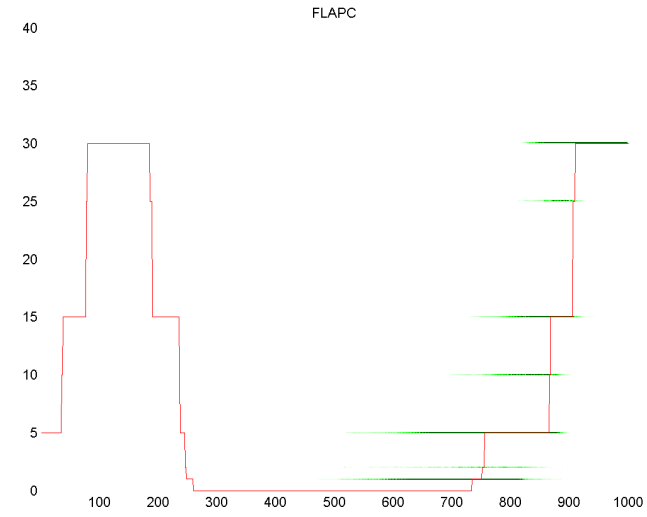


Figure 3.11: Flaps of Flight 2

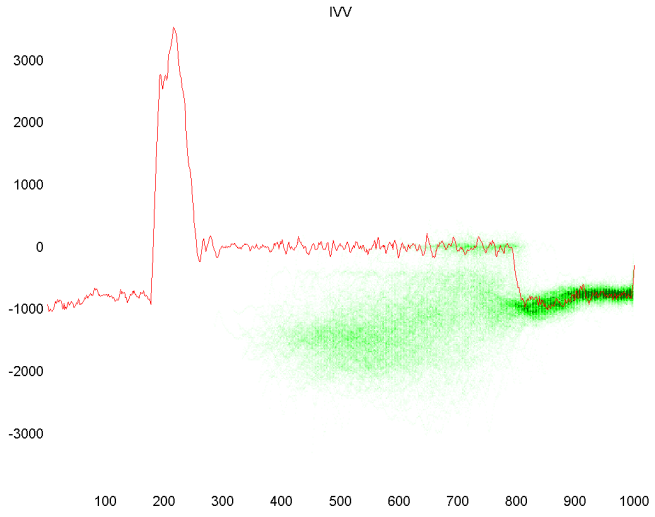


Figure 3.12: Vertical speed of Flight 2

also see in Figure 3.9 that the engine thrust was put to almost the maximum in order to regain altitude, we see the effect in the vertical speed in Figure 3.12.

Flight 3 This flight was detected as an outlier with an very low pvalue ($< 10^{-3}$). It seems that this flight was detected as outlier because of a combination of effects. First this flight has a heading deviation, as can be seen in Figure 3.13. In other words this flight went too far when aligning with the runway and thus found itself on the right hand side of runway (when facing it) where it should not have been. Secondly it has a lower altitude profile, this can also be seen in the graph of Figure 3.13 as the trajectory is cut from 10000 feet we see that the trajectory of this flight begins much sooner than others. We also observe that the speeds (both air and ground) were very high before getting back to a normal range, and moreover the flaps are set very late.

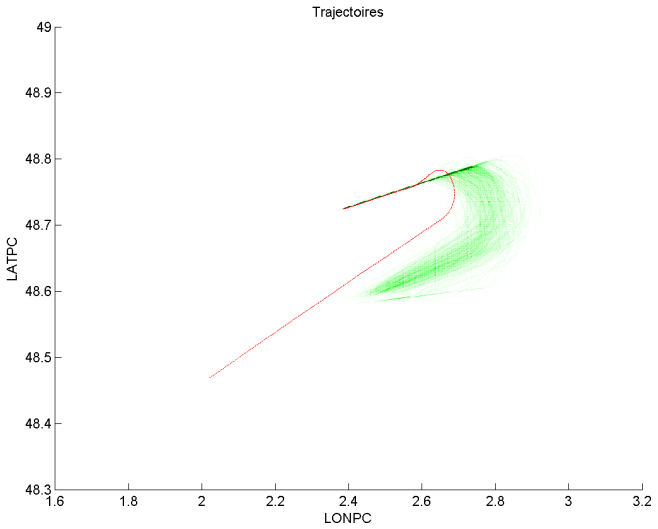


Figure 3.13: Trajectory of Flight 3

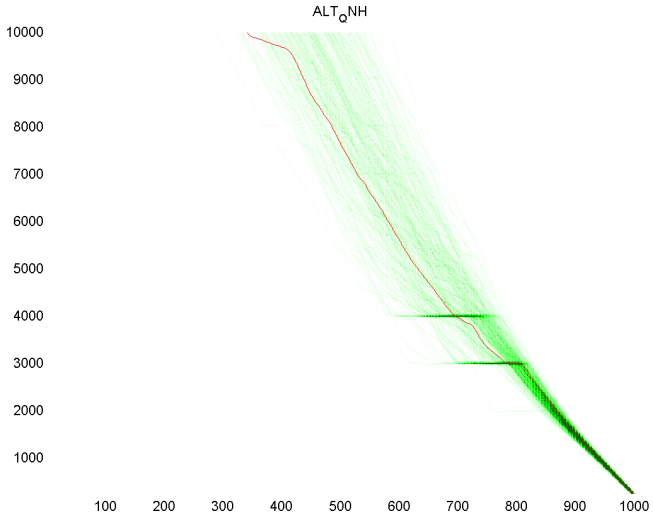


Figure 3.14: Altitude of Flight 3

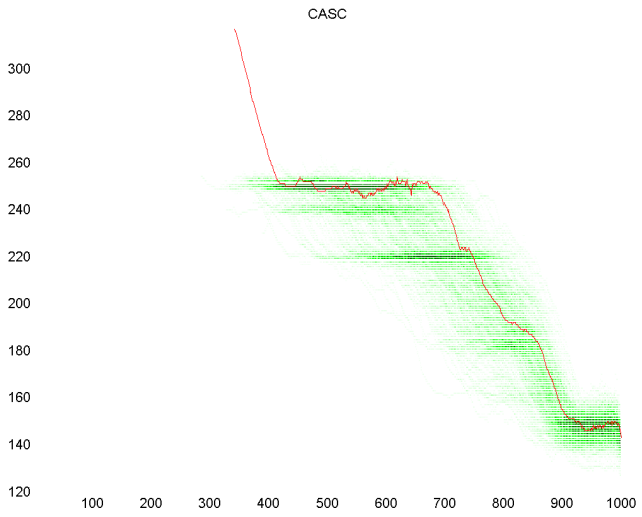


Figure 3.15: Airspeed of Flight 3

Figure 3.16: Ground speed of Flight 3

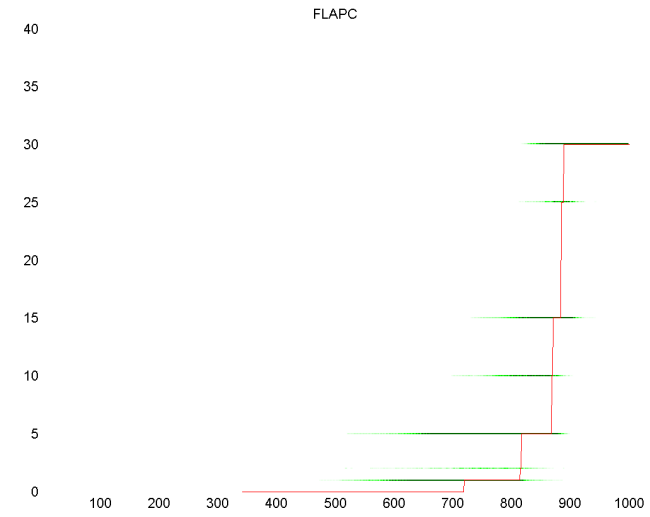


Figure 3.17: Flaps of Flight 3

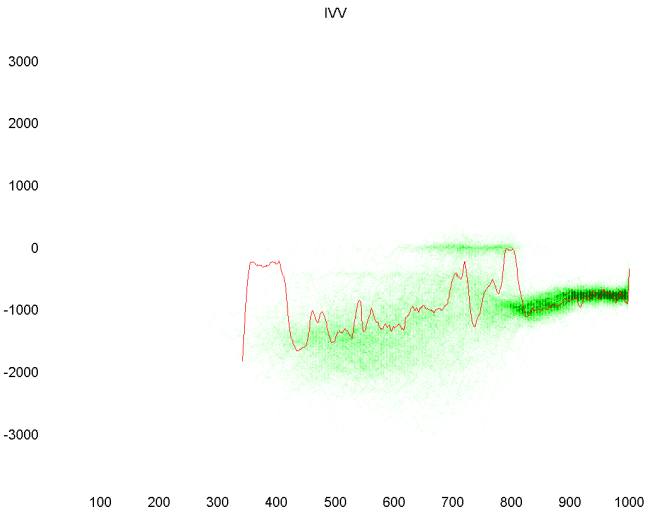


Figure 3.18: Vertical speed of Flight 3

3.6 Conclusion

Overall this first campaign on the data from airline `Airline1` is very positive. Firstly it is safe to say that we have validated the added value of using a statistical approach for FDM, as both FDM analysts in SAGEM and our partners in `Airline1` were very impressed that we were able to find genuinely interesting flights that were completely overlooked by classical methods. Besides, almost all other flights were in accordance with what experts would qualify as "atypical flights" even if not all had issues with safety.

However it is clear that the feature-based approach is insufficient. Although in our method we have addressed some shortcomings of the state-of-art MKAD, such as a more robust novelty detection algorithm than the one-class SVM as described in Chapter 2 and a better treatment of angular parameters as will be described in Chapter 5; there are atypical flights who simply cannot be detected by a feature-based approach. The reasons being first that a problem may be too localized in time to transpire in the feature vector and second that the flight procedure is really defined in terms of a sequence of events that has to be followed in the right order.

It is the subject of the second part of this thesis to extend this method to take into the structure of flight data as described in the introduction, and we hope to build a more principled approach than the one which is currently used in NASA MKAD.

Part II

Structured Data

Chapter 4

About Distances, Similarities, and Related Kernels

4.1 Introduction

In using kernel machines, the view most shared by practitioners is that kernel values reflect some kind of *similarity* between samples. This is in part due to the fact that often data can be represented in an Euclidian space, and one of the most commonly used kernel in this case is the Gaussian kernel:

$$kx, y = \exp -\prod x - y \prod^2$$

This kernel is well-known to be positive definite, and its values can readily be interpreted as similarities: $kx, y = 1$ if and only if $x = y$, and $kx, y \to 0$ when $\prod x - y \prod \to \infty$. By analogy, many practitioners have tried defining new kernels by exponentiation of a distance:

$$kx, y = \exp -dx, y^2$$

Kernels expressed as function of a distance are called *isotropic kernels*, and more generally, kernels with values interpretable as similarity coefficients are sometimes called *generalized radial basis kernels* [Haussler, 1999], here we simply refer to

them as radial basis kernels.

However, one cannot be certain that the resulting kernel is indeed positive definite; in fact some kernels derived from distances have been proved to be *not* positive definite, as we shall see in the following.

In this chapter we will introduce some results on which type of distances yields a positive definite kernel when exponentiated, and to this end we will introduce the concepts of infinitely divisible kernels, conditionally negative definite kernels as well as some other important related theorems. In addition, we shall see how these are linked to radial basis kernels, and how one can normalize any positive definite kernel with positive values to yield a radial basis kernel.

This chapter does not contain new contributions, but rather exposes some important results that can be found in [Berg et al., 1984; Cuturi, 2009; Haussler, 1999].

Many results stated in this chapter will serve as the foundation for the developments in subsequent chapters. In particular, Theorem 8 will be used several times particularly in Chapter 5, as a way to construct a kernel with all the properties that we seek, namely positive definite, radial basis and infinitely divisible.

4.1.1 Importance for the Field of Flight Data Monitoring

From a practical point of view these properties are of prime importance. A positive definite kernel ensures that most optimization problems resulting from the use of kernel machines will be convex and thus will present a global optimum. An infinitely divisible kernel can be scaled at will while still retaining its positive definite property. A radial basis kernel yields a Gram matrix whose coefficients can readily be interpreted as similarity coefficients, which are easily interpreted and understood by domain experts. Consequently, they will be able to validate the algorithm that computes the Gram matrix if they deem that the values are in accordance with their expertise. In this case one can be confident that any atypical flight detected will likely also be abnormal from a domain expertise point of view. Additionally, many kernel methods rely explicitly or implicitly on the radial basis property.

4.2 Distances and Conditionally Negative Definite Kernels

First let us recall the definition of a *distance*, or *metric* on a space $\mathcal{X}$.

Definition 4. *A function d on $\mathcal{X} \times \mathcal{X}$ with non-negative values is a* distance *(or* metric*) if for any x, y, z in $\mathcal{X}$ the following three axioms are true:*

1. $dx, y \geq 0$ *and* $dx, y = 0$ *if and only if* $x = y$,
2. $dx, y = dy, x$ *(symmetry),*
3. $dx, y \leq dx, z + dz, y$ *(triangle inequality)*

Distances are linked to a class of kernels named *conditionally negative definite kernels*. In the literature, they are sometimes called *negative definite kernels* [Haussler, 1999], which can be mistaken for the opposite of positive definite kernels; other times it is the opposite of conditionally negative definite kernels which is of interest and they are called *conditionally positive definite* [Scholkopf, 2001]. In this work we shall use the term *conditionally negative definite kernels* in order to avoid any misunderstanding.

Definition 5. *A function Nx, y on $\mathcal{X} \times \mathcal{X}$ is* conditionally negative definite *if it is symmetric and for any $x_1, \ldots, x_n$ in $\mathcal{X}$ and real $c_1, \ldots, c_n$ such that $\mathcal{P}_{i=1}^{n} c_i = 0$:*

$$\overset{n}{\underset{i,j}{\mathcal{P}}} c_i c_j N x_i, x_j \leq 0$$

Positive Definite and conditionally negative definite kernels Note in the Definition 5 that the condition $\mathcal{P}_{i,j}^{n} c_i c_j N x_i, x_j \leq 0$ is only required for the coefficients that sum to zero. Consequently, for any positive definite kernel k, the kernel $-k$ is conditionally negative definite. However, there exist conditionally negative definite kernels N such that $-N$ is not positive definite. As such, the class of conditionally negative definite kernels is *larger* than the class of positive definite kernels. Although conditionally negative definite kernels do not share the same properties as positive definite ones, many kernel methods can be used with conditionally negative definite kernels as explained by Scholkopf [2001].

Conditionally Negative Definite kernels and distances Let us first define what is a *Hilbertian* norm:

Definition 6. *A metric d on a space $\mathcal{X}$ is Hilbertian if there is an isometric embedding of $\mathcal{X}, d$ into some Hilbert space $\mathcal{H}$.*

Distances which are Hilbertian norms can be identified with conditionally negative definite kernels, as is made explicit by Berg et al. [1984]:

Theorem 5. *Let Ψ be a conditionally negative definite kernel on $\mathcal{X} \times \mathcal{X}$. Then there exist a Hilbert space $\mathcal{H}$, a mapping ϕ from $\mathcal{X}$ to $\mathcal{H}$, and a function $f : \mathcal{X} \to \mathbb{R}$ such that for any x, y in $\mathcal{X}$:*

$$\Psi x, y = \prod \phi x - \phi y \prod^2 + fx + fy \tag{4.1}$$

If the set of pairs x, y such that $\Psi x, y = 0$ is exactly x, x, $x \in \mathcal{X}$ then Ψ is a distance.

Thus, any Hilbertian norm can yield a conditionally negative definite kernel. Moreover, as the function f can be deduced from Ψ using Equation 4.1, $fx = \Psi x, x_{\Uparrow} 2$, then from any conditionally negative definite kernel it is possible to recover a Hilbertian norm. However, not all distances lead to a conditionally negative definite kernel, as not all distances correspond to a Hilbertian norm, for example it has been demonstrated that most variations of the edit-distances on strings are not Hilbertian [Lei and Sun, 2007]. This has important consequences for the study of sequences, which is how we model flights; we shall study in greater details this issue in Chapter 6.

4.3 Introduction to Infinitely Divisible Kernels

In this section we shall give the definition of *infinitely divisible kernels*, and expose how these are related to the concepts of conditionally negative definite kernels, distances and similarities.

Definition 7. *Let K be a positive definite kernel on $\mathcal{X} \times \mathcal{X}$. The kernel K is called* infinitely divisible *if for each positive integer n there exists a positive definite kernel K_n such that $K = K_n^n$.*

The following theorem, from [Berg et al., 1984], establishes a link between infinitely divisible kernels and conditionally negative definite kernels.

Theorem 6. *Let k be a positive definite kernel, and $N = -\log k$. Then the following are equivalent:*

1. *k is infinitely divisible.*
2. *$k^t x, y$ is positive definite for any $t > 0$.*
3. *N is conditionally negative definite.*

Therefore Theorem 6 and Theorem 5 let us partly answer the question asked in the introduction. If d is a *Hilbertian* distance on $\mathcal{X}$, then $k = \exp{-d^2}$ is a positive definite kernel on $\mathcal{X}$. Moreover, this kernel is infinitely divisible, which according to the second property of Theorem 6 means that it can be scaled arbitrarily and still preserve its positive definiteness. In this case, for any $t > 0$, the kernel $k^t = \exp{-t \cdot d^2}$ is still positive definite.

4.4 Similarities and Radial Basis Kernels

We start by defining *radial basis kernels*, which are kernels with values interpretable as similarity coefficients.

Definition 8. *Le k be an arbitrary kernel on $\mathcal{X}$. k is called a* radial basis kernel *if it verifies the following axioms for any x, y in $\mathcal{X}$:*

1. *$0 \leq kx, y \leq 1$,*
2. *$kx, y = 1$ if and only if $x = y$.*

As described by Haussler [1999], it is possible to normalize a positive definite kernel with positive values to yield a radial basis kernel.

Theorem 7. *Let k be a positive definite kernel on $\mathcal{X}$ with* positive *values. In particular, $\forall x \in \mathcal{X}$, $kx, x > 0$. Denote by $\tilde{k}$ the kernel:*

$$\tilde{k}x, y = \frac{kx, y}{kx, xky, y}$$

Then the following hold:

1. *$\tilde{k}$ is positive definite,*
2. *$\tilde{k}$ is a radial basis kernel.*

$\tilde{k}$ is called the radial basis normalization *of k.*

We may finally link all the concepts introduced in this chapter together with this theorem:

Theorem 8. *Let d be a Hilbertian metric on a space $\mathcal{X}$, and for any $t > 0$, let k_t be the kernel on $\mathcal{X}$ defined as:*

$$k_t x, y = \exp -t \cdot d^2 x, y$$

Then we have that:

1. *k_t is positive definite,*
2. *k_t is a radial basis kernel,*
3. *k_t is infinitely divisible.*

Moreover, let N be any conditionally negative definite kernel associated to d, such that there exists a function f which verifies $Nx, y = d^2 x, y + fx + fy$. Denote by $k_{t,N}$ the infinitely divisible kernel associated to N:

$$k_{t,N} x, y = \exp -t \cdot Nx, y$$

Denote by $\tilde{k}_{t,N}$ the radial basis normalization of $k_{t,N}$. Then we have:

$$\tilde{k}_{t,N} = k_t$$

Theorem 8 thus lets us create positive definite kernel from Hilbertian metrics, and these kernels are radial basis so their values can readily be interpreted as similarity coefficients. One may wonder though what is the use of defining a new kernel by exponentiation when one already has a kernel on $\mathcal{X}$: indeed as the metric d is Hilbertian, a dot-product can be resolved from d and used as a kernel, by definition.

The answer is that such a kernel can be used to account for non-linearity in the dataset, such as estimating manifolds instead of principal subspaces. Additionally, kernels defined using Theorem 8 have another interesting property: as these kernels are infinitely divisible, one is able to scale them at will while still retaining the positive definite property. With a small scale practitioners are interested in local properties of the dataset, which are related to its topology, while a larger scale can be used to extract features related to the global structure of the dataset.

4.5 Conclusion

In this chapter we have seen how the concepts of distance and similarity are linked in the context of kernel methods. Most importantly we have presented a way to define a kernel by exponentiation of a distance: provided the distance is Hilbertian, then the resulting kernel will have all the properties that we seek, namely positive definite, radial basis and infinitely divisible. In the sequel we will make use of this theorem to construct kernels suitable for the study of flight data.

Chapter 5

Multivariate Data with Mixed Types

5.1 Introduction

In classical statistics as well as in machine learning, datasets are in the vast majority of cases multivariate. Although multivariate datasets do require special statistic treatments [Anderson, 1954] especially in the high dimensional case [Bühlmann and Van De Geer, 2011], this issue is not a real concern for practitioners.

However, things get a little more complex when features are of heterogeneous (mixed) types, such as for example samples consisting of both continuous and discrete features, or both continuous and circular values. This issue has largely been addressed in the field of nonparametric statistics [Jammalamadaka and Sengupta, 2001; Li and Racine, 2003]. Nevertheless, we wish to dedicate a chapter to this topic because it seems that most other machine learning algorithm dedicated to FDM such as [Amidan and Ferryman, 2005; Das et al., 2010; Smart et al., 2012] did not handle such issue properly, especially the problem of angular data. Moreover, in this chapter we present this issue from a kernel methods perspective, which is necessary for our task.

Importance for the Field of FDM

This chapter is of prime importance for the field of FDM because the data we have to deal with in this field is of several different types. We have distinguished four types of parameters, although most FDM systems only consider two of them (continuous and binary).

1. *Continuous parameters.* Continuous parameters are used to model physical quantities that lie in a defined range. Common continuous parameters include for example the different types of altitude (whether measured using pressure or using radio etc.), different types of speeds (either air or ground etc.), accelerations etc. In the flight recorder, continuous parameters are coded by a fixed number of bits. FDM softwares such as AGS may decode such values because the range and precision of the parameter is known, as defined in the *data frame* which contains all information necessary for the decoding of parameters. These parameters can be mathematically modeled using real values ($\mathbb{R}$), and implemented using floating-point arithmetic.

2. *Angular parameters.* Most FDM systems (including flight recorders) do not make the distinction between angular parameters and general continuous values. They are decoded and processed exactly in the same way as continuous parameters. However, we argue that for using statistical methods it is of paramount importance to make the distinction. Such values are usually called *circular values* in the field of statistics, because from a topological point of view angular values lie on circle. Consequently mathematical concepts such as distances (as explained in Chapter 4), or centroids are not the same, as will be discussed in Section 5.3. We designate the set of angular values as $\mathbb{A}$, and model them as points on the unit circle $\mathbb{S}_1$. Examples of angular parameters are the pitch, the roll, the heading, the latitudes and longitudes etc. Note that these parameters are core parameters that are often analyzed in FDM studies. Additionally certain data like for example the time of the day on a 24 hour cycle can be considered as circular values.

3. *Discrete parameters.* Discrete parameters are parameters which can only be in one of a finite number of states (sometimes also called modes) at a

Autopilot Status		
State	b_0	b_1
OFF	0	0
Flight Director	0	1
Command	1	0
Undefined	1	1

Table 5.1: The 3-state discrete parameter Autopilot Status

time. In the most common case, discrete parameters can only be in one of two states, and are thus called binary. A binary parameter can be used for example to model the state of a switch or an alarm. Some FDM software define discrete parameters that can be in more than two states. This is done by combining several raw binary parameters, and leads to a much easier interpretation as the analyst only has to look at one parameter instead of several. For example, the autopilot status discrete parameter can take one of 3 different states and is defined using two raw binary parameters, as illustrated in Table 5.1. We denote in this case the set of states as $\mathbb{D}$; such that for example for autopilot status, $\mathbb{D}$ = OFF, FD, CMD

4. *Ordered discrete parameters.* Like discrete parameters, ordered discrete parameters can only take a finite number of states. However, these states can be ordered. The easiest way to model this order is to associate to each state a real number, which not only accounts for the order, but can also be used to derive distances and similarities between states, which will be put to use when designing a kernel. In this case, ordered discrete parameters can be considered as continuous parameters. Examples of ordered discrete parameters include the FLAP parameter, which describes the angle of the flaps on the wing of the aircraft. The FLAP parameter can only take values in the $0, 10, 15, 30, 35$ set. We denote in this case the set of states as $\mathbb{O}$, such that for example for the FLAP parameter, $\mathbb{O} = 0, 10, 15, 30, 35$.

In this regard it is important to note that what matters for our studies are not the "raw" parameters as they are recorded in a flight recorder in an aircraft; but rather the high-level engineering values that are computed by a software such as

ALT	8500
AIRSPEED	254
AUTO	ON
PITCH	5°

Table 5.2: Structure of a time-sample

the AGS, and that are the result of several steps of decoding, validation, filtering etc., as briefly explained in Section 1.5.3. It is these high-level engineering values that will be the input of the algorithms that are established in this thesis.

Goal of the Chapter

The goal of this chapter is to establish the necessary mathematical tools such that the method we develop in this thesis is able to handle properly all four types of parameters that we encounter in the field of flight data monitoring. More exactly, using the terms defined in the introduction of this thesis, the goal of this chapter is to construct a kernel k on the space of *time-samples*, that we denote by $\mathcal{X}$. Recall that we call a time-sample the data that is recorded at each instant in an aircraft, and that may contain several parameters of different types, as described for example in Table 5.2.

The kernel should have suitable properties that will be necessary for further developments. In particular it should be a positive definite, infinitely divisible, radial basis kernel. Moreover it should be consistent with the simpler case of real vectors, and if possible it should be compliant with kernel entropy component analysis as defined in Chapter 2.

As explained in Section 1.6, once we have defined a kernel k on the space $\mathcal{X}$ of time-samples, we can then construct a kernel k^* on the space $\mathcal{X}^*$ of *sequences of time-samples*, as will be shown in Chapter 6.

Organization

In Section 5.2 we briefly present two previous attempts at solving this problem in the field of FDM, and address their shortcomings. In Section 5.3 we propose a

different kernel for each parameter type that is encountered in the field of FDM. Recall that $\mathbb{R}$ is the state space for continuous parameters, $\mathbb{D}$ is the state space for discrete parameters, $\mathbb{A}$ is the state space for angular parameters, $\mathbb{O}$ is the state space for ordered discrete parameters; then we will propose the following kernels:

$$\begin{aligned} k_c : &\quad \mathbb{R} \times \mathbb{R} &\to \mathbb{R} \\ k_d : &\quad \mathbb{D} \times \mathbb{D} &\to \mathbb{R} \\ k_a : &\quad \mathbb{A} \times \mathbb{A} &\to \mathbb{R} \\ k_o : &\quad \mathbb{O} \times \mathbb{O} &\to \mathbb{R} \end{aligned}$$

Section 5.4 is the core section of this chapter and explains how using the product approach one can combine the kernels k_c, k_d, k_a, k_o to construct a kernel k on a space $\mathcal{X} = \mathbb{R}^{d_c} \times \mathbb{D}^{d_d} \times \mathbb{A}^{d_a} \times \mathbb{O}^{d_o}$ of a combination of any number of any parameter types. Finally in Section 5.5 we illustrate that the product approach estimates the precise dependency relation between the different components whereas the sum approach is useful when the function to be estimated is the result of the "sum of influences" from the different components.

5.2 Previous Approaches

In the field of FDM, previous studies [Amidan and Ferryman, 2005; Das et al., 2010] have began experimenting different approaches for solving this problem, especially the case of binary parameters.

5.2.1 NASA Morning Report

The NASA *Morning Report* [Amidan and Ferryman, 2005], takes a feature-based approach at solving the problem of the FDM data structure. The first step in the analysis is the creation of a feature vector for each flight, what is called in the article a *mathematical signature*.

For continuous parameters, the mathematical signature consists in a set of 4 statistics (mean value, standard deviation, minimum, maximum) on coefficients of a quadratic model estimated on a sliding window of 11 seconds over the entire flight phase that is studied. As there are 4 coefficients for the quadratic model

(including the error between actual and predicted values) the end result is thus a vector of 16 real values for each continuous parameters that is studied.

For discrete parameters, the mathematical signature is the transition matrix of the parameter. Each entry in the matrix counts the proportion of times that the parameter went from a mode to another. Consequently, the diagonal entries count the proportion of time that a parameter stayed in a particular state. The transition matrix is then reshaped into a vector, so for example a parameter with 3 possible states as illustrated in Table 5.1 will have a 3×3 transition matrix which is then reshaped into a vector of 9 elements.

When studying several parameters the signature of a flight is the vector resulting from the concatenation of all signature vectors of all parameters. For example, when studying 3 continuous parameters and one discrete parameters with 5 states, this results in a real vector of $3 \times 16 + 5^2 = 73$ values.

The great advantage of using such a feature-based approach is that the representation of flights is now a convenient vector of real values, and thus one can use a very large set of tools and methods (the Morning Report uses a principal component analysis and also a clustering in later versions). Moreover this approach is sufficient to extract global and important properties of the flights.

Nonetheless, this approach falls short in the cases where one wants to detect problems that are really localized in time, and which thus do not transpire in the feature vector.

5.2.2 NASA MKAD

The NASA *Multiple Kernel learning for heterogeneous Anomaly Detection* takes a very different approach, and focuses on kernel methods. Firstly, the MKAD uses the SAX representation [Lin et al., 2003] for continuous parameters. The SAX representation consists in first averaging values over fixed length window, and then assigning to each averaged value a symbol according to its deviation from the mean taken over all flights. This is illustrated in Figure 5.1.

A similarity matrix K_c is then created such that for two SAX sequences x_i

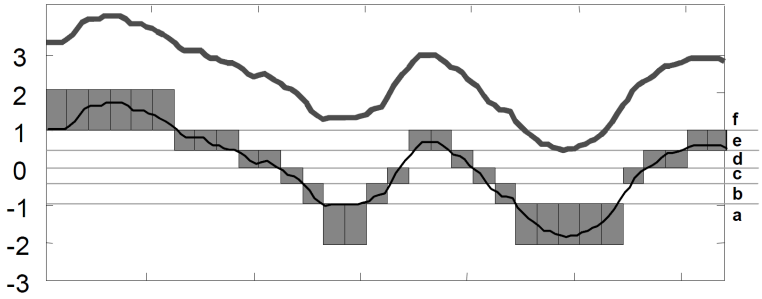


Figure 5.1: Continuous parameter transformed into SAX representation: ffffffeeeddcbaabceedbcaaaaacddee

and x_j of a continuous parameter of any two flights numbered i and j:

$$K_{cij} = \frac{\bigcup \text{LCS} x_i, x_j \bigcup}{\bigcup x_i \bigcup \cdot \bigcup x_j \bigcup}.$$

Where LCS is the *longest common subsequence* [Paterson and Dančík, 1994] between two sequences, where we denote by $\bigcup x \bigcup$ the length of any sequence x. If there are more than one continuous parameters in the study then the resulting similarity matrix is the mean of all similarity matrices.

Similarly for discrete parameters a separate similarity matrix K_d is computed, using once again the length of the longest common subsequence as a similarity measure.

To combine the two matrices K_c and K_d, the MKAD uses the multiple kernel approach as proposed by Bach et al. [2004]:

$$K = \eta K_d + 1 - \eta K_c$$

Where $\eta \in (0, 1]$ is a parameter set by the user and used to weight the influence of the two matrices. The similarity matrix K is then used as a Gram matrix in the context of kernel methods, and used as input to an improved one-class SVM named ν-anomica [Das et al., 2009].

There are however several flaws to this method. First, although the authors claim that the length of the longest common subsequence yields a positive semi-

definite matrix it has been proved otherwise [Vert, 2008]. Secondly, we do not think that summing the Gram matrices is the right approach when dealing with samples of heterogeneous types, as will be explained in Section 5.5. Finally, authors do not make the crucial distinction between continuous and angular parameters.

5.3 The Four Types of Parameters in FDM

In this section we propose a kernel for each of the four types of parameters that is typically encountered in the field of FDM. In addition, when appropriate we also discuss the issues of estimating centroids. We use the term "centroid" to designate an element that is as central as possible with respect to a set of elements. Note that in some cases the centroid may not be an element of the set. It could be seen as a generalization of the mean but for arbitrary types of data. Estimating a centroid will be useful many times in our endeavor. For example when downsampling a time series with a period Δ_t it is better to take the centroids across periods $\big(0, \Delta_t\big($, $\big(\Delta_t, 2\Delta_t\big($, etc. rather than just the values at times $0, \Delta_t, 2\Delta_t$, etc.

5.3.1 Continuous Parameters

Continuous parameters are the most common and easiest to deal with type of parameters. They are simply modeled by numbers of the real line $\mathbb{R}$.

5.3.1.1 Kernel on Continuous Values

The usual Euclidian distance $\big|x_i - x_j\big|$ (note that here x_i and x_j are not vector but simple real values) is of course a Hilbertian metric, and it is widely known that the corresponding kernel:

$$k_c(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2\right)$$

is positive definite and infinitely divisible.

5.3.1.2 Centroid for Continuous Values

The most common way to estimate a centroid in a set $S = x_1, \ldots, x_n$ of n continuous values is the mean:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{5.1}$$

Being a linear function of its entries the mean is very convenient to handle from a mathematical point of view; however it suffers from being very sensitive to outliers [Huber, 1981]. As the robustness is a very important property for any unsupervised novelty detection algorithm as explained in Chapter 2, we shall rather use the median whenever possible. The median is defined as the numerical value separating the higher half of the population from the lower half.

5.3.2 Angular Parameters

As explained in the introduction angular parameters are quite common in the field of FDM. We model them by circular values. Let θ_i and θ_j be two angular values, such that $\theta_i \in [0, 2\pi[$ and $\theta_j \in [0, 2\pi[$.

Point Representation Throughout this subsection we will use a two-dimensional vector representation of angular values. We model angular values θ_i and θ_j by points $\mathbf{x_i}$ and $\mathbf{x_j}$ on the unit circle $\mathbb{S}_1$ of $\mathbb{R}^2$, such that:

$$\mathbf{x_i} = \begin{matrix} \cos\theta_i \\ \sin\theta_i \end{matrix} \quad \text{and} \quad \mathbf{x_j} = \begin{matrix} \cos\theta_j \\ \sin\theta_j \end{matrix}$$

5.3.2.1 Distances on Circular Values

There are at least three different ways to define a distance on circular values. Each has different mathematical properties, and some may be more appropriate for use within an FDM system.

Absolute Angular Difference The most obvious way to define a distance between two angular values θ_i and θ_j is the absolute angular difference, that we

denote d_0:

$$d_0\theta_i, \theta_j = \min_{\bigcup} \theta_j - \theta_i_{\bigcup}, 2\pi - {}_{\bigcup}\theta_j - \theta_i{}_{\bigcup}$$

Such that for any θ_i and θ_j, $d_0\theta_i, \theta_j \in {}_{(}0, \pi{}_{]}$. It is clear that the absolute angular difference between θ_i and θ_j is the shortest geodesic distance along the circle between $\mathbf{x_i}$ and $\mathbf{x_j}$, that we denote by δ:

$$d_0\theta_i, \theta_j = \delta\mathbf{x_i}, \mathbf{x_j}.$$

The absolute angular difference is the most straightforward way to define a distance between angles but there are two other possibilities that have interesting mathematical properties.

Cosine Distance Another possible distance definition is the cosine distance, denoted as d_1:

$$d_1\theta_i, \theta_j = 1 - \cos\theta_i - \theta_j \tag{5.2}$$

The cosine distance is a monotonous increasing function of the absolute angular difference, it ranges from 0 to 2. The interest of this distance from a mathematical point of view is that it is linked to the von Mises probability distribution [Forbes et al., 2011]. With a measure of location μ and a measure of concentration κ, the von Mises distribution can be expressed as:

$$fx_{\bigcup}\mu, \kappa = \frac{e^{\kappa\cos x-\mu}}{2\pi I_0\kappa}$$

Where I_0x is the modified Bessel function of order 0. When κ is close to 0 the distribution is close to uniform, whereas with large κ the distribution is concentrated around the angle μ. The von Mises distribution has many interesting properties which explains that it is also called the *circular normal distribution*. Among them, given an expectation on the circle, it is the distribution that maximizes the entropy. Additionally, it is the distribution on the circle whose location parameter is estimated with maximum likelihood by the sample circular mean, which we shall define in Section 5.3.2.3.

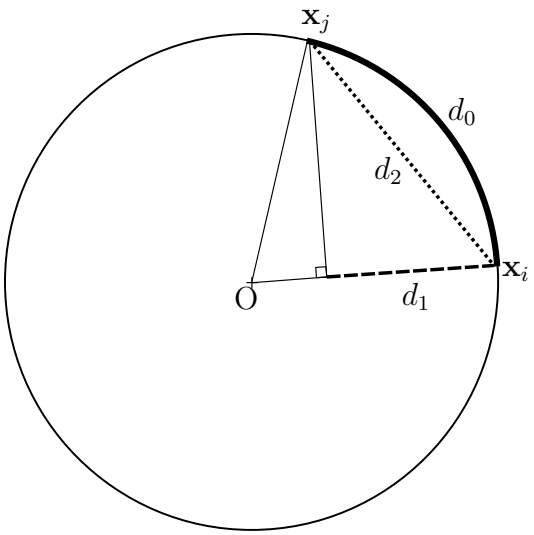


Figure 5.2: Illustration on the unit circle of the three angular distances d_0, d_1 and d_2.

Euclidian Distance A third way to define distances between angular values is simply to consider the Euclidian distance between the vector representations of angular values. We denote this distance by d_2:

$$d_2\theta_i, \theta_j = \prod \mathbf{x_i} - \mathbf{x_j} \prod$$

Note that d_2 can be expressed as

$$d_2\theta_i, \theta_j = 2 \sin \frac{\bigcup \theta_i - \theta_j \bigcup}{2}$$

Similarly the Euclidian distance is a monotonous increasing function of the absolute angular difference, and it ranges from 0 to 2. Note also that for small angular differences,

$$\bigcup \theta_j - \theta_i \bigcup \approx 0 \implies d_2\theta_i, \theta_j \approx d_0\theta_i, \theta_j$$

The difference between these three distances is illustrated in Figure 5.2.

5.3.2.2 Kernel on Circular Values

Using the distances expressed previously it is now possible to define kernels on circular values. We present here two possibilities, using respectively the distances d_1 and d_2. Although it would make sense from a FDM perspective to define a kernel based on the distance d_0, we haven't found any use of such kernel in the literature, and it is unclear if the properties that we seek, especially positive definiteness, are valid in this case.

The zonal kernel The most commonly used kernel for circular values is the *zonal kernel* [Fasshauer, 2011], which uses the cosine distance as defined in Equation 5.2:

$$k'_a\theta_i, \theta_j = \exp\left(-21 - \cos \bigcup \theta_j - \theta_i \bigcup\right]$$

The zonal kernel is linked to the von Mises probability distribution [Forbes et al., 2011], differing only by a multiplicative factor.

The spherical Gaussian kernel Another very simple way to design a kernel that would be positive definite by design is to take the evaluation of the standard 2-dimensional Gaussian kernel on the points $\mathbf{x_i}$ and $\mathbf{x_j}$, in other words to take the exponential kernel associated to the Euclidian distance d_2 between angles:

$$k_a\theta_i, \theta_j = \exp -d_2\theta_i, \theta_j{}^2 = \exp\left(-4\sin^2 \frac{\bigcup \theta_j - \theta_i \bigcup}{2}\right]$$

This kernel is sometimes called the *spherical Gaussian kernel* [Fasshauer, 2011]. The spherical Gaussian kernel is our kernel of choice for dealing with angular values. Firstly, as explained before, for close angles its value is approximately the one that would be obtained using the d_0 distance, which is the most sensible distance from a domain perspective. Secondly, as $\mathbf{x_i}$ and $\mathbf{x_j}$ lie on the unit sphere which is merely a subset of $\mathbb{R}^2$, then by virtue of Theorem 8 we are assured that the kernel k_a has all the properties that we seek: it is a positive definite, radial basis and infinitely divisible kernel.

5.3.2.3 Centroid and Measure of Dispersion

Centroid Obviously one cannot use Equation 5.1 for angular values. Once more we make use of the representation of angular values as points on the unit circle in order to solve this issue. Let us consider a set of n angular values $S = \theta_1, \ldots, \theta_n$ and their associated points on the unit circle $\mathbf{x_1}, \ldots, \mathbf{x_n}$. First we compute the mean of $\mathbf{x_1}, \ldots, \mathbf{x_n}$, which generally does not lie on the unit circle except for the very particular case where all angular values are identical:

$$\bar{\mathbf{x}} = \frac{1}{n} \mathop{\mathcal{P}}_{i=1}^{n} \mathbf{x}_i$$

The coordinates of $\bar{\mathbf{x}}$ can also be expressed using the angular values:

$$\bar{\mathbf{x}} = \begin{matrix} \frac{1}{n} \mathcal{P}_{i=1}^{n} \cos\theta_i \\ \frac{1}{n} \mathcal{P}_{i=1}^{n} \sin\theta_i \end{matrix}$$

A centroid for the set S can then be defined as the angular value associated to $\bar{\mathbf{x}}$, with respect to the origin, whenever $\bar{\mathbf{x}}$ is different from the origin:

$$\bar{\theta} = \text{atan2}\left(\frac{1}{n} \mathop{\mathcal{P}}_{i=1}^{n} \sin\theta_i, \ \frac{1}{n} \mathop{\mathcal{P}}_{i=1}^{n} \cos\theta_i \right] \tag{5.3}$$

Where atan2 is a quadrant-specific version of the arc-tangent function.

Measure of dispersion Note in addition that the norm of $\bar{\mathbf{x}}$ can be used to define a measure of dispersion. The closer $\bar{\mathbf{x}}$ lies to the unit circle, the more we can say that the angular values are concentrated. Conversely, $\bar{\mathbf{x}}$ may be close the origin if the angular values are evenly spread across the unit circle. Consequently, a common measure for the dispersion of angular values is

$$1 - R, \quad \text{where } R = \prod \bar{\mathbf{x}} \prod. \tag{5.4}$$

Angular centroid and dispersion measure are illustrated in Figure 5.3. Note that both this centroid and this measure of dispersion are sensible from a statistical perspective. Let us define $D_1\alpha, S = \frac{1}{\bigcup S \bigcup} \mathcal{P}_{\theta \epsilon S} \, d_1\alpha, \theta$ the mean of the cosine

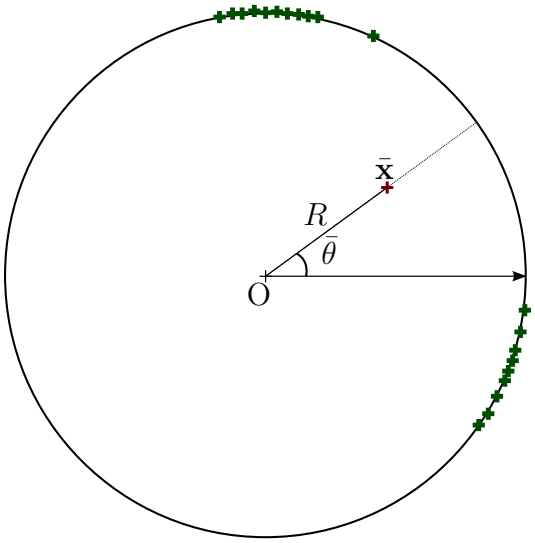


Figure 5.3: Illustration of the centroid and dispersion measure of angular values.

distances between an angle α and all angles in the set S. Then it has been demonstrated [Jammalamadaka and Sengupta, 2001] that for any set S with circular mean $\bar{\theta}$ as defined in Equation 5.3 and dispersion $1 - R$ as defined in Equation 5.4:

$$\min_{\alpha \in (0,2\pi(} D_1\alpha, S = 1 - R$$

$$\operatorname{argmin}_{\alpha \in (0,2\pi(} D_1\alpha, S = \bar{\theta}$$

In other words, the circular mean is the angular value which minimizes the sum of cosine distances, and this sum is related to the dispersion measure.

5.3.3 Discrete Parameters

Dealing with discrete parameters is very different from continuous or angular parameters. In this section we propose a simple way to derive a kernel for discrete values, and we briefly address the problem of defining a centroid for this type of data.

	OFF	FD	CMD
OFF	0	1	1
FD	1	0	1
CMD	1	1	0

Table 5.3: Distance matrix for parameter Autopilot Status

5.3.3.1 Distances on Discrete Values

Distances on a discrete parameter with state space $\mathbb{D}$ can be completely defined by a matrix of size $\bigcup\mathbb{D}\bigcup \times \bigcup\mathbb{D}\bigcup$. For example in the case of the Autopilot Status parameter, one could use a distance matrix as defined in Table 5.3. Of course one could tweak the values of such matrix in order to get closer to a representation that would be sensible from a domain perspective, for example such that the CMD state may be closer to FD than to OFF. However not all values lead to matrix that represents valid distances, much less Hilbertian ones, as defined in Chapter 4. This problem of incorporating information about the "topology" of discrete states, or in other words how much states are close to one another is treated in part in Section 5.3.4, where we define *ordered discrete parameters*. Here we consider the most simple topology, such as the one described in Table 5.3, even if it could be argued in this case that indeed Autopilot Status could be considered as an ordered discrete state.

General case Let us consider a discrete parameter with state space $\mathbb{D}$ of $\bigcup\mathbb{D}\bigcup = q$ states. The distance we consider is thus:

$$d x, y = \begin{array}{ll} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{array}$$

Which means that any two different states are at the same distance. First we shall see that this distance is a Hilbertian distance, and consequently we will be able to design an infinitely divisible kernel by virtue of Theorem 8. Let $\mathbb{D} = x_1, \ldots, x_q$, where each $x_1, \ldots, x_q$ represents a possible state, or mode, of this parameter. We

define a mapping ϕ such that:

$$\phi : \begin{array}{rl} \mathbb{D} & \to \mathbb{R}^q \\ x_i & \mapsto 0, \ldots, \underbrace{1}_{i^{\text{th}} \text{ position}} \underset{\Uparrow}{2}, \ldots, 0 \end{array}$$

Using this mapping, it is trivial to see that we have that for any two states $x, y \in \mathbb{D}$,

$$dx, y = \prod \phi x - \phi y \prod$$

As $\mathbb{R}^q$ is a Euclidian space, then we have by definition that d is a Hilbertian metric on $\mathbb{D}$. We can thus define a kernel k_d on discrete values as:

$$k_d : \begin{array}{rl} \mathbb{D} \times \mathbb{D} & \to \mathbb{R} \\ x, y & \mapsto \exp -dx, y^2 \end{array}$$

By virtue of Theorem 8, k_d is a positive definite, infinitely divisible, and radial basis kernel.

Moreover, as a consequence of the infinite divisibility, and because $x \mapsto \exp -x$ is a bijection from $\left(0, \infty\right($ to $\left]0, 1\right]$ then any kernel of the form:

$$k_d : \begin{array}{rll} \mathbb{D} \times \mathbb{D} & \to \mathbb{R} & \\ x, y & \mapsto \begin{array}{ll} 1 & \text{if } x = y \\ a & \text{if } x \neq y \end{array} & \end{array}$$

where $a < 1$ is also a valid kernel with the same properties enunciated before.

5.3.4 Ordered Discrete Parameters

In order to deal with discrete parameters we use a very simple model. As briefly explained in the introduction of this chapter, we associate a real number to each possible state of the ordered discrete parameter. Sometimes this real number can naturally be defined, for example in the case of the FLAPC parameter, each possible state $0, 10, 15, 30, 35$ is already associated to a real number. However sometimes there are discrete states that are not already associated to a real

number but we still wish to bring an order (or more generally a topology) to these states. For example as described previously, the AUTO parameter can naturally be considered as a discrete (unordered) parameter. But some FDM experts may argue that the CMD state should be closer to the FD state than to the OFF state. Thus an association function ϕ from the state space $\mathbb{O}$ to $\mathbb{R}$ may be for example:

$$\phi : \begin{array}{rl} \mathbb{O} & \to \mathbb{R} \\ \mathsf{OFF} & \mapsto 0 \\ \mathsf{FD} & \mapsto 1 \\ \mathsf{CMD} & \mapsto 2 \end{array}$$

5.3.4.1 Kernel on Ordered Discrete Parameters

Once in $\mathbb{R}$ one can simply use the Gaussian kernel, which has already all the properties we wish our kernel to have (positive definite, infinitely divisible and radial basis):

$$k_o : \begin{array}{rl} \mathbb{O} \times \mathbb{O} & \to \mathbb{R} \\ x, y & \mapsto \exp -\left\| \phi x - \phi y \right\|^2 \end{array}$$

5.3.4.2 Centroid and Measure of Dispersion

Similarly, for a centroid one can simply take the mean of the associated real values; and the standard deviation as a measure of dispersion. Note that in this particular case the centroid does not in the general case belong to the state space $\mathbb{O}$, but this is not an issue as it is easy to deal with real values.

5.4 Combining Kernels for Different Types of Parameters

In this section we present a way to combine the kernels k_c, k_a, k_d, k_o in order to construct a kernel k on the space $\mathcal{X} = \mathbb{R}^{d_c} \times \mathbb{D}^{d_d} \times \mathbb{A}^{d_a} \times \mathbb{O}^{d_o}$ of time-samples. This problem of incorporating data from heterogeneous sources has already been studied in the field of kernel methods, and there are roughly two approaches

to solving it: either consider the sum of the kernels (or more exactly the conic combination) [Bach et al., 2004; Sonnenburg et al., 2006] or the product [Haussler, 1999; Kondor and Lafferty, 2002; Shin and Kuboyama, 2008]. Note that the main goal of [Bach et al., 2004] is to *learn the weight parameters of the kernels*, in a supervised setting, by developing an efficient procedure for solving the resulting quadratically constrained quadratic program. Hence in [Bach et al., 2004] the idea in using multiple kernels is to enhance performance in a supervised setting, not necessarily to incorporate data from heterogeneous sources; as for example multiple kernel matrices may come from the same data sources but with different normalization factors.

As explained before, one of the previous attempts to applying kernel methods to the field of FDM we have studied, the NASA MKAD [Das et al., 2010] uses the conic combination approach to combine data from multiple parameters. It is our belief that this is not the right approach for this problem, and in this section we present the kernel product approach. In Section 5.5 we compare both approaches.

5.4.1 Mathematical Framework

In order to simplify our exposition we restrict ourselves to the case where each sample is a composite structure comprising two elements, one from a space $\mathcal{X}$ and one from a space $\mathcal{Y}$. The extension to a more general case with an arbitrary number of elements is trivial. We denote the composite input space as $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. We thus consider the following dataset:

$$X_1, Y_1, \ldots, X_n, Y_n$$

We assume that each space $\mathcal{X}$ and $\mathcal{Y}$ is endowed with a kernel, respectively k_X and k_Y, and that these kernels are positive definite, radial basis and infinitely divisible. The goal is thus to construct a kernel k_Z on $\mathcal{Z}$ which has the same properties, and which is able to properly estimate the dependency relation between the two variables X and Y.

5.4.2 The Product Kernel

The idea is to consider the *generalized product kernel* k_Z, also called tensor product in some publications [Haussler, 1999]. We use the term *generalized* to emphasize the fact that spaces $\mathcal{X}$ and $\mathcal{Y}$ may be different. The generalized product kernel is defined as follows:

$$\begin{aligned} k_Z : \mathcal{X} \times \mathcal{Y} \times \mathcal{X} \times \mathcal{Y} &\to \mathbb{R} \\ x_0, y_0, x_1, y_1 &\mapsto k_X x_0, x_1 \cdot k_Y y_0, y_1 \end{aligned}$$

We shall now examine the properties of the resulting kernel k_Z.

5.4.2.1 Positive Definite

The positive definiteness of the kernel k_Z is directly inherited from results on the Schür algebra of matrices [Horn and Johnson, 2012].

Considering the samples $X_1, Y_1, \ldots, X_n, Y_n$, let K_x be the Gram matrix constructed using kernel k_X and K_Y the Gram matrix constructed using kernel k_Y. Then the Gram matrix K_Z constructed using the kernel k_Z is the Schür product of the matrices K_X and K_Y:

$$K_Z = K_X \otimes K_Y$$

The Schür product, also known as the Hadamard product, is the entrywise product between two matrices, such that $K_{Zi,j} = K_{Xi,j} \cdot K_{Yi,j}$.

The Schür product theorem [Horn and Johnson, 2012] guarantees that the Schür product of two positive definite Gram matrices is a positive definite matrix. Consequently, the generalized product kernel k_Z is positive definite.

5.4.2.2 Radial Basis

The radial basis property is easier to prove. Let $z_0 = x_0, y_0, z_1 = x_1, y_1$ two samples from $\mathcal{Z}$. As both k_X and k_Y are supposed to be radial basis, then $0 \leq k_X x_0, x_1 \leq 1$ and $0 \leq k_Y y_0, y_1 \leq 1$ which leads to $0 \leq k_Z z_0, z_1 \leq 1$. Additionally $z_0 = z_1$ if and only if $x_0 = x_1$ and $y_0 = y_1$, consequently $z_0 = z_1$ if and only if $k_Z z_0, z_1 = 1$, which

proves this property.

5.4.2.3 Infinitely Divisible

Let $n \in \mathbb{N}^*$. As both k_X and k_Y are supposed to be infinitely divisible, then $k_X^{\frac{1}{n}}$ and $k_Y^{\frac{1}{n}}$ are positive definite. According to the results established in Section 5.4.2.1, the kernel $k_X^{\frac{1}{n}} \cdot k_Y^{\frac{1}{n}} = k_X \cdot k_Y^{\frac{1}{n}} = k_Z^{\frac{1}{n}}$ is positive definite; which proves that k_Z is infinitely divisible.

5.5 Comparison with the Conic Combination Approach

5.5.1 General Principles

We argue that the choice of the method to combine the kernels should depend on the semantic of the data: either each kernel represents a different representation of the data, or the data are of composite structure and each kernel represents a way to compare a component of this structure.

A common illustration is the case of the proteins: a protein can be considered as an amino-acid sequence, a macromolecule with a 3D-structure, an expression level in a DNA-chip or even a node in a biological pathway. Each representation can be associated to a kernel and it is possible to combine all these kernels using a conic combination:

$$k = \sum_{l=1}^{m} \beta_l k_l, \quad \text{with } \sum_{l=1}^{m} \beta_l = 1 \text{ and } \forall l,\ \beta_l > 0$$

A kernel algorithm generally yields a linear combination of point-wise evaluations of the kernel functions:

$$fx = \sum_{i=1}^{n} \alpha_i kx, X_i = \sum_{l=1}^{m} \beta_l \sum_{i=1}^{n} \alpha_i k_l x, X_i$$

Thus it can be seen that it makes sense to use a conic combination when the function to be estimated represents a "sum of influences". In this case, an algorithm such as the one described in [Bach et al., 2004] can be used in a supervised

setting to learn the parameters $\beta_1, \ldots, \beta_l$ as well as the coefficients $\alpha_1, \ldots, \alpha_n$.

On the contrary, in the case where one has to deal with a composite structure, for example samples with mixed data $Z_i = X_i, Y_i$ and the goal is to estimate a function that takes into account the dependency relation between each component, one has to multiply the associated kernels as described in this section.

To understand this let us review the case of radial basis kernels that may represent degrees of similitudes that range from 0 to 1. Two samples $Z_1 = X_1, Y_1$ and $Z_2 = X_2, Y_2$ of composite structure can be considered similar if

$$X_1 \text{ is similar to } X_2 \quad \text{AND} \quad Y_1 \text{ is similar to } Y_2$$

Which translates into the *multiplication* of the similarity functions.

For example in the trivial case of two-dimensional vectors in an Euclidian space it would not occur to a practitioner to use the kernel $\exp\left(-x_1 - x_2^2\right) + \exp\left(-y_1 - y_2^2\right)$ instead of $\exp\left(-\left(x_1 - x_2^2 + y_1 - y_2^2\right)\right) = \exp\left(-x_1 - x_2^2\right) \cdot \exp\left(-y_1 - y_2^2\right)$.

Implications for the number of samples It is very important to understand that when one knows in advance that the function to be estimated can be modeled as a sum of (independent) influences then summing the kernels is the right approach and requires far less samples to yield a proper estimation. We here present some informal arguments to understand this phenomenon.

Suppose for example that the functions associated to the kernels $k_1, \ldots, k_m$ require respectively $n_1, \ldots, n_m$ samples for a sufficiently precise estimation. Then it is clear that the function of interest f would require a number of samples in the order of $\max n_1, \ldots, n_m$ to be correctly estimated.

On the contrary, in the case where the problem is to estimate a proper dependency relation between components of a composite structure (where the product kernel is the right approach) then one cannot escape the *curse of dimensionality*. In this case the number of samples required will be in the order of $n_1 \times \ldots \times n_m$.

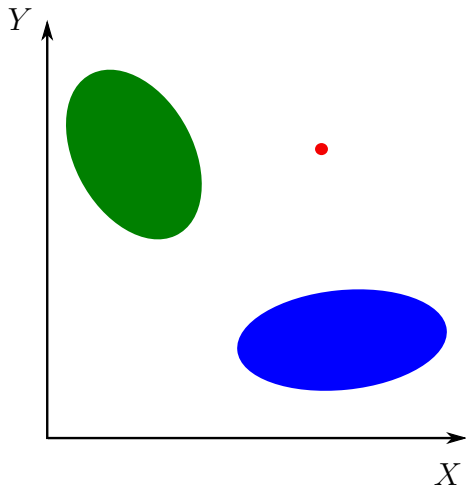


Figure 5.4: Novelty due to "bad synchronization"

5.5.2 Interest for Novelty Detection

We illustrate further the differences between the conic combination and the product approach by studying the implications in the context of novelty detection. Let us consider a novelty that arises due to what is sometimes called a problem of "bad synchronization" between two parameters X and Y. Such a novelty cannot be detected by looking at the univariate distributions, it results from an unusual dependency relation between the two parameters X and Y. This is illustrated in Figure 5.4, where we have two clusters represented by blue and green ellipses, and one red point which is a novelty. It is clear in this figure that the novelty cannot be detected by looking only at univariate histograms for example.

Let us now investigate how such a dataset would be represented using Gram matrices. In Figure 5.5 we represent respectively the Gram matrices K_X, K_Y, $\frac{1}{2}K_X + K_Y$ and $K_X \otimes K_Y$. The indexes of the Gram matrices are all ordered in the same way, the first points are from the green cluster, then comes the novelty, then the points from the blue cluster. This ordering is apparent in the choice of colors. Furthermore the intensity of the color represents the magnitude of the values. To simplify we could say that the clusters are infinitely compact, such that bright colors represents values of 1, washed colors represent values of 0.5 and white represents a value of 0. We can see in Figure 5.5 that when using a conic

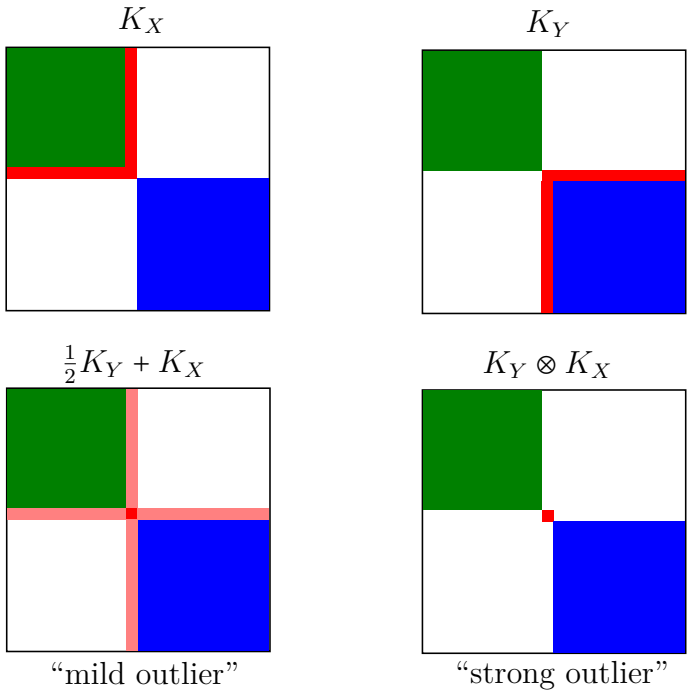


Figure 5.5: Comparison of Gram matrices

combination approach the novelty is "diluted" in the two clusters, which results in a mild outlier. Conversely, using the product approach the novelty is clearly independent from the two clusters and results in a strong outlier.

Using a novelty detection algorithm based on a decomposition of the Gram matrix in eigenvectors such as the one proposed in Chapter 2 would result in each cluster belonging to a subspace of its own in the kernel feature space, these two subspaces being orthogonal. Moreover the novelty itself would be in its own subspace, orthogonal to the subspaces of the two clusters. However, retaining the principal subspace with the highest variance (or entropy as explained in Chapter 2) results in retaining the subspaces corresponding to the two clusters. Consequently the novelty would thus lie far from the principal subspace, and this point would certainly be detected as a novelty based on the reconstruction error.

5.6 Conclusion

We have thus seen that using the product approach yields a kernel that possesses all three important properties defined in Chapter 4, it is a positive definite, radial basis and infinitely divisible kernel. Using this approach we are now able to construct a suitable kernel for the space of time-samples $\mathcal{X} = \mathbb{R}^{d_c} \times \mathbb{D}^{d_d} \times \mathbb{A}^{d_a} \times \mathbb{O}^{d_o}$. In Chapter 6 we describe the next step for the development of our kernel for comparing flights: we build upon the kernel k on $\mathcal{X}$ in order to construct a kernel k^* on $\mathcal{X}^*$ which is the space of *sequences* of elements of $\mathcal{X}$.

Chapter 6

Data as Sequences

6.1 Introduction

Goal of the Chapter In the previous chapter, we have designed a kernel k on a space $\mathcal{X}$ which models the set of all possible *time-samples*. In other words, the kernel k can be used to compare the data that is recorded at one instant with the data that is recorded at another instant. In this chapter, we present the final step in the creation of the kernel on flights. As explained in the introduction of this work, we model a flight as a *sequence of time-samples*. Denote by $\mathcal{X}^*$ the space of finite sequences with elements in $\mathcal{X}$, such that $\mathcal{X}^* = \cup_{i=1}^{\infty} \mathcal{X}^i$. In the literature the kernel k is sometimes called the *base kernel* or *ground kernel*, and the space $\mathcal{X}$ the *ground space*. The goal is now to design a *sequence kernel* k^* on $\mathcal{X}^*$ with suitable properties.

Framework Let $\boldsymbol{x} = x_1, \ldots, x_l$ and $\boldsymbol{x'} = x'_1, \ldots, x'_m$ two elements of $\mathcal{X}^*$. In the general case, these two elements may not have the same length, and thus one cannot use traditional vector-based approaches such as a Gaussian kernel in an Euclidian space to compare these sequences. When dealing with sequences of discrete elements one solution is to compute the minimum number of operations (such as insertion, deletion and modification) needed to obtain one sequence from another. This number of operations can be considered as a distance, and is called the Levenshtein distance [Levenshtein, 1966] or more generally the edit-distance.

Another solution is to define alignments between sequences. An alignment associates elements from one sequence to elements in another sequence such that the order of elements is preserved.

The first attempts to deal with this problem resulted in the well-known dynamic time warping method [Sakoe and Chiba, 1978], which seeks the best alignment between two sequences, and which results in the so-called optimal assignment kernel. Although this kernel has been extensively used by practitioners it has been demonstrated recently that it is in fact not positive definite [Vert, 2008]. Since then some researchers have proposed alternatives, such as for example the global alignment kernel [Cuturi et al., 2007] or the spectrum kernel [Leslie et al., 2002].

Methodology The kernel we propose in this chapter is novel in two regards. Firstly, we only consider a particular kind of alignments with repetitions, these in which *only the shorter sequence can have repeated elements*, hence the name "one-sided". Secondly, instead of only retaining the best alignment like in the optimal assignment we rather consider the mean (in a sense which shall be clarified) of all alignment scores.

In this work we will not use the classical formalism of alignments, but rather refer to what we call "dilatation operators". These will be precisely defined in Section 6.4.2.1, but we can already define them informally: a dilatation operator is a function that maps a finite sequence to a longer finite sequence by repeating one or more of its elements.

Contributions We demonstrate using the theory of infinitely divisible kernels that the proposed kernel is positive definite. We also illustrate many other interesting practical properties: it is a radial basis kernel, has no issues of diagonal dominance, and presents a consistent behavior in the case of time series sub-sampling. We propose an implementation of this kernel using dynamic programming techniques, which result in a complexity in $Ol \times m - l$ for a pair of sequences of respective lengths $l < m$, which is much faster than competing techniques which have a complexity in $Ol \times m$. In the next chapter, we illustrate our approach with promising results in the field of FDM.

```
ATGCCGTGACATGCATTTAAGC
GTG-CGT-ATATG--TTT---C
```

Figure 6.1: Alignments with gaps

```
ATGCCGTGACATGCATTTAAGC
ATGGCGTTACATGGGTTCCCCC
```

Figure 6.2: Alignments with repetitions

6.2 Alignments and Alignment Scores

Let $\boldsymbol{x} = x_1, \ldots, x_l$ and $\boldsymbol{x}' = x'_1, \ldots, x'_m$ two elements of $\mathcal{X}^*$. An alignment associates elements from one sequence to elements in another sequence such that the order of elements is preserved. Alignments can either introduce gaps or repetitions, as illustrated in Figures 6.1 and 6.2. In these figures we use the example of genome sequences for illustration, but of course the concepts presented here are still valid for time series (sequences of real values) or more generally for sequences of structured data. As always the practitioner must choose the right type of alignments based on the application, for example it is commonly agreed that gap alignments are suited to the study of genome sequences whereas alignments with repetitions can be used for trajectory comparisons.

Once an alignment has been found, it is then possible to compute an alignment score between the two sequences, by for example summing the pairwise distances between aligned elements. Consequently, for alignments using gaps there may be elements in one of the two sequences that will not be compared to an element in the other sequence, which does not happen when considering alignments with repetitions.

6.2.1 Global Alignments

In this work we will be interested in alignments that introduce repeating states. Formally we define an alignment π of length p between two sequences of lengths

l and m as a pair π_1, π_2 of p increasing indexes such that:

$$\begin{aligned} 1 = \pi_1 1 \leq \ldots \leq \pi_1 p = l \\ 1 = \pi_2 1 \leq \ldots \leq \pi_2 p = m \end{aligned} \tag{6.1}$$

and

$$\begin{matrix} \pi_1 i + 1 - \pi_1 i \\ \pi_2 i + 1 - \pi_2 i \end{matrix} \in \begin{matrix} 0 \\ 1 \end{matrix}, \begin{matrix} 1 \\ 0 \end{matrix}, \begin{matrix} 1 \\ 1 \end{matrix} \tag{6.2}$$

We denote by $\mathcal{A}\boldsymbol{x}, \boldsymbol{x}'$ the set of all alignments between $\boldsymbol{x}$ and $\boldsymbol{x}'$.

6.2.2 One-Sided Alignments

In this work we will be interested in a particular subset of alignments that we have called one-sided alignments. These are the alignments *where only the shortest sequence can have repeated elements*. Suppose sequence $\boldsymbol{x}$ is shorter than $\boldsymbol{x}'$, such that $l \leq m$; thus the condition on the alignment π becomes:

$$\begin{matrix} \pi_1 i + 1 - \pi_1 i \\ \pi_2 i + 1 - \pi_2 i \end{matrix} \in \begin{matrix} 0 \\ 1 \end{matrix}, \begin{matrix} 1 \\ 1 \end{matrix} \tag{6.3}$$

One should remark that between two sequences of the same length, there exists only one one-sided alignment, which is the trivial alignment $\forall i \in 1 \ldots l$, $\pi_1 i = \pi_2 i = i$. In the general case where $l \leq m$ it can be seen from Equation 6.3 that there are $\binom{m-1}{l-1}$ one-sided alignments between $\boldsymbol{x}$ and $\boldsymbol{x}'$. We denote by $\mathcal{A}^{-}\boldsymbol{x}, \boldsymbol{x}'$ the set of one-sided alignments between $\boldsymbol{x}$ and $\boldsymbol{x}'$.

6.2.3 Representation of Alignments

It is possible to conveniently represent alignments between two sequences of lengths l and m as paths on matrix of size l, m. Note that in this section as well as in the rest of the chapter we shall always represent this matrix with the *shorter sequence as the vertical indexes*. At each step, the vertical position is given by π_1 and the horizontal position is given by π_2. Equation 6.1 means that each path starts at the upper left corner of the matrix and finishes at the lower right corner. Moreover, as is described in Figure 6.3 and 6.4; positions of two

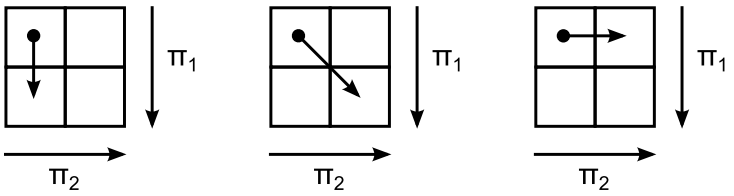


Figure 6.3: Movements in global alignments

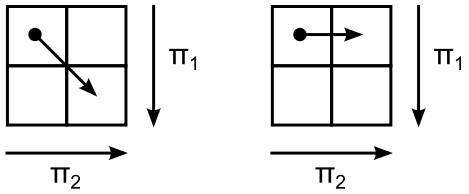


Figure 6.4: Movements in one sided alignments

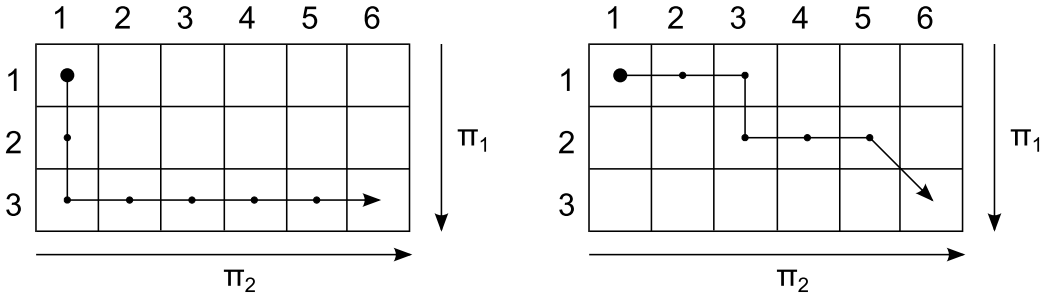


Figure 6.5: Two examples of global alignments

subsequent steps are conditioned by either Equation 6.2 or Equation 6.3 depending on whether the alignment is global or one-sided. Figure 6.5 presents two examples of global alignments between sequences of lengths 3 and 6, and Figure 6.6 presents two examples of one-sided alignments. Note that in Figure 6.6 we have represented some cases of the matrix with stripes: these are cases that are unattainable using one-sided alignments. Because of restrictions in the "movements" as described in Equation 6.3 and illustrated in Figure 6.4, the part of the matrix that can be attained is given by the following equations:

$$\begin{aligned} 1 \leq i \leq l \\ i \leq j \leq m - l + i \end{aligned} \tag{6.4}$$

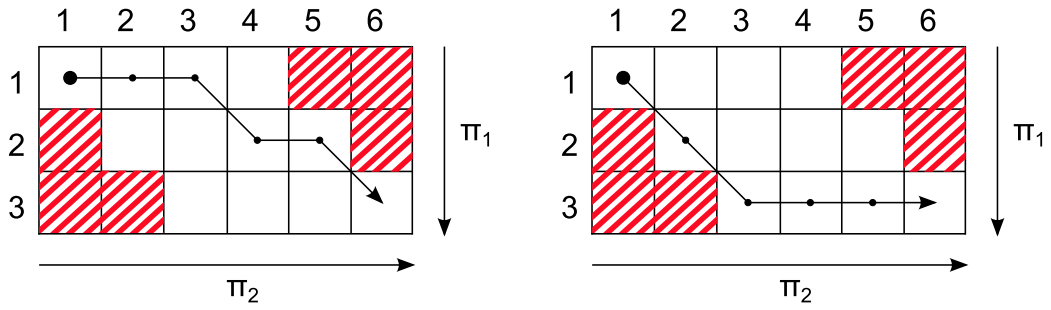


Figure 6.6: Two examples of one-sided alignments

This restriction of the domain will be put to use in Section 6.5 concerning the implementation using dynamic programming: it will be possible to design an algorithm with complexity $Ol \times m - l$ instead of $Ol \times m$.

6.2.4 Examples of Kernels Defined with Alignments

6.2.4.1 Optimal Assignment Kernel

Based on the popular dynamic time warping technique [Sakoe and Chiba, 1978], the optimal assignment kernel considers only the "best" alignment between two sequences. When dealing with continuous values such that $\mathcal{X} = \mathbb{R}^d$ the best alignment is the one that minimizes Euclidian distances:

$$k_{\mathrm{DTW}}\boldsymbol{x}, \boldsymbol{x}' = \exp - \min_{\pi \in \mathcal{A}\boldsymbol{x}, \boldsymbol{x}'} \frac{1}{\bigcup^{\pi}\bigcup} \bigcup_{\mathcal{P}\, i=1}^{\pi} \prod x_{\pi_1 i} - x'_{\pi_2 i} \prod^{2}$$

Which when dealing with a Gaussian ground kernel k is equivalent to the following equation:

$$k_{\mathrm{DTW}}\boldsymbol{x}, \boldsymbol{x}' = \max_{\pi \in \mathcal{A}\boldsymbol{x}, \boldsymbol{x}'} \bigcup_{\mathcal{L}\, i=1}^{\pi} k x_{\pi_1 i}, x'_{\pi_2 i}$$

Although widely used in the literature, Vert [2008] demonstrated that this kernel is in fact not positive definite and thus cannot be used as is in kernel methods.

6.2.4.2 Global Alignment Kernel

To circumvent this issue, Cuturi et al. [2007] have proposed an alternative kernel, named the global alignment kernel. Contrary to the optimal assignment kernel, this one does not only consider the best alignment but rather sums the scores associated to all possible global alignment. As a consequence this kernel may prove more robust to quantify the similarity between two sequences, and they demonstrated that it was indeed positive definite under mild conditions.

$$k_{\mathrm{GAS}}\boldsymbol{x}, \boldsymbol{x}' = \sum_{\pi \in \mathcal{A}\boldsymbol{x},\boldsymbol{x}'} \prod_{i=1}^{\pi} \mathcal{L} kx_{\pi_1 i}, x'_{\pi_2 i} \tag{6.5}$$

6.3 The One-Sided Mean Alignment Kernel

The fact that we use only one-sided alignments for the definition of this kernel will let us introduce another formalism that will not only simplify the notations but also the demonstration of the main theorem of this chapter. The formalism we introduce is that of *dilatation operators*.

A dilatation operator is a function that maps a sequence to a longer sequence by repeating one or more of its elements while still keeping the order. We denote by $\xi_{l\to m}$ the set of dilatation operators that map sequences of length l to sequences of length m. The dilatation operators will be properly defined in later sections, but in the meantime one only has to know that there is a one-to-one correspondence between the set of one-sided alignments $\mathcal{A}^{-}\boldsymbol{x}, \boldsymbol{x}'$ where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are two sequences of respective lengths $l \leq m$ and the set of dilatation operators $\xi_{l\to m}$. Note that consequently the cardinal of this set verifies $\prod \xi_{l\to m} \prod = \binom{m-1}{l-1}$.

Before delving into the technical details we provide in Table 6.1 a set of notations that may be helpful for reading the sequel.

6.3.1 Practical Case: Real Values with Gaussian Kernel

We start by giving an example of the one-sided mean kernel in the case where elements of sequences are real values: $\mathcal{X} = \mathbb{R}$, and where the ground kernel k is the usual one-dimensional Gaussian kernel. This is useful to get a sense of how

$\boldsymbol{x}$	One sequence
$\boldsymbol{x}'$	Another sequence
$\boldsymbol{x}_l$	Shorter sequence between $\boldsymbol{x}$ and $\boldsymbol{x}'$
$\boldsymbol{x}_m$	Longer sequence between $\boldsymbol{x}$ and $\boldsymbol{x}'$
l	Length of the shorter sequence $\boldsymbol{x}_l$
m	Length of the longer sequence $\boldsymbol{x}_m$
x_l	l^{th} element of the sequence $\boldsymbol{x}$
x_m	m^{th} element of the sequence $\boldsymbol{x}$
ϵ_i^l	Operator that repeats the i^{th} element of sequences of length l and leaves unchanged sequences of length other than l
$\xi_{1\to m}$	Set of dilatation operators *without repetition* from sequences of length l to sequences of length m, such that $\bigcup \xi_{l\to m} \bigcup = \binom{m-1}{l-1}$
$\xi'_{1\to m}$	Set of dilatation operators *with repetition* from sequences of length l to sequences of length m, such that $\bigcup \xi'_{l\to m} \bigcup = \frac{m-1!}{l-1!}$
$\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$	N sequences that constitute the dataset
n	Length of the longest sequence in $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$.

Table 6.1: Notation table.

this kernel is represented in most practical cases, before we delve into the more abstract setting of infinitely divisible kernels. As described in Table 6.1, let $\boldsymbol{x}$ and $\boldsymbol{x}'$ two elements of $\mathcal{X}^*$. Furthermore we refer to the shorter and longer elements of $\boldsymbol{x}, \boldsymbol{x}'$ as $\boldsymbol{x}_l$ and $\boldsymbol{x}_m$ respectively, with $l \leq m$ denoting the respective lengths of the sequences. Note that $\boldsymbol{x}_l$ and $\boldsymbol{x}_m$ are not to be mistaken for x_l and x_m which refer respectively to the l^{th} and m^{th} elements of $\boldsymbol{x}$. In the real case the one-sided kernel k^* is defined as:

$$k^*(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{1}{\bigcup \xi_{l\to m} \bigcup} \sum_{\epsilon \in \xi_{l\to m}} \frac{1}{m} \left\| \epsilon \boldsymbol{x}_l - \boldsymbol{x}_m \right\|^2\right). \tag{6.6}$$

Note that as the shorter sequence plays a special role, this equation is not symmetric w.r.t. $\boldsymbol{x}_l$ and $\boldsymbol{x}_m$; however it is indeed symmetric w.r.t. $\boldsymbol{x}$ and $\boldsymbol{x}'$. Additionally, note that when comparing two sequences of the same length m, this

kernel reduces to the usual vector Gaussian kernel:

$$k^*\boldsymbol{x}, \boldsymbol{x}' = \exp -\frac{1}{m}\prod \boldsymbol{x} - \boldsymbol{x}' \prod^2 . \tag{6.7}$$

This example illustrates interesting an interesting difference with the global alignment kernel: as it is defined using means of distances instead of sums, the value of the kernel is bounded even when the length of the sequences compared increases. This has important consequences in terms of consistency when studying sub-sampling of continuous time series.

6.3.2 Abstract Case: Infinitely Divisible Kernels

Definition 9. *Let K be a positive definite kernel on $\mathcal{X} \times \mathcal{X}$. The kernel K is called* infinitely divisible *if for each positive integer n there exists a positive definite kernel K_n such that $K = K_n^n$.*

Next, for any kernel k on $\mathcal{X} \times \mathcal{X}$, and any integer $m \geq 1$ we denote by $k_m : \mathcal{X}^m \times \mathcal{X}^m \to \mathbb{R}$ the product kernel defined as $k_m\boldsymbol{x}, \boldsymbol{x}' = kx_1, x_1' \cdot \ldots \cdot kx_m, x_m'$. We can now state a more general definition of the one-sided mean kernel.

Definition 10. *Let k be a kernel on $\mathcal{X} \times \mathcal{X}$. The one-sided mean kernel k^* is a kernel on $\mathcal{X}^* \times \mathcal{X}^*$ defined as the* geometric mean *of all one-sided alignment scores:*

$$k^*\boldsymbol{x}, \boldsymbol{x}' = \left(\underset{\pi \in \mathcal{A}^-\boldsymbol{x}, \boldsymbol{x}'}{\mathcal{L}} \overset{\pi}{\bigcup}_{i=1} \underset{}{\mathcal{L}} kx_{\pi_1 i}, x'_{\pi_2 i} \overline{\bigcup^{\pi}} \right] \overline{\bigcup^{\frac{1}{\mathcal{A}^-\boldsymbol{x}, \boldsymbol{x}'}}}$$

Using dilatation operators the one-sided mean kernel can be defined in the following way:

$$k^*\boldsymbol{x}, \boldsymbol{x}' = \left(\underset{\epsilon \in \xi_{l \to m}}{\mathcal{L}} k_m \epsilon \boldsymbol{x}_l, \boldsymbol{x}_m{}^{\frac{1}{m}} \right] \overline{\bigcup^{\xi_{l \to m}}} \tag{6.8}$$

As our main contribution we state the following theorem:

Theorem 9. *The one-sided mean kernel k^* verifies the following properties:*

1. *If k is positive definite and infinitely divisible, then k^* is positive definite,*

2. *When comparing two sequences* $\boldsymbol{x}$ *and* $\boldsymbol{x}'$ of the same length m, k^* *reduces to the product kernel:* $k^*\boldsymbol{x}, \boldsymbol{x}' = k_m\boldsymbol{x}, \boldsymbol{x}'^{\frac{1}{m}}$.

3. *If* k *is a radial basis kernel, then* k^* *is a radial basis kernel.*

The first property guarantees that the one-sided mean kernel can be used in the framework of kernel methods [Schölkopf and Smola, 2002], which as briefly explained in the introduction of this thesis, encompasses such diverse tasks as classification, regression, clustering etc. The second property indicates that this kernel on sequences is consistent with a kernel that would be built for dealing with only fixed-length data according to the principles exposed in Section 5.5 for example. This is especially remarkable in the case of real values with a Gaussian kernel where the one-sided mean kernel reduces to a Gaussian kernel on vectors when dealing with fixed-length data as in Equation 6.7. The third property leads to a better interpretation of the entries in the Gram matrix as explained in Chapter 4, additionally many kernel methods are supposed to work with a radial basis kernel.

Of course, the Gaussian kernel $kx, x' = \exp{-x - x'^2}$ where $x, x' \in \mathbb{R}$ is itself infinitely divisible [Berg et al., 1984], and it suffices to express the product as the exponentiation of distances to be lead to Equation 6.6.

6.3.2.1 Expression Using Conditionally Negative Definite Kernels

Let us consider the case where the kernel k on $\mathcal{X}$ is infinitely divisible. According to Theorem 6 of Section 4.3, this kernel can be expressed as the exponentiation of a *conditionally negative definite* kernel, that we denote by N:

$$\forall x, x' \in \mathcal{X}, \quad kx, x' = \exp{-Nx, x'}$$

Much in the same way that we defined k_m from k, we define N_m the kernel on $\mathcal{X}^m$ such that $N_m\boldsymbol{x}, \boldsymbol{x}' = Nx_1, x'_1 + \ldots + Nx_m, x'_m$. As the sum of conditionally negative definite kernels, N_m is itself conditionally negative definite. It is now possible to express the one-sided mean kernel using sums of conditionally negative definite

kernels instead of products of infinitely divisible kernels:

$$k^*\boldsymbol{x}, \boldsymbol{x}' = \exp -\frac{1}{\bigcup^{\xi_{l\to m}}\bigcup} \underset{\epsilon\in\xi_{l\to m}}{\mathcal{P}} \frac{1}{m} N_m \epsilon \boldsymbol{x}_l, \boldsymbol{x}_m \tag{6.9}$$

One can see that Equation 6.9 is very similar to Equation 6.6. Upon seeing Equation 6.8, the practitioner may be worried that a kernel defined with as many products may suffer from issues related to diagonal dominance, as it suffices that one term in the product be close to 0 in order for the whole kernel to be close to 0. In reality as the kernel can be expressed as the exponentiation of a sum, this translates to the majority of terms in the product being close to 1, provided the normalization of distances (or in this case conditionally negative definite kernels) is properly done.

6.4 Demonstration of the Main Theorem

6.4.1 Strategy

In order to prove that the one-sided mean kernel is positive definite we will prove that its evaluation over any finite dataset $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$ of any size N yields a positive definite matrix. We denote by n the length of longest sequence in the dataset. Informally, using the theory of infinitely divisible kernels we will "divide" the values of kernel evaluations $k^*\boldsymbol{x}^i, \boldsymbol{x}^j$ into sufficiently small parts that will be rearranged to expose the fact that the Gram matrix can be expressed as a Schür product[1] of many positive definite matrices. Indeed, one can see that the kernel is already defined as a product of other kernels, however this product is indexed by a set $\xi_{l\to m}$ which depends on the particular pair of samples $\boldsymbol{x}_l, \boldsymbol{x}_m$ being considered. Thus our task shall be to rewrite this product such that it is indexed by $\xi'_{1\to n}$, a set independent of the pair of samples considered.

[1] Recall that the Schür product is the entry-wise product.

$$\begin{array}{rcl} & \epsilon_2^4 & \\ \texttt{AT} & \longmapsto & \texttt{AT} \\ \texttt{ATGC} & \longmapsto & \texttt{AT}\underline{\texttt{T}}\texttt{GC} \\ \texttt{CATC} & \longmapsto & \texttt{CA}\underline{\texttt{A}}\texttt{TC} \\ \texttt{TTGAC} & \longmapsto & \texttt{TTGAC} \end{array}$$

Figure 6.7: Example of applications of ϵ_2^4 to sequences of different lengths.

6.4.2 Tools

6.4.2.1 Formal Definition of Dilatation Operators

We shall define the set of dilatation operators in a recursive manner. First for any positive integer l indicating the length of a sequence, we denote by $\epsilon_i^{0,l}$ the operator that dilates a sequence of length l by repeating once its i^{th} element:

$$\epsilon_i^{0,l} : \begin{array}{ll} \mathcal{X}^l & \to \mathcal{X}^{l+1} \\ a_1 a_2 \dots a_l & \mapsto a_1 a_2 \dots a_i a_i \dots a_l \end{array}$$

For the sake of our demonstration we will have to extend slightly this definition by enlarging the support of $\epsilon_i^{0,l}$ to all of $\mathcal{X}^*$:

$$\epsilon_i^l \boldsymbol{x} = \begin{array}{ll} \epsilon_i^{0,l} \boldsymbol{x} & \text{if } \bigcup \boldsymbol{x} \bigcup = l \\ \boldsymbol{x} & \text{if } \bigcup \boldsymbol{x} \bigcup \neq l \end{array} \tag{6.10}$$

As an illustration, Figure 6.7 presents the application of an extended dilatation operator to examples of sequences. Next we denote by $\xi'_{l \to l+1}$ the set of all dilatation operators that map a sequence of length l to a sequence of length $l+1$. Thus $\xi'_{l \to l+1} = \epsilon_i^l,\ i \in (\!(1, l)\!]$.

Let $l < m$ two integers. In order to define the set of dilatation operators that map a sequence of length l to a sequence of length m, we state that one such operator first dilates a sequence of length l to a sequence of length $l+1$, then to a sequence $l+2$, etc. until a sequence of length m is reached. Recursively it can be defined as:

$$\xi'_{l \to m} = \epsilon' \circ \epsilon, \quad \epsilon \in \xi'_{l \to m-1} \wedge \epsilon' \in \xi'_{m-1 \to m} \tag{6.11}$$

Finally for consistency, we have $\xi'_{l\to l} = \mathrm{Id}_{\mathcal{X}^*}$.

Combinatorial considerations For the sake of the demonstration we consider for example $\epsilon_1^3 \circ \epsilon_2^2$ and $\epsilon_3^3 \circ \epsilon_1^2$ to be two *different* elements of $\xi'_{2\to 4}$ although they are identical in the mathematical sense since they both represent the same input-output relation. Thus in our sense, the cardinal of $\xi'_{l\to m}$ is $\bigcup \xi'_{l\to m} \bigcup = m - 1m - 2 \ldots l = \frac{m-1!}{l-1!}$. We denote by $\xi_{l\to m}$ the set of dilatation operators *without repetition* such that $\bigcup \xi_{l\to m} \bigcup = {m-1 \atop l-1}$ and such that each element in $\xi_{l\to m}$ is repeated exactly $m - l!$ times in $\xi'_{l\to m}$.

Consequently, any mean indexed by $\xi_{l\to m}$ is equal to the mean indexed by $\xi'_{l\to m}$, so in the case of the one-sided mean kernel:

$$\left(\underset{\epsilon\in\xi_{l\to m}}{\mathcal{L}} k_m \epsilon \boldsymbol{x}_l, \boldsymbol{x}_m \right]^{\frac{1}{m}}_{\overline{\bigcup^{\xi_{l\to m}} \bigcup}} = \left(\underset{\epsilon\in\xi'_{l\to m}}{\mathcal{L}} k_m \epsilon \boldsymbol{x}_l, \boldsymbol{x}_m \right]^{\frac{1}{m}}_{\overline{\bigcup^{\xi'_{l\to m}} \bigcup}} \tag{6.12}$$

6.4.3 Developments

We first start by replacing $\boldsymbol{x}_m$ by $\epsilon \boldsymbol{x}_m$ which does not change the values by virtue of Equation 6.10; and then by replacing $\xi_{l\to m}$ by $\xi'_{l\to m}$ which does not change the value of the geometric mean as expressed in Equation 6.12, so that we obtain

$$k^* \boldsymbol{x}, \boldsymbol{x}' = \underset{\epsilon\in\xi'_{l\to m}}{\mathcal{L}} k_m \epsilon \boldsymbol{x}_l, \epsilon \boldsymbol{x}_m \,{}^{\frac{1}{m\cdot \overline{\bigcup^{\xi'_{l\to m}} \bigcup}}}.$$

Next, we change the left index of $\xi'_{l\to m}$ from l to 1, and because both elements $\boldsymbol{x}_l$ and $\boldsymbol{x}_m$ have length strictly superior to $l - 1$ this results according to Equation 6.10 to elements of $\xi'_{l\to m}$ being repeated exactly $l - 1!$ times, which we account for by changing the exponent and which leads to

$$k^* \boldsymbol{x}, \boldsymbol{x}' = \underset{\epsilon\in\xi'_{1\to m}}{\mathcal{L}} k_m \epsilon \boldsymbol{x}_l, \epsilon \boldsymbol{x}_m \,{}^{\frac{1}{m\cdot l-1!\cdot \overline{\bigcup^{\xi'_{l\to m}} \bigcup}}}.$$

Finally, as $\bigcup \xi'_{l\to m} \bigcup = \frac{m-1!}{l-1!}$, we have:

$$k^* \boldsymbol{x}, \boldsymbol{x}' = \underset{\epsilon\in\xi'_{1\to m}}{\mathcal{L}} k_m \epsilon \boldsymbol{x}_l, \epsilon \boldsymbol{x}_m \,{}^{\frac{1}{m!}} \tag{6.13}$$

In the next step we shall prove a lemma.

Lemma 1. *For any integer $m \geq 1$; denote by $\mathcal{K}_m$ the kernel on $\cup_{i=1}^{m} \mathcal{X}^i$ defined as:*

$$\mathcal{K}_m \boldsymbol{x}, \boldsymbol{x}' = \underset{\epsilon \in \xi'_{1\to m}}{\mathcal{L}} k_m \epsilon \boldsymbol{x}, \epsilon \boldsymbol{x}'^{\frac{1}{m!}} \tag{6.14}$$

Then for any two sequences $\boldsymbol{x}_l$ and $\boldsymbol{x}_m$ of respective lengths $l \leq m$, we have that:

$$\forall p \geq m, \quad \mathcal{K}_p \boldsymbol{x}_l, \boldsymbol{x}_m = \mathcal{K}_m \boldsymbol{x}_l, \boldsymbol{x}_m$$

Proof. We shall prove this identity by induction. Let $p \geq m > l$. The goal is to prove that $\mathcal{K}_{p+1} \boldsymbol{x}_l, \boldsymbol{x}_m = \mathcal{K}_p \boldsymbol{x}_l, \boldsymbol{x}_m$. Using Equation 6.11 we can decompose any element of $\xi'_{1\to p+1}$ such that:

$$\underset{\epsilon \in \xi'_{1\to p+1}}{\mathcal{L}} k_{p+1} \epsilon \boldsymbol{x}_l, \epsilon \boldsymbol{x}_m = \underset{\epsilon' \in \xi'_{1\to p}}{\mathcal{L}} \underset{\epsilon'' \in \xi'_{p\to p+1}}{\mathcal{L}} k_{p+1} \epsilon'' \epsilon' \boldsymbol{x}_l, \epsilon'' \epsilon' \boldsymbol{x}_m \tag{6.15}$$

Then by breaking down the definition of k_{p+1} and rearranging the terms in the product one can easily see that for any $\boldsymbol{x}_p$, $\boldsymbol{x}'_p$ in $\mathcal{X}^p$:

$$\underset{\epsilon'' \in \xi'_{p\to p+1}}{\mathcal{L}} k_{p+1} \epsilon'' \boldsymbol{x}_p, \epsilon'' \boldsymbol{x}'_p = k_p \boldsymbol{x}_p, \boldsymbol{x}'_p{}^{p+1} \tag{6.16}$$

By applying Equation 6.16 to $\boldsymbol{x}_p = \epsilon' \boldsymbol{x}_l$ and $\boldsymbol{x}'_p = \epsilon' \boldsymbol{x}_m$ and combining with Equation 6.15 we obtain:

$$\underset{\epsilon \in \xi'_{1\to p+1}}{\mathcal{L}} k_{p+1} \epsilon \boldsymbol{x}_l, \epsilon \boldsymbol{x}_m = \underset{\epsilon \in \xi'_{1\to p}}{\mathcal{L}} k_p \epsilon \boldsymbol{x}_l, \epsilon \boldsymbol{x}_m{}^{p+1}$$

Finally, elevating to the power $\frac{1}{p+1!}$ and using Equation 6.14 leads to:

$$\mathcal{K}_{p+1} \boldsymbol{x}_l, \boldsymbol{x}_m = \mathcal{K}_p \boldsymbol{x}_l, \boldsymbol{x}_m$$

□

According to Equation 6.13 we have that $k^* \boldsymbol{x}, \boldsymbol{x}' = \mathcal{K}_m \boldsymbol{x}_l, \boldsymbol{x}_m$. Recall that n is the length of the longest sequence in the dataset. Applying Lemma 1 for $p = n$

leads to:

$$k^*\boldsymbol{x}, \boldsymbol{x}' = \mathcal{K}_n\boldsymbol{x}_l, \boldsymbol{x}_m$$

As k_n is symmetric we are finally lead to another expression for the one-sided mean kernel:

$$\begin{aligned} k^*\boldsymbol{x}, \boldsymbol{x}' &= \mathcal{K}_n\boldsymbol{x}, \boldsymbol{x}' \\ k^*\boldsymbol{x}, \boldsymbol{x}' &= \underset{\epsilon\in\xi'_{1\to n}}{\mathcal{L}} k_n\epsilon\boldsymbol{x}, \epsilon\boldsymbol{x}'^{\frac{1}{n!}} \end{aligned} \tag{6.17}$$

Recall that n is the length of the longest sequence in the dataset, thus Equation 6.17 is valid for any pair of samples $\boldsymbol{x}, \boldsymbol{x}'$ in the dataset.

6.4.4 Conclusion of the Demonstration

6.4.4.1 First Property: Positive Definiteness

For any $\epsilon \in \xi'_{1\to n}$, denote by $K_{\epsilon,N}$ the $N \times N$ Gram matrix obtained by evaluation of the kernel $k_n^{\frac{1}{n!}}$ over the samples $\epsilon\boldsymbol{x}^1, \ldots, \epsilon\boldsymbol{x}^N$. The Schür product theorem [Horn and Johnson, 2012] guarantees that the product of two positive definite and infinitely divisible kernels is a positive definite and infinitely divisible kernel. As k is a positive definite infinitely divisible kernel, we thus have that $k_n^{\frac{1}{n!}}$ is a positive definite kernel. Thus for any $\epsilon \in \xi'_{1\to n}$, $K_{\epsilon,N}$ is a positive definite matrix.

Now let us denote by K_N the Schür product of the $n-1!$ aforementioned matrices:

$$K_N = \underset{\epsilon\in\xi'_{1\to n}}{\curlywedge} K_{\epsilon,N}$$

Moreover, according to Equation 6.17, we have that

$$K_N = k^*\boldsymbol{x}^i, \boldsymbol{x}^j{}_{i,j}\,.$$

One final application of the Schür product theorem guarantees that K_N is positive definite, which concludes the demonstration of the first property.

6.4.4.2 Second Property: Two Sequences of Same Length

The second property follows from the fact that there is only one one-sided alignment between two sequences of the same length m, which is the trivial alignment of length m: $\forall i \in (\!(1, m]\!]$, $\pi_1 i = \pi_2 i = i$. Equivalently using the formalism of dilatation operators, $\xi'_{m \to m} = \mathrm{Id}_{\mathcal{X}^*}$; which leads to:

$$k^* \boldsymbol{x}, \boldsymbol{x}' = k_m \boldsymbol{x}, \boldsymbol{x}'^{\frac{1}{m}}$$

6.4.4.3 Third Property: Radial Basis Kernel

Firstly, the one-sided mean kernel obviously has only positive values. Secondly, if we suppose that k is a radial basis kernel, then it has values in range $(0, 1]$, consequently for any m, the kernel k_m also has values in range $(0, 1]$. As the geometric mean of values in the range $(0, 1]$, k^* also has values in range $(0, 1]$.

Suppose $\boldsymbol{x} \neq \boldsymbol{x}'$, then at least one of the terms in the definition of k^* will be different from 1 and thus $k^* \boldsymbol{x}, \boldsymbol{x}' < 1$.

On the contrary, if we suppose $\boldsymbol{x} = \boldsymbol{x}'$, then according to the second property, k^* reduces to the generalized product kernel, whose every term is equal to 1, which leads to $k^* \boldsymbol{x}, \boldsymbol{x}' = 1$.

6.5 Implementation Using Dynamic Programming

6.5.1 Introduction

In this section will be presented a practical way to compute the mean alignment kernel as defined in Equation 6.8. As explained in Section 6.2.2, between two sequences $\boldsymbol{x}_l$ and $\boldsymbol{x}_m$ of respective lengths $l \leq m$ there is exactly $\binom{m-1}{l-1}$ one-sided alignments. Consequently, it is absolutely impractical to compute naively the alignment scores for every alignment, except for very small sequence lengths. Fortunately, as is the case with most sequence comparison methods there is a way to implement the computation of the one-sided mean kernel using dynamic

programming techniques, which results in polynomial complexity $Ol \times m - l$ with respect to both time and space.

In order to implement the one-sided mean kernel we shall refer to its expression using conditionally negative definite kernels as is described in Equation 6.9, we shall henceforth refer to this expression as the "additive" form of the one-sided mean kernel. Most often the conditionally negative definite kernels represent distances that are computed before the kernel evaluation, but otherwise as is explained in Section 4.3 one can retrieve the conditionally negative definite kernel corresponding to an infinitely divisible kernel by taking the opposite of its logarithm: $Nx, y = -\log kx, y$. The two reasons for using the additive form are that firstly additions are much faster to compute than multiplications, and secondly the additive form requires only one exponentiation (which is very slow to compute).

6.5.2 Notations

As in previous sections, we denote by $\boldsymbol{x}$ and $\boldsymbol{x}'$ two finite sequences of $\mathcal{X}^*$; $\boldsymbol{x}_l$ and $\boldsymbol{x}_m$ refer respectively to the shorter and longer sequence between $\boldsymbol{x}$ and $\boldsymbol{x}'$, with $l \leq m$ their respective lengths. We shall focus on the argument of the exponential function, that we call the *one-sided mean distance* and that we denote by $D\boldsymbol{x}_l, \boldsymbol{x}_m$:

$$D\boldsymbol{x}_l, \boldsymbol{x}_m = \frac{1}{\bigcup^{\xi_{l\to m}}\bigcup} \mathop{\mathcal{P}}_{\epsilon\in\xi_{l\to m}} \frac{1}{m} N_m \epsilon \boldsymbol{x}_l,\ \boldsymbol{x}_m$$

The index $i \in \llparenthesis 1, l \rrparenthesis$ will refer to an element of the sequence $\boldsymbol{x}_l$ while the index $j \in \llparenthesis 1, m \rrparenthesis$ will refer to an element of the sequence $\boldsymbol{x}_m$. Denote by Ni, j the normalized evaluation of the conditionally negative definite kernel N on the i^{th} element of $\boldsymbol{x}_l$ and the j^{th} element of $\boldsymbol{x}_m$:

$$Ni, j = \frac{1}{m} N x_l^i,\ x_m^j$$

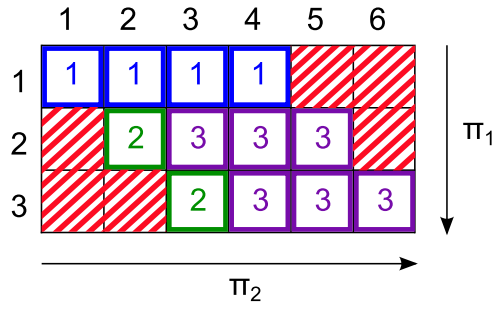


Figure 6.8: Division of the alignment in three areas

Next denote by Di,j the one-sided mean distance between the first i elements of $\boldsymbol{x}_l$ and the first j elements of $\boldsymbol{x}_m$:

$$Di,j = \frac{1}{\bigcup^{\xi_{i\to j}}\bigcup} \underset{\epsilon\in\xi_{i\to j}}{\mathcal{P}} \frac{1}{m} N_j \epsilon x_l^1 \dots x_l^i,\ x_m^1 \dots x_m^j$$

Note that we have to keep the normalization factor to $1_{\Uparrow} m$, which breaks most properties enunciated in previous sections; but this is not an issue as the Di,j for $i < l$ and $j < m$ are merely intermediate results. We are only interested in Dl,m as:

$$Dl,m = D\boldsymbol{x}_l, \boldsymbol{x}_m$$

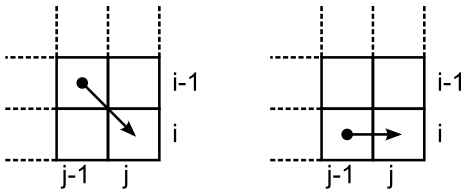
6.5.3 Recursive Formulas

The easiest way to understand the recursive relation is to visualize the alignments as paths on matrices as was done in Section 6.2. We shall divide the domain into three areas, as described in Figure 6.8, and derive a recursive formula on Di,j for each of these areas.

6.5.3.1 First Area

The first area is covered by pair of indexes $1,j$ with $1 \le j \le m-l+1$. In this area there is only one path that goes from $1,1$ to $1,j$, and thus we have that:

$$\begin{aligned} & D1,1 = N1,1, \\ \forall j \in \left(\!\!\left(2, m-l+1\right]\!\!\right], \quad & D1,j = N1,j + D1,j-1 \end{aligned} \tag{6.18}$$



(a) First way (b) Second way

Figure 6.9: Two possible ways to reach i, j

6.5.3.2 Second Area

Similarly, for any pair of indexes i, i, $2 \leq i \leq l$ that belongs to the second area, there is only one path that goes from $1, 1$ to i, i, and thus we have that:

$$\forall i \in [\![2, l]\!], \quad Di, i = Ni, i + Di - 1, i - 1 \tag{6.19}$$

6.5.3.3 Third Area

Finally let us consider i and j such that $2 \leq i \leq j - 1 \leq m - l + i - 1$. When considering one-sided alignments, in order to reach indexes i, j, the path has to cross either $i-1, j-1$ or $i, j-1$. This is illustrated in Figure 6.9. There is exactly $\binom{i-2}{j-2}$ paths that cross $i - 1, j - 1$, and exactly $\binom{i-1}{j-2}$ paths that cross $i, j - 1$; and in total there are $\binom{i-1}{j-1} = \binom{i-2}{j-2} + \binom{i-1}{j-2}$ paths that cross i, j. It then suffices to express Di, j as a partial mean to be lead to:

$$Di, j = \frac{\binom{i-2}{j-2} + \binom{i-1}{j-2}}{\binom{i-1}{j-1}} \cdot Ni, j + \frac{\binom{i-2}{j-2}}{\binom{i-1}{j-1}} \cdot Di - 1, j - 1 + \frac{\binom{i-1}{j-2}}{\binom{i-1}{j-1}} \cdot Di, j - 1$$

By simply using the expression of binomial coefficients with factorials this equation is simplified to:

$$Di, j = Ni, j + \frac{j - i}{j - 1} \cdot Di, j - 1 + \frac{i - 1}{j - 1} \cdot Di - 1, j - 1 \tag{6.20}$$

6.5.3.4 Combining the Three Areas

Note that in Equation 6.20, the factors $\frac{i-1}{j-1}$ and $\frac{j-i}{j-1}$ are either null or equal to 1 in areas 1 and 2 respectively. Thus one can simply extend slightly the domain

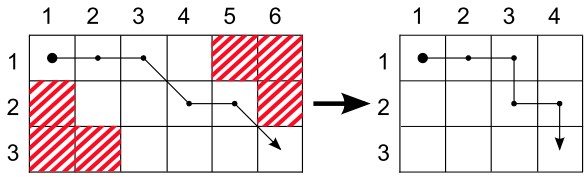


Figure 6.10: Domain transformation

of definition of D by stating for example that $D0, j = 0$ and $Di, 0 = 0$, and as a consequence Equation 6.20 is valid on areas 1 and 2 and replaces Equations 6.18 and Equations 6.19.

6.5.4 Optimizing for Memory Space

It is clear from the equations defining the domain 6.4 that the algorithm has time complexity $Ol \times m - l$. In addition, as the values of the matrix outside the domain are never used, it is not necessary to store them. Fortunately the remaining values can be conveniently stored in a full matrix, as is illustrated in Figure 6.10. This is achieved by a simple change of indexes:

$$\begin{aligned} i &\mapsto i' \\ j - i + 1 &\mapsto j' \end{aligned} \tag{6.21}$$

Consequently, computing the one-sided mean kernel evaluation of two sequences of lengths $l \leq m$ requires the storage of a matrix of size $l \times m - l + 1$. Denote N' the $l \times m - l + 1$ matrix containing pairwise evaluations of N in the new parametrization. We have that:

$$\forall i' \geq 1,\ j' \geq 1, \quad N'i', j' = Ni', j' + i' - 1$$

Furthermore denote D' the new parametrization of matrix D; then equations become:

$$D'1, 1 = N'1, 1,$$

And for any i', j' that verifies:

$$\begin{aligned} 1 \leq & i' \leq l \\ 1 \leq & j' \leq m - l + 1 \\ & i', j' \neq 1, 1 \end{aligned}$$

We have that:

$$D'i', j' = N'i', j' + \frac{j' - 1}{i' + j' - 2} \cdot D'i', j' - 1 + \frac{i' - 1}{i' + j' - 2} \cdot D'i' - 1, j'.$$

6.5.5 Algorithm

It is now possible to properly state the algorithm that computes the evaluation of the one-sided mean kernel. The complete algorithm is specified in Algorithm 1.

6.6 Consistency

In this section we review one of the most important feature of this kernel regarding the application to flight data monitoring: its consistency when dealing with data sampled from continuous processes. The reason for this consistent behavior is that the one-sided mean kernel is defined using means instead of sums like the global alignment kernel for example (see Equation 6.5). In this section we shall illustrate this fact using synthetic data.

6.6.1 Illustration with Toy Data

To illustrate this behavior we have compared two continuously defined functions f_0 and f_1 which are plotted in Figure 6.11(a). Note that f_0 and f_1 correspond to two continuous processes of different durations: f_0 is defined on the interval $(0, 1]$ whereas f_1 is defined on the interval $(0, 1.25]$. This illustrates a common case in flight data monitoring where the same phase of different flights will most often have different durations.

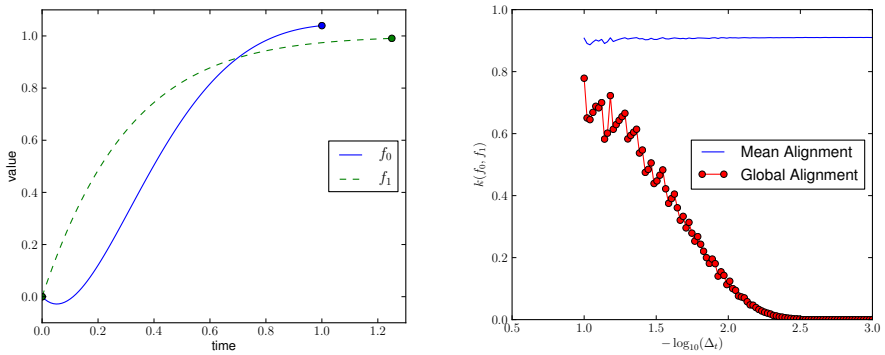
Algorithm 1 Compute the one-sided mean alignment between two sequences

Precondition:

1. $\boldsymbol{x}$ and $\boldsymbol{x}'$ are two sequences;
2. N is the conditionally negative definite kernel corresponding to k, in the Gaussian case, $Na, b = \prod a - b \prod^2$.

```
1  function OneSided(x, x')
2      l ← length(x)
3      m ← length(x')
4      if l > m then                ▷ Swap x and x' such that x is the shorter sequence
5          x, x' ← x', x
6          l, m ← m, l
7      end if

8      D ← matrix0..l, 0..m − l + 1                          ▷ Matrix initialization
9      for i ← 1 to l do
10         D(i, 0⌋ ← 0
11     end for
12     for j ← 1 to m − l + 1 do
13         D(0, j⌋ ← 0
14     end for
15     D(1, 1⌋ ← Nx_1, x'_1
16     for i ← 1 to l do
17         for j ← 1 to m − l + 1 do
18             if i, j ≠ 1, 1 then
19                 A ← i − 1⇑i + j − 2 × D(i − 1, j⌋
20                 B ← j − 1⇑i + j − 2 × D(i, j − 1⌋
21                 D(i, j⌋ ← Nx_i, x'_{i+j−1} + A + B
22             end if
23         end for
24     end for
25     return exp−D(l, m − l + 1⌋⇑m
26 end function
```



(a) Two continuous functions (b) Normalized kernel values

Figure 6.11: Comparison of the one-sided mean and global alignment kernels in the case of continuous time series sampling

The advantage of continuously defined functions is that they can be sampled to infinite precision, which serves nicely the purpose of our illustration; even though in practice one always deals with functions that are already sampled. We shall compare kernel values when the sampling period Δ_t converges to 0.

We have compared the one-sided mean kernel with the global alignment kernel, which is one of the few provably positive definite kernels that has been successfully used for continuous time series analysis [Cuturi and Doucet, 2011]. However, the global alignment kernel is defined with sums with respect to alignments; which means that as the sampling period decreases, the number of alignment increases to infinity, and so does the value of the kernel. Consequently, we have normalized the global alignment kernel so as to yield a radial basis kernel as in Definition 8 of Chapter 4: $\tilde{k}f_0, f_1 = kf_0, f_1_{\Uparrow} kf_0, f_0 \cdot kf_1, f_1$. Both kernels are defined from the same ground kernel: the Gaussian kernel on $\mathbb{R}$.

As one can see in Figure 6.11, as the sampling period Δ_t converges to 0, the one-sided mean kernel evaluation converges to a value close to 0.91 whereas the global alignment kernel converges to 0. To circumvent this problem of diagonal dominance, Cuturi et al. [2007] proposes to take the logarithm of the values of the Gram matrix, but this is known to break the positive definite property.

6.7 Conclusion

The kernel we propose is in the same family as the global alignment kernel; it can just as well handle time series whose elements are structured data because it is defined from a ground kernel k. With only mild requirements on the ground kernel k we have demonstrated that k^* is both positive definite and a radial basis kernel. Because it is defined using means instead of sums the one-sided mean kernel does not suffer from diagonal dominance issues, and thus can readily be used in practice; whereas it is common to take the logarithm of the values of the global alignment kernel which breaks its positive definite property. Of course there are many applications where the fact that only the shorter sequence can have repeated states is an issue, for example one would not want to use the one-sided kernel for applications such as protein sequence analysis. However, when dealing with for example the sampling of continuous processes, as is the case in the field of Flight Data Monitoring, it is our belief that one can obtain meaningful results.

Chapter 7

Results on the `Airline2` dataset

7.1 Introduction

In this chapter we apply the one-sided mean kernel described in Chapter 6 and the novelty detection method described in Chapter 2 to a dataset that was provided to us by partner airline `Airline2`.

As explained in the introduction of this thesis and in Chapter 3 we use the approach of "one flight is one sample". The difference with Chapter 3 is that we will now use a kernel on sequences, which hopefully will let us detect problems that are localized in time.

After a presentation of the dataset we explain why the one-sided mean kernel is suited to the domain of FDM and how we have leveraged some of its properties. We end this chapter by showing promising results on the `Airline2` dataset.

7.2 Presentation of the dataset

Partner airline company `Airline2` provided us with 604 flights of aircraft A320 that flew from [redacted] to [redacted] in 2013. [redacted] is a city on the island of [redacted], part of the Archipelago of the [redacted] in the North Atlantic Ocean and located about [redacted] km west of continental [redacted]. The island is of volcanic origin, and as one can see in Figure 7.1, its peculiar geography with a crater in the center creates difficult wind conditions, which is why [redacted] is considered a very tricky

Figure 7.1: Map of island

Code	Description	Type
LATPC	Latitude	angular
LONPC	Longitude	angular
ALTSTDC	Altitude	continuous
FLAPC	Flaps	ordered discrete
SLATRW	Slats	ordered discrete
IASC	Air speed	continuous
VRTG	Vertical acceleration	continuous
IVVR	Vertical speed	continuous
PITCH	Pitch	angular
AOAL	Angle of attack	angular
HEADMAG	Heading	angular
ROLL	Roll	angular

Table 7.1: List of parameters in this study

runway. The parameters we have considered in this study are listed in Table 7.1, along with descriptions and types.

7.3 Preprocessing of the dataset

7.3.1 Flight phase

Just like for the `Airline1` dataset of Chapter 3 we cut all the flights according to the flight phases. This time we have cut the flights from 10000 until the end of the landing phase, which is defined as the moment when the aircraft stops on the runway or exits the runway.

7.3.2 Sub-sampling

As we model flights as sequences of time-samples, the naive way to compute a Gram matrix would be to consider the data at their native frequency. However there are several problems to this approach, the first being that not all parameters are recorded at the same frequency, for example in this dataset the VRTG parameter is recorded at 8Hz whereas the LATPC is recorded at 1Hz. One could take the parameter at the highest frequency and repeat samples accordingly for other parameters, but this results in a very high number of samples for each flight, even if the flight phase we are considering lasts for about ten minutes.

Another problem is that each comparison of two flights using most kind of alignment technique results in complexity $Ol \times m$, and we have to carry such computation for any pair flights in the dataset, which grows quadratically with respect to the number of flights. Even when using the one-sided alignment kernel with its faster $Ol \times m - l$ complexity, the computation times are too long to be carried on a standard desktop computer as we would like for our experiments.

The third issue is that the information that is carried at such a frequency may not be very interesting for the field of FDM. Rather, we would like to have a more global view on the flight while still retaining the sequential aspect.

Consequently we have decided to sub-sample the data with a period Δ_t. The flight phase considered is divided into chunks of duration Δ_t, and for each parameter and each chunk we compute a centroid, as explained in Section 5.3. It is preferable to use a centroid rather than just the samples at times 0, Δ_t, $2 \cdot \Delta_t$, ... because this approach better captures the information and is less subject to noise.

Note that as illustrated in Section 6.6 using the one-sided mean kernel ensures

that the values of the Gram matrix will converge when $\Delta_t \to 0$. Consequently, when we have more processing power at our disposal we will be able to study the dataset at a finer scale, and be sure that the results will be more and more precise.

On the contrary, when $\Delta_t \to \infty$ the centroids are computed with the data from the whole flight phase, which yields "sequences" of length 1. Because of how the one-sided mean kernel is defined, and because we have chosen the product kernel approach as explained in Section 5.4 then this is in fact equivalent to the feature-based approach presented in Chapter 3.

Thus we see that we are able to carry a whole spectrum of studies in a single framework only by varying the parameter Δ_t.

In this particular study we chose a period of $\Delta_t = 5s$.

7.3.3 Normalization

Contrary to the feature-based case described in Chapter 3 the normalization of structured data and especially sequences can be quite involved, as the data are not vectors. Consequently the approach we have chosen to normalize pairwise distances between flights by their median value, which seems to work very well for both vector and structured data.

7.3.4 Settings for KECA

Unfortunately it is not obvious how to define a measure on spaces of sequences, and therefore we could not properly define a kernel which is simultaneously positive definite and that could be used as an equivalent of a Parzen window for density estimation as is required for KECA. However we think that carrying this procedure still yields sensible results, as the novelty detection algorithm described in Chapter 2 also has a geometric interpretation.

After computing the entropy-values we noticed that only the first principal dimension in the feature space had non-negligible entropy-value so we decided to retain only this dimension as the principal subspace in the feature space.

As explained in Section 2.4 we use the reconstruction error to this principal subspace as a measure for novelty. In the same manner as in Chapter 3 we fit a

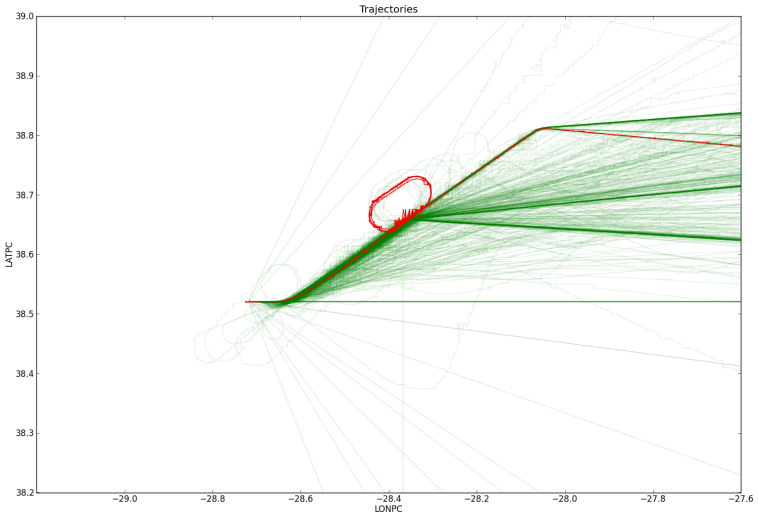


Figure 7.2: Trajectory of Flight 1

Gamma distribution to the reconstruction errors and present flights which have a pvalue smaller than 0.05 according to this distribution.

7.4 Example of atypical flights

Among the 604 flights, 9 were deemed as very atypical with a pvalue $< 10^{-2}$ and 17 others as atypical with a pvalue < 0.05. We present here some of the most interesting flights detected, and try to give an explanation whenever possible. In all the graphical representations the red plot represents the atypical flight, whereas the other flights of the dataset are represented as transparent green plots for easy comparison. Furthermore, the touchdown (moment when the aircraft's landing gear touches the runway) is represented as the 0 time on the horizontal axis as well as a blue vertical line.

Flight 1 The first flight we present was detected as outlier with pvalue equal to 0.0025. It is an example of a holding pattern: the pilot is certainly waiting for instructions from the Air Traffic Control (ATC), so we see in Figure 7.2 that the pilot maneuvers to stay in a defined airspace. This is an example of flight which is atypical but not interesting from a safety perspective. Nevertheless for

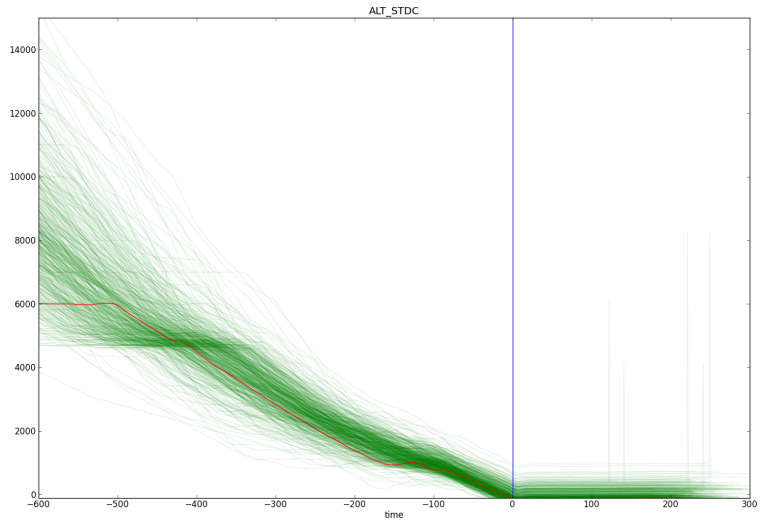


Figure 7.3: Altitude of Flight 1

information we have also provided other graphical representations.

Flight 2 Flight 2 is another interesting flight which was detected as outlier with pvalue equal to 0.0032. This flight is atypical in many respect, first it does not follow the trajectory of other flights, and then near the runway it seems to be in a very long and unusual holding phase, as one can see in Figure 7.6. This is certainly due to cloud ceiling, the pilot may be waiting for clouds to go up in order to have visual conditions to land. As there is no ILS (Instrument Landing System) in this airport the landing on both runways is always visual. Note that there is a slight glitch in the values at the end of the phase, which explains the two straight lines. This abnormality can also be seen in the other parameters we studied.

Flight 3 Flight 3 was detected as outlier with pvalue equal to 0.0089. Flight 3 is another example of go around: the pilot initiated the descent, touched the runway, but for some reason had to take off again and re-initiate a landing. Note that the algorithm implemented in the AGS may not detect all go around, for example if the transition to higher altitude is too smooth. The procedure to detect a go around in the AGS as defined by `Airline2` is defined as follows:

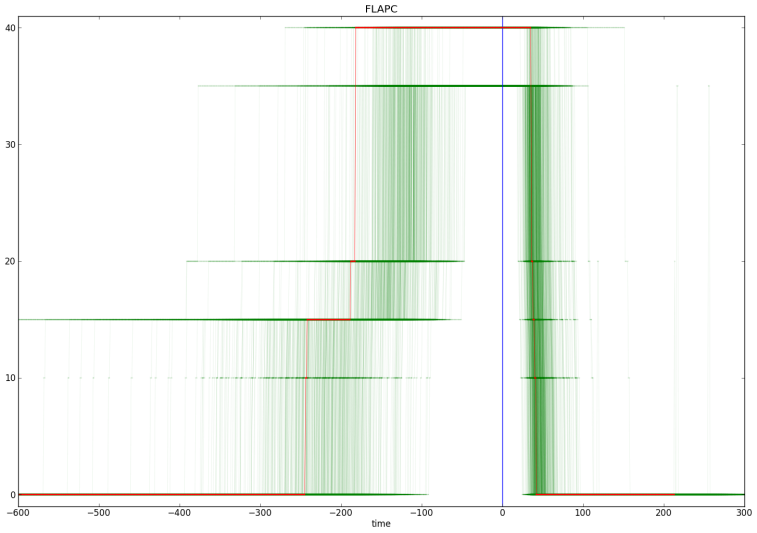


Figure 7.4: Flaps of Flight 1

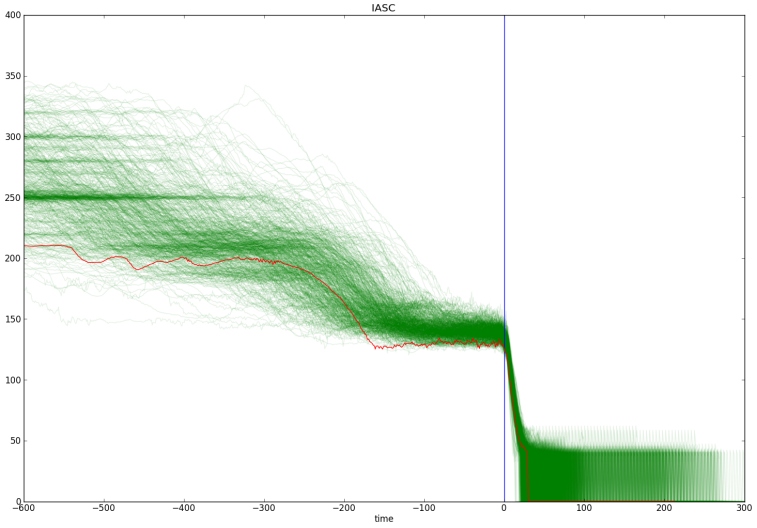


Figure 7.5: Air speed of Flight 1

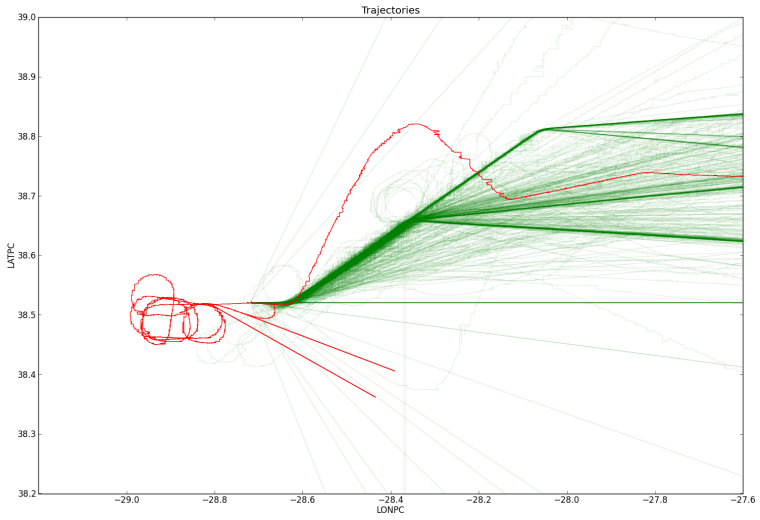


Figure 7.6: Trajectory of Flight 2

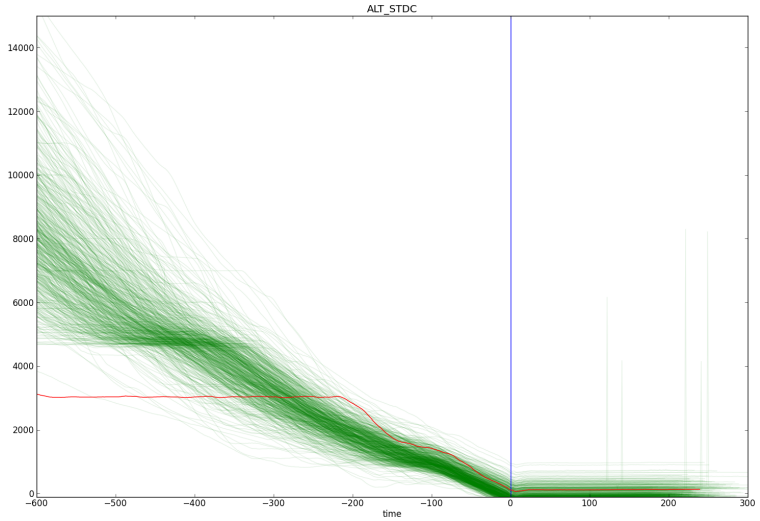


Figure 7.7: Altitude of Flight 2

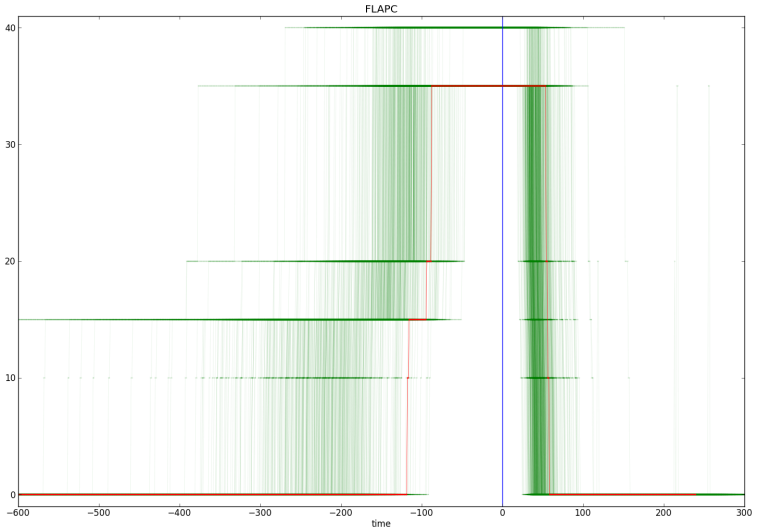


Figure 7.8: Flaps of Flight 2

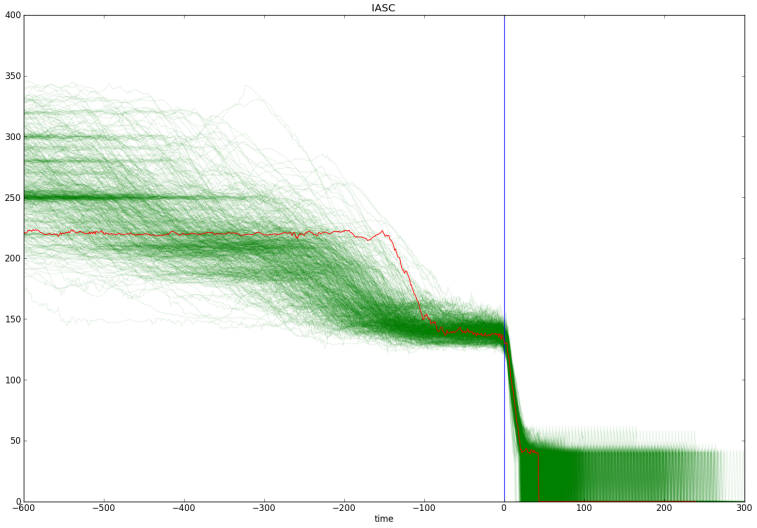


Figure 7.9: Air speed of Flight 2

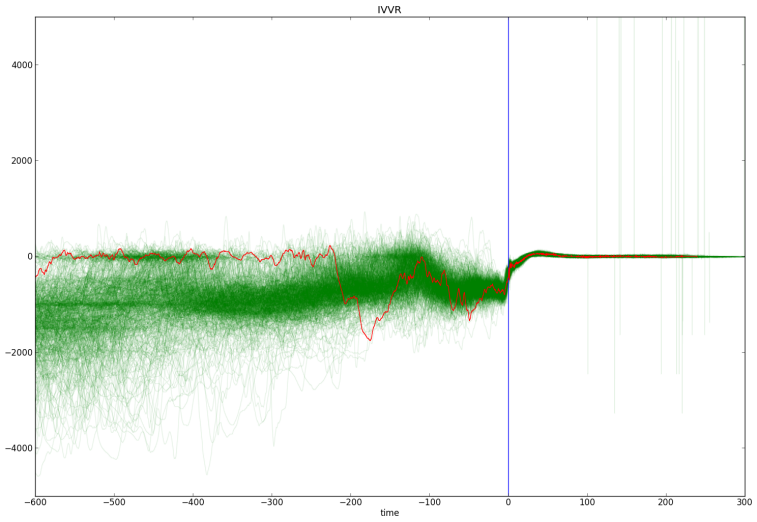


Figure 7.10: Vertical speed of Flight 2

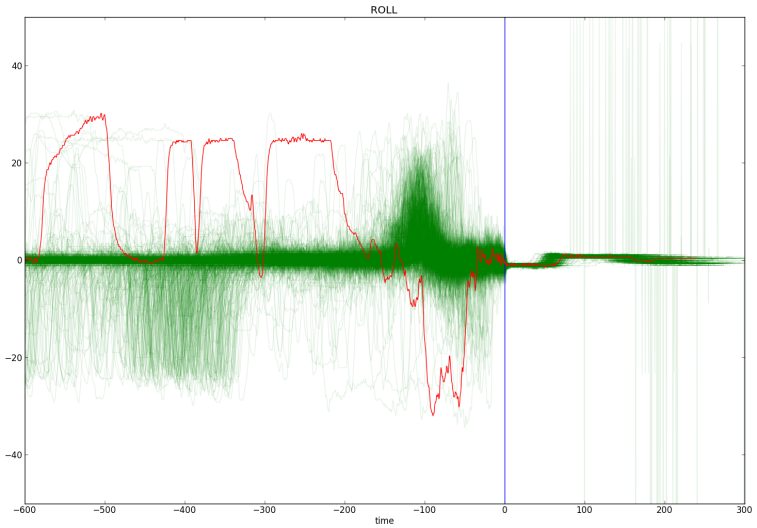


Figure 7.11: Roll of Flight 2

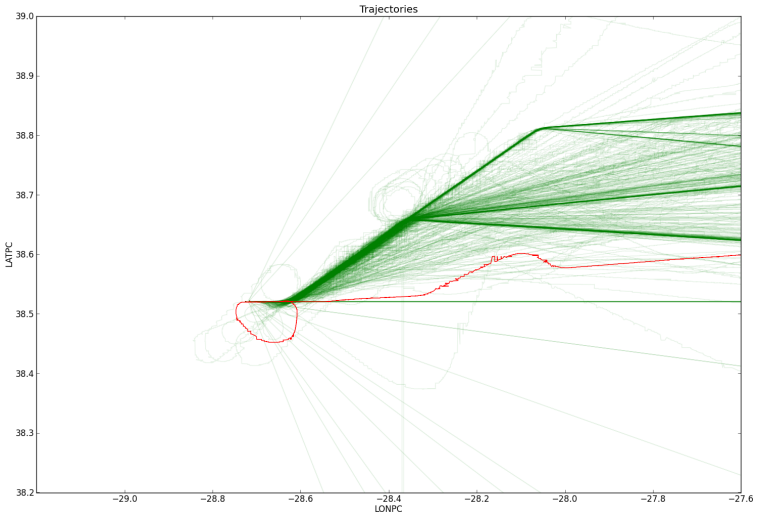


Figure 7.12: Trajectory of Flight 3

Go Around

- Engine not Stopped (N21C > 55% OR N22C>55%).
- Previous value of Flight Phase greater than/equal to APPROACH (pphase ≥ APPROACH).
- Throttle Lever position at TOGA - Engines #1 and #2.

  OR
- Previous value of Flight Phase equal to GO_AROUND (pphase ≥ GO_AROUND).
- Height increasing (ΔHEIGHT > 0 ft) during 3 seconds at least

Experimentally we have seen that our algorithms detected go arounds as atypical flights in almost all cases. This abnormality can also be seen in the other parameters we studied.

7.5 Conclusion

For this dataset we only had access to the data in CSV form, so we could not run the classical analysis on it. Although we did not have the means to make a more

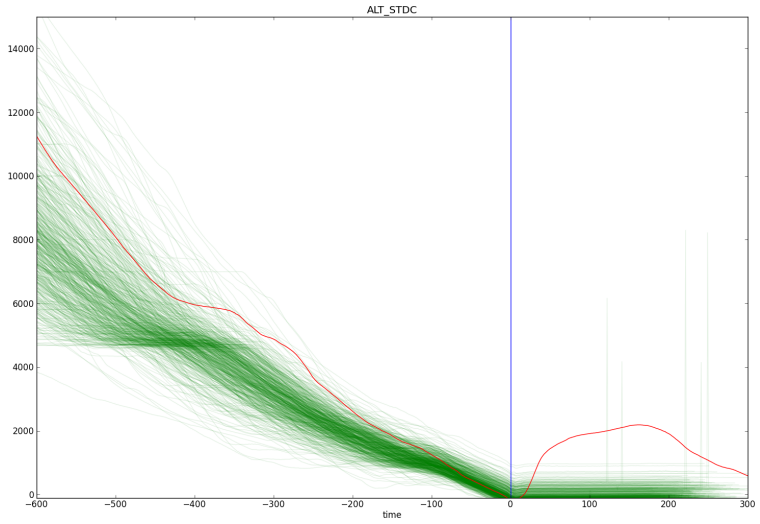


Figure 7.13: Altitude of Flight 3

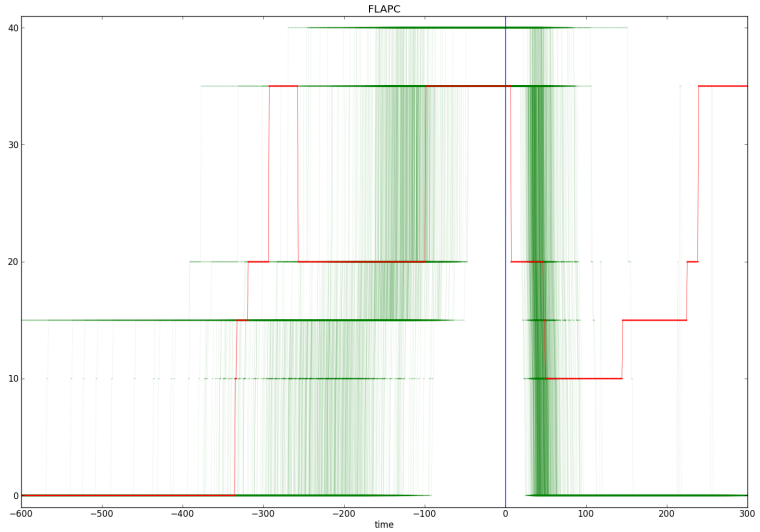


Figure 7.14: Flaps of Flight 3

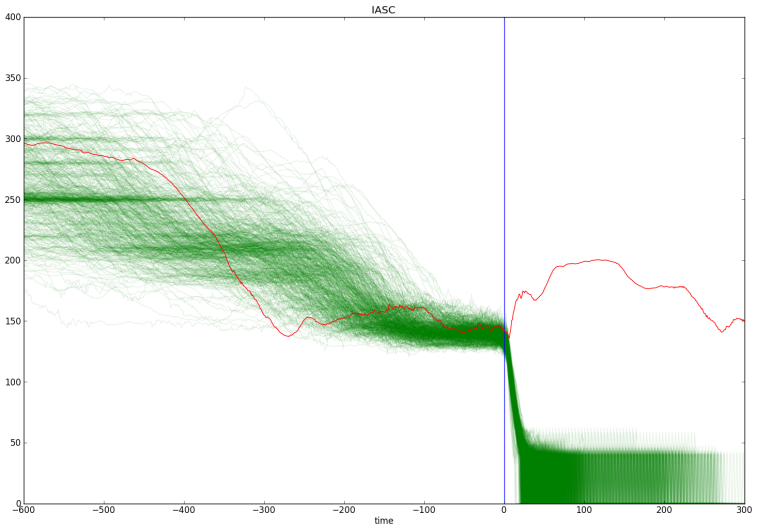


Figure 7.15: Air speed of Flight 3

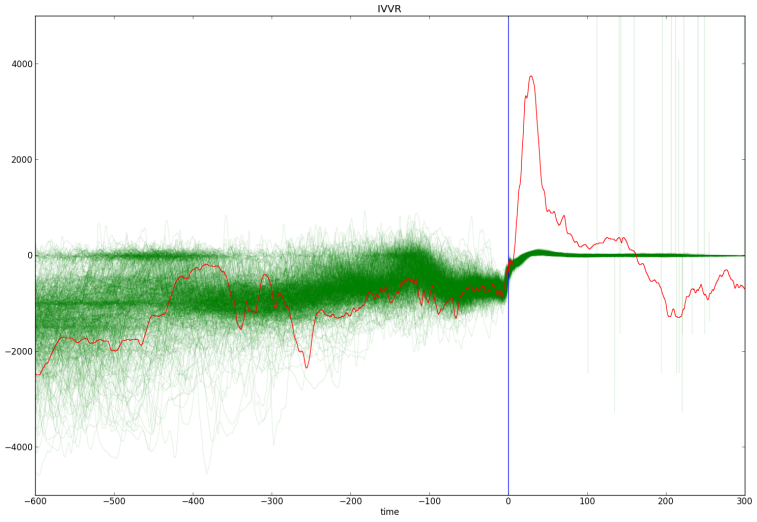


Figure 7.16: Vertical speed of Flight 3

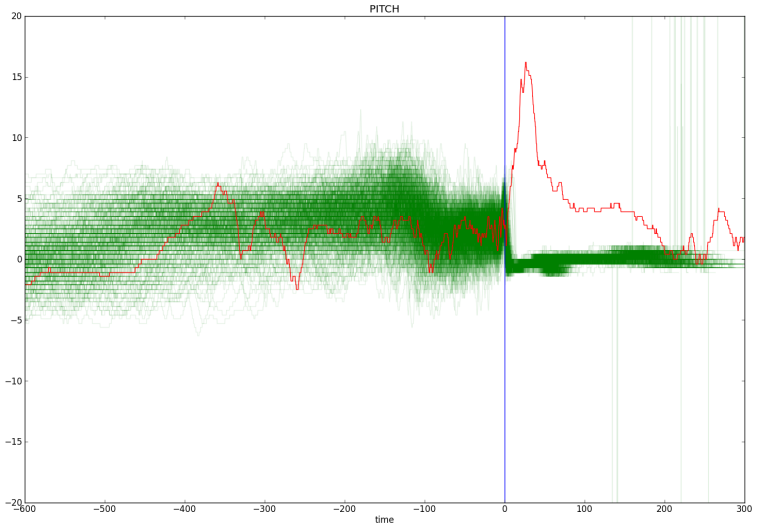


Figure 7.17: Pitch of Flight 3

precise comparison with the classical analysis the feedback we have gotten from our partners in `Airline2` looking at these results was highly positive. They are positive that such a tool would be of tremendous value for FDM operators.

Chapter 8

Conclusions

In the first part of this work, we proposed a novel method for unsupervised novelty detection. This method is interesting in that it uses a criterion from information theory, namely the Rényi entropy, as an *a priori* on the complexity of the distribution of normal data in order to increase detection performance. Using this method we were able to detect outliers in datasets that contained a high proportion of abnormal data.

The second part of this thesis is dedicated to the construction of a kernel suitable for the study of flights modeled as sequences of time-samples. We used a bottom-up approach, by first designing a kernel for each type of parameter that is encountered in FDM, namely continuous, angular, discrete and ordered discrete. Then we explained how one can combine these kernels to define a kernel on the space of time-samples, by using the product kernel approach. Finally we presented a kernel on sequences that is not only provably positive definite but also faster to compute than most other kernels defined using sequence alignments.

Application of these techniques to datasets from `Airline1` and `Airline2` gave very relevant results that prove the usefulness of this approach as a complement to the classical event-based detection systems.

8.1 The right amount of supervision

Traditionally machine learning has been roughly divided into two categories: supervised learning and unsupervised learning [Hastie et al., 2009]. Supervised learning requires a training phase with *labeled data* such as for example `normal` or `abnormal`; after the training phase the algorithm is supposed to be able to determine the label of unseen data. Unsupervised learning does not rely on a training phase nor labeled data, the goal in this case is to understand structure about the data or more generally to infer knowledge about the data.

In supervised learning, given enough samples, enough computation power and enough time, it could be argued that an algorithm may be able to infer the complete relationship between the data and the labels, even down to the feature extraction step. That is essentially the promise of the recent field of deep learning [Bengio, 2009; Le et al., 2012].

Somehow in this work we have taken the opposite approach, we don't have much samples (in the order of thousands), and we don't have much computing resources. In an unsupervised setting, when analyzing large amounts of data there are infinite ways to draw conclusions and to compute statistics; but only a very small subset yields results that are sensible from a domain perspective. That is why in this work and especially in the second part we have tried to "focus" our algorithm on the kind of problems that we want to discover, namely problems in the field of FDMs.

This principle has been formalized in the fields of optimization and machine learning into a set of theorems called the *No Free Lunch* theorems [Wolpert, 1996; Wolpert and Macready, 1997]. The idea is that for an algorithm to give sensible results it has to be somewhat "guided", or in other words it has to be given at least some *a priori* about the data and the class of problems that one wants to solve. In a statistical framework, "*a priori*" can be roughly defined as everything that one knows about the data before seeing the data.

In our case for example, the *a priori* starts with the very definition of the samples: we consider each flight as one sample, we cut them according to well-defined and domain-related events (10000 feet till touchdown) and model them as sequences that are aligned. Furthermore we do not expect to analyze every

parameter but rather choose a set of parameters that we deem to be of interest from an FDM perspective.

In fact it could be argued that the distinction between supervised and unsupervised may not be binary but rather a spectrum, in other words that there is more to supervision than just labeling samples. Any information about the data or the problem that is given to the algorithm during a training phase or event during its conception is a kind of supervision, and as such the distinction between supervised or unsupervised is rather a spectrum.

The goal of course is to find the right place on this spectrum: too far on the unsupervised side and we end up estimating nothing but noise (while needing more samples and more computing power), but too far on the supervised side and we end up doing classical event-based detection, or in other words finding problems we already know...

In a nutshell, it is our belief that the reason the methods described in this work give very meaningful results for FDM practitioners is that they were involved during the conception of the methods, which has allowed us to calibrate our algorithms accordingly.

8.2 Future research

There remains of course many challenges with respect to future research, both practical and theoretical.

Firstly, with respect to the detection of atypical flights we believe that it could be very valuable to be able to point out not only the parameters responsible for the abnormality but also discover precisely the *time* where such abnormality appeared (assuming this abnormality can be defined in time of course). Although it is quite trivial to detect responsible parameters in a feature-based approach, transposing such ideas when dealing with sequences seems much more complicated, and especially with a sufficiently low complexity to be run in a reasonable time.

Secondly, now that we have a positive definite kernel on flights, we could do much more than detecting atypical flights. For example we could implement kernel ridge regression, estimate the landing distance and then try to find the

conditions to minimize it so as to reduce the risk of landing overrun. We could also cluster the dataset and identify the most *typical* flight for each cluster (the "anti-outliers"), that could give valuable information about how pilots in general follow procedures.

Another area of great interest would be an extension of such methods in a semi-supervised setting, such that for example after a first detection phase an FDM practitioner could label the flights (`holding`, `go-around`) such that in the future the method could automatically assign such labels to previously unseen flights. As always, the idea is to minimize the amount of work for practitioners, and in this respect the first phase would still be unsupervised.

Part III

French Abstract

Chapter 9

Résumé en Français

9.1 Introduction

9.1.1 Introduction au contexte industriel

9.1.1.1 La sécurité aérienne

Il est largement admis que l'aviation est l'un des moyens de transport les plus sûrs, au moins en termes de décès par kilomètre. Cependant, le secteur de l'aviation est constamment sous pression pour parvenir à une amélioration de la sécurité. Comme on le voit sur la Figure, le taux d'accident a été relativement stable depuis le début des années 80. Cependant, le volume du trafic aérien a fortement augmenté au cours des deux dernières décennies, passant d'environ vingt-cinq millions d'heures de vol par an à plus de cinquante millions en 2012 [Boeing, 2013] et sera très probablement amené à croître. Cela se traduira par une augmentation globale du nombre d'accidents et de décès.

Cette augmentation du nombre d'accidents est inacceptable, que ce soit pour les constructeurs d'avions ou les compagnies aériennes. Outre les pertes humaines tragiques, chaque accident constitue un énorme coût financier et économique, dû non seulement au coût de remplacement de l'appareil mais aussi à la perte d'exploitation due à l'exposition médiatique après chaque accident.

Les incidents en vol sont souvent dûs à une combinaison de facteurs, soit techniques, comme par exemple un moteur ou une défaillance de la structure ; ou

des événements naturels tels que la foudre, la glace, les impacts d'oiseaux ; ou des facteurs humains tels que des erreurs de l'équipage, un échec de l'organisation, des communications inappropriées etc.

L'industrie dans son ensemble travaille sur ces questions depuis le début de l'exploitation commercial du transport aérien. Les fabricants d'appareils et les équipementiers ont conçu des avions de plus en plus fiables et sûrs, la réglementation a évolué, et un certain nombre d'autres innovations ont été développées telles que les aides à la navigation, le vol aux instruments, etc.

L'une de ces innovations est le développement d'enregistreurs de vols, qui permettent de sauvegarder un grand nombre de paramètres de vols (altitude, vitesse, etc.) tels qu'ils évoluent au cours du temps.

9.1.1.2 L'analyse des données de vol appliquée aux opérations aériennes

Telle que définie par la *Civil Aviation Authority* (CAA), l'analyse des données de vol appliquée aux opérations aériennes, ou *Flight Data Monitoring* en anglais (FDM) est l'analyse systématique, pro-active et non-punitive des données de vols issues des opérations de routine dans le but d'améliorer la sécurité aérienne.

L'idée est d'utiliser les données de vols non pas seulement après un incident pour en comprendre la cause mais aussi de façon générale pour la prévention des incidents.

Depuis 2005 il est désormais obligatoire pour toute compagnie opérant des appareils de plus de 27 tonnes d'avoir un programme de FDM, d'après la législation établie par l'*International Civil Aviation Organization* (ICAO).

9.1.1.3 Limites de l'approche actuelle

L'approche actuelle pour le FDM consiste à surveiller un certain nombre d'événements prédéfinis. Cette approche possède cependant un certain nombre de limites qui ont poussé les opérateurs à rechercher des techniques plus avancées.

La première limitation est qu'un programme de FDM strictement basé sur des événements ne peut par essence détecter que les problèmes qui ont été prévus lors de la création de la table d'événements. Ceci est regrettable car d'un point de vue sécurité, il serait très utile au contraire de détecter les problèmes inattendus.

La deuxième limitation est qu'à mesure que les instruments et les enregistreurs de vol gagnent en sophistication le nombre de paramètres qui sont enregistrés dans les avions modernes devient extrêmement grand, jusqu'à dépasser les 2000 paramètres enregistrés par seconde dans un appareil moderne tel que l'A380.

Il y a donc une augmentation massive du volume des données qui sont enregistrées et qui peuvent être étudiées dans un programme de FDM, cependant une grande partie de ces données est enregistrée mais jamais utilisé dans la plupart des programmes de FDM.

La raison est que le plus souvent les opérateurs s'appuient uniquement sur un petit nombre de paramètres fondamentaux qui sont bien connus par les acteurs de l'industrie. Cela laisse cependant un énorme volume de données disponible mais non exploité, et qui représente, *a priori*, une grande valeur pour les compagnies aériennes.

9.1.2 But de la thèse

Le but de la thèse est de concevoir des techniques statistiques et d'apprentissage automatique avancées pour améliorer les procédures FDM. Plus précisément,

---

Le but de ces travaux est de concevoir une méthode pour détecter les *vols atypiques*, parmi un jeu de données de plusieurs centaines ou milliers de vols.

Un vol atypique est un vol qui est différent en un sens de la plupart des autres vols étudiés. Un vol atypique présente donc probablement des problèmes de sécurité ou opérationnels, et doit donc être étudié par un analyste FDM.

Le but de ce type d'étude statistique n'est pas de remplacer les méthodes "classiques" mais d'apporter un complément d'étude, avec l'espoir que ces nouvelles méthodes détectent des vols qui passent au travers d'approches plus traditionnelles.

Cette méthode doit être non supervisée (c'est à dire sans phase d'entraînement), doit être capable d'étudier n'importe quelle combinaison de paramètres de n'importe quel type (continu, discret, angulaire etc.), doit prendre en compte l'aspect séquentiel du vol et doit pouvoir analyser une flotte de plusieurs centaines de

vols en moins de quelques heures sur un PC standard.

9.1.3 Approche mathématique

9.1.3.1 Jeu de données

Toute procédure statistique commence par le choix d'un *jeu de données*. L'approche que nous avons choisie pour ces travaux est la suivante:

Chaque vol, ou phase de vol, constitue un échantillon.

Ainsi, en étudiant une flotte de 621 vols, nous possédons d'un point de vue statistique 621 échantillons.

Remarquons que ce n'est pas la seule approche valable pour l'analyse des données de vol, par exemple dans le cadre de la maintenance de système il est préférable d'étudier des échantillons temporels, de telle sorte par exemple qu'un essai de 600 secondes échantilloné à 1Hz corresponde à 600 échantillons statistiques. Cependant pour l'analyse des données de vols *d'un point de vue opérationnel* nous avons constaté, et d'autres avant nous [Amidan and Ferryman, 2005; Das et al., 2010] que c'était la meilleure approche.

9.1.3.2 Type de méthode

Pour ces travaux nous avons choisi de nous intéresser à la classe des méthodes à noyau [Hastie et al., 2009], d'abord parce qu'elles sont particulièrement adaptées à l'étude de données structurées telles que définies dans la section suivante, mais aussi parce qu'elles sont assez performantes et reposent sur des bases mathématiques saines [Schölkopf and Smola, 2002].

9.1.4 Structure des données de vol

Ce choix du jeu de données conditionne ainsi la structure mathématique de ces données.

ALT	8500
AIRSPEED	254
AUTO	ON
PITCH	5°

Table 9.1: Structure d'un échantillon temporel

9.1.4.1 Echantillons temporels

En premier lieu, intéressons-nous aux données qui sont enregistrées à chaque instant (les fréquences varient de 0.5Hz à 32Hz) au cours du vol. Nous appelons ces données un *échantillon temporel*. Nous modélisons un échantillon temporel comme une structure composite qui peut contenir un ou plusieurs paramètres de types potentiellement différents. Par exemple, nous pourrions faire une étude avec 4 paramètres: ALT, AIRSPEED, AUTO et PITCH. Dans ce cas, ALT et AIRSPEED sont des valeurs *continues* et peuvent donc être modélisées par des nombres réels. Au contraire, AUTO est un paramètre discret tandis que PITCH est un paramètre angulaire.

Un exemple d'échantillon temporel est illustré en Table 9.1.

Autant la distinction entre paramètres continus et discrets est assez évidente, autant les paramètre angulaires sont très souvent traités comme des paramètres continus. C'est bien sûr une erreur d'un point de vue mathématique, car ces deux espaces possèdent des topologies bien distinctes: les données continues reposent sur une droite alors que les données angulaires reposent sur le cercle unité. Ainsi, de simples statistiques telles que la moyenne perdent totalement leur sens dès lors qu'on traite des données angulaires.

Dans cette thèse l'espace des échantillons temporels est noté $\mathcal{X}$. Le traitement de ces données multivariées et de types hétérogène fait l'objet du Chapitre 5 de la thèse.

9.1.4.2 Structure d'un vol

Nous modélisons ainsi un vol (ou une phase de vol) comme une *séquence d'échantillons temporels*. Il est clair que cette modélisation fait sens, car les données sont enregistrées de façon continue au cours d'un vol.

Vol 1				
ALT	8500	8400	...	2000
AIRSPEED	254	256	...	100
AUTO	ON	ON	...	OFF
PITCH	5°	5°	...	0°
temps	0s	1s	...	612s

Vol 2					
ALT	6300	6200	6100	...	1000
AIRSPEED	120	122	110	...	100
AUTO	ON	ON	ON	...	OFF
PITCH	3°	4°	2°	...	2°
temps	0s	1s	2s	...	598s

Vol 3				
ALT	7300	7200	...	1500
AIRSPEED	254	256	...	200
AUTO	OFF	OFF	...	OFF
PITCH	4°	4°	...	2°
temps	0s	1s	...	703s

Table 9.2: Exemple de structure d'un jeu de données contenant 3 vols.

Ainsi la terminologie "échantillon temporel" permet d'éviter toute ambiguité, sachant que d'un point de vue statistique nous considérons que chaque vol est un échantillon.

Notons aussi que bien qu'il soit possible de considérer des vols entiers, nous avons constaté que nous obtenons de bien meilleurs résultats en focalisant notre étude sur des *phases de vols* bien définies, telles que le décollage, la phase de descente, l'atterrissage etc.

La structure d'un jeu de données complet avec plusieurs vols ressemble donc à ce qui est donné en Table 9.2. Dans cette thèse, l'espace des vols (autrement dit l'ensemble auquel appartiennent chacun de nos échantillons) est noté $\mathcal{X}^*$. Ainsi mathématiquement,

$$\mathcal{X}^* = \cup_{i=1}^{\infty} \mathcal{X}^i. \tag{9.1}$$

Notons que cette structure est bien différente de celle que l'on trouve typiquement

dans des jeux de données standard, où chaque échantillon possède un nombre *fixe* de facteurs. Ainsi, même dans le cas où tous les paramètres seraient de type continu, la difficulté est que nous ne pouvons plus utiliser le formalisme vectoriel des espaces Euclidiens. En effet comme les vols possèdent des durées différentes alors les séquences sont elles aussi de longueurs variables et l'on ne peut plus alors "additionner" par exemple deux échantillons comme on pourrait le faire dans un espace vectoriel. Dans ce cas les méthodes à noyau nous serons d'une grande aide car il suffit de construire un noyau $k^* : \mathcal{X}^* \times \mathcal{X}^* \to \mathbb{R}$ qui soit défini positif pour projeter de façon implicite les échantillons (vols) dans un espace Hilbertien (plus précisément un espace Hilbertien à noyau reproduisant) dans lequel il devient possible de mener un certain nombre de procédures statistiques classiques (analyse en composante principale, analyse discriminante etc.)

9.2 Détection de nouveauté

La première partie de cette thèse concerne la détection d'anomalie, autrement appelée la détection de nouveauté. Le but est de séparer dans un ensemble d'échantillons donné les échantillons normaux des échantillons anormaux [Markou and Singh, 2003a,b].

Une citation de Hawkins [1980] définit assez bien le concept de nouveauté: "Une nouveauté est une observation qui diffèrent tellement des autres observations qu'il est raisonnable d'envisager qu'elle ait été générée par un autre mécanisme"[1].

9.2.1 Détection non supervisée

Le cadre dans lequel nous nous plaçons est celui de la détection *non supervisée*. En effet, la plupart des méthodes de l'état de l'art, telles que le *one-class support vector machine* (OC-SVM) [Schölkopf et al., 2001] ou le *support vector data description* (SVDD) [Tax and Duin, 2004] reposent sur une phase d'apprentissage où les données ne contiennent que (ou alors en majorité) des données normales. Après la phase d'apprentissage, les algorithmes sont alors soumis à des données

[1] Traduction de l'anglais: "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism"

qui peuvent être ou normales ou anormales.

Dans le cadre dans lequel nous nous plaçons, il n'y a pas de phase d'apprentissage, et les données peuvent contenir une proportion significative d'échantillons anormaux. Il est possible de détecter des données anormales dans ce cadre à condition qu'un a priori assez fort soit disponible sur la distribution des données normales.

L'avantage d'une méthode non supervisée est qu'il n'est donc pas nécessaire d'avoir un jeu de données *étiquettées*. Or le processus d'étiquettage d'un jeu de données est long, coûteux et sujet à de possibles erreurs. Ainsi un algorithme non supervisé est un grand avantage d'un point de vue métier.

9.2.2 Methodologie

La plupart des algorithmes de détection de nouveauté reposent sur l'une ou l'autre ou les deux hypothèses suivantes:

1. La plupart des échantillons sont normaux,
2. Les échantillons normaux sont plus *concentrés*.

Les algorithmes de l'état de l'art formalisent la seconde hypothèse selon des termes géométriques comme le SVDD, ou encore probabilistes comme le *minimum volume set* [Scott and Nowak, 2006]. Notre approche est innovante car nous utilisons un formalisme de théorie de l'information [Principe, 2010].

9.2.3 A priori sur la distribution

Plus précisément, nous exprimons un a priori sur la complexité de la distribution en termes d'entropie de Rényi. L'entropie de Rényi [Rényi, 1961] est une forme alternative de mesure de l'entropie de celle de Shannon, dont l'avantage est de pouvoir être facilement estimée à l'aide de statistiques non paramétriques. On considère ainsi qu'une distribution possédant une forte entropie est une distribution "simple" tandis qu'une distribution possédant une faible entropie est une distribution "complexe".

9.2.4 Méthodologie

On suppose que le noyau utilisé est à la fois défini positif et peut être utilisé comme fenêtre de Parzen [Parzen, 1962] pour l'estimation de densité, comme le noyau Gaussien. Nous démontrons dans ce cas en utilisant le formalisme de la théorie des variables aléatoires dans un espace de Hilbert [Blanchard et al., 2007] et en suivant une approche proposée à l'origine par Girolami [2002] que la densité de probabilité des données peut être décomposée en série de termes orthogonaux et qu'à chaque terme correspond une dimension principale dans l'espace de Hilbert à noyau reproduisant.

A chaque terme correspond aussi une quantité de potentiel d'information, il est donc possible de sélectionner les termes les plus importants en fonction de ce critère, tel que décrit dans la méthode proposée par Jenssen [2009], le choix du potentiel d'information résultant correspondant ainsi à un a priori sur la complexité de la distribution.

9.2.5 Contributions

Notre principale contribution est la démonstration d'un lien entre l'erreur de reconstruction de l'image dans par le noyau un point à l'espace principal du noyau est borné par la densité de probabilité tronquée à ce point dans l'espace de départ. Ainsi, classer les points selon leur erreur de reconstruction possède un sens d'un point de vue statistique en plus du sens géométrique.

9.2.6 Résultats sur données diverses

Nous illustrons cette approche en comparant notre méthode à l'état de l'art sur des jeux de données classiques. Nous mettons en évidence que connaître la complexité de la distribution (par exemple savoir à l'avance qu'elle est unimodale) permet d'augmenter la performance de la méthode. Notre approche se compare très favorablement à l'état de l'art.

Classe	Nombre total	Détecté par KECA	Détecté par MKAD
Classe 1	23 vols	11 vols	19 vols
Classe 2	5 vols	3 vols	4 vols
Classe 3	1 vol	0 vol	1 vol

Table 9.3: Comparaison avec l'analyse classique

9.2.7 Résultats sur données de la compagnie *

Nous avons appliqué notre méthode à une flotte de la compagnie [redacted], qui comprend 721 vols de Porto à la piste 26 d'Orly. Nous nous sommes restreints à 13 des paramètres les plus importants en opération aérienne et en avons extrait 22 descripteurs.

Nous avons comparé notre méthode à l'état de l'art pour l'analyse statistique des données de vols appliquée aux opérations aériennes, le NASA *Multiple Kernel Anomaly Detection* [Das et al., 2010], ainsi qu'aux résultats obtenus par l'analyse "classique" avec le logiciel SAGEM AGS. Le logiciel AGS classe les événements détectés en un seuil de sévérité de 1 à 3. Notre méthode détecte 35 vols, tandis que le MKAD en détecte 43, pour un total unique de 64 vols. Nous avons étudié la répartition des événements classiques et comparé avec les approches statistiques en Table 9.3. Plus important, les approches statistiques détectent de vols qui ont été considérés par des analystes comme intéressants et qui n'ont pas du tout été détectés par les méthodes classiques.

Un exemple de vol atypique, dont la pvalue a été estimée à environ 10^{-14}. Ce vol est un parfait exemple de vol atypique, on remarque en Figure que le pilote effectue une boucle très courte juste avant l'atterrissage. Il est possible que cette manœuvre ait été faite car la vitesse de l'appareil était trop grande.

9.3 Distances et similarités

Le Chapitre 4 introduit la deuxième partie de la thèse et donc le sujet des données structurées. Nous nous intéressons aux noyaux *à base radiale*, dont les valeurs peuvent s'interpréter comme des mesures de similarité, et nous interrogeons sur les conditions nécessaires ou suffisantes pour la définie positivité de tels noyau,

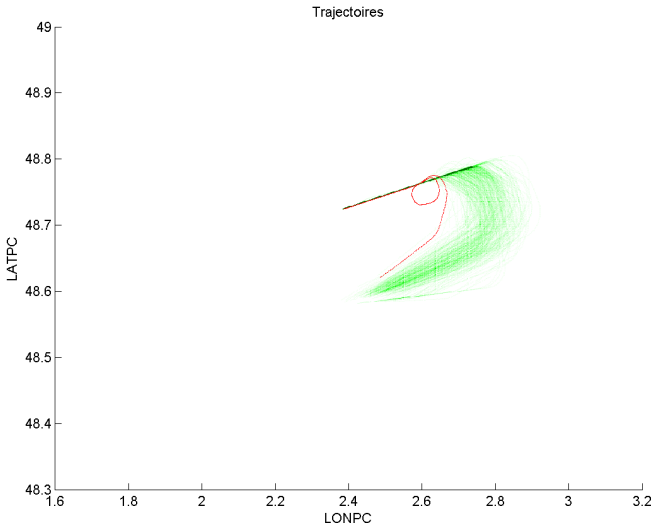


Figure 9.1: Trajectoire du Vol 1

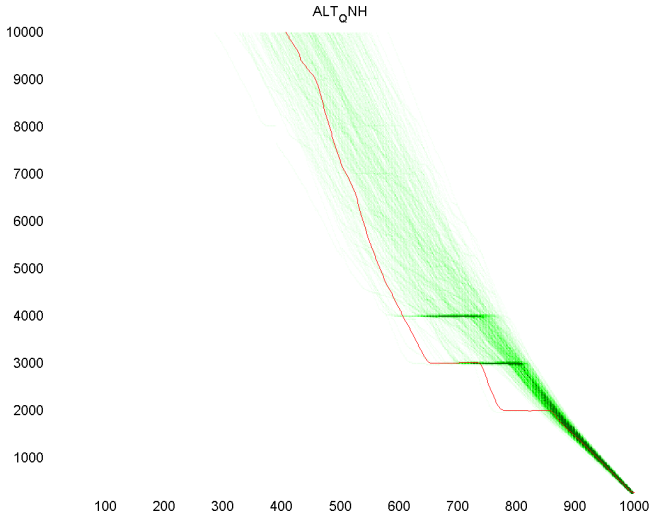


Figure 9.2: Altitude du Vol 1

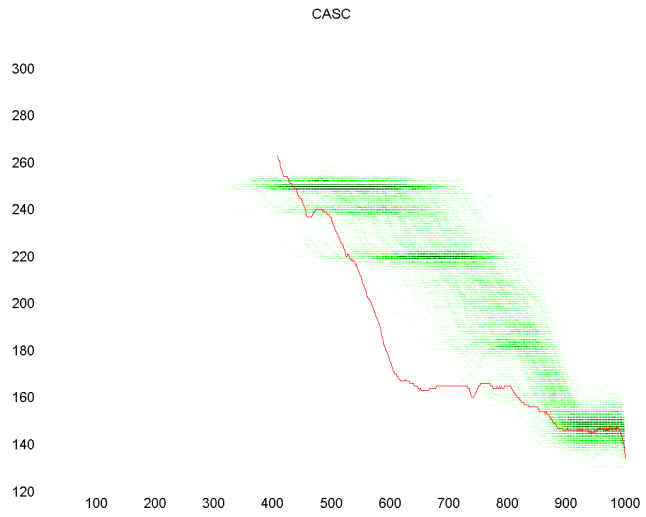


Figure 9.3: Vitesse air du Vol 1

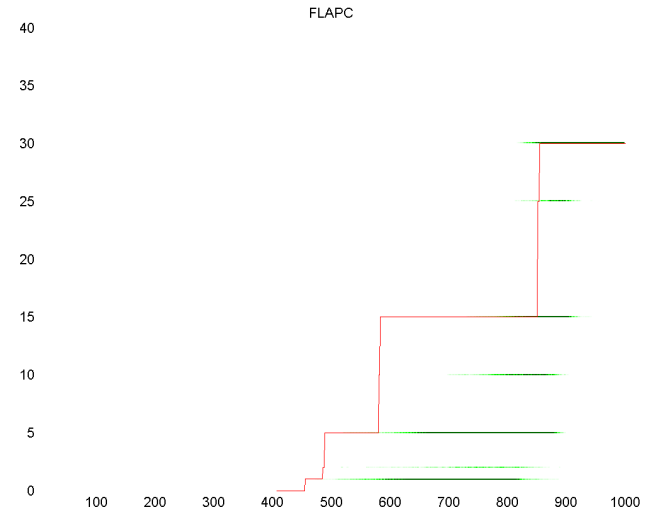


Figure 9.4: Volets du Vol 1

par analogie avec le très connu noyau Gaussien.

Nous introduisons le concept de noyau infiniment divisble, et reformulons ainsi un résultat établi par Berg et al. [1984], qui servira de base aux développements de la deuxième partie de la thèse:

Theorem 10. *Soit d une métrique Hilbertienne sur un espace $\mathcal{X}$, et pout tout $t > 0$, notons k_t le noyau sur $\mathcal{X}$ défini comme:*

$$k_t x, y = \exp -t \cdot d^2 x, y$$

Nous avons alors:

1. *k_t est défini positif,*
2. *k_t est à base radiale,*
3. *k_t est infiniment divisible.*

9.4 Données multivariées et de types hétérogène

Dans le Chapitre 5, nous nous intéressons à l'une des particularité du domaine du FDM, qui est le fait que les paramètres peuvent être de types variés. Il est donc nécessaire de prendre en compte ce fait pour la conception de notre noyau pour la détection de nouveauté, tout en faisant en sorte que celui-ci estime correctement la structure de dépendance entre ces paramètres.

9.4.1 Un noyau par type de données

Pour chacun des types de données de données que l'on rencontre dans le domaine du FDM, plus précisément les données continues $\mathbb{R}$, les données angulaires $\mathbb{A}$, les données discrètes $\mathbb{D}$ et les données discrètes ordonnées $\mathbb{O}$, nous proposons un noyau défini positif et infiniment divisible, ainsi qu'un moyen d'estimer un "centroide" et une mesure de dispersion généralisée.

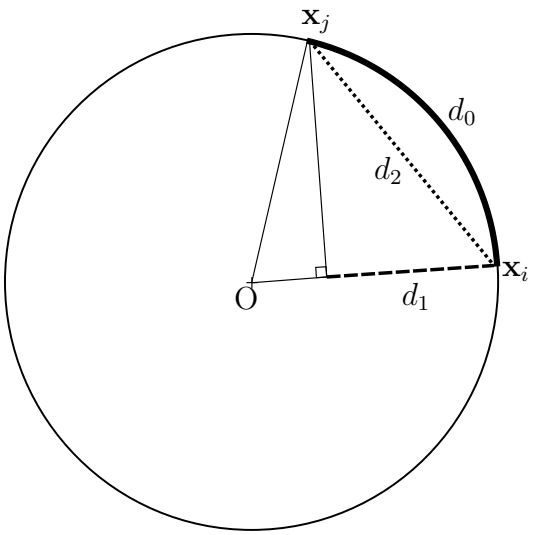


Figure 9.5: Illustration sur le cercle unité de trois possibles distances angulaires: d_0, d_1 et d_2.

Données continues Ces données se modélisent comme des réels et il est tout simplement possible d'utiliser le noyau Gaussien pour les comparer:

$$k_c x_i, x_j = \exp -x_i - x_j{}^2 \tag{9.2}$$

Données angulaires Comme expliqué précédemment, les données angulaires ne doivent pas être confondues avec les données continues même si leur représentation informatique - les nombres à virgule flottante - est la même. En tirant partie de la topologie circulaire de ces données, on propose un noyau défini positif et infiniment divisible, appelé le noyau Gaussien sphérique et qui s'écrit ainsi:

$$k_a \theta_i, \theta_j = \exp -d_2 \theta_i, \theta_j{}^2 = \exp \left(-4 \sin^2 \frac{\cup^{\theta_j - \theta_i} \cup}{2} \right]$$

La distance d_2 est la distance Euclidienne entre les points sur le cercle unité correspondant à chacun des angles, comme illustré sur la Figure 9.5. Notons que les distances d_0 et d_1 auraient aussi pu être utilisées mais possèdent des propriétés mathématiques différentes.

Données discrètes Nous proposons le noyau suivant:

$$k_d : \begin{array}{rcl} \mathbb{D} \times \mathbb{D} & \to & \mathbb{R} \\ x, y & \mapsto & \begin{array}{ll} 1 & \text{if } x = y \\ a & \text{if } x \neq y \end{array} \end{array}$$

Et démontrons que pour tout $a < 1$ ce noyau est défini positif et infiniment divisible.

Données discrètes ordonnées Notre traitement des données discrètes ordonnées consiste à d'abord projeter ces données sur la droite des réels, selon une fonction qui doit posséder un sens d'un point de vue du métier, et ensuite d'utiliser le noyau Gaussien sur les valeurs discrètes. Pour le paramètre AUTO on pourrait par exemple utiliser la projection suivante:

$$\phi : \begin{array}{rcl} \mathbb{O} & \to & \mathbb{R} \\ \mathsf{OFF} & \mapsto & 0 \\ \mathsf{FD} & \mapsto & 1 \\ \mathsf{CMD} & \mapsto & 2 \end{array}$$

Et le noyau s'écrirait donc ainsi:

$$k_o : \begin{array}{rcl} \mathbb{O} \times \mathbb{O} & \to & \mathbb{R} \\ x, y & \mapsto & \exp - \left\| \phi x - \phi y \right\|^2 \end{array}$$

Par construction, ce noyau est défini positif et infiniment divisible.

9.4.2 Combinaison des noyaux

Il ne reste plus qu'à trouver un moyen de combiner les noyaux k_c, k_a, k_d et k_o pour créer un noyau k sur un espace issu de n'importe quelle combinaisons de ces paramètres, du type $\mathcal{X} = \mathbb{R}^{d_c} \times \mathbb{D}^{d_d} \times \mathbb{A}^{d_a} \times \mathbb{O}^{d_o}$.

Nous comparons les deux approches possibles : la combinaison conique [Bach et al., 2004], qui à l'origine a été conçue dans un cadre supervisé avec une procédure d'optimisation pour trouver les paramètres de combinaison optimaux,

et le produit [Haussler, 1999; Shin, 2013].

Nous illustrons le fait que pour pouvoir estimer correctement la relation de dépendance entre tous les paramètres, et pouvoir détecter une anomalie résultant d'une mauvaise "synchronisation" entre paramètres, la bonne approche à choisir est l'approche par produit. Au contraire, l'approche par combinaison conique est la bonne approche lorsque l'on cherche à estimer une fonction qui peut s'interpréter comme la somme de fonction indépendantes.

Dans le cas simple où l'on étudie seulement deux paramètres de types différents, modélisés respectivement par $\mathcal{X}$ et $\mathcal{Y}$, le noyau produit (généralisé) s'écrit ainsi:

$$\begin{aligned} k_Z &: \mathcal{X} \times \mathcal{Y} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \\ x_0, y_0, x_1, y_1 &\mapsto k_X x_0, x_1 \cdot k_Y y_0, y_1 \end{aligned}$$

A l'aide de la théorème de Schür [Horn and Johnson, 2012] il est possible de prouver que le noyau k_Z est défini positif et infiniment divisible dès que k_X et k_Y le sont. Ainsi, par extension à plus de deux paramètres, nous avons ainsi conçu un noyau k défini positif et infiniment divisible capable de comparer deux *échantillons temporels*.

9.5 Etude séquentielle

9.5.1 Cadre

Dans le chapitre 6 en partant du noyau k nous concevons un noyau k^* sur l'espace $\mathcal{X}^*$ des séquences d'échantillons temporels. Soient $\boldsymbol{x} = x_1, \ldots, x_l$ et $\boldsymbol{x'} = x'_1, \ldots, x'_m$ deux éléments de $\mathcal{X}^*$. Dans le cas général les deux séquences peuvent ne pas avoir la même taille, autrement dit $l \neq m$, et il n'est alors plus possible d'utiliser des techniques vectorielles traditionnelles telles qu'un noyau Gaussien. Dans la suite de cette partie, on notera $\boldsymbol{x}$ et $\boldsymbol{x'}$ deux séquences quelconques, $\boldsymbol{x}_l$ la plus courte de ces séquences avec l sa taille, $\boldsymbol{x}_m$ la plus longue des deux séquences avec m sa taille, de telle sorte que $l \leq m$.

Dans le cas de données discrètes une solution est par exemple de calculer le nombre minimal d'opérations (telles que ajout, suppression, modification)

```
ATGCCGTGACATGCATTTAAGC
GTG-CGT-ATATG--TTT---C
```

Figure 9.6: Alignements avec creux

```
ATGCCGTGACATGCATTTAAGC
ATGGCGTTACATGGGTTCCCCC
```

Figure 9.7: Alignements avec répétitions

permettant de passer d'une séquence à une autre. Ce nombre de d'opérations peut être considéré comme une distance et est appelé la distance de Levenshtein [Levenshtein, 1966]. Une autre solution consiste à définir des alignements entre séquences.

9.5.2 Alignements de séquences

Un alignement associe des éléments d'une séquence aux éléments d'une autre séquence tout en préservant l'ordre. Les alignements peuvent introduire ou des creux ou des répétitions comme illustré en Figures 9.6 et 9.7.

Une fois un alignement défini il devient alors possible de calculer un score d'alignement entre les deux séquences, par exemple en additionnant les distances entre chaque entre élément d'une paire.

9.5.3 Formalisme des alignements

Dans ces travaux nous nous intéressons aux alignements qui introduisent des répétitions. Plus précisément, nous considérons les alignements où *seule la séquence la plus courte peut avoir des échantillons répétés.* Nous les appelons les *alignements unilatéraux*, ou *one-sided alignments* en anglais. Supposons que $\boldsymbol{x}$ est plus courte que $\boldsymbol{x}'$, de telle sorte que $l \leq m$, alors l'on peut définir un alignement π

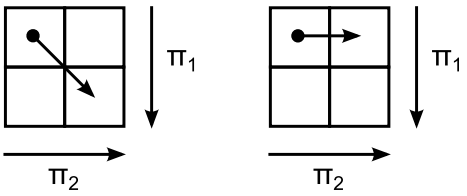


Figure 9.8: Mouvements pour les alignement unilatéraux

comme une paire π_1, π_2 d'indexes croissants tels que:

$$\begin{aligned} 1 = \pi_1 1 \leq \ldots \leq \pi_1 p = l \\ 1 = \pi_2 1 \leq \ldots \leq \pi_2 p = m \end{aligned} \tag{9.3}$$

et

$$\begin{matrix} \pi_1 i + 1 - \pi_1 i \\ \pi_2 i + 1 - \pi_2 i \end{matrix} \in \begin{matrix} 0 & 1 \\ 1 & , 1 \end{matrix} . \tag{9.4}$$

Ainsi, entre deux séquences de même taille il n'existe qu'un seul alignement unilatéral, qui est l'alignement trivial $\forall i \in 1 \ldots l, \pi_1 i = \pi_2 i = i$. Dans le cas général où $l \leq m$ il est clair que le nombre d'alignements unilatéraux est égal à $\binom{m-1}{l-1}$. On note $\mathcal{A}^{-}\boldsymbol{x}, \boldsymbol{x}'$ l'ensemble des alignements unilatéraux entre $\boldsymbol{x}$ et $\boldsymbol{x}'$.

9.5.4 Représentation des alignements

Il est possible de représenter les alignements entre deux séquences de taille l et m comme des chemins sur une matrice de taille l, m. Nous représenterons toujours la séquence la plus courte par les indices verticaux de la matrice. L'équation 9.3 signifie que le chemin commence dans le coin supérieur gauche et se termine dans le coin inférieur gauche. L'équation 9.4 se traduit par le fait qu'il n'y a que deux "mouvements" possible dans le cas unilatéral, qui sont illustrés en Figure 9.8. Ainsi deux exemples de représentation d'alignements unilatéraux sont illustrés en Figure 9.9. Remarquons que dans la Figure 9.9 certaines cases sont hachées : ce sont des cases qui sont inatteignables dans le cas unilatéral à cause des restrictions des Equations 9.3 et 9.4.

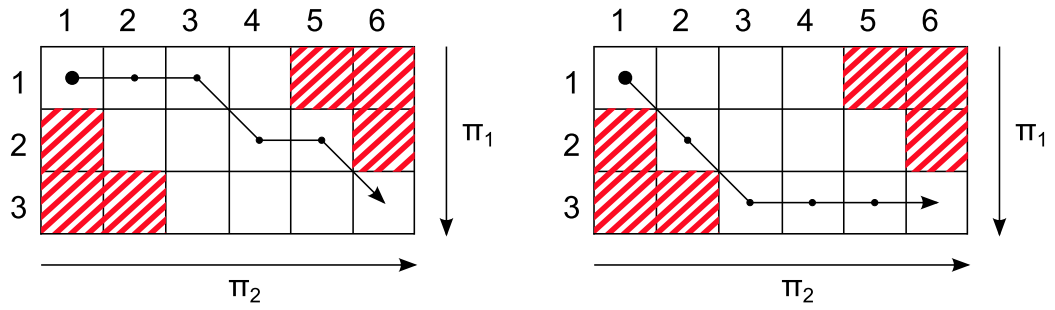


Figure 9.9: Deux exemples d'alignements unilatéraux

9.5.5 Formalisme des dilatations

Alternativement aux alignements nous proposons un formalisme plus simple à manipuler qui s'appelle les *opérateur de dilatation*. Un opérateur de dilatation ϵ associe à une séquence une séquence plus longue en répétant un ou plusieurs de ses éléments sans en changer l'ordre. On note $\xi_{l\to m}$ l'ensemble des opérateurs de dilatation qui permettent de passer d'une séquence de taille l à une séquence de taille m. A chaque alignement unilatéral correspond un unique opérateur de dilatation de la séquence la plus courte vers la plus longue et par conséquent, $\bigcup^{\xi_{l\to m}}\bigcup = {}^{m-1}_{l-1}$.

9.5.6 Le noyau par moyenne d'alignements unilatéraux

9.5.6.1 Cas pratique: séquences réelles

Nous proposons dans cette thèse de définir le noyau par moyenne d'alignements unilatéraux, ou *one-sided mean alignment kernel* en anglais. Nous commençons par présenter son expression dans le cas simple de données séquentielles réelles, de telle sorte que $\mathcal{X} = \mathbb{R}$ et k est le noyau Gaussien à une dimension :

$$kx, x' = \exp{-x - x'^2} \tag{9.5}$$

Dans ce cas, le noyau par moyenne d'alignements unilatéraux s'écrit ainsi:

$$k^*\boldsymbol{x}, \boldsymbol{x}' = \exp{-\frac{1}{\bigcup^{\xi_{l\to m}}\bigcup} \mathop{\mathcal{P}}_{\epsilon\in\xi_{l\to m}} \frac{1}{m}\prod^{\epsilon}\boldsymbol{x}_l - \boldsymbol{x}_m\prod^{2}}. \tag{9.6}$$

Notons que l'équation est bien symétrique par rapport à $\boldsymbol{x}$ et $\boldsymbol{x}'$, mais n'est pas symétrique par rapport $\boldsymbol{x}_l$ et $\boldsymbol{x}_m$, car la séquence la plus courte possède un rôle particulier. En outre, lorsque l'on compare deux séquences de même taille, le noyau se réduit au noyau Gaussien vectoriel usuel:

$$k^*\boldsymbol{x}, \boldsymbol{x}' = \exp -\frac{1}{m}\prod \boldsymbol{x} - \boldsymbol{x}' \prod^2 . \tag{9.7}$$

9.5.6.2 Cas abstrait: noyaux infiniment divisibles

Considérons un noyau k sur l'espace $\mathcal{X}$. Le noyau par moyenne d'alignements unilatéraux peut être défini comme la *moyenne géométrique* des scores d'alignements:

$$k^*\boldsymbol{x}, \boldsymbol{x}' = \left(\underset{\pi \in \mathcal{A}^-\boldsymbol{x}, \boldsymbol{x}'}{\mathcal{L}} \bigcup_{i=1}^{\pi} \bigcup kx_{\pi_1 i}, x'_{\pi_2 i}{}^{\overline{\bigcup^{\frac{1}{\pi}} \bigcup}} \right]^{\overline{\bigcup^{\frac{1}{\mathcal{A}^-\boldsymbol{x}, \boldsymbol{x}'}} \bigcup}}$$

Selon le formalisme des opérateurs de dilatation ce noyau peut aussi s'écrire ainsi:

$$k^*\boldsymbol{x}, \boldsymbol{x}' = \left(\underset{\epsilon \in \xi_{l \to m}}{\mathcal{L}} k_m \epsilon \boldsymbol{x}_l, \boldsymbol{x}_m{}^{\frac{1}{m}} \right]^{\overline{\bigcup^{\frac{1}{\xi_{l \to m}}} \bigcup}} \tag{9.8}$$

9.5.6.3 Théorème principal

Notre principale contribution pour ce chapitre consiste en la démonstration du théorème suivant:

Theorem 11. *Le noyau par moyenne d'alignements unilatéraux k^* vérifie les propriétés suivantes:*

1. *Si k est défini positif et infiniment divisible, alors k^* est défini positif,*
2. *Pour deux séquences $\boldsymbol{x}$ et $\boldsymbol{x}'$ de même taille m, k^* se réduit au noyau produit: $k^*\boldsymbol{x}, \boldsymbol{x}' = k_m \boldsymbol{x}, \boldsymbol{x}'^{\frac{1}{m}}$.*
3. *Si k est un noyau à base radiale, alors k^* est un noyau à base radiale.*

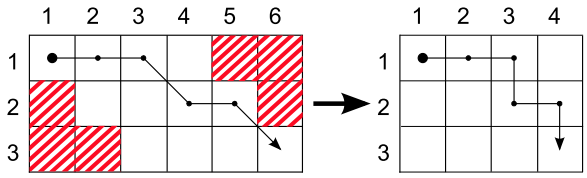


Figure 9.10: Transformation du domaine

9.5.6.4 Démonstration

Le but est de démontrer que toute évaluation du noyau sur un nombre fini d'échantillons conduit à une matrice définie positive. Pour cela nous utilisons la propriété d'infinie divisibilité des noyaux, ce qui nous permet de "diviser" l'évaluation du noyau en parties suffisamment petites qui seront alors ré-arrangées pour faire apparaître le fait que la matrice de Gram peut s'exprimer comme le produit de Schür (produit terme à terme) de matrices de Gram élémentaires dont il est facile de prouver la définie positivité.

9.5.7 Implémentation à l'aide de programmation dynamique

Comme le nombre d'alignements croît en raison factorielle de la taille des séquences, il est déraisonnable de calculer de façon exhaustive les scores associés à chacun des alignements. Au lieu de cela il est possible d'utiliser des techniques de programmation dynamique afin d'aboutir à une complexité polynomiale en $Ol \times m - l$. Remarquons que la plupart des autres algorithmes basés sur des alignements sont en complexité $Ol \times m$, mais le fait que nous ne considérons que les alignements unilatéraux nous permet de diminuer l'espace de recherche comme illustré en Figure 9.10.

9.6 Résultats sur données `Airline2`

Dans le chapitre 7 nous appliquons le noyau conçu dans la seconde partie de la thèse à des données qui nous ont été fournies par la compagnie aérienne [redacted]. Il nous a été fourni les données de 604 vols d'A320, de [redacted] jusqu'à la ville d'[redacted], située sur l'île [redacted] dans l'archipel des [redacted] [redacted] kilomètres à l'Ouest du [redacted].

9.6.1 Prétraitement des données

Comme dans le chapitre 3, nous commençons par découper les vols en phase et nous n'étudions qu'une phase précise de chaque vol, définie comme la partie de la descente commençant à 10000 pieds jusqu'à l'arrêt complet de l'appareil sur la piste d'atterrissage.

Egalement nous ne retenons que 12 paramètres, qui sont des paramètres fondamentaux pour l'étude opérationnelle.

Afin d'obtenir des séquences nous ne prenons pas les données échantillonnées à leur fréquence d'enregistrement, mais nous sous-échantillonnons les données sur des fenêtres de durée de 5s.

9.6.2 Résultats

Parmi les 604 vols, 9 vols ont été considérés comme très atypiques avec une pvalue inférieure à 10^{-2} et 17 autres ont été considérés atypiques avec une pvalue inférieure à 0.05.

Nous présentons deux exemples de vols atypiques.

Le premier vol a été détecté avec une pvalue égale à 0.0032. Comme illustré en Figure 9.11, ce vol présente dans sa trajectoire une phase de *holding* très longue, qui est probablement due à la présence de nuages, le pilote a donc certainement dû attendre d'avoir des conditions visuelles suffisantes pour atterrir. En effet, la piste n'est pas dotée d'un *Instrumental Landing System* permettant l'atterrissage aux instruments.

Le deuxième vol a été détecté avec une pvalue égale à 0.0089. C'est un exemple de *Go Around*, c'est à dire qu'au cours de l'atterrissage le pilote a ré-initié la phase de descente car les conditions optimales n'étaient pas réunies. Dans le cas de ce vol, le pilote a redécollé après avoir touché la piste. This abnormality can also be seen in the other parameters we studied.

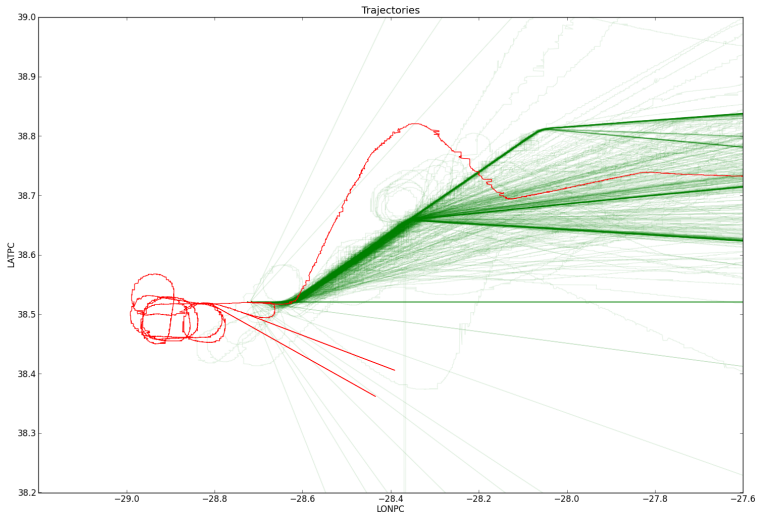


Figure 9.11: Trajectoire du Vol 1

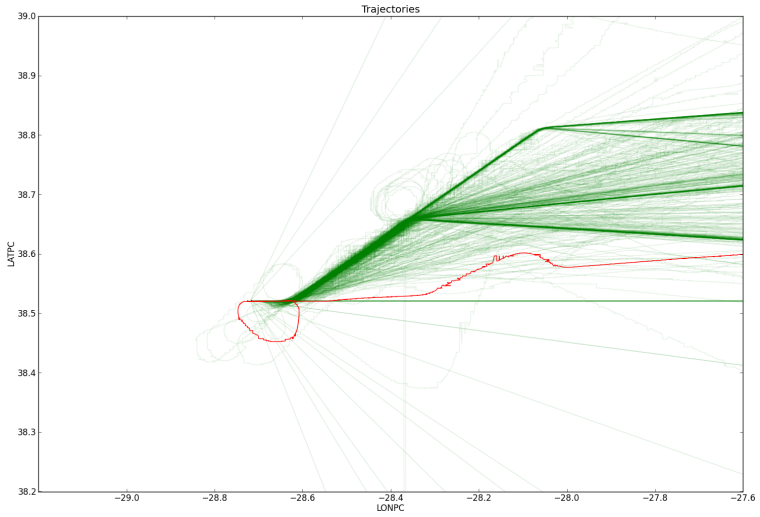


Figure 9.12: Trajectoire du Vol 2

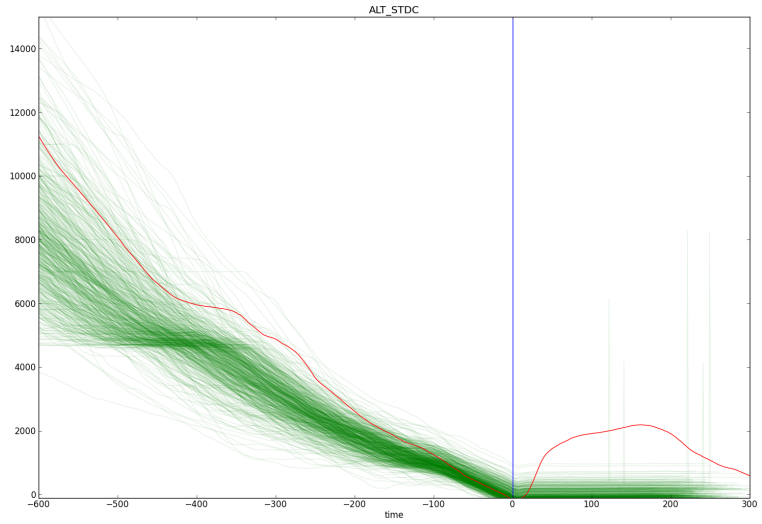


Figure 9.13: Altitude du Vol 2

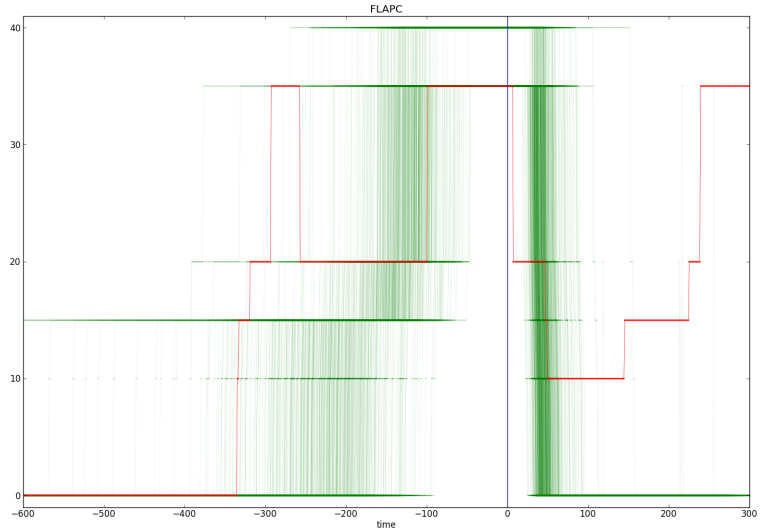


Figure 9.14: Volets du Vol 2

References

A. Aizerman, E.M. Braverman, and L.I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964. 12

B.G. Amidan and T.A. Ferryman. Atypical event and typical pattern detection within complex systems. In *Aerospace Conference, 2005 IEEE*, pages 3620–3631. IEEE, 2005. 53, 78, 82, 148

T.W. Anderson. *An introduction to multivariate statistical analysis*. Wiley Eastern Private Limited; New Delhi, 1954. 78

N. Aronszajn. *Theory of reproducing kernels*. Harvard University, 1951. 13, 25

F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004. 84, 95, 97, 159

Y. Bengio. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009. 141

C. Berg, J.P.R. Christensen, and P. Ressel. Harmonic analysis on semigroups. 1984. 72, 74, 75, 111, 157

G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2):259–294, 2007. 26, 27, 29, 153

Boeing. Statistical summary of commercial jet airplanes accidents, worldwide operations, 1959-2012. http://www.boeing.com/news/techissues/pdf/statsum.pdf, 2013. xiv, 1, 2, 145

B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. 10

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. *Advanced Lectures on Machine Learning*, pages 169–207, 2004. 14

M.L. Braun, J.M. Buhmann, and K.R. Müller. On relevant dimensions in kernel feature spaces. *The Journal of Machine Learning Research*, 9:1875–1908, 2008. 29

P. Bühlmann and S. Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer, 2011. 78

D.A. Clifton, L. Tarassenko, C. Sage, and S. Sundaram. Condition monitoring of manufacturing processes. *Proceedings of condition monitoring 2008*, pages 273–279, 2008. 21

D.A. Clifton, S. Hugueny, and L. Tarassenko. Novelty detection with multivariate extreme value statistics. *Journal of Signal Processing Systems*, pages 1–19, 2010. 21

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995. 10

M. Cuturi. Positive definite kernels in machine learning. *arXiv preprint arXiv:0911.5367*, 2009. 72

M. Cuturi and A. Doucet. Autoregressive kernels for time series. *arXiv preprint arXiv:1101.0673*, 2011. 124

M. Cuturi, J-P Vert, O. Birkenes, and T. Matsui. A kernel for time series based on global alignments. In *Acoustics, Speech and Signal Processing, 2007. ICASSP*

2007. IEEE International Conference on, volume 2, pages II–413. IEEE, 2007. 103, 108, 124

S. Das, K. Bhaduri, N.C. Oza, and A.N. Srivastava. nu-anomica: A fast support vector based novelty detection technique. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 101–109. IEEE, 2009. 84

S. Das, B.L. Matthews, A.N. Srivastava, and N.C. Oza. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 47–56. ACM, 2010. 78, 82, 95, 148, 154

EUROCAE. ED112 - Minimum operational performance specification for crash protected airborne recorder system, 2003. 3

G.E. Fasshauer. Positive definite kernels: past, present and future. *Dolomite Research Notes on Approximation*, 4:21–63, 2011. 89

Flight Safety Foundation, European Region Airline Association, and EUROCONTROL. Go-around safety forum, brussels 2013: Findings and conclusions. http://www.skybrary.aero/bookshelf/books/2325.pdf, 2013. 57

C. Forbes, M. Evans, N. Hastings, and B. Peacock. *Statistical distributions*. John Wiley & Sons, 2011. 87, 89

E. Garnier de Labareyre and T. Donadey. Reducing rate of false fdm events. https://easa.europa.eu/essi/ecast/wp-content/uploads/2013/01/EOFDM_CONFERENCE_2013_LABAREYREDONADEY_False_FDM_events.pdf, 2013. 18

M. Girolami. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 14(3):669–688, 2002. 31, 153

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*, volume 2. Springer, 2009. 141, 148

D. Haussler. Convolution kernels on discrete structures. Technical report, Technical report, Department of Computer Science, University of California at Santa Cruz, 1999. 71, 72, 73, 75, 95, 96, 160

D.M. Hawkins. *Identification of outliers*, volume 11. Chapman and Hall London, 1980. 21, 151

H. Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3): 863–874, 2007. 24, 40, 42

S.A. Hofmeyr and S. Forrest. Architecture for an artificial immune system. *Evolutionary computation*, 8(4):443–473, 2000. 21

P. Honeine and C. Richard. A closed-form solution for the pre-image problem in kernel-based machines. *Journal of Signal Processing Systems*, 40, 2010. 24, 42

R.A. Horn and C.R. Johnson. *Matrix analysis*. Cambridge university press, 2012. 96, 116, 160

P.J. Huber. *Robust statistics*, volume 1. Wiley Online Library, 1981. 86

S.R. Jammalamadaka and A. Sengupta. *Topics in circular statistics*, volume 5. World Scientific Publishing Company Incorporated, 2001. 16, 78, 91

R. Jenssen. Kernel Entropy Component Analysis. *IEEE transactions on pattern analysis and machine intelligence*, pages 847–860, 2009. 23, 25, 30, 36, 39, 153

R. Jenssen. Kernel entropy component analysis: New theory and semi-supervised learning. In *2011 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2011. 30, 38, 40

R.I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002. 95

Q. Le, M.A. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In *International Conference in Machine Learning*, 2012. 141

H. Lei and B. Sun. A study on the dynamic time warping in kernel machines. In *Signal-Image Technologies and Internet-Based System, 2007. SITIS'07. Third International IEEE Conference on*, pages 839–845. IEEE, 2007. 74

C.S. Leslie, E. Eskin, and W.S. Noble. The spectrum kernel: A string kernel for svm protein classification. In *Pacific symposium on biocomputing*, volume 7, pages 566–575. World Scientific, 2002. 103

V.I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966. 102, 161

Q. Li and J. Racine. Nonparametric estimation of distributions with categorical and continuous data. *journal of multivariate analysis*, 86(2):266–292, 2003. 78

J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003. 56, 83

M. Markou and S. Singh. Novelty detection: a review–part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, 2003a. 22, 151

M. Markou and S. Singh. Novelty detection: a review–part 2: neural network based approaches. *Signal Processing*, 83(12):2499–2521, 2003b. 22, 151

H. Mendes. *Study of Mathematical Algorithms to Identify Abnormal Patterns in Aircraft Flight Data*. PhD thesis, Universidade Técnica de Lisboa, 2012. 3

E.A. Nadaraya. On non-parametric estimates of density functions and regression curves. *Theory of Probability & Its Applications*, 10(1):186–190, 1965. 32

NASA. Dahshlink, a web-based collaboration tool for those interested in data mining and systems health. https://c3.nasa.gov/dashlink/, 2014. 56

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*, page 81. ACM, 2004. 30

E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962. 30, 153

M. Paterson and V. Dančík. *Longest common subsequences*. Springer, 1994. 84

J.C. Principe. *Information Theoretic Learning: Renyi's entropy and kernel perspectives*. Springer Verlag, 2010. 23, 152

A. Rényi. On measures of entropy and information. In *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, pages 547–561, 1961. 23, 35, 152

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978. 103, 107

B. Scholkopf. The kernel trick for distances. *Advances in neural information processing systems*, pages 301–307, 2001. 73

B. Schölkopf and A.J. Smola. *Learning with kernels*. MIT Press, 2002. 111, 148

B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K.R. Müller. Kernel PCA pattern reconstruction via approximate pre-images. 1998a. 42

B. Schölkopf, A. Smola, and K.R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998b. 13

B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12(3):582–588, 2000. 45

B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001. 21, 47, 151

C.D. Scott and R.D. Nowak. Learning minimum volume sets. *The Journal of Machine Learning Research*, 7:665–704, 2006. 23, 152

J. Shawe-Taylor, C.K.I. Williams, N. Cristianini, and J. Kandola. On the eigenspectrum of the Gram matrix and the generalization error of kernel PCA. *Information Theory, IEEE Transactions on*, 51(7):2510–2522, 2005. 29

T. Shi, M. Belkin, and B. Yu. Data spectroscopy: Eigenspaces of convolution operators and clustering. *Annals of Statistics*, 37(6B):3960–3984, 2009. 38, 40

K. Shin and T. Kuboyama. A generalization of haussler's convolution kernel: mapping kernel. In *Proceedings of the 25th international conference on Machine learning*, pages 944–951. ACM, 2008. 95

Kilho Shin. A new frontier of kernel design for structured data. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 401–409, 2013. 160

B.W. Silverman. Weak and strong uniform consistency of the kernel estimate of a density and its derivatives. *The Annals of Statistics*, 6(1):177–184, 1978. 32

E. Smart, D. Brown, and J. Denman. Combining multiple classifiers to quantitatively rank the impact of abnormalities in flight data. *Applied Soft Computing*, 12(8):2583–2592, 2012. 78

B. Sofman, J.A. Bagnell, and A. Stentz. Anytime online novelty detection for vehicle safeguarding. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1247–1254. IEEE, 2009. 21

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006. 95

I. Steinwart, D. Hush, and C. Scovel. An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels. *Information Theory, IEEE Transactions on*, 52(10):4635–4643, 2006. 13, 31

L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady. Novelty detection for the identification of masses in mammograms. In *Artificial Neural Networks, 1995., Fourth International Conference on*, pages 442–447. IET, 1995. 21

D.M.J. Tax and R.P.W. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004. 22, 23, 47, 151

J-P Vert. The optimal assignment kernel is not positive definite. *arXiv preprint arXiv:0801.4061*, 2008. 85, 103, 107

D.H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996. 141

D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997. 141

M. Wu and J. Ye. A small sphere and large margin approach for novelty detection using training data with outliers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(11):2088–2092, 2009. 22

Z. Zhang, G. Wang, D-Y Yeung, and J.T. Kwok. Probabilistic kernel principal component analysis. Technical report, Citeseer, 2004. 42

Nicolas CHRYSANTHOS

Doctorat : Optimisation et Sûreté des Systèmes

Année 2014

Méthodes à noyau pour l'analyse de données de vols appliquée aux opérations aériennes

L'analyse de données de vols appliquée aux opérations aériennes ou "Flight Data Monitoring" (FDM), est le processus par lequel une compagnie aérienne recueille, analyse et traite de façon régulière les données enregistrées dans les avions, dans le but d'améliorer de façon globale la sécurité.
L'objectif de cette thèse est d'élaborer dans le cadre des méthodes à noyau, des techniques pour la détection des vols atypiques qui présentent potentiellement des problèmes qui ne peuvent être trouvés en utilisant les méthodes classiques.
Dans la première partie, nous proposons une nouvelle méthode pour la détection d'anomalies. Nous utilisons une nouvelle technique de réduction de dimension appelée analyse en entropie principale par noyau afin de concevoir une méthode qui est à la fois non supervisée et robuste.
Dans la deuxième partie, nous résolvons le problème de la structure des données dans le domaine FDM. Tout d'abord, nous étendons la méthode pour prendre en compte les paramètres de différents types tels que continus, discrets ou angulaires.
Ensuite, nous explorons des techniques permettant de prendre en compte l'aspect temporel des vols et proposons un nouveau noyau dans la famille des techniques de déformation de temps dynamique, et démontrons qu'il est plus rapide à calculer que les techniques concurrentes et est de plus défini positif.
Nous illustrons notre approche avec des résultats prometteurs sur des données réelles des compagnies aériennes comprenant plusieurs centaines de vols.

Mots clés : noyaux (analyse fonctionnelle) -analyse discriminante - information, théorie de l' - structures de données – aéronautique, mesures de sécurité.

Kernel Methods for Flight Data Monitoring

Flight Data Monitoring (FDM), is the process by which an airline routinely collects, processes, and analyses the data recorded in aircrafts with the goal of improving the overall safety or operational efficiency.
The goal of this thesis is to investigate machine learning methods, and in particular kernel methods, for the detection of atypical flights that may present problems that cannot be found using traditional methods. Atypical flights may present safety of operational issues and thus need to be studied by an FDM expert.
In the first part we propose a novel method for anomaly detection that is suited to the constraints of the field of FDM. We rely on a novel dimensionality reduction technique called kernel entropy component analysis to design a method which is both unsupervised and robust.
In the second part we solve the most salient issue regarding the field of FDM, which is how the data is structured. Firstly, we extend the method to take into account parameters of diverse types such as continuous, discrete or angular.
Secondly, we explore techniques to take into account the temporal aspect of flights and propose a new kernel in the family of dynamic time warping techniques, and demonstrate that it is faster to compute than competing techniques and is positive definite.
We illustrate our approach with promising results on real world datasets from airlines comprising hundreds of flights.

Keywords: kernel functions - discriminant analysis - information theory -data structures (computer science) - aeronautics, safety measures.

Thèse réalisée en partenariat entre :

SAFRAN Sagem

utt université de technologie Troyes

Ecole Doctorale "Sciences et Technologies"