EXPECTED SIMILARITY ESTIMATION
FOR LARGE-SCALE ANOMALY DETECTION

Dissertation zur Erlangung des Doktorgrades
Dr. rer. nat.
der Fakultät für Ingenieurwissenschaften, Informatik und Psychologie
der Universität Ulm

von Markus Schneider aus Wangen im Allgäu | 2016

Universität Ulm
Institut für Neuroinformatik

Colophon

This document was typeset in L^AT_EX
based on Edward Tufte's *Beautiful Evidence*
using the Garamond typeface and the Euler fonts for math.

Markus Schneider (2016) *Expected Similarity Estimation for Large-Scale
Anomaly Detection*, PhD dissertation

Last revision: January 17, 2017

Amtierender Dekan: Prof. Dr. Frank Kargl

Gutachter: Prof. Dr. Günther Palm

Gutachter: PD Dr. Friedhelm Schwenker

Gutachter: Prof. Dr. Wolfgang Ertel

Tag der Promotion: 21. Dezember 2016

Abstract

Anomalies are patterns in data or events which are unlikely to appear under normal conditions. It is of central interest to detect such anomalous instances to prevent damage or to extract valuable information from data. While statistics and machine learning developed several excellent key techniques to perform anomaly detection, most of them suffer poor algorithmic scalability when applied to large-scale datasets since the computational complexity and memory requirements become the limiting factor of these algorithms.

This dissertation makes several contributions to the problem of large-scale anomaly detection centered on a novel method we introduce named `EXPPOSE` which estimates the similarity between a new unseen observation and the distribution data under normal conditions. That way `EXPPOSE` measures the likelihood for a new observation to be anomalous. Its core is based on the kernel embedding of distributions which maps a probability measure into a reproducing kernel Hilbert space where it can be manipulated efficiently. The kernel embedding representation requires no parametric assumptions or explicit description of the probability measure. This constitutes an important advantage since the distributions of normal and anomalous instances are in general unknown.

The main contributions of this work are efficient algorithms to train and evaluate the `EXPPOSE` anomaly detector. This can be achieved with computational complexities and memory requirements independent of the dataset size which is the key to solve large-scale machine learning problems. The dependence on the reproducing kernel function as a similarity measure enables the application to many domains and introduces the possibility to incorporate domain and expert knowledge into the modeling process.

The key technologies are further developed to online and streaming anomaly detection where instances arrive in a possible infinite sequence of observations. A crucial requirement in these applications is the ability to make predictions as data arrive based on the information obtained from previous observations. One of the major challenges is the non-stationary nature of streams in which our understanding of what is normal and anomalous change over time. This introduces the necessity to adapt to such changes *e.g.* by forgetting outdated information while incorporating new knowledge.

The simplicity of the proposed methodologies facilitates a theoretical analysis to provide guarantees in terms of convergence rates and probabilistic bounds.

To my family

Contents

I Prologue

- 1 Introduction 3

II Large-Scale Anomaly Detection

- 2 Foundations of Anomaly Detection 13
- 3 Anomaly Detection Algorithms 25
- 4 Hilbert Spaces & Kernel Functions 51
- 5 Expected Similarity Estimation 61
- 6 Convex Expected Similarity Estimation 87
- 7 Online & Streaming Anomaly Detection 103
- 8 Concentration Inequalities 117
- 9 Distance-Based Feature Maps 131

III Epilogue

- 10 Conclusion 139

IV Appendix

- A Probability Theory 145
- B Functional Analysis 149
- C Convex Analysis 153
- Bibliography 155
- Index 173

Nomenclature

Anomaly Detection

C_A	class of anomalous data (def. on page 15.)
C_N	class of normal data (def. on page 15.)
η	classifier or predictor (def. on page 16.)
$\hat{\Psi}$	an estimator (def. on page 17.)
Θ	parameter space (def. on page 16.)
Ψ	hypothesis space (def. on page 16.)
$(\Omega, \mathcal{F}, \mathbb{P})$	underlying source of events (def. on page 15.)
$(\mathcal{Q}, \mathcal{D})$	prediction or decision space (def. on page 15.)
$(\mathcal{X}, \mathcal{X})$	input space (def. on page 15.)
$(\mathcal{Y}, \mathcal{Y})$	label or space (def. on page 15.)
$(\mathcal{Z}, \mathcal{Z})$	sample space (def. on page 17.)
$X = x$	x is the realization of X (def. on page 15.)
$Z = z$	training set z sampled from Z (def. on page 17.)

Probability

$\nu \ll \mu$	absolute continuous measure with respect to μ (def. on page 147.)
$\mathbb{B}$	Borel σ -algebra (def. on page 145.)
δ_ω	Dirac measure (def. on page 147.)
$\mathbb{E}[X]$	expectation of X (def. on page 148.)
$\mathbb{E}[X \mid \mathcal{G}]$	conditional expectation of X with respect to $\mathcal{G}$ (def. on page 148.)
$f \sim g$	f and g are equivalent functions (def. on page 146.)
$[f]$	equivalence class of f (def. on page 146.)
$\mathbb{I}_A$	indicator function (def. on page 147.)
a. e.	almost everywhere (def. on page 146.)
a. s.	almost surely (def. on page 146.)
$\mathcal{B}(\mathcal{X})$	set of all finite Borel measures on $\mathcal{X}$ (def. on page 62.)
$\mathcal{B}^1(\mathcal{X})$	set of all Borel probability measures on $\mathcal{X}$ (def. on page 62.)
$\otimes_{i=1}^n \mathcal{F}_i$	$\sigma(\times_{i=1}^n \mathcal{F}_i)$, given the σ -algebras $\mathcal{F}_i$ (def. on page 145.)
$\sigma(S)$	smallest σ -algebra containing S (def. on page 145.)
$\text{supp } \mu$	support of measure μ (def. on page 146.)

Functional Analysis

$B(p, \epsilon)$	open ball about p with radius ϵ (def. on page 150.)
$C(X, Y)$	set of continuous functions $f: X \rightarrow Y$ (def. on page 151.)
$\langle f, x \rangle_*$	the dual map (def. on page 152.)
$\mathcal{F}^*$	the dual space of $\mathcal{F}$ (def. on page 152.)
$\text{Hom}(\mathcal{F}, \mathcal{G})$	set of all linear operators from $\mathcal{F}$ to $\mathcal{G}$ (def. on page 152.)
$\langle \cdot, \cdot \rangle_{\mathcal{F}}$	inner product on $\mathcal{F}$ (def. on page 152.)
ι_A	see $A \hookrightarrow B$ (def. on page 150.)
$A \hookrightarrow B$	inclusion map (def. on page 150.)
$\mathcal{L}(\mathcal{F}, \mathcal{G})$	set of all continuous linear operators from $\mathcal{F}$ to $\mathcal{G}$ (def. on page 152.)
$\ \cdot \ _{\mathcal{F}}$	norm on $\mathcal{F}$ (def. on page 151.)
$\ \cdot \ _{\mathcal{L}^p}$	norm with respect to $\mathcal{L}^p$ (def. on page 146.)
$\ \cdot \ _{\infty}$	supremum or uniform norm (def. on page 152.)
$\Pi_H(w)$	projection of w onto H (def. on page 89.)
S_k	integral operator induces by the kernel k (def. on page 55.)
L^p	space of equivalence classes of $\mathcal{L}^p$ (def. on page 146.)
ℓ^2	space of square summable sequences (def. on page 51.)
$\mathcal{L}^p$	space of all p integrable functions (def. on page 146.)
$\mathcal{H}_1 \otimes \mathcal{H}_2$	tensor product of $\mathcal{H}_1$ with $\mathcal{H}_2$ (def. on page 58.)
$\overline{U}$	closure of U (def. on page 150.)
$\overset{\circ}{U}$	interior of U (def. on page 150.)
$w_t \rightharpoonup w$	w_t converges weakly to w (def. on page 91.)

Miscellaneous

$\hat{i}$	$\sqrt{-1}$ (def. on page 56.)
-----------	--------------------------------

Part I

PROLOGUE

CHAPTER ONE

Introduction

SYNOPSIS · This chapter introduces the problem of anomaly detection and provides an overview of applications in which these techniques help to prevent financial or human damage. We discuss the properties of large-scale machine learning and conclude with the contributions and outline of this thesis followed by a list of previously published material.

1.1 *Anomaly Detection*

What is an anomaly? The dictionary defines an anomaly as something that deviates from the common rule; that is abnormal or does not fit in. The machine learning and data mining literature also provides various characterizations as in the following excerpts:

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. These non-conforming patterns are often referred to as anomalies [...]

— Chandola et al., *Anomaly detection: A survey*

An [anomaly] is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.

— Hawkins, *Identification of Outliers*

These interpretations reflect our intuition that anomaly detection always requires a *context* which determines what is considered to be *normal*. It is therefore arguable if normality is an absolute concept or rather relative. At this point we provide the following probabilistic definition of “anomaly”.

DEFINITION 1.1. An *anomaly* is an event or a pattern which is unlikely to be observed under *normal* conditions represented by the probability distribution $\mathbb{P}_X$.

This definition explicitly declares that anomalies are not necessary impossible events, but unlikely under normal conditions. However, it is unrealistic to assume that we know the actual probability distribution $\mathbb{P}_X$, which fully describes the state of normality. It is therefore necessary to resort to a set of observations or samples from this distribution to estimate the likelihood of an instance or event. These samples should represent typical observations under normal conditions. We then identify anomalies as instances whose properties differ significantly from the majority of other elements in this set. The necessity of a *significant* deviation acknowledges the fact that normal instances can of course be different from each other, but only to a certain

a·nom·a·ly *noun*
[əˈnɒməli]
(*plural*) -lies

“Everyone loves a mystery, and at the heart of it, that’s what anomaly detection is — spotting the unusual, catching the fraud, discovering the strange activity. Anomaly detection has a wide range of useful applications, from banking security to natural sciences to medicine to marketing. Anomaly detection carried out by a machine-learning program is actually a form of artificial intelligence.”
— Dunning et al., *Practical Machine Learning: A New Look at Anomaly Detection*.

degree. Moreover, since the set can also contain unlikely events (according to the distribution $\mathbb{P}_X$) we require only a deviation from the *majority* of other elements in order to qualify an instance as anomalous.

THE CHALLENGE in detecting anomalies is that anomalous instances or events are by definition *rare*. Even if we had access to a set of observed anomalies it would be unreasonable to assume that these are representative for all types of anomalous events that might happen. A typical example is air traffic safety where (fortunately) only relatively few accidents occur. Hence the sensor recordings of these few events are not sufficient to build a statistical model to detect all possible failures, malfunctions and defects of future flights. Consequently it is the objective of statistics and machine learning to extract the concept of normality from a set of observations and use this knowledge to derive a classifier or predictor which can distinguish between normal and anomalous elements or events. These algorithms can be characterized based on how this concept of normality is modeled and how a deviation from the norm is defined.

IN THE FOLLOWING example we collect global temperature records to identify unusual phenomena.¹ A weather anomaly can be defined as a significant deviation from the average temperature. We would then discover an unusual rise of the globally averaged temperatures caused by El Niño in 1998.

¹ MSU/AMSU data are produced by Remote Sensing Systems. Data are available at www.remss.com.

This example also illustrates that an anomaly is not necessarily an impossible event possibly due to measurement or processing errors, but an unusual extreme and nevertheless *real* deviation.

The average temperature is a relative simple model and raises several questions: How much deviation from the norm is considered to be *significant*? Is it reasonable to use the average over all measurements or would it better to inspect shorter time frames to compensate for natural temperature drifts? Can we obtain additional features *e.g.* the rate of temperature changes, which could improve our model? In a subsequent chapter we review various anomaly detection techniques and discuss how individual approaches respond to such questions. All these algorithms have in common that they estimate (implicitly or explicitly) how normal instances are distributed. Furthermore, they all classify a new observation based on the distance or proximity to this estimate.

After the detection of an anomaly or an unlikely event it is necessary to initiate an appropriate reaction or counteraction which depends on the anomaly detection application. An overview of the most common applications is provided in the following section.

1.2 *Anomaly Detection Applications*

Anomaly detection has a long history with early statistical studies in the 19th century.² Today, anomaly detection is applied to various fields with different requirements and demands. Typical applications are network intrusion detection, credit card fraud detection, medical diagnosis and failure detection in industrial environments.

THE Center for Strategic and International Studies estimates that the United States, China, Japan and Germany incur a total loss of about \$200 billion per year due to cybercrime and economic espionage.³ The global costs of cybercrime are estimated to be between \$375 and \$575 billion every year. To defeat the increasing number of attacks on computer-based networks, traditional intrusion detection methods which are based on knowledge provided by experts can be complemented or replaced by systems which automatically detect unusual network behavior.⁴ Network intrusion detection is especially challenging since the pattern of both normal traffic and attacks can change over time and hence requires algorithms with the ability to continuously adapt to such changes.

Credit card transactions which significantly differ from the usual shopping behavior of the card owner can indicate that the credit card was stolen or a compromise of data associated with the account occurred.⁵ Here, unusual behavior often refers to purchases with abnormal prices, uncommon goods and services or suspicious geographical locations of the credit card use. Similar techniques can be applied to detect other frauds such as identity theft, insider trading and insurance frauds. All of these are time-critical applications of anomaly detection since a delayed counteraction can cause significant damages.

In medical diagnosis, human professionals can be supported by automated systems to detect breast cancer in mammographic image analysis⁶ or the automatic segmentation of brain tumors from MR images using anomaly detection techniques.⁷ Other applications in this domain focus on the detection of disease outbreaks where seasonal variation in the data additionally complicates the analysis.

Unplanned downtime of production lines caused by failing components is a serious concern in many industrial environments. In such situations it is possible to utilize anomaly detection to identify unusual sensor readings and predict possible faults in advance which enables condition-based maintenance.⁸ One of the major challenges for this kind of application is to distinguish between real anomalies and sensor noise caused by background interferences.

Anomaly detection is not only used to prevent damage, but can also extract valuable information from data *e.g.* in astronomical surveys⁹ (SDSS). The aim of the SDSS project is to filter sky signals and find astronomical objects with unusual properties which may lead to new discoveries.

For a more detailed literature review of anomaly detection applications we refer to the work of Chandola et al.¹⁰ and continue with a discussion of large-scale anomaly detection and the limitations of existing techniques.

² Edgeworth, *On discordant observations*, (1887).

³ McAfee, *Net Losses: Estimating the Global Cost of Cybercrime*.

⁴ Kumar, *Parallel and distributed computing for cybersecurity*.

⁵ Aleskerov et al., *Cardwatch: A neural network based database mining system for credit card fraud detection*.

⁶ Spence et al., *Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model*.

⁷ Prastawa et al., *A brain tumor segmentation framework based on outlier detection*.

⁸ Zhang et al., *A probabilistic fault detection approach: application to bearing fault detection*.

⁹ Kent, *Sloan digital sky survey*.

¹⁰ Chandola et al., *Anomaly detection: A survey*.

1.3 *Large-Scale Problems*

The focus of this work is anomaly detection in the context of large-scale applications and the aim of this section is to discuss the importance of this topic and provide some background information.

THE AMOUNT OF DATA which has to be processed constantly rises, but where does it come from? One provider is the internet whose content developed from text and images to videos, now producing enormous amounts of information. Also data gathering is ubiquitous in a world where sensors are cheap and every mobile device is essentially a computer. It is reasonable to store this data since it can contain valuable information which only has to be found.

WHAT are *large-scale* machine learning applications? Jacobs aims to provide a time and technologically independent description (of big data) as

[...] data whose size forces us to look beyond the tried-and-true methods that are prevalent at that time.

— Jacobs, *The pathologies of big data*

The application of machine learning and data mining techniques to large-scale problems is still challenging. For example, most distance-based approaches, such as the *local outlier factor*, *nearest neighbor data description* or *distance-based outlier detection* rely on fast nearest neighbor lookups. Although storing data in an appropriate tree structure accelerates this lookup when operating in low dimensions, the benefit of a search tree vanishes as the dimension grows. Notably, it can be shown that a nearest neighbor lookup in high dimensions is not better than an exhaustive search with linear time complexity unless a data storage is used which grows exponentially in size.¹¹ However, even when operating in low dimensions, most distance-based techniques require storing the entire dataset which is often already impossible on its own. Direct density estimation as performed by statistical models such as the *Gaussian mixture model* or the non-parametric *kernel density estimator* is notoriously challenging for high-dimensional spaces. In general, the sample size must increase exponentially with the dimensionality of the space for a reliable statistical performance. The *one-class support vector machine*, which works well in high-dimensional applications, does not scale with the number of instances in the dataset. In particular, not only the cubic time complexity to train the support vector machine is problematic, but also its dependence on the number of support vectors which can be shown to increase linearly with the dataset size.¹²

We see that the computational complexity and memory requirements of classical algorithms are the limiting factor when both the dataset sizes and the data dimensions grow. Moreover, not only the volume is increasing, but also the velocity at which data is generated, stored and required to be processed constantly rises. This forces the development of more efficient algorithms which can detect hidden value in data before its worth expires or uncover potential threats immediately.

“What is ‘big data’ anyway? Gigabytes? Terabytes? Petabytes?”

— Jacobs, *The pathologies of big data*.

“From 2005 to 2020, the digital universe [a measure of all the digital data created, replicated, and consumed in a single year] will grow by a factor of 300, from 130 exabytes to 40 000 exabytes, or 40 trillion gigabytes (more than 5200 gigabytes for every man, woman, and child in 2020). From now until 2020, the digital universe will about double every two years.”

— Reinsel et al., *The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East*.

¹¹ Goodman et al., *Handbook of discrete and computational geometry*, Chapter 39: Nearest neighbors in high-dimensional spaces.

¹² Steinwart, *Sparseness of support vector machines*.

THERE ARE certain requirements for large-scale machine learning techniques from which we identify the following key properties for algorithms of this problem class.

- A (close to) linear training computational complexity. Ideally the algorithm is completely independent of the dataset size and input dimension.
- It should be possible to parallelize computations without a loss of prediction quality to take advantage of modern hardware and large computing clusters.
- Once a classifier is learned, it should be possible to make fast predictions in *constant* time, independent of the original training dataset size.
- It should be possible to learn online (incremental) such that information can be extracted from real-time data streams. This requires a constant computational complexity per instance for both the learning operations and predictions.

“Efficient and effective tools and analysis methods for dealing with the ever-growing amount of data in different applications and fields are of paramount need.”

— Gama et al., *A survey on concept drift adaptation*.

1.4 Contributions

This dissertation presents a novel algorithm called *EXpected Similarity Estimation* (EXPOSE) for anomaly detection on very large datasets and data streams. In summary, the proposed approach achieves the same predictive performance as common anomaly detection algorithms while being significant faster than techniques with the same discriminative power.

WE show that the EXPOSE classifier can be estimated from a training set within a predefined accuracy in *constant* time. Moreover, we show that the memory requirements and the time per prediction are constant — independent of the sample size. We argue that such constant time algorithms are key to solving large-scale anomaly detection problems. In addition, all derivations are accompanied by a theoretical analysis and we provide convergence rates and probabilistic bounds for various quantities of interest.

THE EFFECTIVENESS of the proposed method is demonstrated by a series of benchmarks comparing leading and established anomaly detection techniques.

1.5 *Organization*

This chapter described the theme of this dissertation. We introduced the problem of anomaly detection and glanced at the challenges caused by large-scale applications.

Chapter 2 discusses the lack of label availability and the resulting requirement for unsupervised or semi-supervised machine learning algorithms. Furthermore, it introduces the theoretic foundations of anomaly detection which are essential for subsequent chapters. The final part of this chapter explains how statistical hypothesis testing can be applied to evaluate and compare anomaly detection algorithms.

Chapter 3 comprises a review of various anomaly detection techniques with focus on *scalability*, *robustness* and *configurability*. The strengths and weaknesses of each algorithm are analyzed which reveals the necessity for more efficient anomaly detection algorithms.

Chapter 4 reviews some elementary reproducing kernel Hilbert spaces theory and their properties, notably the theorems due to Mercer and Bochner. In this context, the reproducing kernel function is of central interest in its role as a similarity measure.

Chapter 5 derives the EXPOSE anomaly detector and reviews the kernel embedding of distributions as an integral part of the proposed method. It shows how EXPOSE can efficiently be estimated from samples and be processed in parallel or distributed environments. It includes benchmarking results comparing EXPOSE to several anomaly detection algorithms on multiple datasets and validates the scalability claims experimentally.

Chapter 6 reformulates EXPOSE as a stochastic optimization problem. This yields an algorithm which is able to approximate the EXPOSE model within a predefined accuracy in *constant* time, independent of the training dataset size.

Chapter 7 extends the EXPOSE classifier to online and streaming anomaly detection problems. The primary focus lies on efficient techniques for concept drift adaptation which requires to learn new concepts while forgetting outdated information. The chapter includes an experimental evaluation of several streaming anomaly detectors.

Chapter 8 deals with probabilistic bounds in form of concentration inequalities for the empirical kernel embedding of distributions. The problem of sampling without replacement is studied in the context of random variables in separable Hilbert spaces.

Chapter 9 presents an approximate feature map construction for kernel functions which are based on a distance metric.

The dissertation concludes with *Chapter 10*, which summarizes its contributions to the problem of large-scale anomaly detection. The appendix comprises important notions on functional analysis, measure theory and convex analysis.

1.6 *Previously Published Material*

This dissertation contains material previously published in the following papers, written with Wolfgang Ertel, Fabio Ramos and Günther Palm.

- M. Schneider, W. Ertel, and G. Palm (2015a). IJCNN 2015
 “Expected Similarity Estimation for Large-Scale Anomaly Detection”. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2015)*. IEEE, pp. 1–8
- M. Schneider, W. Ertel, and G. Palm (2015b). KI 2015
 “Kernel Feature Maps from Arbitrary Distance Metrics”. In: *KI 2015: Advances in Artificial Intelligence*. Ed. by S. Hölldobler et al. Springer International Publishing, pp. 137–150
- M. Schneider (2016). AISTATS 2016
 “Probability Inequalities for Kernel Embeddings in Sampling without Replacement”. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS 2016)*, pp. 66–74
- M. Schneider, W. Ertel, and F. Ramos (2016a). MLJ 2016
 “Expected Similarity Estimation for Large-Scale Batch and Streaming Anomaly Detection”. In: *Machine Learning* 105.3, pp. 305–333
- M. Schneider, W. Ertel, and F. Ramos (2016b). ICML 2016
 “Kernel Embeddings for Large-Scale Anomaly Detection”. In: *International Conference on Machine Learning (ICML 2016): Anomaly Detection Workshop*
- M. Schneider, W. Ertel, and G. Palm (2016). ECAI 2016
 “Constant Time EXpected Similarity Estimation for Large-Scale Anomaly Detection”. In: *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*. IOS Press, pp. 12–20

Part II

LARGE-SCALE ANOMALY
DETECTION

CHAPTER TWO

Foundations of Anomaly Detection

SYNOPSIS · In this chapter we argue that anomaly detection is primarily an unsupervised or semi-supervised machine learning problem caused by the lack of labeled data. Furthermore, we introduce the theoretic foundations of anomaly detection which are essential for subsequent chapters. Most definitions are inspired by statistical decision theory *i.e.* decision making in the presence of statistical knowledge. The main objective of this chapter is the introduction of related definitions from a probabilistic perspective to facilitate a theoretical analysis. We formally define the process of *learning* as the estimation of a classifier from a training set and explain the differences between *batch*, *online* and *streaming* anomaly detection. This chapter concludes with a review of statistical hypothesis testing and how these methods can be applied to evaluate and compare anomaly detection algorithms. The introduced methodology will be used throughout the thesis in all experimental evaluations.

2.1 *Terminology*

Typical machine learning techniques require a labeled dataset to learn a predictor (classifier) which can distinguish between individual classes. A dataset is said to be *labeled* if it has a class label assigned to each instance. However, the creation of a representative labeled dataset for all types of anomalies is often too expensive or even impossible. The former situation occurs when the labeling has to be done by a human expert or is obtained through costly experiments.¹ In some applications anomalies are also very rare as mentioned before *e.g.* in air traffic safety or space missions. Hence, anomaly detection is different from classical supervised machine learning with labeled data. Consequently, literature distinguishes various forms of anomaly detection from completely supervised to unsupervised.²

¹ Hodge et al., *A survey of outlier detection methodologies*.

² Aggarwal, *Outlier analysis*; Chandola et al., *Anomaly detection: A survey*.

2.1.1 *Supervised Anomaly Detection*

Supervised anomaly detection is similar to the classical machine learning task of classification. Given a training set which contains labeled instances for the normal and anomaly classes, these algorithms build (learn) a predictor or model to distinguish between those classes. This model is then used to make predictions for new instances which classifies them as either normal or anomalous.

In contrast to the traditional classification task, where all classes are equally important, anomalies are typically underrepresented as they are far less frequent than normal instances. This problem is known as *class imbalance* and has to be considered appropriately.³ The bigger challenge however, is that there are several types (or different sources) of anomalies, all collected under the anomaly class label. This is problematic since learning a good classifier requires a representative number of samples from each plausible anomaly class. This requirement is often difficult to satisfy as argued before. Therefore, supervised anomaly detection techniques must be modified to cope with this limitation or other algorithms without such a deficit have to be designed.

³ Chawla et al., *Editorial: special issue on learning from imbalanced data sets*.

2.1.2 Unsupervised Anomaly Detection

In contrast to supervised approaches, unsupervised anomaly detection techniques do not resort to a training set to build a predictor, but try to directly identify anomalies in a given unlabeled dataset. For example many clustering algorithms can identify anomalies as a by-product of the cluster finding process. Since unsupervised anomaly detection methods do not yield a predictor or classifier we will refer to them as *outlier selection* algorithms.

The absence of class labels requires additional assumptions on the dataset. In unsupervised anomaly detection it is implicitly assumed that the dataset contains primarily normal instances and only very few anomalies. In general, outlier selection algorithms will not be able to deliver satisfactory results if this assumption is violated. Furthermore, there is no dedicated learning phase with the consequence that these techniques cannot afford to spend additional time and resources to create a good predictor beforehand as supervised algorithms can. The lack of this possibility makes their application to time-critical or high-frequency applications more difficult as it can be expected that outlier selection methods take more time to make adequate predictions.

2.1.3 Semi-Supervised Anomaly Detection

Semi-supervised anomaly detection algorithms are in between the previous two concepts. As in the unsupervised case we cannot resort to class labels, however a dedicated (training) dataset is used to create a predictor or classifier. Unlike outlier selection algorithms, this classifier can be used to make predictions for arbitrary observations and is not restricted to identify anomalies in a specific dataset. The requirement that the dataset contains only a few anomalies is only necessary for the training set, but not for the sets on which the predictor is applied. It is reasonable to assume that a training set with this property can be created as it should most often be possible to collection large amounts of data under normal conditions *e.g.* usual credit card transactions or ordinary network traffic. Semi-supervised anomaly detection is therefore widely applicable and we will see in the following that outlier selection is only a special case thereof.

IN THIS work we solely focus on *semi-supervised* anomaly detection using an unlabeled dataset in which normal instances are far more frequent than anomalies.

Figure 2.1: The difference between supervised classification (top) and anomaly detection (bottom). The former creates a decision boundary based on both classes: normal (•) and anomalous (▲). The query point (■) in the upper right corner would not be detected as an anomaly by a supervised classifier, even though this point is far from other normal observations. This is caused by the assumption that there are only two classes and consequently, the query point is more likely to be normal. In the same situation, an anomaly detection algorithm creates a decision boundary which is close to the normal training instances.

2.2 Theoretical Foundations

The aim of this section is to describe the problem of anomaly detection more formally through a statistical framework. In summary, *learning* is the estimation of an anomaly detector from a set of samples of the input space distribution. Such an anomaly detector is a mapping from the *input* to the *prediction* space. It is therefore natural to model the framework by measurable spaces and random variables.

The most important component in machine learning is *data* represented by instances of the input space $\mathcal{X}$. For the moment we make no restrictive assumptions about the input space other than it should be measurable. Consequently, this space can comprise data of any form *e.g.* simple numerical vectors, images, video data or more complicated structures. We assume that these instances are generated by an unknown underlying *source of events* modeled by the measure space $(\Omega, \mathcal{F}, \mathbb{P})$ and observable through the random variable X mapping from Ω to $\mathcal{X}$.

DEFINITION 2.1 (Input Space). The measurable space $(\mathcal{X}, \mathcal{X})$ containing all values that X might take is called *input space*. The realization after measurement of the random variable X is denoted by $X = x$.

We assume that there is a *true* (but unknown) probability distribution

$$\mathbb{P}_X: \mathcal{X} \rightarrow [0, 1]$$

of the data. Since the goal of a classifier is to separate the instances which are normal from anomalous elements, this distribution is of central importance. However, the true class assignment of an instance provided by the class *label* is also unknown to us.

DEFINITION 2.2 (Label Space). An observation $X = x$ can either belong to the class of normal data (C_N) or to the class of anomalies (C_A). This is called *label* of the instance and modeled through the random variable Y . The collection of all labels is denoted by the measurable space $(\mathcal{Y}, \mathcal{Y})$ called *label space* or *output space*.

The observation $X = x$ is stochastic, depending on the label Y and is therefore distributed according to $\mathbb{P}_{X|Y}$. We notice that both, the observations $X: (\Omega, \mathcal{F}) \rightarrow (\mathcal{X}, \mathcal{X})$ and the labels $Y: (\Omega, \mathcal{F}) \rightarrow (\mathcal{Y}, \mathcal{Y})$ are random variables on the underlying source of events $(\Omega, \mathcal{F}, \mathbb{P})$ as illustrated in Fig. 2.2.

As argued before, it is impossible to perform anomaly detection without additional assumptions as the true class labels are never observed. Here, we require that most observations are realizations from the distribution of normal data *i.e.* $Y = C_N$. Under this assumption we can use the input space distribution $\mathbb{P}_X$ as a proxy for the distribution of $X | Y = C_N$ (the instances which are normal). Once an estimate for this probability distribution is obtained, we can *predict* the likelihood of an element being normal.

DEFINITION 2.3 (Prediction and Decision Space). Based on the outcome $X = x$ of an observation, the objective of an anomaly detection algorithm is to make a prediction $\vartheta \in \mathcal{Q}$, where the measurable space $(\mathcal{Q}, \mathcal{Q})$ is called the *prediction space* or sometimes *decision space*.

The appendix contains a collection of definitions related to measure and probability theory. We also recommend Olav Kallenberg's *Foundations of modern probability*.

Figure 2.2: The source of events and its connection to the input and label spaces.

The prediction space $(\mathcal{Q}, \mathcal{Z})$ is not necessarily equal to label space $(\mathcal{Y}, \mathcal{V})$. Especially in anomaly detection and general classification tasks, many algorithms calculate a probability or a score for a label. Such a score is called *anomaly score* if it quantifies the likelihood of x belonging to C_A and *normal score* if it determines the degree of certainty to which x has the label C_N .

Scoring based algorithms are more flexible than techniques which assign hard class labels since anomalies can be ranked and prioritized according to their score. Then a domain specific discrimination threshold can be applied to separate anomalies from normal data *e.g.* we can define a mapping $\tau_\theta: (\mathcal{Q}, \mathcal{Z}) \rightarrow (\mathcal{Y}, \mathcal{V})$ as

$$\tau_\theta(\vartheta) = \begin{cases} C_N & \text{if } \vartheta \geq \theta \\ C_A & \text{else} \end{cases}$$

based on the threshold θ . Such domain specific thresholds depend on the costs of *false positives* (an anomaly is reported when the observation is normal) and *false negatives* (no anomaly is reported when the observation is anomalous).

DEFINITION 2.4 (Classifier and Predictor). A measurable function from the input space to the prediction space $\eta: (\mathcal{X}, \mathcal{X}) \rightarrow (\mathcal{Q}, \mathcal{Z})$ is called a *classifier* or *predictor* and sometimes *decision-rule*.

A classifier makes a prediction for an observation $X = x$ with the objective to distinguish normal from anomalous instances (Fig. 2.3). The set of possible classifiers are specified beforehand and is called the hypothesis space.

DEFINITION 2.5 (Hypothesis Space and Model). A *hypothesis space* or *statistical model* Ψ is a collection of classifiers $\eta: (\mathcal{X}, \mathcal{X}) \rightarrow (\mathcal{Q}, \mathcal{Z})$. We say the hypothesis space is *parameterized* with *parameter space* Θ , if there exists a surjective map $\Theta \rightarrow \Psi$.

The surjectivity in the parametric hypothesis space is required so that the set of classifiers is fully described by parameter space Θ . If this map is injective, the parameterization is *identifiable*. Consider, for instance, the set of classifiers on $\mathcal{X} = \mathbb{R}^d$ defined as

$$\Psi = \{\eta_c: \mathcal{X} \rightarrow \mathcal{Q} \mid \eta_c(x) = \langle c, x \rangle, c \in \Theta = \mathbb{R}^d\},$$

where each c gives rise to a different predictor. Our objective is then to find a c such that η_c has a good discriminative performance.

Interestingly, it can be shown that if the decision space is a finite set of k elements $\mathcal{Q} = \{C_1, \dots, C_k\}$, then a classifier defines a finite partition of the input space $\mathcal{X}$ as

$$\mathcal{X}_i = \{x \in \mathcal{X} \mid \eta(x) = C_i\} \quad (1 \leq i \leq k)$$

where $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \dots \cup \mathcal{X}_k$ and $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$ if $j \neq i$.

DEFINITION 2.6 (Hypothesis Space Dimension). We say that a hypothesis space Ψ is of dimension $d \in \mathbb{N}$ if $\Theta \subset \mathbb{R}^d$ and call it *parametric hypothesis space*. If Θ needs to be infinite to parameterize Ψ we call the model *non-parametric*.

Figure 2.3: A prediction for an instance of the input space by the classifier η thresholded by τ_θ to obtain a label.

The definition of non-parametric seems to be counter-intuitive since such a hypothesis space has indeed parameters, but it is common terminology, especially in Bayesian statistics.

CONSIDERING that the actual data and label distributions are not directly accessible, the only available information is a collection of unlabeled samples known as the *training set*.

DEFINITION 2.7 (*Sample Space and Training Set*). For any number of samples $n \in \mathbb{N}$, the probability space defined as the Cartesian product of n input spaces

$$\mathcal{Z} = \prod_{i=1}^n \mathcal{X}, \quad \mathcal{Z} = \bigotimes_{i=1}^n \mathcal{X} \quad \text{and} \quad \mathbb{P}_Z = \bigotimes_{i=1}^n \mathbb{P}_X$$

is called *sample space* and denoted by $(\mathcal{Z}, \mathcal{Z}, \mathbb{P}_Z)$. The realization $Z = z$ with $z = (x_1, \dots, x_n)$ is known as the (statistical) *training set*.

An elementary characteristic of machine learning is the estimation of a functional relationship between the input space $\mathcal{X}$ and the prediction space $\mathcal{Q}$ based on data from the statistical training set. This estimation is called *learning* (Fig. 2.4).

DEFINITION 2.8 (*Learning*). The selection of a classifier η from the hypothesis space Ψ based on a training set $z = (x_1, \dots, x_n)$ through the estimator

$$\begin{aligned} \hat{\Psi}: \mathcal{Z} &\rightarrow \Psi \\ z &\mapsto \eta \end{aligned}$$

is called *learning* or *training*.

Learning is often guided by a *loss function* which measure the performance of a classifier.

DEFINITION 2.9 (*Loss Function*). A measurable function $L: \mathcal{Y} \times \mathcal{Q} \rightarrow \mathbb{R}$ which quantifies the consequence of the prediction $\vartheta \in \mathcal{Q}$ for an instances with label $y \in \mathcal{Y}$ is called *loss function*.

While the loss function measures the costs of a single label-prediction pair, the risk function is defined as the loss we can *expect* under the data distribution if the classifier η is used to make predictions. Ideally $\hat{\Psi}$ selects the predictor with the lowest risk.

DEFINITION 2.10 (*Risk Function*). The *risk function* is defined as the expected loss under $\mathbb{P}_X \otimes \mathbb{P}_Y$, that is

$$\Psi \rightarrow \mathbb{R}: \eta \mapsto \int_{\Omega} L(Y, \eta(X)) d(\mathbb{P}_X \otimes \mathbb{P}_Y)$$

using the predictor η and loss function L .

Figure 2.4: Estimation of a classifier as an element from the hypothesis space from samples.

The Nature of Statistical Learning Theory by Vladimir Vapnik discusses the empirical minimization of risk functions and provides theoretical bounds on the performance of learning algorithms.

2.3 *Anomaly Detection*

There are two common techniques to train a classifier which are referred to as *batch* and *online* learning. Batch algorithms assume that the training set is permanently available at the learning phase and can be accessed in a random order. It is therefore possible to process each instance multiple times and even enables the possibility to incorporate the training set into the predictor as done *e.g.* by nearest neighbor-based algorithms. For anomaly detection in particular we make the following assumptions.

- The objective is to estimate a classifier η based on an unlabeled training set $z = (x_1, \dots, x_n)$ with a good predictive performance.
- The training set z comprises independent samples from the input space distribution $\mathbb{P}_X$ where the majority of instances are normal and anomalies are rare.
- It is assumed that the algorithm has full access to all elements of the dataset at once and is not required to make predictions before the training phase is completed.
- We may have access to a small labeled dataset of independent samples to configure the algorithm's parameters.

The labeled subset from the last assumption is used to select an appropriate hypothesis space *e.g.* the choice of a metric or distance function or other parameters which are not estimated from the sample z and therefore not included in Θ . These parameters are chosen to minimize the *empirical risk function*

$$\Psi \rightarrow \mathbb{R}: \eta \mapsto \frac{1}{m} \sum_{i=1}^m L(y_i, \eta(x_i))$$

evaluated on the labeled sample $(x_1, y_1), \dots, (x_m, y_m)$ with $m \in \mathbb{N}$.

AS MENTIONED before, we distinguish between *anomaly detection* and *outlier selection* algorithms. While anomaly detection techniques use the training set to learn a classifier from observations, *i. e.*

$$\begin{aligned} \hat{\Psi}: \mathcal{Z} &\rightarrow \Psi \\ (x_1, \dots, x_n) &\mapsto \eta, \end{aligned}$$

outlier selection algorithms assign a score or label to each element in the dataset as

$$\begin{aligned} \mathcal{Z} &\rightarrow \bigtimes_{i=1}^n (\mathcal{X} \times \mathcal{Q}) \\ (x_1, \dots, x_n) &\mapsto ((x_1, \vartheta_1), \dots, (x_n, \vartheta_n)). \end{aligned}$$

Notably, outlier selection is a special case of anomaly detection as the classifier η can always be applied to the instances in the training set.

2.4 Online & Streaming Anomaly Detection

In contrast to batch learning, where the full dataset $z = (x_1, \dots, x_n)$ is permanently available, *online* learning algorithms observe each instance only once and in a sequential order. Typically these algorithms have limited memory and thus can only store a small set of previously observed samples. It is therefore necessary to continuously update the model based on new observations. Formally,

$$\begin{aligned} \hat{\Psi}: \mathcal{X} \times \Psi &\rightarrow \Psi \\ (x_t, \eta_t) &\mapsto \eta_{t+1} \end{aligned}$$

for all $t = 1, 2, \dots$ and some initial model $\eta_1 \in \Psi$.

THERE ARE two situations in which online learning is commonly used: *incremental* and *streaming* anomaly detection. The former can be applied to the standard batch setting in situations where it is impractical to access the training set at once *e.g.* the dataset is distributed on large clusters where random access is expensive. The training set observed from the sample space $(Z, \mathcal{Z}, \mathbb{P}_Z)$ is still assumed to be independent and identically distributed (i.i.d.) which implies that $\mathbb{P}_Z = \bigotimes_{i=1}^n \mathbb{P}_X$. The sequential order in which each instance is accessed or observed is in this case irrelevant.

In contrast to incremental learning, the i.i.d. assumption does not hold in general for *streaming anomaly detection* where data arrives in a sequence $x_1, x_2, x_3, \dots$ of observations which is potentially unbounded in size. It is desirable to detect anomalous events early which enables the initiation of appropriate reactions or counteractions in order to prevent damage. A reasonable requirement is therefore to rank or classify each instance of the stream after it has been observed. Since the unboundedness of the stream makes it impossible to store the entire sequence or pass through it multiple times this problem is a suitable candidate to apply online learning techniques.

In the streaming setting, the input and label space distributions may additionally evolve over time, a problem known as *concept drift*. In this case it is also said that the environment is *non-stationary* (Fig. 2.5). That means our understanding of what is normal and anomalous changes over time. Obviously this implies now a temporal ordering of the instances in the stream. The random variables of the input and label spaces are now modeled through the (discrete-time) stochastic processes

$$\{X_t \mid t \in \mathbb{N}\} \quad \text{and} \quad \{Y_t \mid t \in \mathbb{N}\}$$

and their corresponding distributions. The presence of concept drift introduces the necessity to identify and remove those previously observed instances which are no longer consistent with the current state of the environment. Most algorithms realize this by forgetting outdated information while incorporating new knowledge.⁴ This is done under the assumption that recent observations carry more relevant information than older data.

Figure 2.5: Taxonomy: online, incremental and streaming anomaly detection.

⁴ Gama et al., *A survey on concept drift adaptation*.

IN SUMMARY, we identify the following key characteristics of *streaming anomaly detection*:

- A data stream is possibly unbounded in size and therefore impossible to store and process at once.
- In the context of anomaly detection, as considered here, we assume that most instances of the stream belong to the class of normal data and anomalies are rare events.
- The stream can evolve over time, forcing algorithms to adapt to changes in the data distribution.
- Only a small fraction of data at the beginning of the stream is available to configure the algorithm's parameters.
- An instance of the stream has to be classified or ranked as soon as it is observed which requires a fast processing rate.

2.5 *Statistical Comparison of Algorithms*

A considerable part of this thesis is concerned with the analysis of machine learning methods and the introduction of new techniques. This requires a theoretically sound methodology for comparison and evaluation of algorithms which we briefly summarize at this point.

2.5.1 *Performance Metrics*

Machine learning algorithms are evaluated based on their predictive performance on a dedicated labeled *test set*. Based on the algorithm's predictions and the actual label of the instance, there are several common performance measures which are often used to evaluate anomaly detection techniques. Recall that the class of normal data is denoted by C_N and the class of anomalies by C_A .

THE simplest metric is the *accuracy* which is simply the rate of correctly classified instances. An accuracy of 100% means that the predictions are exactly the same as the true class labels. This seems to be a natural choice, but there are several disadvantages in using the accuracy as a metric. Most significantly it is very sensitive to imbalanced (skewed) class distributions as they occur in anomaly detection. Since anomalies are by definition rare events, a classifier which simply predicts C_N all the time would yield a high accuracy which is obviously not desirable. Another disadvantage is that this metric is only applicable to classifiers which output a hard class label which is either C_N or C_A and not to such predicting a score.

SINCE many anomaly detection algorithms make probabilistic predictions or output continuous scores their performance varies with the chosen discrimination threshold. The ROC curve is a graphical plot which illustrates the performance with respect to this threshold. It distinguishes between four possible outcomes. If a prediction is *normal* and the actual label is *normal*, we call it a *true-positive* (TP). On the contrary, if the actual label for this prediction was *anomaly*, then it is called *false-positive* (FP). Conversely,

“One of the earliest adopters of ROC graphs in machine learning was Spackman (1989), who demonstrated the value of ROC curves in evaluating and comparing algorithms. Recent years have seen an increase in the use of ROC graphs in the machine learning community, due in part to the realization that simple classification accuracy is often a poor metric for measuring performance. In addition to being a generally useful performance graphing method, they have properties that make them especially useful for domains with skewed class distribution and unequal classification error costs. These characteristics have become increasingly important as research continues into the areas of cost-sensitive learning and learning in the presence of unbalanced classes.”

— Fawcett, *An introduction to ROC analysis*.

a *true-negative* (TN) is a classification as *anomaly* when the actual label was *anomaly* and *false-negative* (FN) when it was *normal* (Fig. 2.6).

The ROC curve plots the true-positive rate TPR (the number of TP divided by the number of *normal* instances) against the false-positive rate FPR (the number of FP divided by the number of *anomalies*) *i. e.*

$$\text{TPR} = \frac{\sum \text{TP}}{\sum C_N} \quad \text{and} \quad \text{FPR} = \frac{\sum \text{FP}}{\sum C_A}.$$

The curve can be summarized as a single number — the *area under the ROC curve* (AUC) obtained by integration. This AUC is a score between zero and one, where a higher value is considered to be better. The area under the curve can be interpreted as the probability that a classifier will rank a randomly chosen positive (normal) instance higher than a randomly chosen negative (anomalous) instance.

AN OFTEN used alternative to the receiver operating characteristic curves are *precision-recall* (PR) curves. The *precision* (PPV) and *recall* (which is equivalent to the true-positive rate TPR) are defined as

$$\text{PPV} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FP}} \quad \text{and} \quad \text{TPR} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FN}},$$

respectively. As before we can calculate the area under this curve as the *area under the PR curve* (AUCPR) to obtain a single number.

THE *F-score* is more often used in the field of information retrieval and defined as the harmonic mean of precision and recall

$$\text{F-score} = 2 \frac{\text{PPV} \cdot \text{TPR}}{\text{PPV} + \text{TPR}}.$$

However, as accuracy and precision, the F-score is not appropriate to evaluate the performance of anomaly detection algorithms due to its sensitivity to the class imbalance problem as observed by Fawcett:

ROC curves have an attractive property: they are insensitive to changes in class distribution. If the proportion of positive to negative instances changes in a test set, the ROC curves will not change [...]. Metrics such as accuracy, precision, lift and F-score use values from both columns of the confusion matrix. As a class distribution changes these measures will change as well, even if the fundamental classifier performance does not. ROC graphs are based upon true-positive rate and false-positive rate, in which each dimension is a strict columnar ratio, so do not depend on class distributions.⁵

Interestingly, Davis et al. observe that “a deep connection exists between ROC space and PR space, such that a curve dominates in ROC space if and only if it dominates in PR space.”⁶ We therefore focus primarily on the AUC as a performance measure in the experimental evaluations.

Figure 2.6: The four possible classification outcomes. The illustration shows the distribution of anomalies on the left and the distribution of normal data on the right. The vertical line represents the decision boundary.

Figure 2.7: Receiver operating characteristic curve.

“The AUC has an important statistical property: the AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. This is equivalent to the Wilcoxon test of ranks.”

— Fawcett, *An introduction to ROC analysis*.

⁵ Fawcett, *An introduction to ROC analysis*.

⁶ Davis et al., *The relationship between Precision-Recall and ROC curves*.

2.5.2 Statistical Hypothesis Testing

When comparing multiple (anomaly detection) algorithms over multiple datasets one cannot simply compare the raw numbers obtained from the area under *receiver operating characteristic* or *precision-recall* curves. In particular, Webb warns against averaging these numbers as “it is debatable

whether error rates in different domains are commensurable, and hence whether averaging error rates across domains is very meaningful”.⁷ Also Demšar points out that it is dangerous to use tests which are designed to compare two algorithms for more than a pairwise comparison: “A common example of such questionable procedure would be comparing seven algorithms by conducting all 21 paired t-tests [...]. When so many tests are made, a certain proportion of the null hypotheses is rejected due to random chance, so listing them makes little sense.”⁸

In a paired t -test we compare two algorithms under the null hypothesis that both perform equally well. After a k -fold cross-validation we would calculate the pairwise differences on some performance criteria *e.g.* the classification error. Call these differences $u_1, u_2, \dots, u_m$. Let $\bar{u}$ be the sample mean and $\bar{s}$ be the sample variance of $u_1, u_2, \dots, u_m$. The test statistic

$$T = \frac{\sqrt{m}(\bar{u} - 0)}{\bar{s}}$$

then follows a t -distribution with $m - 1$ degrees of freedom under the null hypothesis which is that the differences are distributed according to a normal distribution with zero mean. We can then lookup a significance level *e.g.* 5% and reject the null hypothesis if the observed p-value is less than this significance level ($p < 0.05$). A p-value is the probability of a result arising under the null hypothesis by chance and explains why Demšar’s concern is justified.

To avoid the limitations of the paired t -test when comparing multiple algorithms, Demšar suggests to use the Friedman test with the corresponding post hoc Nemenyi test for comparison of several classifiers over multiple data sets. The Friedman test only determines if the null hypothesis can be rejected, but not which algorithms perform differently. This is the task of a post hoc test.

The Friedman test⁹ is a non-parametric statistical test which ranks algorithms for each dataset individually starting from 1 as the best rank. Its purpose is to examine whether there is a significant difference between the performances of individual algorithms with the null hypothesis that all of them are equivalent and therefore their ranks should be equal. Let’s assume we compare k algorithms on m datasets and let r_{ij} be the rank of the j -th algorithm on the i -th dataset. We use $\bar{r}_j$ to denote the average rank of algorithm j *i.e.* $\bar{r}_j = m^{-1} \sum_i r_{ij}$. The Friedman statistic

$$\chi_F^2 = \frac{12m}{k(k+1)} \left(\sum_{j=1}^k \bar{r}_j^2 - \frac{k(k+1)^2}{4} \right)$$

is undesirably conservative, which is why Iman et al.¹⁰ suggest to use the modified statistic

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

which is distributed according to the (Fisher–Snedecor) F-distribution with $k - 1$ and $(k - 1)(m - 1)$ degrees of freedom. If the null hypothesis (all algorithms are equivalent) is rejected one can proceed with a post hoc test.

⁷ Webb, *Multiboosting: A technique for combining boosting and wagging*.

⁸ Demšar, *Statistical comparisons of classifiers over multiple data sets*.

⁹ Friedman, *The use of ranks to avoid the assumption of normality implicit in the analysis of variance*.

¹⁰ Iman et al., *Approximations of the critical region of the Friedman statistic*.

THE NEMENYI TEST¹¹ is a post hoc test to compare all (anomaly detection) algorithms with each other. Two methods then perform significantly different if their average ranks differ by at least

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

called the *critical difference*. Here the *critical values* q_α are obtained from the Studentised range statistic divided by $\sqrt{2}$ and have been tabulated in many textbooks on statistics (table 2.1).

k	2	3	4	5	6	7	8
NEMENYI	1.960	2.343	2.569	2.728	2.850	2.949	3.031
BONFERRONI	1.960	2.241	2.394	2.498	2.576	2.638	2.690

¹¹ Nemenyi, *Distribution-free multiple comparisons*.

Table 2.1: Critical values for the Nemenyi and Bonferroni-Dunn tests for $q_{0.05}$.

When comparing multiple classifiers with a control algorithm, the Nemenyi test gets less powerful the more classifiers are used since it operates under the assumption that we are performing $\frac{1}{2}k(k+1)$ comparisons. We then can either apply a Bonferroni-Dunn correction¹² or use the more powerful Holm test.¹³ The latter is a step-down procedure and orders the test statistics for comparing the i -th and j -th classifier

$$z = \frac{\bar{r}_i - \bar{r}_j}{\sqrt{\frac{k(k+1)}{6m}}}$$

which is distributed according to the normal distribution. Holm's test starts with the most significant p -value and rejects a null hypothesis if $p_i < \frac{\alpha}{(k-i)}$ for $i = 1, \dots, k-1$. As soon as the first hypothesis cannot be rejected, the procedure stops and all remaining hypotheses are retained.

¹² Dunn, *Estimation of the Medians for Dependent Variables*.

¹³ Holm, *A simple sequentially rejective multiple test procedure*.

DEMŠAR ALSO SUGGESTS to visually represent the results of the Nemenyi test in a critical difference diagram (Fig. 2.8). In this exemplar diagram we compare 5 algorithms on 20 datasets against each other. An algorithm is represented by a colored line. The position on the horizontal axis denotes the algorithms average rank across all compared techniques. The null hypothesis is that the average rank of all methods is the same. Algorithms for which this null hypothesis cannot be rejected based on the experimental evidence are connected by a bar. Therefore, only algorithms which are not connected by a bar can be considered significantly different from each other.

Figure 2.8: Visualization of the post hoc Nemenyi test in form of a critical difference diagram. The position on the line indicates the algorithms's average rank. Algorithms which are significantly different (at $p < 0.05$) are *not* connected with a bar. The critical difference (CD) is given by the double arrow in the top left.

CHAPTER THREE

Anomaly Detection Algorithms

SYNOPSIS · The previous chapter introduced the theoretic foundations and definitions of anomaly detection. In this part we inspect and compare various anomaly detection algorithms. We will start with classical statistical approaches (§3.2) which fit a probability distribution to normal data and then classify based on the likelihood of an unseen instance under this distribution. This is based on the definition of an anomaly as an unlikely event. We continue with distance-based methods (§3.3) which provide good classification performance, but are relatively slow. These techniques assume that normal instances occur in dense neighborhoods while anomalies are isolated. Discriminative models (§3.4) are the heart of machine learning theory and are well studied. Members of this category are standard classifiers such as the support vector machine and decision trees, modified to meet the requirements of semi-supervised anomaly detection. In contrast, clustering-based models are not designed for anomaly detection but often detect anomalies as a side effect (§3.6). They make either the assumption that anomalies do not belong to any cluster formed by normal instances or that anomalies form small clusters with low densities while normal instances belong to large and dense clusters. We finish the review with a preview of the proposed approach named **EXPPOSE** (§3.7). Using concepts of *kernel embedding* and Hilbert space theory, the method is able to estimate the similarity between observations and the input space distribution based solely on a similarity measure — the kernel function.

3.1 *Preliminaries*

The dataset shown in Fig. 3.1 is used exemplarily to illustrate various properties of different anomaly detection algorithms reviewed in this chapter. The dots ($\bullet$) represent a training set $\mathbf{z} = (x_1, \dots, x_n)$ from the sample space $(\mathcal{Z}, \mathcal{X}, \mathbb{P}_{\mathcal{Z}})$ as defined in the previous chapter. This dataset consists of one dense cluster and several other instances which are distributed around the cluster in a half-circle of lower density. Obviously, this set is not representative for large-scale problems in high dimensions, however it will serve to study and convey the underlying ideas on which the following algorithms are based on.

THERE are several desirable properties of anomaly detection techniques to keep in mind when comparing different approaches. These are *scalability*, *robustness* and *configurability*.

Figure 3.1: Distribution of normal data. We combine a scatterplot ($\bullet$) and plots of marginal distributions as axes ($\|$) as suggested by Tufte et al., *The visual display of quantitative information*.

Scalability Since we are interested in large-scale anomaly detection, scalability is our main concern. Algorithms of practical interest should roughly operate linearly in both dataset size and dimension.

Robustness Semi-supervised and unsupervised anomaly detection operate on unlabeled data, but implicitly assume that normal instances are far more frequent than anomalies. An algorithm should not react too sensitive if the dataset contains more anomalies than expected.

Configurability The number of parameters required to define the hypothesis class is crucial for the applicability of an algorithm. Examples are the number of components for mixture models, the depth of a decision tree or the number of nearest neighbors to consider. It is often hard to tune more than two or three such parameters.

3.2 Statistical Approaches

Statistical techniques fit a probability distribution to the (normal) data which enables them to classify between normal and anomalous based on the likelihood of a new instances under this distribution. This approach is very intuitive as anomalies are defined to be unlikely observations.

3.2.1 Parametric Models

Parametric approaches start the modeling process with the choice of an appropriate statistical hypothesis class $\Psi = \{\mathbb{P}_\theta \mid \theta \in \Theta\}$ which comprises a collection of candidate probability measures $\mathbb{P}_\theta: \mathcal{Z} \rightarrow [0, 1]$ on the sample space $(\mathcal{Z}, \mathcal{Z})$.

Assume we want to fit the mean μ and variance σ^2 of a normal distribution $N(\mu, \sigma^2)$ to real-valued observations $(X_1, \dots, X_n)$. Then the sample space is $(\mathcal{Z}, \mathcal{Z}) = (\mathbb{R}^n, \mathbb{B}^n)$ and the hypothesis class is given by

$$\Psi = \{N(\mu, \sigma^2) \mid \mu \in \mathbb{R}, \sigma^2 \in (0, \infty)\}$$

with parameter space $\Theta = \mathbb{R} \times (0, \infty)$. It remains to estimate the mean and variance from data. After these parameters have been found, all instances which are unlikely under this model *i.e.* deviate more than a certain distance from the mean (for example 3σ) are then declared to be anomalous (Fig. 3.2).

Figure 3.2: Under the normal distribution, the probability that an observation is within one standard deviation σ of the mean μ is ≈ 0.68 , while the probability for two standard deviations is ≈ 0.95 .

IN machine learning, the hypothesis class is often defined in terms of probability densities rather than measures. We then have to ensure that each measure can be represented by a density function which is provided by the Radon-Nikodym theorem as long as Ψ is a *dominated* class.

DEFINITION 3.1 (*Dominated*). A hypothesis class Ψ is called *dominated* if there exists a σ -finite measure $\mu: \mathcal{X} \rightarrow [0, \infty]$ such that every $\mathbb{P}_\theta \in \Psi$ has a μ -density f_θ , $f_\theta \ll \mu$.

The distribution $\mathbb{P}_\theta$ can then be identified as $\mathbb{P}_\theta(\mathcal{D}) = \int_{\mathcal{D}} f_\theta(x) d\mu(x)$ for all $\mathcal{D} \in \mathcal{X}$. Such a representation is not always unique, but this does not cause any problems as most techniques output the density directly as a *normal score*, that is, $\eta(y) = f_\theta(y)$. The *anomaly score* is therefore proportional to the inverse of the density function. Consequently, the prediction space is induced by the density and hence $\mathcal{Q} = \mathbb{R}_+$.

IF THE HYPOTHESIS CLASS is dominated, then a common method in statistics to estimate the parameter $\theta \in \Theta$ is the *maximum likelihood* principle. As the name indicates, the goal is to maximize a mapping L_z , the *likelihood function*.

DEFINITION 3.2 (*Likelihood function*). Let $\Psi = \{\mathbb{P}_\theta \mid \theta \in \Theta\}$ be a dominated class of probability measures on $(\mathcal{Z}, \mathcal{Z})$ with densities f_θ . For a training set $z \in \mathcal{Z}$, the mapping

$$L_z: \Theta \mapsto \mathbb{R}_+ \quad \text{with} \quad L_z(\theta) = f_\theta(z) \quad (\theta \in \Theta)$$

is called the *likelihood function* of z .

In contrast to the density function, the likelihood function takes the parameters θ as an argument and not the sample z . However, the likelihood function is always defined with respect to such a sample. The maximum likelihood *estimator* then chooses the density as a classifier (predictor) for which the likelihood is maximized.

DEFINITION 3.3. The map $\hat{\Psi}: \mathcal{Z} \rightarrow \Theta$ with

$$L_z[\hat{\Psi}(z)] = \sup_{\theta \in \Theta} L_z(\theta), \quad \text{a.s.}$$

is called *maximum likelihood estimator* (MLE) for θ .

In the context of the previously defined anomaly detection framework, *learning* is simply a maximum likelihood estimate with $z \mapsto \eta = f_{\hat{\theta}}$ with $\hat{\theta} = \sup_{\theta \in \Theta} L_z(\theta)$.

AN APPROACH TO VISUALIZE the quantiles of a distribution is a so-called *box plot*¹ (Fig. 3.3). The box plot directly marks all data outside the whiskers as anomalies (outliers). The lower and upper whiskers are calculated as $q_1 - 1.5 \cdot \text{IQR}$ and $q_3 + 1.5 \cdot \text{IQR}$ respectively, where IQR stands for *interquartile range* and is defined as the difference between the upper and lower quantile $q_3 - q_1$.

Figure 3.3: Example of a box plot. Design by Tufte et al., *The visual display of quantitative information*.

¹ McGill et al., *Variations of box plots*.

3.2.2 *Mixture Models*

The assumption of parametric models that the distribution of interest can be modeled using a simple parametric hypothesis class such as the Poisson, binomial or normal distribution is often too restrictive. In these cases an (often finite) mixture of such a parametric hypothesis class is used, called *mixture model*.² The idea is to use a combination of simple models to form a richer, more complex one which can better capture the characteristics of the true distribution.

A mixture model of $k \in \mathbb{N}$ components can be created as follows. Let $\{\mathbb{P}_{\theta_i} \mid \theta_i \in \Theta_i\}$ be a parametric family of distributions for $1 \leq i \leq k$, where k is known as the *order* of the mixture. Each candidate distribution $\mathbb{P}_{\theta}$ in the hypothesis class is then a mixture of k distributions

$$\mathbb{P}_{\theta} = \sum_{i=1}^k \pi_i \mathbb{P}_{\theta_i}$$

with weights $(\pi_1, \dots, \pi_k) \in \mathbb{R}^k$. The parameter space of the hypothesis class $\Psi = \{\mathbb{P}_{\theta} \mid \theta \in \Theta\}$ is given by

$$\Theta = \left\{ (\theta_1, \dots, \theta_k, \pi_1, \dots, \pi_k) \mid \theta_i \in \Theta_i, \pi_i \in \mathbb{R}_+, \sum_{i=1}^k \pi_i = 1 \right\}$$

for all $1 \leq i \leq k$.

THE MOST POPULAR mixture model comprises a collection of multivariate normal distributions, called the *Gaussian mixture model* (GMM). These models are applied, for example, to detect anomalies in object motion patterns³ or to calculate extreme-value statistics for novelty detection in biomedical data.⁴

The Gaussian mixture model is defined in terms of density functions and hence assumes that the hypothesis class is dominated by a σ -finite measure. The densities associated with the distributions $\mathbb{P}_{\theta_i}$ are given by multivariate normal distributions

$$f_{\mu, \Sigma}(z): \mathbb{R}^d \rightarrow \mathbb{R}_+$$

$$f_{\mu, \Sigma}(z) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(z - \mu)^T \Sigma^{-1}(z - \mu)\right),$$

where $(\mu, \Sigma) \in \Theta_i = (\mathbb{R}^d \times \mathbb{R}_{++}^{d \times d})$ and $\mathbb{R}_{++}^{d \times d}$ is the space of positive definite matrices. The parameters μ and Σ are the *mean* and *covariance* of the density respectively (see Fig. 3.4). A sufficient number of components can give rise to complex densities which are able to model various continuous distributions. The GMM is typically trained by a maximum likelihood estimation. However, there is no closed-form analytical solution for the maximum likelihood and therefore numerical optimization techniques have to be applied. Alternatively, *expectation maximization*⁵ (EM), an approach from Bayesian statistics can be applied to fit the model parameters. We refer to the literature for more details about EM.

THE main disadvantage of mixture models is the dimension of the parameter space. From the previous equation we see that the GMM has to fit one

² Lindsay, *The geometry of mixture likelihoods: a general theory*; Redner et al., *Mixture densities, maximum likelihood and the EM algorithm*.

Gaussian mixture models

³ Basharat et al., *Learning object motion patterns for anomaly detection and improved object detection*.

⁴ Roberts, *Extreme value statistics for novelty detection in biomedical data processing*.

⁵ Dempster et al., *Maximum likelihood from incomplete data via the EM algorithm*.

Figure 3.4: A density estimation using a Gaussian mixture model with 3 components. The arrows ($\longrightarrow$) show the eigenvectors of the covariance matrices for each component and the observations $Z = z$ are drawn as black dots ($\bullet$).

d -dimensional mean vector, one $d \times d$ covariance matrix and one weight per mixture component. Since the covariance matrix has to be positive definite, these are $k(d + d(d + 1)/2 + 1) = \mathcal{O}(kd^2)$ parameters in total. This becomes intractable as the dimension d of the input space increases. To overcome this problem some authors⁶ suggest to use the same covariance matrix for all mixture components or even restrict the covariance matrix to be diagonal or a scaled identity matrix. However, such approaches forfeit much of the approximation capability, which was the sole purpose of introducing mixture models.

The memory requirements and the computational complexity for a single prediction are both $\mathcal{O}(kd^2)$, independent of the dataset size n , but not practical in high dimensions. Also, a single iteration of the maximum likelihood optimization routine takes $\mathcal{O}(knd^2)$ which is often too expensive. Since the number of iterations is unknown, it is not possible to calculate the computational complexity of the learning phase exactly.

⁶ Tax, *One-class classification*, Section 3.3.2.

3.2.3 Kernel Density Estimation

An alternative to simple parametric models and mixture distributions are non-parametric approaches such as the *kernel density estimator* (KDE) introduced by Murray Rosenblatt and Emanuel Parzen — therefore also known as the *Parzen density estimator*.⁷ As the name suggests it is a tool to estimate (continuous) Lebesgue densities using a kernel function.

Let $(\mathcal{Z}, \mathcal{Z}, \mathbb{P}_Z)$ be the sample space and $z = (x_1, \dots, x_n)$ be the training set with instances from the input space $(\mathcal{X}, \mathcal{X})$. The kernel density estimator f is a non-parametric approach to estimate the density of the probability measure $\mathbb{P}_X$ given Z . If $(\mathcal{X}, \mathcal{X}) = (\mathbb{R}, \mathbb{B})$ the estimator is given by

$$f_z: \mathcal{X} \rightarrow \mathbb{R}_+$$

$$f_z(x) = \frac{1}{nh} \sum_{i=1}^n k\left(\frac{x - x_i}{h}\right),$$

where $h \in \mathbb{R}_+$ is called *bandwidth* (Fig. 3.5). Here, k is a non-negative (kernel) function that integrates to one and has zero mean as

$$\int k(u) du = 1 \quad \text{and}$$

$$k(-u) = k(u)$$

kernel density estimator

⁷ Parzen, *On estimation of a probability density function and mode*; Rosenblatt, *Remarks on some nonparametric estimates of a density function*.

for all $u \in \mathbb{R}$. The bandwidth h can be optimized using, for example, a maximum likelihood approach. It can be shown⁸ that, under certain conditions, as the number of samples goes to infinity, the sequence of kernel density estimators (f_n) converges almost surely to the true density $f_X \ll \mathbb{P}_X$ as

$$\mathbb{P}\left(\left\{z \in \mathcal{Z} \mid \lim_{n \rightarrow \infty} \sup_{y \in \mathcal{X}} |f_n(y) - f_X(y)| = 0\right\}\right) = 1.$$

Here, f_n is a kernel density estimate from a set with n samples.

⁸ Nadaraya, *On non-parametric estimates of density functions and regression curves*.

Figure 3.5: A non-parametric density estimation using KDE. The black dots ($\bullet$) mark the realizations $Z = z$. The bandwidth h is indicated by the dashed circle (---).

A multivariate kernel density estimator in d dimensions ($\mathcal{X} = \mathbb{R}^d$) is defined as

$$f_z: \mathcal{X} \rightarrow \mathbb{R}_+$$

$$f_z(x) = \frac{1}{n\sqrt{|H|}} \sum_{i=1}^n k\left(H^{-\frac{1}{2}}(x - x_i)\right),$$

where H is a $d \times d$ symmetric and positive definite matrix and $|H|$ its determinant. Figure 3.5 illustrates the kernel density estimation with the popular kernel function

$$k(u) = \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{1}{2}\langle u, u \rangle\right)$$

where H is represented by a diagonal matrix with entries equal to the bandwidth h as indicated by the dashed circle. As before, the prediction space is defined through the density function as $\mathcal{Q} = \mathbb{R}_+$ since the classifier is now the kernel density estimation *i.e.* $\eta = f_z$.

THE kernel density estimator can be seen a continuous generalization of a simple histogram estimation as used, for example by Endler for intrusion detection.⁹ Here the (univariate) estimator f_z is defined as

$$f_z(x) = \frac{1}{nh} \sum_{i=1}^n \mathbb{I}_{I_i}(x_i) \mathbb{I}_{I_i}(x)$$

where $I_i = (ih, (i+1)h]$ are the intervals on which the histogram is calculated (Fig. 3.6).

Figure 3.6: Histogram estimator. The height of each bar is proportional to the number of instances which fall into the corresponding intervals.

⁹ Endler, *Intrusion detection. applying machine learning to Solaris audit data*.

EVEN THOUGH the kernel density estimator has well studied consistency guarantees¹⁰ it is often impractical to apply to large-scale problems. First, density estimation in high dimensions is known to be problematic in general since it requires an enormous amount of data. This is also observed *e.g.* by Friedman et al.

The direct extension of these methods to multivariate settings, however, has not been as successful in practice. This can partly be attributed to their deteriorating statistical performance caused by the so-called “curse of dimensionality”¹¹ which requires very large spans (radii of neighborhoods) in order to achieve sufficient counts. The resulting estimates are then highly biased. In addition, these methods do not provide any comprehensible information about the structure of the multivariate point cloud.

— Friedman et al., *Projection Pursuit Density Estimation*

Second, the memory requirements and prediction times are prohibitively large since the complete training set needs to be processed to predict the likelihood for a single observation. Consequently, both quantities grow linearly with the dataset size and dimension.

¹⁰ Wied et al., *Consistency of the kernel density estimator: A survey*.

¹¹ Bellman, *Adaptive control processes: A guided tour*.

¹² Tukey, *Mathematics and the picturing of data*.

¹³ Eto, *The Concept of Depth In Statistics*.

¹⁴ Mosler, *Depth Statistics*.

¹⁵ Zuo et al., *General notions of statistical depth function*.

3.2.4 Data Depth

The *data depth* concept is also a statistical approach, but different from the previous techniques as it does not estimate a density or a distribution, but the *depth* of an instance relative to other observations in a dataset.

DEFINITION 3.4 (Data Depth). The depth of an instance $y \in \mathcal{X}$ with respect to a dataset $z = (x_1, \dots, x_n)$ is defined as the smallest number of instances lying in a closed half-space whose boundary is passing through the point y .¹²

Instances with high depth are more central in the set and therefore less likely anomalous. However, while conceptually simple and elegant data depth is computationally challenging:

For the concept of data depth to be really useful, its computation has to be possible and efficient. A collaborative effort of statisticians and computational geometers is the foundation for ongoing research on this subject. Although there are already some satisfactory algorithms for computing depth functions in the bivariate case, there is still much work to do in higher dimensions.¹³

This intuitive idea to define depth in terms of half-spaces has been extended in several directions, generalizing the notion of statistical depth.¹⁴ Zuo et al.¹⁵ identifies the following “desirable” properties of general depth functions:

DEFINITION 3.5 (Depth Function). Let $\mathcal{B}^1(\mathcal{X})$ denote the set of all Borel probability measures on $\mathcal{X} = \mathbb{R}^d$. A bounded mapping

$$D: \mathcal{X} \times \mathcal{B}^1(\mathcal{X}) \rightarrow \mathbb{R}_+$$

which satisfies

Affine invariance The depth should be independent of the underlying coordinate system. That is $D(y, \mathbb{P}_X) = D(Ay + b, \mathbb{P}_{AX+b})$, where A is a bounded operator and b is a vector in $\mathcal{X}$.

Maximality at center If a uniquely defined “center” y^* of the distribution exists, then the depth function should be maximal at this center.
 $D(y^*, \mathbb{P}_X) = \sup_{y \in \mathcal{X}} D(y, \mathbb{P}_X)$.

Monotone on rays If a point y^* has maximal depth, then the depth should decrease monotonically along any ray through the center. Then it holds that $D(y, \mathbb{P}_X) \leq D(y^* + \alpha(y - y^*), \mathbb{P}_X)$ for $\alpha \in [0, 1]$

Vanishing at infinity $D(y, \mathbb{P}_X) \rightarrow 0$ as $\|y\| \rightarrow \infty$ for each $\mathbb{P}_X \in \mathcal{B}^1(\mathcal{X})$.

is called *statistical depth function*.

It is arguable if a uniquely defined center is appropriate for anomaly detection as it ignores multimodality properties of the underlying distribution as illustrated in Fig. 3.7.

Figure 3.7: Tukey's half-space depth.

3.3 Distance-Based Models

Nearest neighbor-based anomaly detection techniques assume that normal instances occur in dense neighborhoods, while anomalies are isolated. There are different ways to define such a *dense neighborhood*. Most common is to declare the distance to the k -nearest neighbor of an instance as its neighborhood. The number of points within this neighborhood is then used to approximate a density. The advantage of these algorithms is that they only require a metric. They can therefore be applied to a variety of domains such as videos, images, trajectories or more complicated structures like graphs. This section reviews some frequently applied techniques and discusses their advantages and disadvantages.

DEFINITION 3.6 (k -Nearest Neighbors Distance). Let $(\mathcal{X}, d)$ be a metric space and $x \in \mathcal{X}$ be an instance of the input space. An $\epsilon > 0$ is called *k -nearest neighbors distance* of x with respect to the sample $z = (x_1, \dots, x_n)$, if and only if

$$\begin{aligned} |\overset{\circ}{B}(x, \epsilon)| &< k \quad \text{and} \\ |\overline{B}(x, \epsilon)| &\geq k + 1 \end{aligned}$$

where $B(x, \epsilon)$ is the open ball¹⁶ with center x and radius ϵ . The k -nearest neighbors distance is then denoted by $\tau_z(x)$.

Figure 3.8: The k -nearest neighbors distance for $k = 2$.

¹⁶ See (§B) for the definition of open ball $B(\epsilon, x)$, interior, boundary and closure.

DEFINITION 3.7 (*k-Nearest Neighbors*). In continuation of Def. 3.6, the *k-nearest neighbors* of x with respect to z , written as $\tilde{n}_z^k(x)$, is defined as

$$\tilde{n}_z^k(x) = \partial B(x, \tau_z(x))$$

which are all elements on the boundary of the ball with radius $\tau_z(x)$.

The definition of *k-nearest neighbors* of x in terms of interior and boundary is necessary since it is possible that more than one element in z has the same distance to x as shown in Fig. 3.8. For $k = 1$ we will simply say *nearest neighbors distance* and *nearest neighbors* of x .

3.3.1 Distance-based Outlier Detection

A straightforward way to build an anomaly detector based on the neighborhood is to use the *k-nearest neighbors distance* as an anomaly score

$$\begin{aligned} \hat{\Psi}: \mathcal{Z} &\rightarrow \Psi \\ z &\mapsto \eta = \tau_z, \end{aligned}$$

as suggested by Ramaswamy et al.¹⁷ The assumption made here is that the *k-nearest neighbor distance* for an anomaly is much large than for normal instances.

THERE exists different variations of this simple idea. Knorr et al. propose a *distance-based outlier detection*¹⁸ algorithm which classifies an instance as anomalous (distance-based outlier) if at least a fraction of α of the objects in the training have a distance from this instance that is larger than δ .

DEFINITION 3.8 (*Distance-based Outlier Detection*). Let $(\mathcal{X}, d)$ be a metric space and $z = (x_1, \dots, x_n)$ be a training set. An instance $y \in \mathcal{X}$ is called a *distance-based outlier* if

$$|\{x_i \in z \mid d(y, x_i) > \delta\}| \geq \alpha n$$

where $\delta \in \mathbb{R}$, $\alpha \in [0, 1]$ and $|\cdot|$ denotes the cardinality of a set.

The biggest disadvantage of this definition is the difficulty to choose an appropriate distance δ . Interestingly, this is equivalent to the *k-nearest neighbor distance* approach proposed by Ramaswamy using $k = \alpha n$ (rounded to the next $k \in \mathbb{N}$) and the classifier $\tau_z(y) > \delta$.

THERE are several extensions of the distance-based outlier detection algorithm to streaming data based on a sliding window. The window is a buffer which keeps track of past observations. Whenever a new observation is added, the oldest one expires and is removed from the buffer (Fig. 3.9). The distance-based outlier detection is now restricted to the observations in the sliding window and all other instances are not relevant for the classification. Most proposed algorithms aim at improving the nearest neighbor search within the sliding window using an appropriate data structure and try to reduce the memory requirements with summary statistics. Two implementations are the *stream outlier miner*¹⁹ (STORM) and *continuous outlier detection*²⁰ (COD).

¹⁷ Ramaswamy et al., *Efficient algorithms for mining outliers from large data sets*.

¹⁸ Knorr and Ng, *Algorithms for mining distance-based outliers in large datasets*; Knorr, Ng, and Tucakov, *Distance-based outliers: algorithms and applications*.

Figure 3.9: Illustration of a sliding window of size 4 over a stream.

¹⁹ Angiulli et al., *Distance-based outlier queries in data streams: the novel task and algorithms*.

²⁰ Kontaki et al., *Continuous monitoring of distance-based outliers over data streams*.

3.3.2 Nearest Neighbor Data Description

The *nearest neighbor data description*²¹ (NNDD) operates under the assumption that the density of the input space distribution varies and therefore the prediction score should be adapted to this property. This can be realized by a comparison of the density at a certain point to the density at its neighbors. The authors propose a score based on the ratio of the k -nearest neighbors distances of an instance and its nearest neighbor.

DEFINITION 3.9 (*Nearest Neighbor Data Description*). Let $(\mathcal{Z}, \mathcal{L}, \mathbb{P}_Z)$ be the sample space and $z = (x_1, \dots, x_n)$ a random sample from the input space $(\mathcal{X}, \mathcal{X})$ with metric d . The decision-rule

$$\eta: \mathcal{X} \rightarrow \mathbb{R}_+$$

$$\eta(x) = \min\{\tau_z(y) \mid y \in \tilde{n}_z^k(x)\}^{-1} \tau_z(x)$$

is called *nearest neighbor data description*.

The definition above accounts for the possibility that there is more than one nearest neighbor of x . In this case the smallest nearest neighbors distance is used.

²¹ Ridder et al., *An experimental comparison of one-class classification methods*. nearest neighbor data description

Figure 3.10: Contour plot of the (negative) nearest neighbor data description score. The sensitivity to the density in a neighborhood can cause non-smooth decision boundaries.

NNDD is not very robust in the sense that the nearest neighbor ratios often produce rough edges in the score surface as illustrated in Fig. 3.10. Such effects are mostly undesirable as a small shift in the input space can have a huge effect on the anomaly detection score. Therefore, a smoother prediction function is in general preferred.

3.3.3 Local Outlier Factor

Another algorithm which considers local densities is the *local outlier factor*²² (LOF). It is especially designed to work on datasets with non-uniform densities. LOF is related to the nearest neighbor data description, but tries to overcome its weaknesses and creates a smooth decision boundaries with a concept called *reachability distance*.

DEFINITION 3.10 (*Reachability Distance*). Let $(\mathcal{X}, d)$ be a metric space and $z = (x_1, \dots, x_n)$ be a sample. The k -reachability distance of $x \in \mathcal{X}$ with respect to z is defined as

$$\tilde{d}_z: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$$

$$\tilde{d}_z(x, y) = \max\{\tau_z(y), d(x, y)\}$$

where τ_z is the k -nearest neighbors distance as defined previously.

local outlier factor
²² Breunig et al., *LOF: identifying density-based local outliers*.

The reachability distance is not a metric since it is asymmetric in the parameters and introduced to replace the k -nearest neighbors distance by something more robust. When y is in a sufficiently dense area and the distance between x and y is large, then the reachability distance $\tilde{d}_z$ is equal to the original distance d . However, if the distance between x and y is small, then the k -nearest neighbors distance τ_z is used as a smoother. This smoothing factor increases with k .

Figure 3.11: The local outlier factor for $k = 5$. We used the strictly monotone mapping $t \mapsto \exp(-t)$ to transform the LOF to a *normal* score.

DEFINITION 3.11 (k -Neighborhood). The k -neighborhood of x is defined as

$$n_z(x) = \{x_i \in z \setminus \{x\} \mid d(x_i, x) \leq \tau_z(x)\}$$

and contains all points within the closed ball of radius $\tau_z(x)$ without x itself.

DEFINITION 3.12 (Neighborhood Density). The neighborhood density of x , denoted by ω , is given by

$$\omega(x) = \left(\frac{1}{|n_z(x)|} \sum_{x_i \in n_z(x)} \tilde{d}_z(x, x_i) \right)^{-1}$$

which is inverse to the mean reachability distance of all points in the neighborhood of x .

The local outlier factor is an anomaly score and therefore is interpreted as the likelihood to which an $y \in \mathcal{X}$ is an anomaly. It is given by the ratio between the mean neighborhood densities of all neighbors and the neighborhood density of y itself.

DEFINITION 3.13 (Local Outlier Factor). Let $(\mathcal{X}, d)$ be a metric space and denote the training set by $z = (x_1, \dots, x_n)$. The predictor

$$\eta(y) = \frac{1}{|n_z(y)|} \sum_{x_i \in n_z(y)} \frac{\omega(x_i)}{\omega(y)}$$

is called *local outlier factor* of x with respect to z .

As a consequence, a point with similar density as the data around it will receive a *local outlier factor* of approximately 1. If the point is not located

within a cluster, it will have a relatively low neighborhood density and hence assigned a higher outlier factor.

The local outlier factor is successfully applied in various applications with promising results. However, the local outlier factor performance depends on how fast nearest neighbor queries can be made. For high-dimensional data LOF approaches $\mathcal{O}(n^2)$ training time as “[...] we need to use a sequential scan or some variant of it [...], with a complexity of $\mathcal{O}(n)$, leading to a complexity of $\mathcal{O}(n^2)$ for the materialization [learning] step.”²³ The memory required to store the neighborhood density is of order $\mathcal{O}(n)$. Another disadvantage is the requirement to choose the parameter k which has a major impact on the algorithm’s performance. The authors suggest to choose k to be between 10 and 50, but it is not clear if this is sufficient on large-scale and high-dimensional data.

²³ Breunig et al., *LOF: identifying density-based local outliers*.

3.3.4 Stochastic Outlier Selection

An algorithm called *stochastic outlier selection* (SOS) was introduced in the context of maritime outlier detection.²⁴ SOS can also operate on a metric space using the notion of affinity.

We use the standard problem setup with the metric space $(\mathcal{X}, d)$ and the training set $z = (x_1, \dots, x_n)$. The stochastic outlier selection algorithm defines the *affinity* of two instances from the input space $\mathcal{X}$ as

$$\alpha: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$$

$$\alpha(x, y) = \begin{cases} \exp(-d(x, y)/2\sigma_x^2) & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases}$$

where σ_x is a function of x called *bandwidth*. Consequently, the affinity to distant observations is small. Moreover, the higher the bandwidth is, the less is the affinity influenced by the distance d . We also note that σ_x introduces an asymmetry into the function α as it only depends on x and not y . The authors introduce a normalization of the affinity function with respect to the training set z as

$$b_z(x, y) = \frac{\alpha(x, y)}{\sum_{x_i \in z} \alpha(x, x_i)}$$

called *binding probabilities*. Obviously, $b_z(x, y) \in [0, 1]$ for all $x, y \in z$. After the normalization step, SOS predicts a *normal* score²⁵ for the observation $y \in z$ as

$$\eta(y) = 1 - \prod_{\substack{x_i \in z \\ x_i \neq y}} [1 - b_z(x_i, y)].$$

The problem is that η gets close to 1 as soon as $b_z(x_i, y)$ is close to 1 for a single $x_i \in z$. The implication is that a single point in the training set suffices to cause a high normal score in its neighborhood. In fact, this is one of the main disadvantages of stochastic outlier selection since a single wrongly labeled point in the training set has a major influence on the decision-rule as shown in Fig. 3.13.

²⁴ Janssens, Postma, et al., *Density-Based Anomaly Detection in the Maritime Domain*.
stochastic outlier selection

²⁵ In the original publication the so-called *outlier probability* defined as $1 - \eta(y)$ are used.

sos on a dataset with only normal instances.

We injected an anomaly into the training set which causes a major change of the sos scores.

Figure 3.13: Stochastic outlier selection and its sensitivity to noise in the training set.

SO FAR we did not address the problem on how to find the bandwidth σ_x . The idea is to set σ_x depending on its neighbors such that each neighbor has “sufficient” influence on x . The bandwidth σ_x are adapted to the local density of a data point, in such a way that a higher density leads to a lower bandwidth, causing the affinity to its neighbors to decay faster. Given a *perplexity* parameter $h > 0$, the sos algorithm determines σ_x for each $x \in \mathcal{Z}$ such that

$$2^{H(x)} = h \quad \text{with} \\ H(x) = - \sum_{\substack{x_i \in \mathcal{Z} \\ x_i \neq x}} b_z(x, x_i) \log_2 b_z(x, x_i),$$

where $H(x)$ is the Shannon entropy²⁶ of x . “The effect of such an adaptive variance is that $[x]$ distributes, in general, around 90% of its affinity to its h nearest neighbors.”²⁷

Since the bandwidth can, in general, not be determined in closed-form we have to solve n optimization problems, one for each $x_i \in \mathcal{Z}$, to make a single prediction. This results in a computational complexity of $\mathcal{O}(n^2)$ per classification and $\mathcal{O}(n)$ memory consumption which makes stochastic outlier selection impractical for large-scale problems.

²⁶ Shannon, *A mathematical theory of communication*.

²⁷ Janssens, Huszár, et al., *Stochastic outlier selection*.

3.3.5 Limitations of Distance-Based Methods

The common problem of distance-based techniques for large-scale anomaly detection is the time required to perform a nearest neighbor lookup. This can be accelerated if a tree structure *e.g.* a k-d tree²⁸ is used. Building such a tree from n elements takes $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ memory. If the input space is of low dimension and n is much larger than 2^d then finding r nearest neighbor in a k-d tree with randomly distributed points takes $\mathcal{O}(\log n)$ time on average. However, this does not hold in high dimensions as observed by Goodman et al.:

²⁸ Bentley, *Multidimensional binary search trees used for associative searching*.

Unfortunately, as the dimension grows, the algorithms become less and less efficient. More specifically, their space or time requirements grow exponentially in the dimension. In particular, the nearest-neighbor problem has a solution with $\mathcal{O}(d^{O(1)} \log n)$ query time, but using roughly $n^{O(d)}$ space. Alternatively, if one insists on linear or near-linear storage, the best known running time bound for random input is of the form $\min(2^{O(d)}, dn)$, which is essentially linear in n even for moderate d . Worse still, the exponential dependence of space and/or time on the dimension (called the “curse of dimensionality”) has been observed in applied settings as well. Specifically, it is known that many popular data structures (using linear or near-linear storage), exhibit query time linear in n when the dimension exceeds a certain threshold [...]

— Goodman et al., *Handbook of discrete and computational geometry*

As a consequence, the lookup in high dimensions is not better than an exhaustive search with $\mathcal{O}(n)$ if a data storage is used which does not grow exponentially.

3.4 Discriminative Models

Many classification-based anomaly detection algorithms are variants of their classical counterpart for two or multiple classes. Examples are *kernel fisher discriminants*²⁹ and the *support vector machine*³⁰ (SVM).

²⁹ Roth, *Outlier detection with one-class kernel Fisher discriminants*.

³⁰ Schölkopf and Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*.

³¹ Schölkopf, Platt, et al., *Estimating the support of a high-dimensional distribution*.

one-class support vector machine

³² Tax and Duin, *Support vector data description*.

3.4.1 Support Vector Machines

The concept of the *one-class support vector machine*³¹ (OC-SVM) was independently published by Tax and Duin³² under the name *support vector data description*. Like the standard support vector machine this technique has the advantage that it works well on high-dimensional datasets, *e.g.* documents, images or videos. Here, the OC-SVM attempts to find a hyperplane such that most of the observations are separated from the origin with maximum margin.

IN THE following let $(\mathcal{X}, \mathcal{X})$ be a separable input space. Since the data is often not linearly separable from the origin in the original space $\mathcal{X}$, the OC-SVM transforms each instance $x \in \mathcal{X}$ into a high-dimensional (reproducing kernel) Hilbert space $\mathcal{H}$ via a map $\phi: \mathcal{X} \rightarrow \mathcal{H}$ (§4.1). Given a training set $z = (x_1, \dots, x_n)$ from the sample space $(\mathcal{Z}, \mathcal{Z}, \mathbb{P}_Z)$, a linear hyperplane in $\mathcal{H}$ which separates the samples z from the origin with a large margin can be found by solving the optimization problem

$$\begin{aligned} \min_{w, \xi, \rho} \quad & \frac{1}{2} \|w\|^2 + \frac{1}{\nu n} \sum_{i=1}^n \xi_i - \rho & w \in \mathcal{H}, \rho \in \mathbb{R}, \xi \in \mathbb{R}^n \\ \text{s.t.} \quad & \langle w, \phi(x_i) \rangle \geq \rho - \xi_i, & i = 1, \dots, n \\ & \xi_i \geq 0, & i = 1, \dots, n \end{aligned}$$

where the parameter $\nu \in (0, 1)$ controls the trade-off between the fraction of instances from training set classified as anomalies and the fraction of support vectors. The so-called *slack variables* $\xi = (\xi_1, \dots, \xi_n) \in \mathbb{R}^n$ allow instances to be within the margin. The slack variables are necessary since

there exist feature maps (induced by universal kernel functions³³ such as the squared exponential and Laplacian) for which *every* set can be perfectly separated by a hyperplane in $\mathcal{H}$. This is not always desirable as it leads, especially in the presence of noise, to overfitting.

If w and ρ are solutions the optimization problem above, then the classifier given by

$$\eta(y) = \text{sgn}(\langle w, \phi(y) \rangle - \rho)$$

will be positive for most x_i contained in the training set z . The function assigns a positive label to all instances in the affine half-space above $\langle w, \phi(y) \rangle - \rho$ and a negative label to all instances in the other affine half-space (Fig. 3.14). Therefore, the hypothesis class of support vector machines comprises all affine hyperplanes

$$\Psi = \{ \langle w, \phi(x) \rangle - \rho = 0 \mid w \in \mathcal{H}, \rho \in \mathbb{R} \}$$

with parameter space $\Theta = \mathcal{H} \times \mathbb{R}$.

NOTICE THAT the OC-SVM optimization problem is equivalent to the one of the standard (two class) v-SVM³⁴ given by

$$\begin{aligned} \min_{w, \xi, \rho} \quad & \frac{1}{2} \|w\|^2 - \nu \rho + \frac{1}{n} \sum_{i=1}^n \xi_i \quad w \in \mathcal{H}, \rho \in \mathbb{R}, \xi \in \mathbb{R}^n \\ \text{s.t.} \quad & y_i (\langle w, \phi(x_i) \rangle + b) \geq \rho - \xi_i, \quad i = 1, \dots, n \\ & \xi_i \geq 0, \quad i = 1, \dots, n \end{aligned}$$

with $y_i = 1$ for all $1 \leq i \leq n$ and $b = 0$. Since b is equal to zero, this optimization problem finds a hyperplane through the origin in $\mathcal{X}$ such that the distance of all x_i to the margin of the hyperplane is maximized. A connection to the original C-SVM³⁵ is the so-called SVM *without offset*.³⁶

IN PRACTICE a direct optimization in the often infinite-dimensional Hilbert space $\mathcal{H}$ is avoided using a Lagrange approach to solve the *dual* problem

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(x_i, x_j) \quad \alpha_i \in [0, (\nu n)^{-1}] \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i = 1 \end{aligned}$$

using the kernel property $\langle \phi(x_i), \phi(x_j) \rangle = k(x_i, x_j)$. If α is a solution of the optimization problem, the decision function can be defined in terms of kernel expansions as

$$\eta(y) = \text{sgn} \left(\sum_{i=1}^n \alpha_i k(x_i, y) - \rho \right),$$

which is sometimes called the *dual decision boundary* (Fig. 3.15). Note that in general the above representation is not unique and depends on the

³³ Universal kernel functions are dense in the space of continuous functions (Def. 5.6).

Figure 3.14: The OC-SVM finds a hyperplane such that the observations are separated from the origin with a large margin. The instances on the boundary are called *support vectors*.

³⁴ Schölkopf, Smola, et al., *New support vector algorithms*.

³⁵ Schölkopf and Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*.

³⁶ Scott, *The equivalence between the one-class and paired support vector machines for nonseparable data*.

algorithm used to solve the optimization problem. Nevertheless, this representation leads an identifiable parameterization $\Theta \rightarrow \Psi$ of the hypothesis class with

$$\Theta = \{(\alpha_1, \dots, \alpha_n, \rho, x_1, \dots, x_n) \in \mathbb{R}^n \times \mathbb{R} \times \mathcal{X}^n\}$$

which is now *non-parametric* as its dimension depends on the number of samples n .

Figure 3.15: The decision boundary calculated by the one class support vector machine.

THIS OC-SVM classifier can be turned into a scoring function by considering the signed distance to the separating hyperplane using

$$\eta(y) = \sum_{i=1}^n \alpha_i k(x_i, y) - \rho.$$

This implies that the further an instance is from the origin in $\mathcal{H}$, the more likely it belongs to the class of normal data (Fig. 3.15).

A disadvantage of the one class support vector machine is that it does not scale to very large datasets. As shown by Steinwart, the number of support vectors can grow linearly with the number of instances in the training set:

Let us assume that the kernel used in [the optimization problem] is universal. [...] With probability tending to 1 when $n \rightarrow \infty$ the fraction of support vectors is essentially greater than the Bayes risk. [...] Therefore, when a noisy classification problem is learned, we should expect that the number of support vectors increases linearly in the sample size.

— Steinwart, *Sparseness of support vector machines*

Since, the memory requirements and the computational complexity per prediction depend on the number of support vectors, both are of order $\mathcal{O}(n)$. That means even if we would be able to afford the $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space complexity to solve the optimization problem at the training stage, the time to make a prediction may render the OC-SVM ineligible for large-scale applications.

3.4.2 Isolation Forest

The *Isolation Forest*³⁷ (iForest) is an anomaly detection algorithm based on random trees. The idea is to separate (isolate) all instances using a binary tree

³⁷ Liu et al., *Isolation-based anomaly detection*.

structure called isolation tree (*iTree*). The assumption is that anomalies are rare and that they obey different properties than normal instances. Hence it should be easier to isolate them which implies that anomalies are closer to the *iTree*'s root and therefore have a shorter path length, denoted by $h(x)$. On the contrary, normal instances are more difficult to isolate since they lie in dense regions of the input space and are therefore deeper located within the tree.

Figure 3.16: The isolation forest anomaly detector. We used the mapping $t \mapsto \exp(-t)$ to convert the anomaly score to a *normal* score.

The input space is assumed to be Euclidean *i.e.* $(\mathcal{X}, \mathcal{X}) = (\mathbb{R}^d, \mathbb{B}^d)$ and $z = (x_1, \dots, x_n)$. An isolation tree is created on a random subsample $\tilde{z}$ of size $m \leq n$ from the training set z by recursively partitioning $\tilde{z}$ until all instances are isolated. Both the attribute to partition and the split point are selected randomly. An *iForest* is simply an ensemble of $k \in \mathbb{N}$ such isolation trees. The anomaly score is calculated as

$$\eta(y) = -\frac{\sum_{i=1}^k h_i(y)}{c(m)},$$

which is the normalized average path length of each tree $h_1(x), \dots, h_k(x)$. Here, the normalization term

$$c(m) = \begin{cases} 2 \sum_{i=1}^{m-1} \frac{1}{i} - 2 \frac{m-1}{m} & \text{for } m > 2 \\ 1 & \text{for } m = 2 \\ 0 & \text{otherwise} \end{cases}$$

is introduced to make models with different subsample sizes m comparable.

Isolation forests achieve a constant training time and space complexity by subsampling the training set to a fixed size and can therefore be applied to datasets with a large number of samples. However, the random attribute and split point selection can become problematic in high dimensions as the splitting can occur at suboptimal attributes. It would be better to exploit the knowledge contained in the data to guide this selection.

3.4.3 Tree-Based Methods for Streaming Anomaly Detection

*Streaming half-space-trees*³⁸ (HSTA) randomly construct a binary tree structure upfront without any data. Here, the proposed approach selects a dimension at random and splits it at half. Each tree then counts the number of instances from the training set assigned to each node which is referred to as “mass”. The score for a new instance is defined to be proportional

³⁸ Tan et al., *Fast anomaly detection for streaming data*.

to the mass of the leaf which this instance hits after passing down the tree. Obviously, an ensemble of such trees can be built in constant time and the training grows linearly with the dataset size. However, randomly splitting a very high-dimensional space will not produce a tree which is sufficiently fine-grained for anomaly detection.

THE *RS-Forest*³⁹ is a modification of *HSTA* in which each dimension is not split in half, but at a random cut-point. However, the assumption that “[...] once each instance is scored, streaming *RS-Forest* will receive the *true* label of the instance [...]”⁴⁰ is not realistic in anomaly detection applications.

³⁹ Wu et al., *RS-Forest: A Rapid Density Estimator for Streaming Anomaly Detection*.

⁴⁰ *Ibid.*

3.5 Angle-Based Models

The angle-based outlier detection algorithm for high-dimensional data⁴¹ (*ABOD*) proposed by Kriegel et al. is based on the assumption that “comparing the angles between pairs of distance vectors to other points helps to discern between points similar to other points and [anomalies]”. Here, the angle variances for observations within a cluster (the normal data) will be large compared to the angle variances of anomalies which are far away from normal instances.

⁴¹ Kriegel et al., *Angle-based outlier detection in high-dimensional data*.

IN THE FOLLOWING let p be the set of all pairs from the training set $z = (x_1, \dots, x_n)$, that is

$$p = \{(x_i, x_j) \mid x_i, x_j \in z\}$$

which is of size n^2 . Let

$$\alpha(y, (x_i, x_j)) = \frac{\langle y - x_i, y - x_j \rangle}{\|y - x_i\|^2 \|y - x_j\|^2}$$

be the angle between the two line segments defined by (y, x_i) and (y, x_j) weighted by their length. The angle-based outlier factor (*ABOF*) is then given by

$$\eta(y) = n^{-2} \sum_{(x_i, x_j) \in p} \left(\alpha(y, (x_i, x_j)) - n^{-2} \sum_{(x_i, x_j) \in p} \alpha(y, (x_i, x_j)) \right)^2,$$

which is the sample variance over all angles. If $y \in \mathcal{X}$ is an anomaly, then this variance of angles between pairs of the training set should be small. As the score decreases, the likelihood of y being an anomaly increases. This is illustrated in Fig. 3.17, however at least in low dimensions, the weighted distance has more influence on the score than the variance of angles.

Even though *ABOD* is designed for data in high dimensions and is free of parameters, its $\mathcal{O}(n^2)$ time per prediction and $\mathcal{O}(n)$ memory requirements are prohibitive for large datasets. The authors therefore suggest to estimate the variance of angles not from all pairs in p but only from the set of k -nearest neighbors of the query point y . This new approach is called *F-ABOD* (fast *ABOD*) and reduces the computational complexity to $\mathcal{O}(n + k^2)$, since the nearest neighbor search will take $\mathcal{O}(n)$ in high dimensions as argued before (§3.3.5).

Variance in angles
 $\langle y - x_i, y - x_j \rangle$

Variance in distances
 $\|y - x_i\|^2 \|y - x_j\|^2$

A combination of both plots
 on the left (ABOF).

Figure 3.17: Angle-based outlier detection.

3.6 Clustering-Based Models

Clustering-based anomaly detection techniques group similar data into clusters. A way to interpret this clusters is to see them as representatives of a group of data and use a nearest neighbor approach which considers the distance to the nearest cluster instead of the nearest instance. This has the advantage that the nearest neighbor search is considerably faster and noise (also anomalies wrongly labeled as normal) does not perturb to the prediction. These techniques make the assumption that normal instances are close to a cluster centroid, while anomalies are distant to their closest cluster centroid.

3.6.1 k-means

One of the most popular clustering algorithms is *k-means*.⁴² The input space is assumed to be Euclidean *i.e.* $(\mathcal{X}, \mathcal{X}) = (\mathbb{R}^d, \mathbb{B}^d)$. Let $\|\cdot\|$ be the Euclidean norm and $z = (x_1, \dots, x_n)$ be the training set.

The objective of *k-means* is to find a partition $S = (S_1, S_2, \dots, S_k)$ of z such that $z = S_1 \cup S_2 \cup \dots \cup S_k$ and $S_i \cap S_j = \emptyset$ if $j \neq i$ which minimizes

$$\sum_{i=1}^k \sum_{x \in S_i} \|x - m_i\|^2,$$

where m_i the sample mean over all instances in S_i .

Lloyd's algorithm⁴³ (often just called *k-means*) starts with k groups $(S_1, S_2, \dots, S_k)$, where each group S_i consists of a single random point $x_i \in z$ determining the mean of the group as $m_i = x_i$. All remaining points are added to the group whose mean is closest to the respective point. After this step the mean of the group is adjusted using the sample mean of all points in that group. Formally the algorithms iterates over $t = 1, 2, \dots$ starting with initial centroids $m_1^{(1)}, \dots, m_k^{(1)} \in z$. Set

$$S_i^{(t)} = \left\{ x \in z \mid \|x - m_i^{(t)}\|^2 \leq \|x - m_j^{(t)}\|^2, j = 1, 2, \dots, k \right\}$$

⁴² MacQueen, *Some methods for classification and analysis of multivariate observations*.

⁴³ Lloyd, *Least squares quantization in PCM*.

and update

$$m_i^{(t+1)} = |S_i^{(t)}|^{-1} \sum_{x \in S_i^{(t)}} x$$

until the partition S no longer changes. A common technique to initialize the cluster centroids is called k -means++ proposed by Arthur et al.⁴⁴ and is used to reduce the number of iterations.

Given the k centroids $m = (m_1, m_2, \dots, m_k)$ obtained by a clustering algorithm such as the one described above, we can perform anomaly detection by using the distance to the closest centroid as an anomaly score, *i. e.* for $k = 1$ we can write

$$\eta(y) = \tau_m(y)$$

where $\tau_m(y)$ is the k -nearest neighbors distance with respect to the set m . This approach assigns a high anomaly score to instances without a cluster center nearby.

20 clusters.

50 clusters.

Difference between the scores using 50 clusters and the scores using 20 clusters.

Figure 3.18: k -means anomaly detection for different values of k . We used the strictly monotone mapping $t \mapsto \exp(-t)$ to transform the distance score to a normal score.

IN CONTRAST to the clustering problem (finding real clusters in a set), the anomaly detector based on k -means is not very sensitive if the parameter k is specified too large as shown in Fig. 3.18. The reason is that the distance to the nearest cluster does not change significantly if more centroids are used. The time complexity per iteration is $\mathcal{O}(kn)$, however since k -means clustering is an iterative algorithm the overall training time complexity cannot be determined exactly. The memory requirements and prediction time are relatively small with $\mathcal{O}(kd)$ and $\mathcal{O}(k)$ respectively. The k -means algorithm is implicitly based on pairwise Euclidean distances between points since it minimizes the within-cluster variance. An approach to perform clustering in general metric spaces is *e.g.* k -medoids using the *partitioning around medoids* (PAM) algorithm.⁴⁵ k -medoids techniques represent a cluster by one of its instances. However, these techniques do not scale to large datasets.

⁴⁴ Arthur et al., *k-means++: The advantages of careful seeding*.

⁴⁵ Kaufman et al., *Finding groups in data: An introduction to cluster analysis*.

3.7 Proposed Approach

In this dissertation we present a novel algorithm for anomaly detection named *EXpected Similarity Estimation* (EXPOSE) and this section is a preview of the proposed approach which is discussed in detail in the following chapters.

GIVEN a kernel function $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ which measures the similarity or closeness between instances of the input space $\mathcal{X}$, EXPOSE can be seen as the average similarity between y and the realizations of X *i.e.* $\mathbb{E}[k(y, X)]$ defined as

$$\eta(y) = \int_{\mathcal{X}} k(y, x) d\mathbb{P}_X(x).$$

However, this formulation has several drawbacks, in particular, the evaluation of η is computationally inefficient. We therefore propose several modifications to this initial approach. Specifically, we will exploit the fact that every kernel can be written as an inner product in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ as

$$k(x, y) = \langle \phi(x), \phi(y) \rangle,$$

where $\phi: \mathcal{X} \rightarrow \mathcal{H}$ is called *feature map*. Consider the simple example with $x, y \in \mathbb{R}$ and the kernel

$$k(x, y) = xy = \begin{bmatrix} \frac{x}{\sqrt{2}} & \frac{x}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{y}{\sqrt{2}} \\ \frac{y}{\sqrt{2}} \end{bmatrix},$$

where we can define a feature map as $\phi(u) = \begin{bmatrix} \frac{u}{\sqrt{2}} & \frac{u}{\sqrt{2}} \end{bmatrix} \in \mathbb{R}^2$. Another feature map which fulfills the requirement $k(x, y) = \langle \phi(x), \phi(y) \rangle$ would simply be $\phi(u) = u \in \mathbb{R}$. Importantly, many kernel functions cannot be represented as finite-dimensional vectors and therefore implicitly defined through the kernel as $\phi(u) = k(\cdot, u)$. This is also a valid feature function as $k(x, y) = \langle k(\cdot, x), k(\cdot, y) \rangle$. Here, the dot $\cdot$ indicates that this argument of k is not bound to a variable, *i.e.* for any fixed $u \in \mathcal{X}$, $\phi(u)$ is a function from $\mathcal{X}$ to $\mathbb{R}$.

EXPOSE is defined in §5 on page 61.

See §4.1 on page 51 for a discussion on feature maps and their properties.

3.7.1 Hilbert Space Mapping

In the course of this thesis, we show that the EXPOSE anomaly detector can be rewritten as

$$\begin{aligned} \eta(y) &= \int_{\mathcal{X}} k(y, x) d\mathbb{P}_X(x) \\ &= \langle \phi(y), \mu[\mathbb{P}_X] \rangle_{\mathcal{H}}, \end{aligned}$$

with the intention to break the bond or tie between y and x such that it is not necessary to evaluate the integral for every prediction. The function μ is called *kernel embedding* or *kernel mean map* of $\mathbb{P}_X$ and is defined as

$$\mathbb{P}_X \mapsto \mu[\mathbb{P}_X] = \int_{\mathcal{X}} \phi(x) d\mathbb{P}_X(x),$$

where $\mu[\mathbb{P}_X]$ is an element of $\mathcal{H}$. Given a kernel k , the kernel mean map can

The kernel mean map is defined in §5.1 on page 62.

be used to embed a probability distribution (or a measure) into a RKHS.

The inner product $\langle \phi(y), \mu[\mathbb{P}_X] \rangle_{\mathcal{H}}$ grows inversely with the distance of $\phi(y)$ to the mean $\mu[\mathbb{P}_X]$ and can therefore be seen as the closeness or similarity to the mean. It is important to realize that even though the notion of a unique center is inappropriate for anomaly detection in the original input space $\mathcal{X}$ the situation is different in the RKHS induced by the kernel and can therefore be interpreted as a *statistical depth function*. This is illustrated in Fig. 3.19, where we use multidimensional scaling⁴⁶ (MDS) to visualize the embedding of the training set into the infinite-dimensional RKHS.

⁴⁶ Torgerson, *Multidimensional scaling*.

Figure 3.19: Visualization of the feature map embedding. The feature map projects the data into an RKHS where the closeness to the center is an appropriate feature for anomaly detection.

WHENEVER the input space distribution $\mathbb{P}_X$ is unknown, but a sample $z = (x_1, \dots, x_n)$ from the sample space $(\mathcal{Z}, \mathcal{Z}, \mathbb{P}_Z)$ is available, the empirical distribution $\mathbb{P}_n$ can be used to construct an approximation $\mu[\mathbb{P}_n]$ of $\mu[\mathbb{P}_X]$ as

$$\mu[\mathbb{P}_X] \approx \mu[\mathbb{P}_n] = \frac{1}{n} \sum_{i=1}^n \phi(x_i)$$

which is called *empirical kernel embedding* (§5.3).

AS MENTIONED before, there is in general no explicit representation of the feature map ϕ . Since $\mu[\mathbb{P}_X]$ is defined in terms of ϕ , the kernel mean map is affected by the same problem. We therefore aim to find an *approximate* feature map (§5.5), *i.e.* a function $\hat{\phi}$ such that

$$k(x, y) \approx \langle \hat{\phi}(x), \hat{\phi}(y) \rangle$$

and $\hat{\phi}(x) \in \mathbb{R}^r$ for some $r \in \mathbb{N}$. Combining the empirical mean map $\mu[\mathbb{P}_n]$ and approximate feature maps yields

$$\begin{aligned} \hat{\Psi}: \mathcal{Z} &\rightarrow \Psi \\ z &\mapsto \eta(y) = \left\langle \hat{\phi}(y), \frac{1}{n} \sum_{i=1}^n \hat{\phi}(x_i) \right\rangle \end{aligned}$$

as a computationally efficient EXPOSE anomaly detector (Fig. 3.20).

Figure 3.20: EXPOSE scores calculated by $\eta(y) = \langle \phi(y), \mu[\mathbb{P}_n] \rangle$. The empirical mean map $\mu[\mathbb{P}_n]$ is an estimator for $\mu[\mathbb{P}_X]$ based the sample $(\bullet)$.

We see that, using approximate features maps, $\mu[\mathbb{P}_n]$ requires only a constant amount of memory to be stored and can be calculated in $\mathcal{O}(n)$ time. This linear computational complexity can be further improved, estimating an ϵ -accurate approximation of $\mu[\mathbb{P}_X]$ in *constant time*, independent of the datasets size, by means of convex optimization (§6). The calculation of a score $\eta(y)$ at the prediction stage is reduced to the evaluation of an inner product which also takes only constant time.

3.8 Summary

We reviewed several anomaly detection approaches with roots in statistics (§3.2), based on distance (§3.3), discriminative models (§3.4), angle-based (§3.5), clustering techniques (§3.6) and we gave a preview of the EXPOSE anomaly detector (§3.7).

Statistical methods provide the most information if the correct class of distribution is chosen. However, especially in anomaly detection we have to assume that we do not know the input space distribution. The alternative of non-parametric statistical models are impractical due to the high computational costs. Distance-based methods can operate on any metric space and are hence widely applicable. However, the limiting factors are the slow nearest neighbor search and the high memory requirements caused by the necessity to store all training data. The angle-based method reviewed in this chapter suffers from the same limitations. Some clustering approaches avoid the shortcomings of distance-based methods by summarizing the data information in terms of cluster exemplars. Once the clusters had been found, all training data can be discarded and predictions are made solely based on the cluster exemplars. Similarly, discriminative approaches build a model

based on the training set. Only this model is then needed to classify new instances. In the group of discriminative techniques we reviewed kernel-based methods *e.g.* the support vector machine and approaches based on decision trees such as *iForest*.

Figure 3.21 is a preview of the experimental section (§5.8) comparing the runtime and predictive quality of each algorithm. The “indefinite” marker indicates that LOF, F-ABOD and KDE were not able to produce a result on some of the larger datasets caused by an infeasible memory consumption or runtime. In cases where an algorithms take the same processing time for multiple datasets, the markers overlap. On large-scale, high-dimensional datasets EXPoSE is several orders of magnitude faster than OC-SVM and LOF. Furthermore, we will show that EXPoSE performs significantly better in terms of prediction quality than *iForest*, F-ABOD and KDE.

Figure 3.21: Runtime vs. rank (predictive performance) on several datasets. The color indicates the dataset size. The result of a large dataset overlays the result of a smaller one if the algorithm’s runtime is the same. Experiments with the same dataset are connected by a line.

CHAPTER FOUR

Hilbert Spaces & Kernel Functions

SYNOPSIS · This chapter introduces reproducing kernel Hilbert spaces (RKHS) and their properties. We review some elementary theory of reproducing kernel functions notably the theorems due to Mercer (§4.3) and Bochner (§4.4). Reproducing kernels can be seen as a notion of similarity between elements of the inputs space and are an integral part of this work. One of our concerns will be the measurability of such kernel functions which is necessary for the probabilistic approach taken in subsequent chapters.

4.1 Reproducing Kernel Hilbert Spaces

Kernel methods are popular tools to solve various problems in machine learning, computational biology and statistics. Their core is a positive definite function called *kernel* which implicitly defines a so-called reproducing kernel Hilbert space (RKHS) — an inner product space with special properties. Kernels offer a way to define a similarity between two instances of the input space. A distance or similarity function is often easier to provide than an input space transformation to a suitable feature space in which anomaly detection can be performed effectively. Consequently, kernel functions are applicable to various domains such as images, documents, projections and even distributions. The selection of a suitable kernel function is critical for the accuracy and performance of the algorithm. This section provides the necessary background information and definitions of kernel functions and their corresponding reproducing kernel Hilbert spaces.

WE CAN EITHER start with a positive definite kernel function k and show that k defines an RKHS or we start with a Hilbert space where the evaluation functional is continuous and show that there is a unique corresponding kernel. In this chapter, we follow the latter approach.

DEFINITION 4.1 (Hilbert Space). A Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ over a non-empty set $\mathcal{X}$ is an inner product space which is complete with respect to the norm induced by the inner product, *i.e.* it is a Banach space with an inner product.

We will just write $\langle \cdot, \cdot \rangle$ for the inner product if the dependence on $\mathcal{H}$ is unambiguous. Examples of popular Hilbert spaces are the Euclidean space $\mathbb{R}^d$ with the standard vector dot product and the space of *square summable sequences* ℓ^2 *i.e.* sequences with elements $(x_i)_{i \in \mathbb{N}}$ such that $\sum_{i \in \mathbb{N}} |x_i|^2 < \infty$ and inner product $\langle x, y \rangle = \sum_{i \in \mathbb{N}} x_i y_i$. Also the space of square-integrable

“Hilbert spaces are perfect generalizations of euclidean spaces, to the point of being almost trivial as geometrical objects. The deeper theory (and the fruitful applications) is, however, concerned with the *operators* on Hilbert space.”
— Pedersen, *Analysis now*.

functions $\mathcal{L}^2(a, b)$ on the interval (a, b) is a Hilbert space with inner product $\langle f, g \rangle = \int_a^b f(x)g(x) dx$ for functions $f, g \in \mathcal{L}^2(a, b)$.

Only real Hilbert spaces are relevant for the remainder of this work and we therefore solely consider these. Nevertheless most of the theory derived in this chapter can also be applied to complex Hilbert spaces. Notably, not all Hilbert spaces are reproducing kernel Hilbert spaces as this requires certain properties on the evaluation functional.

DEFINITION 4.2 (Evaluation Functional). Let $\mathcal{X}$ be a nonempty set and $\mathcal{H}$ be a Hilbert space of functions $f: \mathcal{X} \rightarrow \mathbb{R}$. For any $x \in \mathcal{X}$, the linear map $\tilde{\delta}_x: f \mapsto f(x)$ is called *evaluation functional*.

Notice that all evaluation functionals are linear.

We are mainly interested in spaces where the evaluation functional is not only linear, but also *continuous*.

DEFINITION 4.3 (Reproducing Kernel Hilbert Space). A Hilbert space $\mathcal{H}$ of functions $f: \mathcal{X} \rightarrow \mathbb{R}$ defined on a nonempty set $\mathcal{X}$ is said to be a *reproducing kernel Hilbert space* (RKHS) if $\tilde{\delta}_x: f \mapsto f(x)$ is continuous for all $x \in \mathcal{X}$.

A linear functional $f \in \mathcal{H}$ is continuous if and only if it is bounded, *i.e.* there exists a positive constant $c > 0$ such that

$$|f(x)| \leq c \|f\|_{\mathcal{H}}$$

for all $x \in \mathcal{X}$.

Reproducing kernel Hilbert spaces have the useful property that if two functions f and g are close to each other in the RKHS, then their function values $f(x)$ and $g(x)$ are also close (by continuity of the evaluation functional). Another desirable property which can be derived from continuous linear functionals is given by Riesz representation theorem, which tells us that every element in an RKHS can be represented as an inner product.

DEFINITION 4.4 (Riesz Representation). Let $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ be a Hilbert space (not necessary an RKHS) and $\mathcal{H}^*$ its dual of continuous linear functionals from $\mathcal{H}$ into the field $\mathbb{R}$ (Def. B.24). Every element in $\mathcal{H}^*$ is uniquely of the form $\langle \cdot, y \rangle$ for some $y \in \mathcal{H}$ (the representer) and hence all continuous linear functionals are *inner product functionals*. More precisely,¹ the map $\varphi: \mathcal{H} \rightarrow \mathcal{H}^*$ given by $y \mapsto \langle \cdot, y \rangle$ is a conjugate linear isometric isomorphism.

¹ Pedersen, *Analysis now*, Proposition 3.1.9.

So the inner product representation provides an isometric isomorphism between $\mathcal{H}$ and its dual. We can therefore switch between both spaces and choose the one which is more convenient for the problem at hand.

Notably, in an RKHS $\mathcal{H}$, the evaluation functional is by definition a continuous linear operator in $\mathcal{H}$. Therefore, for any $x \in \mathcal{X}$ and $f \in \mathcal{H}$ we can find a representer $k_x \in \mathcal{H}$ such that $f(x) = \langle f, k_x \rangle$. The converse does not hold in general *i.e.* inner product functionals on an RKHS are not necessarily point evaluations.

isomorphisms: bijective bounded linear operators with bounded inverse.

isometric isomorphisms: isomorphisms which preserve norms.

THERE is an interesting connection between RKHS and symmetric positive definite functions, called reproducing kernels. A symmetric function $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be *positive definite* if

$$\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(x_i, x_j) \geq 0$$

for all $\alpha_1, \dots, \alpha_n \in \mathbb{R}$, $x_1, \dots, x_n \in \mathcal{X}$ and $n \geq 1$. The function k is called *strictly positive definite* if, for mutually distinct $x_1, \dots, x_n \in \mathcal{X}$, the equality only holds when $\alpha_1 = \dots = \alpha_n = 0$. For the remainder of this work we refer to kernel functions which are symmetric, positive definite and real-valued if not stated otherwise.

DEFINITION 4.5 (*Reproducing Kernel*). Let $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ be a reproducing kernel Hilbert space on $\mathcal{X}$. A function $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a (*reproducing*) *kernel* if and only if for all $x \in \mathcal{X}$ and all $f \in \mathcal{H}$

$$k(x, \cdot) \in \mathcal{H} \quad \text{and} \\ \langle f, k(x, \cdot) \rangle = f(x),$$

where the last condition is called *reproducing property*.

The reproducing property tells us that the function evaluation of f can be written as an inner product between f and $k(x, \cdot)$ which qualifies $k(x, \cdot)$ as a representer of $f(x)$. In particular, the reproducing property implies that

$$\langle k(x, \cdot), k(y, \cdot) \rangle = k(x, y)$$

for all $x, y \in \mathcal{X}$ since $k(x, \cdot)$ is itself an element of $\mathcal{H}$.

Notably, it can be shown that k is a (reproducing) kernel if and only if it is symmetric and positive definite.² Moreover, if $\mathcal{H}$ is a reproducing kernel Hilbert space, then there *exists* a *unique* reproducing kernel k as stated by the Moore-Aronszajn theorem.³ In particular, every kernel defines a unique RKHS, and every RKHS has exactly one corresponding kernel.

DEFINITION 4.6 (*Feature Map*). Let $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ be an RKHS on $\mathcal{X}$ with kernel k . The map $\phi: \mathcal{X} \rightarrow \mathcal{H}$, $\phi: x \mapsto k(x, \cdot)$ with the property that

$$k(x, y) = \langle \phi(x), \phi(y) \rangle$$

is called (canonical) *feature map* or *Aronszajn map* of k .

THROUGHOUT this work we assume that the reproducing kernel Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ is *separable* such that ϕ is measurable and to avoid other technical difficulties.⁴

LEMMA 4.7 (*Separable RKHSs*). Let $\mathcal{X}$ be a separable topological space and k be a continuous kernel on $\mathcal{X}$. Then the RKHS of k is separable.⁵

We therefore make the following assumptions throughout the remainder of this thesis:

- (A-1) The input space $\mathcal{X}$ is a separable topological space and $\mathcal{X}$ is the Borel σ -algebra containing all open sets.
- (A-2) The reproducing kernel k on $\mathcal{X}$ is continuous.

By the following two lemmas, these assumptions are sufficient to ensure the measurability of the kernel and the feature map.

LEMMA 4.8 (*RKHSs of measurable kernels*). Let $(\mathcal{X}, \mathcal{X})$ be a measurable space and k be a kernel on $\mathcal{X}$ with RKHS $\mathcal{H}$. Then all $f \in \mathcal{H}$ are measurable if and only if $k(\cdot, x): \mathcal{X} \rightarrow \mathbb{R}$ is measurable for all $x \in \mathcal{X}$.⁶

² Steinwart and Christmann, *Support vector machines*, Theorem 4.16.

³ Berlinet et al., *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Theorem 3.

⁴ “[...] in Probability and Statistics, separability has powerful effects. For example, for separable metrizable spaces we have: $\mathbb{B}(\mathcal{X} \times \mathcal{X}) = \mathbb{B}(\mathcal{X}) \times \mathbb{B}(\mathcal{X})$ for the Borel σ -algebras, [...] Moreover, separable Hilbert spaces are Polish so that we have all the machinery of descriptive set theory available, regular conditional probabilities exist, Bochner integration is simple, and all probability measures on them are tight. Most importantly, by a classical result, [...], all separable Hilbert spaces are isomorphic with l^2 .”

— Owhadi et al., *Separability of reproducing kernel Hilbert spaces*.

⁵ Steinwart and Christmann, *Support vector machines*, Lemma 4.33.

⁶ *Ibid.*, Lemma 4.24.

LEMMA 4.9 (*Measurability of the canonical feature map*). Let $(\mathcal{X}, \mathcal{X})$ be a measurable space and k be a kernel on $\mathcal{X}$ such that $k(\cdot, x): \mathcal{X} \rightarrow \mathbb{R}$ is measurable for all $x \in \mathcal{X}$. If the RKHS $\mathcal{H}$ of k is separable, then both the canonical feature map $\phi: \mathcal{X} \rightarrow \mathcal{H}$ and $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ are measurable.⁷

⁷ Steinwart and Christmann, *Support vector machines*, Lemma 4.25.

The kernels we consider in this work are continuous, and hence measurable as stated in the following lemma.

LEMMA 4.10 (*Continuity and Measurability*). Any continuous mapping between two topological spaces $\mathcal{S}$ and $\mathcal{T}$ is measurable with respect to the Borel σ -fields $\mathbb{B}(\mathcal{S})$ and $\mathbb{B}(\mathcal{T})$.⁸

⁸ Kallenberg, *Foundations of modern probability*, Lemma 1.5.

If we would not require $\mathcal{H}$ to be separable, there would be no reproducing kernel as shown in the following corollary.

PROPOSITION 4.11 (*Non separable Hilbert spaces*). A non separable Hilbert space $\mathcal{H}$ of continuous functions on a separable topological space $\mathcal{X}$ has no reproducing kernel.⁹

⁹ Berline et al., *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Corollary 3.

4.2 Applications

Why are reproducing kernel Hilbert spaces relevant for anomaly detection and machine learning in general? Many classical algorithms make decisions based on the value of a linear combination of the input vector characteristics *e.g.* $g(\langle w, x \rangle)$, where w and x are elements in $\mathbb{R}^d$ and g is an arbitrary transformation function. A few examples of such algorithms are the perceptron,¹⁰ the (one-class) support vector machine,¹¹ logistic regression¹² and ordinary linear regression.

One can enhance the predictive performance of these methods by transforming the raw inputs x (non-linearly) into another space using the mapping $x \mapsto \varphi(x)$ and learn $g(\langle w, \varphi(x) \rangle)$. The function φ is then a so-called *feature function*. An example would be linear regression where we can take a polynomial instead of a line to fit the data using $\varphi: \mathbb{R} \rightarrow \mathbb{R}^4$, $\varphi(x) = [1 \ x \ x^2 \ x^3]$. The critical question is how to choose the function φ as this choice has a significant influence on the classifier's modeling and predictive performance.

Now assume that an algorithm can entirely be expressed in terms of inner products $\langle \varphi(x), \varphi(y) \rangle$ between the inputs x and y . Instead of choosing a transformation φ we can select a reproducing kernel k and replace the inner product by the kernel value, *i.e.* $\langle \varphi(x), \varphi(y) \rangle = k(x, y)$. This approach implicitly uses the kernel feature map ϕ as an input transformation *i.e.* $\varphi = \phi$ and has several advantages. First, it is possible to use input transformations which cannot be expressed explicitly. This happens if the kernel feature map has no analytic form or cannot be represented by today's computer architectures. Second, a kernel $k(x, y)$ can (intuitively) be seen as a similarity measure between its arguments x and y . This can be useful as it is sometimes easier to define how similar two object are than to design a suitable feature transformation. Furthermore, it is not necessary to make any assumptions about the input space $\mathcal{X}$ other than it being a set. For instance, we can work with images and strings directly and do not need a vector representation of these instances.

¹⁰ Rosenblatt, *The perceptron: a probabilistic model for information storage and organization in the brain*.

¹¹ Cortes et al., *Support-vector networks*.

¹² Cox, *The regression analysis of binary sequences*.

4.3 Mercer's Theorem

One of the fundamental results underlying reproducing kernel Hilbert space theory is Mercer's theorem which is stated in this section.

DEFINITION 4.12 (Integral Operator). Let $(\mathcal{X}, \mathcal{X})$ be a measurable space, ν a σ -finite Borel measure and $\mathcal{H}$ be a separable RKHS with reproducing kernel $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ satisfying $\|x \mapsto k(x, x)\|_{\mathcal{L}^2(\nu)} < \infty$. Recall that $L^2(\nu)$ is the space of equivalence classes of functions which are equal ν -a.s. For every $f \in L^2(\nu)$, the kernel induces a Hilbert-Schmidt operator

$$S_k: L^2(\nu) \rightarrow \mathcal{H}$$

$$S_k f = \int_{\mathcal{X}} k(x, \cdot) f(x) d\nu(x)$$

and $T_k = S_k^* S_k$ with $T_k: L^2(\nu) \rightarrow L^2(\nu)$ is the positive, self-adjoint, compact *integral operator* induced by k .

Since $S_k^* = \mathcal{H} \hookrightarrow L^2(\nu)$ [is the inclusion of $\mathcal{H}$ in $L^2(\nu)$], one may be tempted to think that the operators T_k and S_k are the same modulo their image space. However, recall that in general $L^2(\nu)$ does not consist of functions, and hence $S_k f(x)$ is defined, while $T_k f(x)$ is not. [...] In particular, evaluations $f(x)$ of elements $f \in L^p(\nu)$ are usually not defined, whereas for $f \in \mathcal{L}^p(\nu)$ such evaluations always make perfect sense.

— Steinwart and Christmann, *Support vector machines*

Following Steinwart and Christmann,¹³ the spectral theorem for self-adjoint compact operators tells us that there exists an at most countable orthonormal basis of $\mathcal{H}$ consisting of eigenfunctions $(e_i)_{i \in I}$ of T_k and corresponding eigenvalues $(\lambda_i)_{i \in I}$. Since T_k is a positive compact self-adjoint operator all eigenvalues are non-negative. The following theorem due to James Mercer¹⁴ plays an important role in the theory of integral equations, stochastic processes and functional analysis.

THEOREM 4.13 (Mercer's Theorem). Let k be a continuous reproducing kernel and $(\mathcal{X}, \mathcal{X}, \nu)$ be a compact measure space with finite Borel measure ν . If $(e_i)_{i \in I}$ and $(\lambda_i)_{i \in I}$ are the eigenfunctions and eigenvalues of T_k , then k can be represented as

$$k(x, y) = \sum_{i \in I} \lambda_i e_i(x) e_i(y) \quad x, y \in \mathcal{D}$$

where $\mathcal{D} = \text{supp } \nu$ and the convergence is absolute and uniform. This series is called *Mercer representation* of k .

We see that Mercer's theorem provides a series representation for continuous reproducing kernels on compact measurable spaces. In later chapters we work with the truncated version of this series expansions for kernel functions which has several desirable properties.

For certain kernel functions, such as the Laplacian, there are analytic expressions for the eigenvalues and eigenfunctions available.¹⁵ In Fig. 4.2 we illustrated the first six eigenfunctions of the Laplacian kernel and the Mercer representation. The plot on the right shows the difference between a series expansion truncated after ten terms and the kernel function.

Figure 4.1: Integral operator factorization as in Steinwart and Christmann, *Support vector machines*, p. 149.

¹³ *Ibid.*, Theorem 4.49.

¹⁴ Mercer, *Functions of positive and negative type, and their connection with the theory of integral equations*.

¹⁵ Ghanem et al., *Stochastic finite elements: a spectral approach*.

4.4 Bochner's Theorem

Salomon Bochner's theorem¹⁶ for translation invariant kernels provides a connection between positive definite functions and their Fourier transform. A kernel $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is called *translation invariant* if there exists a function $g: \mathcal{X} \rightarrow \mathbb{R}$ such that $k(x, y) = g(x - y)$ for all $x, y \in \mathcal{X}$. Hence the kernel value only depends on the difference between its arguments.

THEOREM 4.14 (Bochner). Let $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a translation invariant kernel. Then there exists a finite non-negative Borel measure¹⁷ λ on $\mathcal{X}$ such that

$$\begin{aligned} k(x, y) &= \int_{\omega} e^{i\langle \omega, x-y \rangle} \lambda(\omega) \\ &= \int_{\omega} \psi_{\omega}^*(x) \psi_{\omega}(y) \lambda(\omega) \end{aligned}$$

with $\psi_{\omega}(x) = e^{i\langle \omega, x \rangle}$. Here, ψ_{ω}^* is the conjugate transpose of ψ_{ω} and $\hat{\imath}$ is defined as $\sqrt{-1}$.

Informally, the translation invariant kernels have non-negative Fourier transforms.

4.5 Popular Kernel Functions

This section reviews some popular translation invariant kernel functions defined on $\mathbb{R}^d$. Their Fourier transform is known and can be found in literature *e.g.* the work of Sriperumbudur.¹⁸

THE *squared exponential* (SE) kernel is one of the most popular functions and has the form

$$\begin{aligned} k(x, y) &= \exp\left(-\frac{\|x - y\|_2^2}{2\sigma^2}\right) \quad \sigma > 0 \\ \lambda(\omega) &= (2\pi)^{-\frac{d}{2}} \sigma^d \exp\left(-\frac{\sigma^2 \|\omega\|_2^2}{2}\right) \end{aligned}$$

¹⁶ Bochner, *Vorlesungen über Fouriersche Integrale*.

¹⁷ A Borel measure is any measure λ defined on the Borel measurable space $(\mathcal{X}, \mathbb{B}(\mathcal{X}))$ (Def. A.1).

¹⁸ Sriperumbudur, *Reproducing kernel space embeddings and metrics on probability measures*.

Figure 4.3: Graphs of the five kernel functions introduced in this section.

where the parameter σ is known as the *characteristic length-scale*. Since this kernel function is infinitely differentiable and therefore very smooth, it may be unrealistic to model certain physical processes where the Matérn kernel can yield better results.

THE *Matérn* kernel is named after Bertil Matérn¹⁹ and defined by

¹⁹ Matérn, *Spatial variation*.

$$k(x, y) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} \|x - y\|_2}{\sigma} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu} \|x - y\|_2}{\sigma} \right) \quad \nu, \sigma > 0$$

$$\lambda(\omega) = \frac{2^{\frac{d}{2}+\nu} \Gamma(\nu + \frac{d}{2}) \nu^\nu}{\Gamma(\nu) \sigma^{2\nu}} \left(\frac{2\nu}{\sigma^2} + 4\pi^2 \|\omega\|_2^2 \right)^{\nu + \frac{d}{2}},$$

where Γ is the gamma function and K_ν is the modified Bessel function of the second kind. As $\nu \rightarrow \infty$, the Matérn kernel converges to the squared exponential.

ANOTHER special case of the Matérn kernel can be obtained by setting $\nu = 1/2$ is called *Laplacian* or *Ornstein-Uhlenbeck* kernel defined as

$$k(x, y) = \exp(-\sigma \|x - y\|_1) \quad \sigma > 0$$

$$\lambda(\omega) = \pi^{-d} \prod_{i=1}^d \frac{\sigma}{\sigma^2 + \omega_i^2},$$

where $\omega = (\omega_1, \dots, \omega_d)$.

AN example of a simple periodic function is the *sinc* kernel

$$k(x, y) = \prod_{i=1}^d \frac{\sin \sigma(x_i - y_i)}{x_i - y_i} \quad \sigma > 0$$

$$\lambda(\omega) = \left(\frac{\pi}{2} \right)^{\frac{d}{2}} \prod_{i=1}^d \mathbb{I}_{[-\sigma, \sigma]}(\omega_i),$$

where $\mathbb{I}$ denotes the indicator function (Def. A.13).

BASED on the early work of Hardy²⁰ the *inverse multiquadric* (IMQ) kernel is often used in geodesy, geophysics and mapping applications. Its function is given by

²⁰ Hardy, *Multiquadric equations of topography and other irregular surfaces*.

$$k(x, y) = (c^2 + \|x - y\|_2^2)^{-\beta} \quad c > 0, \beta > \frac{d}{2}$$

$$\lambda(\omega) = \frac{2^{1-\beta}}{(2\pi)^{\frac{d}{2}} \Gamma(\beta)} \left(\frac{\|\omega\|_2}{c} \right)^{\beta - \frac{d}{2}} K_{\frac{d}{2}-\beta}(c\|\omega\|_2),$$

where Γ and K_ν are defined as before for the Matérn kernel.

4.5.1 Combination of Kernel Functions

Existing kernel functions can be combined in various ways to create new ones. It is just necessary to show that the newly created kernel is again a positive definite function. In the following we review some approaches which are known to produce valid kernel functions.

LEMMA 4.15 (Sum and Scaling of Kernels). Let k_1, k_2 be kernel functions on $\mathcal{X}$ and $\alpha, \beta \geq 0$, then the *conical sum* $\alpha k_1 + \beta k_2$ is also a kernel function.

Since the sum of kernel functions is again a reproducing kernel, there is a corresponding RKHS which has the following characteristics.

LEMMA 4.16 (Sum of Reproducing Kernel Hilbert Spaces). Let k_1 and k_2 be reproducing kernels on $\mathcal{X}$ with corresponding RKHS $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively. Then $k = k_1 + k_2$ is the reproducing kernel of the RKHS

$$\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 = \{f_1 + f_2 \mid f_1 \in \mathcal{H}_1, f_2 \in \mathcal{H}_2\}$$

with norm $\|\cdot\|_{\mathcal{H}}$ defined by

$$\|f\|_{\mathcal{H}}^2 = \min \{ \|f_1\|_{\mathcal{H}_1}^2 + \|f_2\|_{\mathcal{H}_2}^2 \mid f = f_1 + f_2 \}$$

for all $f \in \mathcal{H}$.²¹

²¹ Berlinet et al., *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Theorem 5.

IN addition to the sum of kernel functions it is also possible to form their products. This product is formulated in terms of the tensor product of the respective RKHS.

DEFINITION 4.17 (Tensor Product of Hilbert spaces). The *tensor product* of the Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ is a Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_2$ defined by the bilinear form

$$\cdot \otimes \cdot : \mathcal{H}_1 \times \mathcal{H}_2 \rightarrow \mathcal{H}_1 \otimes \mathcal{H}_2$$

such that

$$\langle f_1 \otimes f_2, g_1 \otimes g_2 \rangle = \langle f_1, g_1 \rangle \langle f_2, g_2 \rangle$$

and the closed linear span of $\{f \otimes g \mid f \in \mathcal{H}_1, g \in \mathcal{H}_2\}$ is equal to $\mathcal{H}_1 \otimes \mathcal{H}_2$.

If both $\mathcal{H}_1$ and $\mathcal{H}_2$ are reproducing kernel Hilbert spaces, their tensor product is again an RKHS with a corresponding kernel function.

LEMMA 4.18 (*Product Kernels*). Let k_1, k_2 be kernel functions on $\mathcal{X}_1$ and $\mathcal{X}_2$ respectively, then

$$\begin{aligned} k_1 \otimes k_2: (\mathcal{X}_1 \times \mathcal{X}_2)^2 &\rightarrow \mathbb{R} \\ ((x_1, x_2), (y_1, y_2)) &\mapsto k_1(x_1, y_1)k_2(x_2, y_2) \end{aligned}$$

is a kernel function on $\mathcal{X}_1 \times \mathcal{X}_2$.²²

²² *Ibid.*, Theorem 13.

We will use such a product kernel to create a spatio-temporal kernel which operates on time series (§7). Consequently, it is immediate that if $\mathcal{X}_1 = \mathcal{X}_2$, then $(x, y) \mapsto k_1(x, y)k_2(x, y)$ is a kernel on $\mathcal{X}_1$.

ADDITIONALLY, it is possible to restrict the input space $\mathcal{X}$ to a nonempty subset $\mathcal{X}_0$ of $\mathcal{X}$ which yields an RKHS of functions which are themselves restricted to $\mathcal{X}_0$.

LEMMA 4.19 (*Restriction of the Input Space*). Let $\mathcal{H}$ be an RKHS on $\mathcal{X}$ with kernel k and let $\mathcal{X}_0$ be a nonempty subset of $\mathcal{X}$. The restriction k_0 of k to $\mathcal{X}_0$ is a reproducing kernel of the RKHS $\mathcal{H}_0$ with norm $\|\cdot\|_{\mathcal{H}_0}$ defined by

$$\|f_0\|_{\mathcal{H}_0} = \min\{\|f\|_{\mathcal{H}} \mid f \in \mathcal{H}, f|_{\mathcal{X}_0} = f_0\},$$

where $f|_{\mathcal{X}_0}$ is the restriction of f to $\mathcal{X}_0$.²³

²³ *Ibid.*, Theorem 6.

CLEARLY, there are many *families* of kernel functions and combinations thereof. So which kernel should be chosen for a particular problem? Selecting an appropriate kernel is often one of the main difficulties and there is no best generic answer to this question. The squared exponential kernel is often taken as a first choice. One of the main reasons is that the RKHS of the squared exponential kernel is dense in the space of continuous functions.²⁴ A consequence thereof is that for any continuous function $g: \mathcal{X} \rightarrow \mathbb{R}$, there exist a function $f \in \mathcal{H}$ such that

$$\|f - g\|_{\infty} \leq \epsilon$$

for all $\epsilon > 0$ i.e. f approximates g arbitrarily well. A kernel whose corresponding RKHS is dense in the space of continuous functions is called *universal* (Def. 5.6). However, we emphasize that other kernel functions pose a possibility to include domain and expert knowledge into the machine learning modeling process.

²⁴ Steinwart and Christmann, *Support vector machines*, Definition 4.52.

4.6 Summary

This chapter reviewed the theoretical foundations of reproducing kernel Hilbert spaces which will be essential in subsequent chapters. The feature map defined through the kernel function allows us to work directly in the corresponding RKHS which has several advantages. Bochner's and Mercer's theorems form the basis of approximate feature maps used in later chapters to improve the computational complexity and memory requirements of EXPOSE.

CHAPTER FIVE

Expected Similarity Estimation

SYNOPSIS · This chapter builds on the previous part which introduced reproducing kernel Hilbert spaces and their basic properties. We present the `EXPOSE` anomaly detector and show that it can be reformulated as an inner product between a feature map and the embedding of an input space distribution into an RKHS. We discuss how `EXPOSE` can efficiently be estimated from samples (§5.3), be processed in parallel and in distributed environments (§5.4). The experimental evaluation (§5.8) shows that `EXPOSE` can be used as an efficient anomaly detection algorithm with the same predictive performance as the best state of the art methods while being significantly faster than techniques with the same discriminative power. Some technical preliminaries appear in the appendices (§A) and (§B).

THE chapter content was partly published in

- M. Schneider, W. Ertel, and G. Palm (2015a).
“Expected Similarity Estimation for Large-Scale Anomaly Detection”. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2015)*. IEEE, pp. 1–8
- M. Schneider, W. Ertel, and F. Ramos (2016a).
“Expected Similarity Estimation for Large-Scale Batch and Streaming Anomaly Detection”. In: *Machine Learning* 105.3, pp. 305–333

IN THE FOLLOWING let X be a random variable taking values in the measurable input space $(\mathcal{X}, \mathcal{X})$. We assume that these instances (realizations) of $\mathbb{P}_X$ are mostly normal and contain only very few anomalies. Our goal is to collect information about this distribution such that we can estimate the likelihood (score) of an instance $y \in \mathcal{X}$ belonging to $\mathbb{P}_X$. Instances with a low likelihood are probably anomalies. If we assume that normal instances are more similar to each other than to anomalies, we can calculate such a score using the average similarity of y to the realizations under the distribution $\mathbb{P}_X$ as in the following.

DEFINITION 5.1 (*Expected Similarity Estimation*). The *expected similarity* of $y \in \mathcal{X}$ with respect to the (probability) distribution $\mathbb{P}_X$ is defined as

$$\eta(y) = \int_{\mathcal{X}} k(y, x) d\mathbb{P}_X(x),$$

where $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a continuous reproducing kernel.

Intuitively speaking, $\eta(y)$ defines the average similarity or closeness between y and X through $\mathbb{E}[k(y, X)]$ using the reproducing kernel k as a similarity measure. The evaluation of this integral makes predictions computationally demanding. The aim of this chapter is therefore a more efficient representation of `EXPPOSE` as $\eta(y) = \langle \phi(y), \mu[\mathbb{P}_X] \rangle$, *i.e.* an inner product between the feature map $\phi(y)$ and the *kernel mean map* $\mu[\mathbb{P}_X]$ of the distribution $\mathbb{P}_X$. This reformulation is of central importance and the key to `EXPPOSE`'s efficiency.

5.1 The Kernel Mean Embedding

The *kernel embedding of distributions* or *kernel mean map*¹ is a popular technique to handle probability measures. It was originally introduced around 1975 and studied in the early 80's.² The underlying idea is to embed a probability distribution or a measure into a reproducing kernel Hilbert space (RKHS) where it is then in a more accessible form and can be manipulated efficiently. The kernel embedding approach has been successfully applied to a wide variety of machine learning applications. For example, Gretton et al. use the kernel mean map to perform statistical hypotheses testing in high-dimensional spaces. They formulate a two-sample test using the difference between the kernel embeddings of two distributions as a test statistic. This statistic is called the *maximum mean discrepancy* (MMD).³ Gretton et al. apply the MMD to the problem of independence testing using the *Hilbert-Schmidt independence criterion* (HSIC) as associated test statistic. The HSIC can also be applied to feature selection by choosing a subset of features which maximizes the dependence between data and labels.⁴ Furthermore, it can be used to derive a non-parametric independence test for random processes.⁵ A new representation of *hidden Markov models* (HMM) for structured and non-Gaussian continuous observation distributions can be defined in a way such that the model updates can be performed entirely in the RKHS in which these distributions are embedded.⁶ Representing prior and conditional probabilities through the kernel mean map allows the *kernel Bayes' rule* to perform inference of the posterior distribution in the same RKHS without an explicit parametric form.⁷ Song, Gretton, et al. derived a kernelized belief propagation algorithm using the kernel embedding of conditional distributions.⁸

THE kernel mean map was primarily defined for probability distributions, but can be extended for arbitrary Borel measures. We therefore introduce the following notations. We will denote the set of all finite Borel measures on the input space $(\mathcal{X}, \mathcal{X})$ by $\mathcal{B}(\mathcal{X})$ and the set of all Borel probability measures by $\mathcal{B}^1(\mathcal{X})$.

DEFINITION 5.2 (Kernel Embedding). The *kernel embedding* or *kernel mean map* of a finite Borel measure $\nu \in \mathcal{B}(\mathcal{X})$ is given in terms of the Bochner integral⁹

$$\begin{aligned} \mu: \mathcal{B}(\mathcal{X}) &\rightarrow \mathcal{H} \\ \nu &\mapsto \mu[\nu] = \int_{\mathcal{X}} k(\cdot, x) d\nu(x), \end{aligned}$$

where $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is the associated reproducing kernel.

¹ Smola et al., *A Hilbert space embedding for distributions*; Sriperumbudur, Gretton, et al., *Injective Hilbert space embeddings of probability measures*.

² "Some of the first studies considering inner products on sets of measures and applications in Probability and Statistics were carried out in the years 1975-1980 at the University of Lille under the leadership of Bosq. Guilbart (1977-1979) studied the relationships between reproducing kernels and inner products on the space $\mathcal{M}$ of signed measures on a measurable space. He exploited the embedding of $\mathcal{M}$ in an RKHS and characterized the inner products inducing the weak topology on sets of measures. Guilbart also proved the continuity of projections with respect to the reproducing kernel defining the inner product. He proved a Glivenko-Cantelli theorem that he applied to estimation and hypothesis testing." — Berlinet et al., *Reproducing Kernel Hilbert Spaces in Probability and Statistics*.

³ Gretton et al., *A kernel method for the two-sample-problem*.

⁴ Song, Smola, et al., *Feature selection via dependence maximization*.

⁵ Chwialkowski et al., *A Kernel Independence Test for Random Processes*.

⁶ Song, Boots, et al., *Hilbert space embeddings of hidden Markov models*.

⁷ Fukumizu, Song, et al., *Kernel Bayes' rule: Bayesian inference with positive definite kernels*.

⁸ Song, Huang, et al., *Hilbert space embeddings of conditional distributions with applications to dynamical systems*.

⁹ The Bochner integral is given in Def. 5.8.

We see that μ maps every measure or distribution to a point in the RKHS $\mathcal{H}$ and we will shortly explain under what conditions this map is injective.

An interesting example is the embedding of the Dirac measure δ_x . The equality

$$\int f(t) d\delta_x(t) = f(x)$$

holds for any function $f: \mathcal{X} \rightarrow \mathbb{R}$. Thus integrating with respect to the Dirac measure is equivalent to function application. If $f \in \mathcal{H}$ we can use the reproducing property of k and write $f(x) = \langle f, k(\cdot, x) \rangle$. Therefore

$$\int f(t) d\delta_x(t) = \int \langle f, k(\cdot, t) \rangle d\delta_x(t) = \left\langle f, \int k(\cdot, t) d\delta_x(t) \right\rangle$$

and hence the embedding $\delta_x \mapsto \int k(\cdot, t) d\delta_x(t) = k(\cdot, x)$ is the representer of $f(x)$ in $\mathcal{H}$.

A RELEVANT QUESTION for us is under what conditions $\mu[\nu]$ exists. This is subject of the following lemma.

LEMMA 5.3. Let k be a measurable kernel on $\mathcal{X}$ and $\nu \in \mathcal{B}(\mathcal{X})$. If

$$\int_{\mathcal{X}} \sqrt{k(x, x)} d\nu(x) < \infty,$$

then $\mu[\nu]$ exists and is an element of the RKHS $\mathcal{H}$.¹⁰

The inequality in lemma 5.3 is weaker than the assumption of k being bounded, which is $|k(x, y)| < \infty$ for all $x, y \in \mathcal{X}$. This follows from the facts that ν is a finite measure and that k is bounded if and only if $\|k\|_{\infty} = \sup_{x \in \mathcal{X}} \sqrt{k(x, x)} < \infty$.¹¹ If not stated otherwise we use the following assumption:

(A-3) For all kernel functions k and measures $\nu \in \mathcal{B}(\mathcal{X})$ it holds that $\int_{\mathcal{X}} \sqrt{k(x, x)} d\nu(x) < \infty$.

Consequently, we can assume that $\mu[\nu]$ exists for all measures ν considered in this work.

¹⁰ Sejdinovic et al., *Equivalence of distance-based and RKHS-based statistics in hypothesis testing*, Page 8.

¹¹ Steinwart and Christmann, *Support vector machines*, Section 4.3.

5.1.1 Maximum Mean Discrepancy

The maximum mean discrepancy was introduced to calculate the distance between the kernel embeddings of two measures which can be used as a test statistic for two-sample tests.¹² The discrepancy is defined in terms of the Hilbert space norm as follows.

DEFINITION 5.4 (*Maximum mean discrepancy*). Let $\mathbb{P}$ and $\mathbb{Q}$ be elements in $\mathcal{B}^1(\mathcal{X})$. The RKHS distance between the embeddings of probability measures $\mathbb{P}$ and $\mathbb{Q}$ given by

$$\gamma(\mathbb{P}, \mathbb{Q}) = \|\mu[\mathbb{P}] - \mu[\mathbb{Q}]\|_{\mathcal{H}}$$

is called *maximum mean discrepancy* (MMD).

We use the MMD solely for visualization purposes as for example in Fig. 5.1 where we show the kernel embedding of several normal distributions with

¹² Gretton et al., *A kernel method for the two-sample-problem*.

mean in the interval $[-5, 5]$ and variance in the interval $[0.1, 5]$. We use the squared exponential kernel for which closed-form expressions of the MMD are available.¹³ Since it is not possible to plot the reproducing kernel Hilbert space $\mathcal{H}$, we calculate the pairwise distances between the embedded distributions and apply multidimensional scaling.¹⁴ This allows us to project the embeddings into $\mathbb{R}^2$ using the first two principal coordinates. Notably, distributions with the same mean get projected close to each other, forming ray like clusters. The higher the variance the less influential is the mean as a criterion to distinguish two normal distributions. This property can be observed from the projections closer to the center.

IT IS WORTH noting that the MMD is a pseudo metric since $\gamma(\mathbb{P}, \mathbb{Q}) = 0$ does not necessary imply $\mathbb{P} = \mathbb{Q}$. This is obviously the case if the kernel embedding is not injective, *i.e.* two different distributions $\mathbb{P}$ and $\mathbb{Q}$ get mapped to the same value. If the embedding is injective, we say the kernel is *characteristic*.¹⁵

DEFINITION 5.5 (Characteristic Kernel). A bounded measurable kernel k is *characteristic* to a set $\mathcal{D}(\mathcal{X}) \subset \mathcal{B}^1(\mathcal{X})$ of probability measures defined on $\mathcal{X}$ if for all $\mathbb{P} \in \mathcal{D}(\mathcal{X})$

$$\mathbb{P} \mapsto \mu[\mathbb{P}] = \int_{\mathcal{X}} k(\cdot, x) d\mathbb{P}(x)$$

is injective. k is simply said to be characteristic if it is characteristic to $\mathcal{B}^1(\mathcal{X})$. The RKHS $\mathcal{H}$ induced by such a k is called a characteristic RKHS.

Conditions under which kernels are characteristic have been studied by various authors.¹⁶ Notably, there is also a strong connection between universal and characteristic kernels *i.e.* the embedding is injective if the kernel is universal.

¹³ Sriperumbudur, *Reproducing kernel space embeddings and metrics on probability measures*, Example 5.16.

¹⁴ Torgerson, *Multidimensional scaling*.

¹⁵ Sriperumbudur, *Reproducing kernel space embeddings and metrics on probability measures*, Definition 3.6.

¹⁶ Fukumizu, Gretton, et al., *Characteristic kernels on groups and semigroups*; Sriperumbudur, Gretton, et al., *Injective Hilbert space embeddings of probability measures*.

DEFINITION 5.6 (*Universal Kernel*). A continuous kernel k on a compact metric space $\mathcal{X}$ is called *universal* if the RKHS $\mathcal{H}$ of k is dense in the space of continuous functions $g: \mathcal{X} \rightarrow \mathbb{R}$.¹⁷

¹⁷ Steinwart and Christmann, *Support vector machines*, Definition 4.52.

This implies that for *any* continuous function g there exists an element f in the RKHS $\mathcal{H}$ which is arbitrarily close to g , that is $\|g - f\|_\infty \leq \epsilon$ for all $\epsilon > 0$. Examples of universal kernels are the widely used squared exponential and Laplacian kernel functions.

WE EMPHASIZE that the characteristic property is not relevant for the EXPOSE anomaly detection algorithm and we therefore explicitly allow kernel mean embeddings which are not injective.

5.2 Embedding of EXPOSE

After the review of kernel mean embeddings we can now continue to introduce the central theorem of our work which allows us to reformulate EXPOSE as an inner product between the feature map $\phi(y)$ and the kernel mean map $\mu[\mathbb{P}_X]$ of $\mathbb{P}_X$.

THEOREM 5.7. Let $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ be an RKHS on a topological space $\mathcal{X}$ with Borel σ -algebra $\mathcal{X}$ and continuous reproducing kernel $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. The expected similarity of $y \in \mathcal{X}$ with respect to the distribution $\mathbb{P}_X$ can be expressed as

$$\begin{aligned} \eta(y) &= \int_{\mathcal{X}} k(y, x) d\mathbb{P}_X(x) \\ &= \langle \phi(y), \mu[\mathbb{P}_X] \rangle, \end{aligned}$$

where $\mu[\mathbb{P}_X]$ is the kernel embedding of $\mathbb{P}_X$.

This reformulation has several desirable properties *e.g.* we see how the EXPOSE classifier can make prediction in *constant* time: After the kernel mean map $\mu[\mathbb{P}_X]$ of the input space distribution $\mathbb{P}_X$ is learned, EXPOSE only needs to evaluate a single inner product in $\mathcal{H}$ to calculate the score. However, there are some crucial aspects to consider. Since, in Hilbert spaces, integrals and continuous linear forms are not in general interchangeable we need some additional tools to prove this theorem.

DEFINITION 5.8 (*Strong Integral*). Let $(\mathcal{X}, \mathcal{X}, \nu)$ be a finite measure space and let $\phi: \mathcal{X} \rightarrow \mathcal{H}$ be a measurable map. Then ϕ is *strong integrable* (*Bochner integrable*) over a set $\mathcal{D} \in \mathcal{X}$ if and only if its norm $\|\phi\|$ is Lebesgue integrable over $\mathcal{D}$, that is,

$$\int_{\mathcal{D}} \|\phi\| d\nu < \infty.$$

If ϕ is strong integrable over each $\mathcal{D} \in \mathcal{X}$ we say that ϕ is *strong integrable*.¹⁸

¹⁸ Aliprantis et al., *Infinite dimensional analysis: a hitchhiker's guide*, Theorem II.44.

DEFINITION 5.9 (*Weak Integral*). Let $(\mathcal{X}, \mathcal{X}, \nu)$ be a finite measure space. A function $\phi: \mathcal{X} \rightarrow \mathcal{H}$ is *weakly integrable* over a set $\mathcal{D} \in \mathcal{X}$ if there exists some $\lambda \in \mathcal{H}$ satisfying

$$\langle f, \lambda \rangle = \int_{\mathcal{D}} \langle f, \phi(x) \rangle d\nu(x)$$

for each $f \in \mathcal{H}$. The weak integral is denoted by

$$\lambda = \oint_{\mathcal{D}} \phi d\nu$$

and the unique element $\lambda \in \mathcal{H}$ is called *weak integral* of ϕ over $\mathcal{D}$. If the integral exists for each $\mathcal{D} \in \mathcal{X}$ we say that ϕ is *weakly integrable*.¹⁹

The weak integral is also known as *Pettis integral* and the following lemma shows its connection to the strong integral.

LEMMA 5.10. If ϕ is strong (Bochner) integrable then ϕ is weak (Pettis) integrable and the two integrals coincide.²⁰

We are now in the position to proof theorem 5.7.

PROOF OF THEOREM 5.7. By definition of the feature map ϕ we have

$$\int_{\mathcal{X}} k(x, y) d\nu(x) = \int_{\mathcal{X}} \langle \phi(y), \phi(x) \rangle d\nu(x).$$

From (A-3) it follows that

$$\int_{\mathcal{X}} \|\phi(x)\| d\nu(x) = \int_{\mathcal{X}} \sqrt{k(x, x)} d\nu(x) < \infty$$

and therefore ϕ is strongly integrable and hence weakly integrable (lemma 5.10).

By definition of the weak integral, it holds that

$$\begin{aligned} \int_{\mathcal{X}} \langle \phi(y), \phi(x) \rangle d\nu(x) &= \left\langle \phi(y), \oint_{\mathcal{X}} \phi d\nu \right\rangle \\ &= \left\langle \phi(y), \int_{\mathcal{X}} \phi d\nu \right\rangle \\ &= \langle \phi(y), \mu[\nu] \rangle \end{aligned}$$

for all $\nu \in \mathcal{B}(\mathcal{X})$ and especially for $\nu = \mathbb{P}_{\mathcal{X}}$. $\square$

An alternative proof can be derived by an argument which shows that the linear functional $T_{\mathbb{P}_{\mathcal{X}}}: \mathcal{H} \rightarrow \mathbb{R}, f \mapsto \int_{\mathcal{X}} f(x) d\mathbb{P}_{\mathcal{X}}(x)$ is bounded.²¹ This follows from

$$\begin{aligned} \left| \int_{\mathcal{X}} f d\mathbb{P}_{\mathcal{X}} \right| &\leq \int_{\mathcal{X}} |f(x)| d\mathbb{P}_{\mathcal{X}}(x) \\ &= \int_{\mathcal{X}} |\langle f, k(\cdot, x) \rangle| d\mathbb{P}_{\mathcal{X}}(x) \\ &\leq \int_{\mathcal{X}} \sqrt{k(x, x)} \|f\|_{\mathcal{H}} d\mathbb{P}_{\mathcal{X}}(x) \end{aligned}$$

and hence

$$\|T_{\mathbb{P}_{\mathcal{X}}}\| = \sup_{f \in \mathcal{H}, f \neq 0} \frac{|T_{\mathbb{P}_{\mathcal{X}}}[f]|}{\|f\|_{\mathcal{H}}} < \infty$$

whenever the kernel k is bounded. By the Riesz representation theorem (Def. 4.4) there exists a unique $\lambda \in \mathcal{H}$ such that $T_{\mathbb{P}_{\mathcal{X}}}[\phi(y)] = \langle \phi(y), \lambda \rangle$ for all $\phi(y) \in \mathcal{H}$ and $y \in \mathcal{X}$ which implies $\lambda = \int_{\mathcal{X}} \phi d\mathbb{P}_{\mathcal{X}} = \mu[\mathbb{P}_{\mathcal{X}}]$.

¹⁹ Aliprantis et al., *Infinite dimensional analysis: a hitchhiker's guide*, Section 11.10.

²⁰ *Ibid.*, Theorem 11.50.

²¹ Sriperumbudur and Gretton, *Hilbert space embeddings and metrics on probability measures*, Theorem 1.

ANOTHER INTERESTING insight can be gained if we look again at the embedding of the Dirac measure

$$\mu[\delta_x] = \int_{\mathcal{X}} k(\cdot, t) d\delta_x(t) = \phi(x).$$

Inserted into theorem 5.7 we obtain

$$\begin{aligned} \eta(y) &= \langle \phi(y), \mu[\mathbb{P}_X] \rangle \\ &= \langle \mu[\delta_y], \mu[\mathbb{P}_X] \rangle \\ &= \int_{\mathcal{X}} k(\cdot, \cdot) d(\delta_y \otimes \mathbb{P}_X) \end{aligned}$$

which is the integral of the kernel k with respect to $\delta_y \otimes \mathbb{P}_X$ — the product measure of δ_y and $\mathbb{P}_X$.

5.3 Sample Estimates

As argued before, in anomaly detection we cannot assume to know the input space distribution $\mathbb{P}_X$. Instead we have access to a sample $Z = z$ with $z = (x_1, \dots, x_n)$ from the sample space $(\mathcal{Z}, \mathcal{Z}, \mathbb{P}_Z)$.

THE empirical measure after n draws

$$\mathbb{P}_n = \frac{1}{n} \sum_{i=1}^n \delta_{X_i},$$

associated with the random sample $Z = (X_1, \dots, X_n)$ can act as a proxy for $\mathbb{P}_X$. Consequently, $\mathbb{P}_n$ is used to construct an approximation $\mu[\mathbb{P}_n]$ of $\mu[\mathbb{P}_X]$ as

$$\mu[\mathbb{P}_n] = \int_{\mathcal{X}} \phi(t) d\mathbb{P}_n(t) = \frac{1}{n} \sum_{i=1}^n \phi(X_i)$$

which is called *empirical kernel embedding*.²² We also know that the mapping

$$\begin{aligned} (\Omega, \mathcal{F}, \mathbb{P}) &\rightarrow (\mathcal{H}, \mathcal{H}) \\ \omega &\mapsto \frac{1}{n} \sum_{i=1}^n \phi(X_i(\omega)) \end{aligned}$$

is measurable since ϕ is measurable (lemma 4.9). The empirical kernel embedding is reasonable a choice since the strong law of large numbers implies that $\mu[\mathbb{P}_n]$ converges almost surely to $\mu[\mathbb{P}]$ as n tends to infinity. Moreover, it is possible to derive a Hilbert space version of the Central Limit Theorem such that the sequence $(\Lambda_n)_{n \in \mathbb{N}}$ with

$$\Lambda_n = \sqrt{n}(\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X])$$

converges weakly to a centered Gaussian variable.²³

²² Smola et al., *A Hilbert space embedding for distributions*.

²³ Berlinet et al., *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Theorem 108.

WE MAY ASK how well $\mu[\mathbb{P}_n]$ approximates $\mu[\mathbb{P}_X]$ for any finite $n \in \mathbb{N}$. This question is answered by theorem 8.1 (Chapter 8) in terms of a *concentration inequality* which tells us that for any $\epsilon > 0$

$$\mathbb{P}(\|\mu[\mathbb{P}_X] - \mu[\mathbb{P}_n]\|_{\mathcal{H}} \geq \epsilon) \leq 2 \exp\left(-\frac{n\epsilon^2}{8d^2}\right)$$

holds if $\|\phi(x)\| \leq d$ for all $x \in \mathcal{X}$. This inequality bounds the probability that the empirical kernel embedding $\mu[\mathbb{P}_n]$ differs from the actual embedding $\mu[\mathbb{P}_X]$ using n independent samples by more than ϵ . This is (up to some constants) a Hilbert and Banach space version of Hoeffding's inequality.²⁴

²⁴ Hoeffding, *Probability inequalities for sums of bounded random variables*.

5.3.1 The EXPoSE Sample Estimator

The empirical kernel embedding $\mu[\mathbb{P}_n]$ allows us to formulate an empirical version of EXPoSE by substituting $\mu[\mathbb{P}_X]$ with $\mu[\mathbb{P}_n]$ whenever the distribution $\mathbb{P}_X$ is not directly accessible and has to be estimated from samples $z = (x_1, \dots, x_n)$. We overload notation and write

$$\begin{aligned} \eta(y) &= \langle \phi(y), \mu[\mathbb{P}_n] \rangle \\ &= \left\langle \phi(y), \frac{1}{n} \sum_{i=1}^n \phi(x_i) \right\rangle \end{aligned}$$

for the (empirical) EXPoSE anomaly detector. The estimation of the empirical kernel embedding $\mu[\mathbb{P}_n]$ is responsible for the *linear* training time complexity of EXPoSE. Since the learning stage only involves the empirical kernel embedding, we simply refer to $\mu[\mathbb{P}_n]$ as the *EXPoSE model*.

This is obviously not improvable without further assumptions. In the subsequent chapter we argue that a high accuracy sample estimate of $\mu[\mathbb{P}_X]$ by $\mu[\mathbb{P}_n]$ is not necessary to achieve a good predictive performance. Instead we show that it is possible to calculate an approximation which is arbitrarily close to $\mu[\mathbb{P}_X]$ with a predefined accuracy, independent of the dataset size. This can be achieved by means of convex optimization.

Figure 5.2: The EXPoSE predictor estimated from samples using a squared exponential kernel. Query points close to the data obtain a higher score than query points with no training data nearby.

ONE of the important observations is that this representation through the kernel mean map requires no parametric assumptions or analytic description of the input space distribution. This is an advantage over other statistical approaches that try to approximate the distribution directly with parametric models. Nevertheless, the kernel embedding captures all characteristics of the distribution or the mapping would not be (under certain conditions) injective.

5.4 Parallel & Distributed Processing

Parallel and distributed data processing is the key to scalable machine learning algorithms. The formulation of EXPoSE as $\eta(y) = \langle \phi(y), \mu[\mathbb{P}_n] \rangle$ is especially appealing for this kind of operations as we can apply an SPMD (single program, multiple data) technique to achieve parallelism. One of the first programming paradigms on this line is Google's *MapReduce* for processing large data sets on a cluster.²⁵

²⁵ Dean et al., *MapReduce: simplified data processing on large clusters*.

Assume a partition of the dataset $z = (x_1, \dots, x_n)$ into $m \leq n$ distinct collections $s_1, \dots, s_m$ which can be distributed on different computational nodes. Obviously, the feature map ϕ can be applied in parallel to all instances in z . We also note that the partial sums

$$p(s_i) = \sum_{x \in s_i} \phi(x)$$

can be calculated without any communication or data-sharing across concurrent computations. Solely the partial sums $p(s_i)$, which are elements of $\mathcal{H}$, need to be transmitted and combined as

$$\mu[\mathbb{P}_n] = \frac{1}{n} \sum_{i=1}^m p(s_i)$$

by a central processing node. Here, the following requirements on ϕ and p play an important role in most concurrent and parallel programming models.

Determinism A deterministic or *stateless* function is a function in the mathematical sense. Given the same inputs it produces the same outputs on every call. If a function is not deterministic, it is called a *non-deterministic* function. Both ϕ and p are deterministic.

Commutativity A function is commutative if a change of the order of the operands does not change the result. This concerns the function $p(s_i)$, where order of elements in the sets s_i do affect the output. Also the result of the function $\mu[\mathbb{P}_n]$ does not depend on the partition of the dataset or the order in which the partial sums are combined.

Associativity The function p is also associative since

$$p(\{ p(\{u, v\}), w \}) = p(\{ u, p(\{v, w\}) \}) \quad \forall u, v, w \in \mathcal{H}$$

and therefore can be cascaded easily.

Figure 5.3: EXPoSE in a distributed data processing environment.

We see that it is not only straightforward to calculate the EXPoSE model $\mu[\mathbb{P}_n]$ in a distributed way, but we can also achieve this with no additional computational costs and still obtain the exact same result as the original proposed sequential approach.

IN SUMMARY, this section introduced the EXPoSE anomaly detector and discussed its properties. We argued that a single prediction takes *constant* time while estimating the model $\mu[\mathbb{P}_n]$ can be done in *linear* time and with *constant* memory.

5.5 Approximate Feature Maps

We showed in the previous part how `EXPPOSE` can be expressed as the inner product $\langle \phi(y), \mu[\mathbb{P}_X] \rangle$ between the kernel mean map and the feature map of a query point $y \in \mathcal{X}$. However, there is in general no explicit representation of ϕ and hence $\mu[\mathbb{P}_X]$ available since both are elements in a Hilbert space of functions defined through $\phi(x) = k(\cdot, x)$. We therefore aim to find an *approximate feature map* $\hat{\phi}$ for a given kernel function k such that

$$k(x, y) \approx \langle \hat{\phi}(x), \hat{\phi}(y) \rangle$$

and $\hat{\phi}: \mathcal{X} \rightarrow \mathbb{R}^r$ for some $r \in \mathbb{N}$. In the following we present several approaches which allow us to derive approximate feature maps for various kernel functions.

5.5.1 Random Kitchen Sinks

An efficient way to create an approximate feature map $\hat{\phi}$ from a given kernel k is known as *random kitchen sinks*²⁶ (RKS). The random kitchen sinks approximation is based on Bochner's theorem (§4.4) for translation invariant kernels such as the Laplacian, Matérn, squared exponential and others. We recall that Bochner's theorem allows us to write translation invariant kernels as

$$k(x, y) = \int_{\omega} \psi_{\omega}^*(x) \psi_{\omega}(y) \lambda(\omega)$$

with ψ_{ω} defined as

$$\psi_{\omega}(x) = e^{i\langle \omega, x \rangle}$$

and finite non-negative Borel measure λ . Using an appropriate scaling, we can turn λ into a probability measure $\tilde{\lambda}$ and use a Monte Carlo approximation of the integral to estimate the expression above as

$$k(x, y) = \mathbb{E} \left[\frac{1}{r} \sum_{i=1}^r \psi_{\omega_i}^*(x) \psi_{\omega_i}(y) \right] \quad \text{with} \quad \omega_i \sim \tilde{\lambda}$$

from $r \in \mathbb{N}$ samples. Using the vector notation

$$\hat{\phi}(x) = \frac{1}{\sqrt{r}} \left[\psi_{\omega_1}(x) \dots \psi_{\omega_r}(x) \right]^T \in \mathbb{R}^r$$

we obtain

$$k(x, y) = \mathbb{E} [\langle \hat{\phi}(x), \hat{\phi}(y) \rangle].$$

Hence we can use random kitchen sinks to find a r dimensional feature space that approximates any shift-invariant kernel.

²⁶ Rahimi et al., *Random features for large-scale kernel machines, Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning.*

WE HAVE SEEN that for kernels such as the squared exponential

$$k(x, y) = \exp\left(-\frac{\|x - y\|_2^2}{2\sigma^2}\right) \quad \sigma > 0$$

the measure λ can be found with the help of the inverse Fourier transform, which shows that the ω_i are normally distributed according to $\mathcal{N}(0, \sigma^2)$. This allows us to calculate the approximate feature maps $\hat{\phi}(x)$ as

$$\hat{\phi}(x) = \frac{1}{\sqrt{r}} \exp(iZx)$$

with

$$Z_{ij} \sim \mathcal{N}(0, \sigma^2) \quad \text{and} \quad Z \in \mathbb{R}^{r \times d}$$

where d is the input space dimension (see algorithm 5.1). Approximate feature maps for other kernel functions can be derived in a similar way.

REQUIRE:

- 1: r : the number of feature expansion
- 2: d : the input space dimension
- 3: σ^2 : the kernel bandwidth

ALGORITHM:

- 4: FOR $i \leftarrow 1, 2, \dots, r$ DO
 - 5: Sample entries of $\omega_i \in \mathbb{R}^d$ from $\mathcal{N}(0, \sigma^2)$
 - 6: Set $\hat{\phi}_i(x) = \frac{1}{\sqrt{r}} \exp(i\langle \omega_i, x \rangle)$
 - 7: RETURN $\hat{\phi} = [\hat{\phi}_1 \dots \hat{\phi}_r]$
-

The parameter $r \in \mathbb{N}$ determines the number of kernel expansions (Monte Carlo samples) and is typically around 20 000. Larger r result in better kernel approximations as the Monte Carlo estimate becomes more accurate (Fig. 5.4). A theoretical guarantee that

$$\|\langle \hat{\phi}(x), \hat{\phi}(y) \rangle - k(x, y)\|_{D \times D} \rightarrow 0 \quad \text{as} \quad r \rightarrow \infty$$

for any compact set $D \subset \mathbb{R}^d$ is provided by Rahimi et al.²⁷ More precisely, let $\rho(D) = \sup\{d(x, y) \mid x, y \in D\}$ for any subset D of a metric space (M, d) , then the authors derived a convergence rate of $\mathcal{O}(\rho(D) \sqrt{r^{-1} \log r})$. Sriperumbudur and Szabó²⁸ further improved this bound to

$$\mathcal{O}\left(\sqrt{r^{-1} \log \rho(D)}\right)$$

for nonempty compact sets $D \subset \mathbb{R}^d$.

Algorithm 5.1: Approximate Feature Maps for the squared exponential kernel using the random kitchen sinks approach.

²⁷ Rahimi et al., *Random features for large-scale kernel machines*, Claim 1.

²⁸ Sriperumbudur and Szabó, *Optimal rates for random Fourier features*, Theorem 1.

²⁹ Le et al., *Fastfood: approximating kernel expansions in loglinear time*.

5.5.2 Approximating Kernel Expansions

Le et al. proposed an approximation of the matrix Z which comprises the samples ω such that the product Zx can be calculated in $\mathcal{O}(r \log d)$ time complexity instead of $\mathcal{O}(rd)$ while requiring only $\mathcal{O}(r)$ rather than $\mathcal{O}(rd)$ storage.²⁹ This approach exploits the idea that a random matrix with normally distributed entries behaves similar to a Hadamard matrix when applied to a diagonal matrix with normally distributed entries. Hadamard matrices do not require any storage and furthermore, the multiplication of a $d \times d$ Hadamard matrix with a vector can be performed in $d \log d$ operations similar to a discrete Fourier transform.

Figure 5.4: A comparison between the values calculated by a squared exponential kernel $k(x, y)$ and the RKs approximation $\langle \hat{\phi}(x), \hat{\phi}(y) \rangle$ for random images of the MNIST dataset. The individual plots show how the number of kernel expansion r affect the approximation quality. Color indicates the density.

IN THIS section we only sketch the relevant ideas and refer to the original literature for a more detailed explanation. The proposed algorithm requires that the input space dimension is $d = 2^l$ for some $l \in \mathbb{N}$. However, one can simply pad the input vector with zeros until this requirement holds. We then construct $\frac{r}{d}$ matrices V_i for $1 \leq i \leq \frac{r}{d}$ like

$$V_i = \frac{1}{\sigma\sqrt{d}} \text{SHGITHB} \in \mathbb{R}^{d \times d}$$

where

- $\Pi \in \{0, 1\}^{d \times d}$ is a random permutation matrix.
- $H \in \mathbb{R}^{d \times d}$ is the Walsh-Hadamard matrix.
- $B \in \mathbb{R}^{d \times d}$ has random $\{-1, 1\}$ entries on its main diagonal.
- $G \in \mathbb{R}^{d \times d}$ is a diagonal matrix with normally distributed $(\mathcal{N}(0, 1))$ entries.
- S is a diagonal scaling matrix with $S_{ii} = s_i \|G\|_{\text{Frob}}^{-\frac{1}{2}}$ and $s_i \sim \chi_d^2$.

These matrices V_i are then stacked as

$$Z \approx V = \begin{bmatrix} V_1 & \dots & V_{\frac{r}{d}} \end{bmatrix}^T$$

to resemble the original matrix Z . The approximate feature map is then

$$\hat{\phi}(x) = \frac{1}{\sqrt{r}} \exp(iVx),$$

where the RKS matrix Z is replaced by V .

The convergence rate closely follows the original random kitchen sink rate of $\mathcal{O}(\rho(D) \sqrt{r^{-1} \log r})$. It has to be shown if the optimal rate given by Sriperumbudur and Szabó³⁰ can be applied as well. However, the advantages are noticeable. The Fast Walsh-Hadamard Transform³¹ (FWHT) takes $\mathcal{O}(r \log d)$ operations, $\mathcal{O}(d \log d)$ for each of the r/d blocks. All other matrices are diagonal with d entries. Since the Hadamard matrix itself requires no storage V consumes only $\mathcal{O}(r)$ memory.

³⁰ Sriperumbudur and Szabó, *Optimal rates for random Fourier features*.

³¹ Fino et al., *Unified matrix treatment of the fast Walsh-Hadamard transform*.

5.5.3 Nyström's Approximation

An alternative to random kitchen sinks are Nyström methods³² which are based on the eigenfunction decomposition of kernel functions. Recall that Mercer's theorem (§4.3) allows us, under certain conditions, to write the kernel function as

$$k(x, y) = \sum_{i \in I} \lambda_i e_i(x) e_i(y) \quad x, y \in \mathcal{X}$$

in terms of eigenfunctions $(e_i)_{i \in I}$ and corresponding eigenvalues $(\lambda_i)_{i \in I}$ of the integral operator. This provides us with another possibility to create a feature map as

$$\begin{aligned} \phi: \mathcal{X} &\rightarrow \ell^2 \\ \phi(x) &= (\sqrt{\lambda_i} e_i(x))_{i \in I} \quad x \in \mathcal{X} \end{aligned}$$

³² Williams et al., *Using the Nyström method to speed up kernel machines*.

which takes values in l^2 , the space of square summable sequences. Moreover, it holds that

$$\int_{\mathcal{X}} k(x, y) e_i(x) d\nu(x) = \lambda_i e_i(y)$$

for all eigenfunctions $(e_i)_{i \in I}$ with an appropriate probability measure ν . We can now approximate the integral by an empirical estimate

$$\frac{1}{n} \sum_{j=1}^n k(x_j, y) e_i(x_j) \approx \lambda_i e_i(y)$$

where $z = (x_1, \dots, x_n)$ are independent samples from ν . From the equation above we obtain n linear equations by using $y = x_i$ for $1 \leq i \leq n$ and derive an eigenvalue decomposition

$$KU = U\Lambda$$

where $K_{ij} = k(x_i, x_j)$, the matrix Λ is diagonal comprising the eigenvalues and U is the matrix of eigenvectors. We then get

$$\begin{aligned} e_i(x_j) &\approx U_{ji} \\ \lambda_i &\approx \frac{1}{n} \Lambda_{ii} \end{aligned}$$

and hence

$$\hat{\phi}_i(y) = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^n U_{ji} k(x_j, y) \approx e_i(y), \quad 1 \leq i \leq r,$$

where $\hat{\phi}(x) = (\hat{\phi}_1(x), \dots, \hat{\phi}_r(x))$ is an approximate feature map.

REQUIRE:

- 1: r : the number of feature expansion
- 2: k : a reproducing kernel
- 3: $(x_1, \dots, x_n)$: a sample

ALGORITHM:

- 4: Randomly choose a subset $s = (s_1, \dots, s_r)$ from $(x_1, \dots, x_n)$
 - 5: Compute kernel matrix $K_{ij} = k(s_i, s_j)$
 - 6: Perform eigenvalue decomposition of $KU = U\Lambda$
 - 7: **FOR** $i \leftarrow 1, 2, \dots, r$ **DO**
 - 8: Set $\hat{\phi}_i(x) = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^n U_{ji} k(s_j, x)$
 - 9: **RETURN** $\hat{\phi} = [\hat{\phi}_1 \dots \hat{\phi}_r]$
-

Algorithm 5.2: Nyström feature map approximation.

THE computational complexity of the eigenvalue decomposition grows cubically with the datasets size. It is therefore suggested to use only a random subset of z of size $r < n$. The method is then known as the *Nyström feature map* and summarized in algorithm 5.2. Using the subsampling scheme yields a feature map approximation with $\mathcal{O}(r^3)$ computational complexity and $\mathcal{O}(r^2)$ memory requirements. The Nyström approximation needs in general less basis functions (r) than the RKS approach (typically around 1000). However, the approximation is data dependent and hence becomes erroneous if the underlying distribution changes or when we are not able to get independent samples from the dataset.

5.5.4 EXPoSE & Approximate Feature Maps

All approximate feature maps introduced previously can be incorporated into EXPoSE to explicitly represent the feature function ϕ and consequently also the empirical mean map $\mu[\mathbb{P}_n]$ yielding

$$\eta(y) \approx \left\langle \hat{\phi}(y), \frac{1}{n} \sum_{i=1}^n \hat{\phi}(x_i) \right\rangle$$

for the EXPoSE predictor.

WE EMPHASIZE that with an efficient approximation of ϕ , as shown here, the training time of this algorithm is *linear* in the number of samples n and an evaluation of the predictor $\eta(y)$ takes only *constant time*. Moreover, we need only $\mathcal{O}(r)$ memory to store the model which is also *independent* of n and the input dimension d .

5.6 Related Work

In this section we review two techniques related to EXPoSE. The first one (the orthogonal series density estimator) has its roots in non-parametric density estimation and the second (centre of mass) is based on Hilbert space distances.

5.6.1 Orthogonal Series Density Estimation

There are several parallels with the orthogonal series density estimator introduced by Chentsov.³³ Suppose that the random variable X has support $D = [0, 1]$ and that the density f of X is square-integrable. Then

$$f(x) = \sum_{i=-\infty}^{\infty} \theta_i \varphi_i(x) \quad \text{with} \quad \theta_i = \int_0^1 f(x) \varphi_i(x) dx$$

for all $x \in [0, 1]$, where $\{\varphi_i\}$ is a complete orthogonal system of functions on D . This infinite series is approximated by a partial sum

$$\hat{f}_r(x) = \sum_{i=0}^r \theta_i \varphi_i(x)$$

where $r \in \mathbb{N}$ is called *cutoff*. “[...] the main statistical issue will be how to choose a cutoff $[r]$ [...]”.³⁴ There are several possible choices for $\{\varphi_i\}$. One of the most common is a system of orthogonal polynomials on $[0, 1]$

$$\left\{ \varphi_i(x) = \sum_{k=0}^i a_{ik} x^k \mid i = 0, 1, 2, \dots \right\},$$

where the coefficients a_{ik} have to be chosen such that the φ_i are orthogonal e.g. by the recursion $\varphi_0(x) = 1$ and

$$\varphi_i(x) = \frac{x^i - \sum_{k=0}^{i-1} \left(\int_0^1 u^i \varphi_k(u) du \right) \varphi_k(x)}{\left(\int_0^1 \left(v^i - \sum_{k=0}^{i-1} \left(\int_0^1 u^i \varphi_k(u) du \right) \varphi_k(v) \right)^2 dv \right)^{\frac{1}{2}}}$$

as described by Efromovich.³⁵ If f has compact support, for instance $D = [0, 1]$,

³³ Chentsov, *Evaluation of an unknown distribution density from observations*.

³⁴ Efromovich, *Nonparametric curve estimation: methods, theory, and applications*.

³⁵ *Ibid.*, Section 2.1.

then other orthogonal basis functions *e.g.* trigonometric, Fourier, spline, or wavelet, can be used as well. Other, such as the normalized Hermite and Laguerre functions can be applied to model densities with unbounded support $D = \mathbb{R}$ or $D = [0, \infty)$.

Multivariate densities with support $D \subset \mathbb{R}^d$ can be estimated using a *tensor product* basis. For simplicity let $d = 2$ and $\{\varphi_{i1}\}, \{\varphi_{i2}\}$ be a two complete orthogonal systems. Then, for all $x = (u, w) \in D$

$$\{\varphi_{ij}(u, w) = \varphi_{i1}(u)\varphi_{j2}(w) \mid i, j = 0, 1, \dots\}$$

forms the tensor product basis. The orthogonal series density estimator is then given by

$$f(u, w) = \sum_{i,j=-\infty}^{\infty} \theta_{ij} \varphi_{ij}(u, w)$$

with

$$\theta_{ij} = \int_0^1 \int_0^1 f(u, w) \varphi_{ij}(u, w) du dw$$

which can be generalized to $d > 2$.

THERE IS another challenge besides the choice of the cutoff r . Given a sample $z = (x_1, \dots, x_n)$, estimating θ_i by

$$\hat{\theta}_i = \frac{1}{n} \sum_{j=1}^n \varphi_i(x_j)$$

“[...] may not be well defined: It has infinite variance and is not consistent in ISE [integrated squared error] sense”.³⁶ Here one has to use additional weights w_i in the partial sum

$$\hat{f}_r(x) = \sum_{i=-r}^r w_i \hat{\theta}_i \varphi_i(x)$$

where for all $-r \leq i \leq r$, the weights are required to be symmetric *i.e.* $w_{-i} = w_i$, $0 < w_i < 1$ and it has to hold that $\sum_{i=-r}^r |w_i| < \infty$. This is then known as the *tapered orthogonal series estimator*. Efromovich remarks³⁷ that for the case of a d -dimensional space one has to estimate 5^d to 10^d coefficients $\hat{\theta}_i$ which makes the density estimation in high dimensions nearly impossible.

UNDER CERTAIN restrictions we can derive the orthogonal series density estimator as a special case of EXPOSE. However, our approach is not restricted to orthogonal basis functions and we explicitly allow every feature map expansion which can be derived from a positive definite kernel. As we have seen, some of the approximate feature maps are especially designed to work efficiently in high dimensions. In general, the EXPOSE predictor

$$\eta(y) = \int_{\mathcal{X}} k(y, x) d\mathbb{P}_X(x)$$

does not integrate to 1 and is therefore not a density estimation as it was never meant to be one. Additionally, the input space $(\mathcal{X}, \mathcal{X})$ can be any measurable, separable topological space. We will also see that the treatment of $\phi(x)$ as a member of an RKHS $\mathcal{H}$ allows us to apply various tools from functional analysis and convex optimization.

³⁶ Izenman, *Recent developments in nonparametric density estimation*.

³⁷ Efromovich, *Nonparametric curve estimation: methods, theory, and applications*, Chapter 6.

5.6.2 Centre of Mass

Shawe-Taylor et al.³⁸ suggests to use the distance from the *centre of mass* (COM) in an RKHS $\mathcal{H}$ as an anomaly score.

[...] a simple novelty detection algorithm that checks if a new data point is further from the true mean than the furthest training point. The chances of this happening for data generated from the same distribution can be shown to be small, hence when such points are found there is a high probability that they are outliers.³⁹

³⁸ Shawe-Taylor et al., *Kernel methods for pattern analysis*, Section 5.1.1, *Estimating the moments of a random vector with applications*.

³⁹ *Ibid.*

The centre of mass with respect to a set $z = (x_1, \dots, x_n)$ is defined as

$$\phi_z = \frac{1}{n} \sum_{i=1}^n \phi(x_i),$$

which is essentially the empirical kernel mean map. The proposed algorithms does not consider the application of approximate feature maps. However, the norm distance between the feature map $\phi(y)$ of a query point $y \in \mathcal{X}$ and the centre of mass can be calculated explicitly as

$$\begin{aligned} \|\phi(y) - \phi_z\|^2 &= \langle \phi(y), \phi(y) \rangle - 2\langle \phi(y), \phi_z \rangle + \langle \phi_z, \phi_z \rangle \\ &= k(y, y) + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n k(x_i, x_j) - \frac{2}{n} \sum_{i=1}^n k(y, x_i), \end{aligned}$$

using the identity $k(x, y) = \langle \phi(x), \phi(y) \rangle$. Obviously, this formulation has a quadratic computational complexity per prediction, which is prohibitive for large-scale problems.

IT WOULD be reasonable to assume that the application of approximate feature maps can overcome the computational complexity using as it does for EXPOSE. However, this is problematic when using data dependent feature map approximations such Nyström's technique. Recall that Nyström's feature maps are of the form

$$\begin{aligned} \hat{\phi}(y) &= (\hat{\phi}_1(y), \dots, \hat{\phi}_r(y)) \quad \text{with} \\ \hat{\phi}_i(y) &= \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^n u_{ji} k(x_j, y) \end{aligned}$$

where the x_j are random instances from the training set. The value of $\hat{\phi}(y)$ goes to zero if y is not close to any x_j as $k(x_j, y) \rightarrow 0$. In this case,

$$\|\hat{\phi}(y) - \hat{\phi}_z\|^2 = \underbrace{\langle \hat{\phi}(y), \hat{\phi}(y) \rangle}_{\rightarrow 0} - 2 \underbrace{\langle \hat{\phi}(y), \hat{\phi}_z \rangle}_{\rightarrow 0} + \underbrace{\langle \hat{\phi}_z, \hat{\phi}_z \rangle}_{=\text{const}}$$

which is fatal since points far from the training set get a low anomaly score and hence classified as normal (Fig. 5.5). This is exactly the *opposite* of how an anomaly is defined.

In the same scenario, EXPOSE behaves as expected: its (normal) scores decrease if the distance to the training set increases and correctly reports an anomalous instance.

Original Centre of Mass
without approximate
feature maps.

Centre of Mass using
Nyström's approxima-
tion.

Figure 5.5: The Centre of Mass technique with and without Nyström's feature maps. Instances far from the normal data (•) get a low anomaly score if Nyström's method is used and hence falsely classified as normal.

5.7 Empirical Analysis on Synthetic Data

The aim of this section is to gain insight into the strength and weaknesses of EXPOSE on different problem classes. These are *point difficulty*, *anomaly frequency* and *anomaly density* as explained in what follows.

POINT DIFFICULTY evaluates the predictive performance on different difficulty levels of the query points (test set). We assume that an anomalous instance is more difficult to detect if the probability that this instance is a sample of the normal distribution is *high*. Consequently, we first estimate the class probabilities using a kernel density classifier. The training set does not change, however we choose the anomalous instances of the test set such that their probability of being samples of the normal distribution lies in the intervals $[0.0 - 0.1)$, $[0.05 - 0.15)$, $\dots$, $[0.9 - 1.0)$ as illustrated in the left and middle plots of Fig. 5.6. Each experiment was repeated five times and we show the AUC values for each repetition together with their mean and confidence intervals (Fig. 5.6 right). As expected, the AUC decreases as the point difficulty increases. It is questionable if anomalies with a very high likelihood under the distribution of normal data (*e.g.* ≥ 0.5) can be detected without further information, provided that most actual normal instances obey a lower probability.

Point difficulty levels as the probability (or likelihood) of an anomalous instance under the distribution of $X | Y = c_N$

Easy: probability $\in [0, 0.15)$

Normal: probability $\in [0.15, 0.3)$

Hard: probability $\in [0.3, 0.5)$

Challenging: probability $\in [0.5, 1]$

ANOMALY FREQUENCY is the fraction of anomalies in the training set which is varied from 0 to 50 percent by uniformly sampling anomalous instances (Fig. 5.7). The test set is not affected by this problem class. A higher fraction of anomalies violates the assumption that the training set comprises mostly normal data which is reflected in the drop of the predictive performance (AUC). Nevertheless, the empirical evaluation indicates that EXPOSE is relatively robust against anomalies in the training set as, on average, an anomaly frequency of 0.23 causes the AUC to drop only 10%. Even though the fraction of anomalies exceeds 50% the AUC is still above 0.5 as the normal data is denser than the uniformly sampled anomalies.

Low difficulty:
Probability of anomalous query
points being normal is ≤ 0.3 .

High difficulty:
Probability of anomalous query
points being normal is ≥ 0.7 .

Figure 5.6: Point difficulty. The plots on the left and middle show examples of a low and high point difficulty, respectively. The AUC for different difficulty levels is shown in the right.

Training set and prediction scores
for an anomaly frequency of 0.1.

Training set and prediction scores
for an anomaly frequency of 0.5.

Figure 5.7: Anomaly frequency. A higher fraction of anomalies in the training set causes a drop in AUC.

Training set with an anomaly
frequency of 0.25.

A random test set of 1000
instances.

Figure 5.8: Anomaly density. Anomalies in the training set are not uniformly sampled but form a dense cluster.

ANOMALY DENSITY also varies the fraction of anomalies in the training set, however, unlike previously the anomalous instances are not sampled uniformly but are concentrated in one cluster (Fig. 5.8). Even though the anomalies which are inside or close to the cluster get a high normal score, there is a significant amount of instances in the test whose ranking is still acceptable. However, the influence of the cluster on the scores depends on its size. We can therefore expect the predictive performance to decrease even more if the cluster size increases.

5.8 Experimental Evaluation

In this section we show in several experiments how EXPOSE compares to other state of the art anomaly detection techniques in prediction and runtime evaluations. We first explain how datasets for anomaly detection can be generated from multi-class datasets.

5.8.1 Constructing Anomaly Detection Datasets

One difficulty in comparing anomaly detection frameworks is that there is no objective way to tell if an element of a set is anomalous in comparison with the other elements in the same set without the availability of labels. A common alternative to manual labeling is the creation of an artificial anomaly detection dataset from a labeled set with multiple classes. A *multi-class dataset* contains samples from three or more classes, for example $\mathcal{Y} = \{c_1, \dots, c_m\}$ for some $m > 2$. Assume that the dataset z comprises n labeled pairs *i.e.* $z = ((x_1, y_1), \dots, (x_n, y_n))$. We can then construct m new labeled anomaly detection datasets from z , where the k th set is defined as

$$\underbrace{\{(x, y) \mid (x, y) \in z, y = c_k\}}_{\text{normal}} \cup \underbrace{\{(x, y) \mid (x, y) \in z, y \neq c_k\}}_{\text{anomalous}}$$

for each $1 \leq k \leq m$. Using multi-class datasets with several classes reflects the typical situation in which we observe only one class (the normal data) and have to discriminate this class against several other unknown sources of anomalies. Since, by definition, anomalies are rare events we require the training set to reflect this property. We therefore suggest to sample the training set randomly such that it contains only a small percentage (e.g. 1 % or 0.1 %) of anomalies. This can simulate sensor noise or wrongly labeled instances and can help to assess the robustness of the evaluated technique. However, as we consider semi-supervised anomaly detection algorithms (§2.1), it is important that majority of elements in the training set is still sampled from the class of normal data.

5.8.2 Datasets

For performance analysis and evaluation of EXPOSE we take datasets which are established in literature for the comparison of anomaly detection algorithms.⁴⁰ There are several smaller benchmark datasets with known anomaly classes *i.e.* Ionosphere, Arrhythmia, Pima, Satellite, Shuttle,⁴¹ Biomed and Wisconsin Breast Cancer (Breastw).⁴² These datasets are set up as described in the work of Liu et al.⁴³ where all nominal and binary

⁴⁰ Liu et al., *Isolation-based anomaly detection*; Schölkopf, Platt, et al., *Estimating the support of a high-dimensional distribution*; Tax and Duin, *Support vector data description*.

⁴¹ Lichman, *UCI Machine Learning Repository*.

⁴² Tax and Duin, *Support vector data description*.

⁴³ Liu et al., *Isolation-based anomaly detection*.

attributes are removed. One of the larger datasets are the *knowledge discovery and data mining* network intrusion data (KDDCUP) and Forest Cover Type (ForestCover). For KDDCUP instances we follow the setup of Yu et al.⁴⁴ and obtain a total of 127 attributes. Furthermore, we create two large and high-dimensional image datasets: MNIST⁴⁵ and the Google Street View House Numbers (SVHN)⁴⁶ with HOG features.⁴⁷ We use the methodology described before (§5.8.1) to create anomaly detection benchmarks from the MNIST and SVHN multi-class datasets. Table 5.1 summarizes the dataset properties and how the anomaly classes are defined.

	SIZE	DIM.	ANOMALY CLASS	$c_A\%$
KDDCUP	1036 241	127	“attack”	0.3%
FORESTCOVER	286 048	10	class 4 vs. 2	9%
MNIST (1,...,9)	90 924	784	class i vs. rest	1%
SVHN (1,...,9)	54 488	2592	class i vs. rest	1%
SHUTTLE	58 000	9	classes 2,3,4,5,7	6%
SATELLITE	6435	36	classes 2,4,5	32%
PIMA	768	8	“pos”	35%
BREASTW	683	9	“malignant”	35%
ARRHYTHMIA	452	274	classes 3,4,5,7,8,9,14,15	14%
IONOSPHERE	351	32	“bad”	36%
BIOMED	194	5	“carrier”	34%

⁴⁴ Yu et al., *Classifying large data sets using SVMs with hierarchical clusters*.

⁴⁵ Chang et al., *LIBSVM: A library for support vector machines*.

⁴⁶ Netzer et al., *Reading digits in natural images with unsupervised feature learning*.

⁴⁷ Vondrick et al., *Hoggles: Visualizing object detection features*.

Table 5.1: Dataset properties. Each dataset is listed with its size (n), dimension (d), the anomaly class (c_A) and the anomaly proportion ($c_A\%$) within the dataset.

In the experiments we provide a dedicated labeled random subset of 1% or 2000 instances (whichever is smaller) to configure the algorithm’s parameters via cross-validation. Examples of such parameters are the number of nearest neighbors in the local outlier factor (LOF) or the kernel bandwidth of the algorithms EXPOSE, the kernel density estimator and the one-class support vector machine. We do not optimize over different distance metrics and kernel families, but use the most common Euclidean distance and squared exponential kernel, respectively. Each experiment is repeated 5 times and their area under ROC (AUC) scores are used to perform the Friedman test (§2.5). If not stated otherwise we combine EXPOSE with Nyström’s approximation (§5.5.3).

5.8.3 Evaluation

The average scores and runtimes for each experiment are reported in tables 5.2 and 5.3. Some algorithms failed on the larger datasets *e.g.* LOF was not able to process the KDDCUP dataset due to the high memory requirements of the tree data structure. However, the advantage of a tree for nearest neighbor lookup in low dimensions can be seen when comparing the runtime of LOF on ForestCover and MNIST. Even though the ForestCover dataset has more than 2.5 times the size of MNIST, it takes only a fraction of the time to be processed. This advantage vanishes in higher dimensions. KDE and F-ABOD exhibit a good anomaly detection performance on the small datasets, however fail as soon we apply them to medium-sized problems. Figure 5.9 is a combination of the average ranks and runtimes from tables 5.2 and 5.3. The “indefinite” marker indicates that LOF, F-ABOD and KDE were not able to produce a result on these datasets ($\diamond$ and $\star$ entries in

the tables). In cases where an algorithms take the same processing time for multiple datasets, the markers overlap.

The strength of `EXPose` and `OC-SVM` are clearly the high-dimensional image datasets. This is not surprising as kernelized approaches are intended to work in such scenarios. We partly attribute the success of `EXPose` to its simplicity as (in this case) only the kernel bandwidth has to be configured through cross-validation. This is much easier than the estimation of two or three parameters as required by the `OC-SVM` or `LOF`. We therefore recommended to allocate significantly more instances and computational time to configure these algorithms which may increases their prediction performance. The number of `EXPose`'s kernel expansions is uncritical to choose as a higher value yields a better feature map approximation. Consequently it is recommended to use a relatively large number of expansions — as large as permitted by the computational limitations.

Figure 5.9: Runtime vs. rank (predictive performance) on several datasets. The color indicates the dataset size. The result of a large dataset overlays the result of a smaller one if the algorithm's runtime is the same. Experiments with the same dataset are connected by a line. (Figure is repeated from page 49)

THE Friedman test confirms a statistical significant difference between the performances of the individual algorithms at a p-value of 0.05 and we can apply one of the post hoc tests (§2.5). We use the Nemenyi test for pairwise comparisons off *all* algorithms and inspect the critical difference diagram in Fig. 5.11. It shows that `EXPose` performs significant better than `iForest`, `F-ABOD` and `KDE`. While no significant difference in terms of anomaly detection between `EXPose`, `OC-SVM` and `LOF` can be confirmed by the Nemenyi test, `EXPose` is several orders of magnitude faster on large-scale, high-dimensional datasets.

Figure 5.10: Figure 3.21 with logarithmic runtime axis.

Figure 5.11: Critical difference diagram of the batch anomaly detection performance. Algorithms which are significantly different (at $p < 0.05$) are *not* connected with a bar.

	EXPOSE	IFOREST	OC-SVM	LOF	KDE	F-ABOD
KDDCUP	<u>1.00</u>	0.99	<u>1.00</u>	◇	★	★
FORESTCOVER	0.83	0.85	<u>0.89</u>	0.56	★	★
MNIST 1	<u>1.00</u>	0.99	<u>1.00</u>	0.97	★	★
MNIST 2	0.79	0.67	0.80	<u>0.85</u>	★	★
MNIST 3	0.86	0.70	0.80	<u>0.88</u>	★	★
MNIST 4	0.88	0.80	<u>0.93</u>	0.87	★	★
MNIST 5	<u>0.89</u>	0.68	0.82	<u>0.89</u>	★	★
MNIST 6	<u>0.94</u>	0.84	<u>0.94</u>	0.89	★	★
MNIST 7	<u>0.92</u>	0.88	<u>0.92</u>	0.89	★	★
MNIST 8	0.78	0.64	0.79	<u>0.84</u>	★	★
MNIST 9	0.89	0.81	<u>0.90</u>	<u>0.90</u>	★	★
SVHN 1	0.90	0.88	<u>0.91</u>	0.85	★	★
SVHN 2	<u>0.88</u>	0.76	0.78	0.78	★	★
SVHN 3	<u>0.72</u>	0.58	0.59	0.71	★	★
SVHN 4	<u>0.85</u>	0.72	0.75	0.83	★	★
SVHN 5	<u>0.83</u>	0.72	0.73	0.74	★	★
SVHN 6	0.84	0.76	0.80	<u>0.87</u>	★	★
SVHN 7	<u>0.89</u>	0.85	0.86	0.87	★	★
SVHN 8	0.83	0.75	0.75	<u>0.88</u>	★	★
SVHN 9	0.85	0.77	0.80	<u>0.87</u>	★	★
SHUTTLE	<u>0.99</u>	<u>0.99</u>	0.91	0.55	<u>0.99</u>	0.78
SATELLITE	<u>0.79</u>	0.70	0.62	0.57	0.78	0.74
PIMA	<u>0.68</u>	<u>0.68</u>	0.62	0.59	0.67	0.65
BREASTW	<u>0.99</u>	<u>0.99</u>	0.81	0.45	<u>0.99</u>	<u>0.99</u>
ARRHYTHMIA	<u>0.79</u>	<u>0.79</u>	0.71	0.68	0.74	<u>0.79</u>
IONOSPHERE	0.92	0.84	0.66	0.89	0.81	<u>0.93</u>
BIOMED	0.87	0.82	0.76	0.69	<u>0.88</u>	<u>0.88</u>
AVERAGE RANK	<u>1.87</u>	3.48	2.90	3.18	4.87	4.67

Table 5.2: Anomaly detection performances (area under ROC). Algorithms marked with a ◇ run out of memory and a ★ means the execution did not finish in less than two full days.

	EXPOSE	IFOREST	OC-SVM	LOF	KDE	F-ABOD
KDDCUP	44	71	22 213	◇	★	★
FORESTCOVER	29	25	25 901	47	★	★
MNIST 1	12	9	1976	23 760	★	★
MNIST 2	11	10	2773	18 717	★	★
MNIST 3	11	10	1991	20 109	★	★
MNIST 4	11	9	1159	17 412	★	★
MNIST 5	10	9	1892	15 324	★	★
MNIST 6	10	9	3091	18 208	★	★
MNIST 7	11	9	2727	20 153	★	★
MNIST 8	11	9	1607	18 217	★	★
MNIST 9	10	8	2383	18 542	★	★
SVHN 1	24	11	9311	18 192	★	★
SVHN 2	20	10	10 371	18 144	★	★
SVHN 3	16	8	14 122	28 508	★	★
SVHN 4	13	7	4044	19 247	★	★
SVHN 5	14	7	10 348	22 359	★	★
SVHN 6	11	6	5389	13 166	★	★
SVHN 7	11	6	6790	14 906	★	★
SVHN 8	9	5	4210	9741	★	★
SVHN 9	9	5	3831	9290	★	★
SHUTTLE	3	6	38	24	690	631
SATELLITE	0	3	3	4	1	55
PIMA	1	2	0	0	0	9
BREASTW	1	3	0	0	0	4
ARRHYTHMIA	1	1	0	0	0	4
IONOSPHERE	1	3	0	0	0	0
BIOMED	0	2	0	0	0	0

Table 5.3: Runtimes in seconds. Algorithms marked with a ◇ run out of memory and a ★ means the execution did not finish in less than two full days.

CHAPTER SIX

Constant Time Convex Expected Similarity Estimation

SYNOPSIS · The previous chapter introduced the `EXPOSE` anomaly detector which relies on an efficient estimation of the kernel embedding. In this part we view the sample estimate of the kernel mean map from the perspective of an optimization problem. This yields an algorithm which is able to approximate the `EXPOSE` model within a predefined accuracy in *constant* time, independent of the training dataset size. The assumption is that a high accuracy sample estimate of the `EXPOSE` model does not improve the anomaly detection performance.

The following content was partly published in

- M. Schneider, W. Ertel, and G. Palm (2016).
“Constant Time EXpected Similarity Estimation for Large-Scale Anomaly Detection”. In: *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*. IOS Press, pp. 12–20

6.1 Preliminaries

Recall that the `EXPOSE` anomaly detector estimates the similarity between a new unseen observation and the input space distribution using

$$\eta(y) = \langle \phi(y), \mu[\mathbb{P}_X] \rangle$$

where $\mu[\mathbb{P}_X]$ is the kernel mean map of $\mathbb{P}_X$. Commonly, in anomaly detection applications, the input space distribution $\mathbb{P}_X$ is not accessible, but a sample $z = (x_1, \dots, x_n)$ of size n . Hence, the sample estimate $\mu[\mathbb{P}_n]$ is used as a proxy for $\mu[\mathbb{P}_X]$ in the `EXPOSE` predictor. We already showed that `EXPOSE` can be trained in *linear* time where the empirical kernel mean map $\mu[\mathbb{P}_n] = \frac{1}{n} \sum_{i=1}^n \phi(x_i)$ is the dominating factor with $\mathcal{O}(n)$. The question arises whether it is possible to improve the linear training time complexity and create an algorithm which is completely *independent* of the dataset size.

The answer to this question is positive if a high accuracy sample estimate $\mu[\mathbb{P}_n]$ of $\mu[\mathbb{P}_X]$ does not improve the anomaly detection performance. We follow Bousquet et al.¹ who observes that for most machine learning applications there is no need to optimize below the statistical error. The authors argue that a high accuracy minimizing an empirical cost function does not gain much since it is itself an approximation of the expected costs and therefore contains errors.

“[...] convex optimization problems (beyond least-squares and linear programs) are more prevalent in practice than was previously thought. Since 1990 many applications have been discovered in areas such as automatic control systems, estimation and signal processing, communications and networks, electronic circuit design, data analysis and modeling, statistics, and finance. Convex optimization has also found wide application in combinatorial optimization and global optimization, where it is used to find bounds on the optimal value, as well as approximate solutions.”
— Boyd et al., *Convex optimization*.

¹ Bousquet et al., *The tradeoffs of large scale learning*.

IN THE FOLLOWING we derive a methodology to build an ϵ -accurate model w of $\mu[\mathbb{P}_X]$ using only a random subset of the training data by means of stochastic optimization.

DEFINITION 6.1 (ϵ -accurate). We say an algorithm finds an ϵ -accurate solution w of a real-valued objective function f if

$$f(w) \leq \inf f + \epsilon$$

for a given accuracy $\epsilon > 0$.

We will show that for the proposed objective function f we can achieve a rate of convergence of $\mathcal{O}(1/t)$ that is

$$\mathbb{E}[f(w_t) - f(\mu[\mathbb{P}_X])] \leq \mathcal{O}\left(\frac{1}{t}\right),$$

where w_t only needs access to $t \in \mathbb{N}$ independent random samples of $\mathbb{P}_X$. The key observation is that for any given accuracy $\epsilon > 0$, we can obtain $\mathbb{E}[\|w_t - \mu[\mathbb{P}_X]\|_{\mathcal{H}}] < \epsilon$ in a *fixed* number of iterations *independent* of the dataset size n . Moreover, it can be shown that (without further assumptions) the $\mathcal{O}(1/t)$ rate is optimal for stochastic optimization.²

6.2 Stochastic Optimization

Stochastic optimization (SO) and especially, stochastic gradient (SG) methods,³ are widely used to train machine learning models on very large-scale datasets, because a single iteration of SG requires only little computational time. More sophisticated optimization routines can converge with fewer iterations, however a single step of these methods often take more time than several thousand steps of SG.

The challenge with large-scale applications is that even basic linear algebra like matrix decomposition and matrix-matrix or matrix-vector multiplication becomes computationally too expensive to be performed at each iteration of an optimization routine. The following table (which is similar to the one presented by Nesterov⁴) exemplifies the feasible operations for each problem class ranging from small to large-scale problems. For small

	COMPLEXITY	OPERATIONS	DIM. d	SIZE n	
SMALL	$\mathcal{O}(n^4 d^4)$	All	10^2	10^3	
MEDIUM	$\mathcal{O}(nd^3)$	A^{-1}	10^3	10^4	
BIG	$\mathcal{O}(d^2)$	Ax	10^6	10^{10}	■
LARGE	$\mathcal{O}(d \log d)$	$x + y$	10^8	10^{12}	■

problem sizes we are able to apply all kinds of sophisticated operations and even solve an optimization problem per instance.⁵ We can apply polynomial time operations to medium-size problems which includes for example, matrix inversions. These kind of operations are already intractable for big datasets where we are limited to $\mathcal{O}(d^2)$ operations. At a large-scale we have to further reduce the computational complexity per iteration step to linear time operations where stochastic gradient methods are a viable option.

Figure 6.1: Steps of the gradient descent algorithm (—) on the contours of the objective function with minimum at the center.

² Agarwal et al., *Information-Theoretic Lower Bounds on the Oracle Complexity of Convex Optimization* Convex optimization.

³ Bousquet et al., *The tradeoffs of large scale learning*; Roux et al., *A stochastic gradient method with an exponential convergence rate for finite training sets*.

⁴ Nesterov, *Subgradient methods for huge-scale optimization problems*.

Table 6.1: Problem sizes, complexity and example operations per iteration. A is a matrix of size $d \times d$. x and y are two vectors of size d .

⁵ Janssens, Huszár, et al., *Stochastic outlier selection*.

6.2.1 The Stochastic Approximation Algorithm

Let $\mathcal{H}$ be a separable Hilbert space and H be a closed and convex subset of $\mathcal{H}$. The classic stochastic approximation algorithm⁶ tries to solve

$$\min_{w \in H} \mathbb{E}[f(w, X)] = \min_{w \in H} \int_{\mathcal{X}} f(w, x) d\mathbb{P}_X(x)$$

where X is a random element taking values in the measurable space $(\mathcal{X}, \mathcal{X})$ with distribution $\mathbb{P}_X$. All expectations are taken with respect to X , need to exist and be well defined for any $w \in H$. To solve this problem, stochastic approximation techniques make the following two assumptions.

- (A-4) It is possible to generate independent samples $x_1, x_2, \dots$ from the probability distribution $\mathbb{P}_X$.
- (A-5) Given a point $(w, x) \in \mathcal{H} \times \mathcal{X}$, we assume to have access to a *stochastic subgradient* of f at w .

A subgradient is the generalization of a gradient to non-differentiable functions. It can be defined for general normed vector spaces in terms of their dual.

DEFINITION 6.2 (Subgradient, Subdifferential). Let $(\mathcal{F}, \|\cdot\|)$ be a normed vector space, $\mathcal{F}^*$ its dual (Def. B.24) and $f: \mathcal{X} \rightarrow \mathbb{R}$. An element $\zeta(w) \in \mathcal{F}^*$ is called *subgradient* of f at $w \in \text{dom}(f)$ if

$$f(w) - f(v) \geq \langle \zeta(w), w - v \rangle_*$$

for all $v \in \mathcal{F}$. The set of all subgradients of f at $w \in \mathcal{F}$ is called the *subdifferential* and denoted by $\partial f(w)$.

If the function f is (Gâteaux) differentiable, then there is only one subgradient which is equal to the classical gradient of f . This means for any $w \in \text{dom}(f)$ we have $\partial f(w) = \{\nabla f(w)\}$. Notably, in a Hilbert space $\mathcal{H}$ the subgradient can be identified with an element in $\mathcal{H}$ by Riesz representation theorem (Def. 4.4).

A *stochastic subgradient* $\zeta(w, X)$ is a random variable whose expectation is a subgradient *i.e.* it has the property that $\zeta(w) = \mathbb{E}[\zeta(w, X)] \in \partial f(w)$.

STARTING at some $w_1 \in H$, stochastic approximation algorithm then creates a sequence $(w_t)_{t \in \mathbb{N}}$ using

$$w_{t+1} = \Pi_H(w_t - \gamma_t \zeta(w_t, X))$$

with the aim to minimize $\mathbb{E}[f(w, X)]$. Here (γ_t) is a sequence of positive step sizes and the operator

$$\Pi_H(w) = \arg \min_{v \in H} \|w - v\|$$

is called the *projection* of w onto H . It is important to realize that $\Pi_H(w)$ is unique in H . Moreover, the function $w \mapsto \Pi_H(w)$ is *non-expanding* (or Lipschitz continuous with Lipschitz constant 1) which means that

$$\|\Pi_H(w) - \Pi_H(v)\| \leq \|w - v\|$$

for all w and v in $\mathcal{H}$. We will denote the optimal solution to the optimization problem by $\check{w} = \min_{w \in H} f(w)$. Notice that $\check{w}$ may not exist and is not necessarily unique.

⁶ Robbins et al., *A stochastic approximation method*.

Figure 6.2: Subgradients of f at w .

Most literature is concerned with the optimization of an objective function defined on some finite-dimensional Euclidean vector space. Nemirovski et al.⁷ for example consider H to be a nonempty closed bounded convex subset of $\mathcal{H} = \mathbb{R}^d$ and show that stochastic approximation can obtain a $\mathcal{O}(1/t)$ convergence rate if the objective function f is differentiable and α -strongly convex on $\mathcal{H}$.

DEFINITION 6.3 (α -Strongly Convex Function). Let H be a convex subset of a normed space. A function $f: H \rightarrow \mathbb{R} \cup \{+\infty\}$ is called α -strongly convex if for some $\alpha > 0$

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{\alpha}{2} \lambda(1 - \lambda) \|x - y\|^2$$

for all $x, y \in \text{dom}(f)$ and $\lambda \in [0, 1]$.

This implies that a function f is α -strongly convex if and only if the function

$$x \mapsto f(x) - \frac{\alpha}{2} \|x\|^2$$

is convex. Nemirovski et al. require in addition, that the stochastic subgradients $\zeta(w)$ are bounded in expectation *i.e.* there exists a constant $c > 0$ such that

$$\mathbb{E}[\|\zeta(w, X)\|^2] \leq c^2$$

for all $w \in H$. Under these conditions, Nemirovski et al. demonstrate that

$$\begin{aligned} \mathbb{E}[f(t) - f(\check{w})] &\leq \mathcal{O}\left(\frac{1}{t}\right) \quad \text{and} \\ \mathbb{E}[\|w_t - \check{w}\|^2] &\leq \mathcal{O}\left(\frac{1}{t}\right). \end{aligned}$$

Significantly, without further assumptions on the objective function f , these convergence rates are unimprovable.⁸

⁷ Nemirovski et al., *Robust stochastic approximation approach to stochastic programming*.

⁸ Agarwal et al., *Information-Theoretic Lower Bounds on the Oracle Complexity of Convex Optimization Convex optimization*.

6.3 Infinite-Dimensional Stochastic Optimization

This section lays the theoretical foundation to prove the rate of convergence for the EXPOSE optimization problem. We will see that if a minimizer exists and the gradient of the objective function f is bounded, we can achieve rates similar to the ones derived by Nemirovski et al. (theorems 6.7 and 6.8). However, since our optimization problem is infinite-dimensional, the main challenge will be to overcome the technical difficulties that arise in such spaces.

IN finite-dimensions, closed and bounded sets are *compact*, guaranteed by the Heine–Borel theorem. Compact sets have several desirable properties *i.e.* compactness of a nonempty set L suffice to show that a continuous function $f: L \rightarrow \mathbb{R}$ attains its minimum on L which means there is some $\check{w} \in L$ such that $f(\check{w}) = \inf_{w \in L} f(w)$. Furthermore, if (w_t) is a *minimizing sequence* for f , that is

$$\lim_{t \rightarrow \infty} f(w_t) = \inf f$$

and if (w_t) converges to $\check{w}$, then $\check{w} \in \arg \min f$. This is easy to see from the Bolzano–Weierstrass theorem: Since (w_t) is bounded, $f(w_t)$ is bounded and there exists a subsequence (w_{t_k}) which converges to some $\check{w}$ and because L is closed, $\check{w}$ is an element of L . Since (w_t) is a minimizing sequence, it follows from the continuity of f that $f(w_{t_k}) \rightarrow f(\check{w}) = \inf f$.

minimizing sequence

HOWEVER, the situation is entirely different in infinite-dimensional spaces like the Hilbert space $\mathcal{H}$ we have to consider in our optimization problem.

In infinite-dimensional Banach spaces, closed balls are not compact.

— Zeidler, *Applied Functional Analysis*

Moreover, in this situation, closed and bounded sets need not even be *sequentially compact*. That means not every sequence in the set has a converging subsequence that converges to a point in that set⁹. For example, consider the following sequences (u_t) in l^2 with $\|u_t\| = 1$ defined as

$$\begin{aligned} u_1 &= (1, 0, 0, \dots) \\ u_2 &= (0, 1, 0, \dots) \\ u_3 &= (0, 0, 1, \dots) \\ &\vdots \\ u_t &= (0, 0, 0, \dots, 0, 1, 0, \dots) \end{aligned}$$

which is closed and bounded, but no subsequence can converge since $\|u_t - u_k\|^2 = 2$ whenever $t \neq k$. In infinite-dimensional optimization, we can try to surrogate Bolzano-Weierstrass theorem by the property that a bounded sequence in a reflexive Banach space always has a *weakly* convergent subsequence.¹⁰

DEFINITION 6.4 (Weak Convergence). A sequence (w_t) in a Banach space $\mathcal{H}$ converges *weakly* to w if

$$\lim_{t \rightarrow \infty} \langle u^*, w_t \rangle_* = \langle u^*, w \rangle_*$$

for all u^* in the dual of $\mathcal{H}^*$ of $\mathcal{H}$ and is denoted by $w_t \rightharpoonup w$.

⁹ In finite-dimensions this is guaranteed by the Bolzano-Weierstrass theorem

¹⁰ Zeidler, *Applied Functional Analysis*, Theorem 2.C.

6.3.1 Strongly Convex Convergence Rates

We start with deriving convergence rates for general strongly convex functions with Lipschitz continuous gradients and then apply them to `EXPPOSE`. The derivation closely follows the one in finite-dimensional optimization literature¹¹ using additional assumptions wherever necessary to overcome the difficulties of general reproducing kernel Hilbert spaces.

The following lemma plays an important role in the derivation of the central theorems about the convergence rate of strongly convex functions with domain in infinite-dimensional Hilbert spaces (theorems 6.7 and 6.8).

LEMMA 6.5. Let $\mathcal{H}$ be an (infinite-dimensional) Hilbert space, $H \subset \mathcal{H}$ and $f: H \rightarrow \mathbb{R}$ be continuous and α -strongly convex. If $\check{w}$ is a global minimizer of f , then

$$\langle \zeta(v), v - \check{w} \rangle \geq \alpha \|v - \check{w}\|^2$$

for all v in H and any subgradient $\zeta(v) \in \partial f(v)$.

¹¹ Boyd et al., *Convex optimization*; Nemirovski et al., *Robust stochastic approximation approach to stochastic programming*; Nesterov, *Introductory lectures on convex optimization*.

PROOF. By definition of α -strong convexity we know that¹²

$$f(w) \geq f(v) + \langle \zeta(v), w - v \rangle + \frac{\alpha}{2} \|w - v\|^2$$

and therefore,

$$\begin{aligned} f(v) - f(w) &\leq \langle \zeta(v), v - w \rangle - \frac{\alpha}{2} \|w - v\|^2 \\ \langle \zeta(v), v - w \rangle &\geq f(v) - f(w) + \frac{\alpha}{2} \|w - v\|^2 \end{aligned}$$

for all v, w in H and $\zeta(v) \in \partial f(v)$. If $\check{w}$ is a global minimizer of f , then

$$f(v) - f(\check{w}) \geq 0$$

for all $v \in H$ which concludes the proof. $\square$

Lemma 6.5 leads directly to a second important inequality which bounds the distance from the minimizer in terms of the subgradient.

LEMMA 6.6. Under the prerequisites of Lemma 6.5

$$\|v - \check{w}\|^2 \leq \frac{\|\zeta(v)\|^2}{\alpha^2}$$

for all $v \in H$.

PROOF. By the Cauchy–Schwarz inequality we obtain

$$\begin{aligned} \alpha^2 \|v - \check{w}\|^4 &\leq \langle v - \check{w}, \zeta(v) \rangle^2 \\ &\leq \|v - \check{w}\|^2 \|\zeta(v)\|^2 \\ \alpha^2 \|v - \check{w}\|^2 &\leq \|\zeta(v)\|^2 \end{aligned}$$

which concludes the proof. $\square$

THEOREM 6.7. Let $\mathcal{H}$ be a Hilbert space, H a nonempty closed convex subset of $\mathcal{H}$ and $f: H \rightarrow \mathbb{R}$ be α -strongly convex. Assume the minimizer $\check{w} \in H$ exists and $\zeta(w_t, X)$ is strongly (Bochner) integrable, then the stochastic approximation sequence (w_t)

$$w_{t+1} = \Pi_H(w_t - \gamma_t \zeta(w_t, X))$$

converges strongly to $\check{w}$ with rate

$$\mathbb{E}[\|w_t - \check{w}\|^2] \leq \frac{\|\zeta(w_t)\|^2}{\alpha^2 t}$$

for any $w_t \in H$ and $t \in \mathbb{N}$.

¹² Peygouquet, *Convex Optimization in Normed Spaces: Theory, Methods and Examples*, Proposition 3.23.

PROOF.

$$\begin{aligned}
\|w_t - \check{w}\|^2 &= \|\Pi_H(w_t - \gamma_t \zeta(w_t, X)) - \check{w}\|^2 \\
&= \|\Pi_H(w_t - \gamma_t \zeta(w_t, X)) - \Pi_H(\check{w})\|^2 && \text{since } \check{w} \in H \\
&\leq \|w_t - \gamma_t \zeta(w_t, X) - \check{w}\|^2 && \text{since } \Pi_H \text{ is non-expanding} \\
&= \|w_t - \check{w}\|^2 + \gamma_t^2 \|\zeta(w_t, X)\|^2 - 2\gamma_t \langle w_t - \check{w}, \zeta(w_t, X) \rangle
\end{aligned}$$

We will take expectations on both sides. As $\zeta(w_t, X)$ is assumed to be strongly integrable we can interchange the inner product and the expectation operator (lemma 5.10) and therefore

$$\mathbb{E}[\langle w_t - \check{w}, \zeta(w_t, X) \rangle] = \langle w_t - \check{w}, \mathbb{E}[\zeta(w_t, X)] \rangle.$$

Furthermore, by Jensen's inequality

$$\mathbb{E}[\|\zeta(w_t, X)\|^2] \leq \|\mathbb{E}[\zeta(w_t, X)]\|^2$$

and hence applying lemma 6.5 yields

$$\begin{aligned}
\mathbb{E}[\|w_t - \check{w}\|^2] &\leq \|w_t - \check{w}\|^2 + \gamma_t^2 \|\zeta(w_t)\|^2 - 2\gamma_t \alpha \|\check{w} - v\|^2 \\
&= (1 - 2\gamma_t \alpha) \|w_t - \check{w}\|^2 + \gamma_t^2 \|\zeta(w_t)\|^2 \\
&= (1 - 2t^{-1}) \|w_t - \check{w}\|^2 + \frac{\|\zeta(w_t)\|^2}{\alpha^2 t^2}
\end{aligned}$$

where we used $\gamma_t = \frac{1}{\alpha t}$ in the last step. For $t = 1$ we know that

$$\mathbb{E}[\|w_1 - \check{w}\|^2] \leq \frac{\|\zeta(w_1)\|^2}{\alpha^2}$$

by lemma 6.6. The claim follows by an induction argument. $\square$

Theorem 6.7 implies a $\mathcal{O}(1/\sqrt{t})$ rate of convergence for the minimizing sequence $(w_t) \rightarrow \check{w}$, whenever $\|\zeta(w_t)\|^2$ is bounded. Using the previous argumentation we can also derive a bound for the objective function f .

THEOREM 6.8. Assume the prerequisites of theorem 6.7 are fulfilled. Assume additionally, that f is Fréchet differentiable with β -Lipschitz continuous stochastic gradient $\zeta(w) = \mathbb{E}[\zeta(w, X)] \in \partial f(w) = \{\nabla f(w)\}$ and let $\zeta(w, X)$ be strongly integrable as before, then

$$\mathbb{E}[f(w_t) - f(\check{w})] \leq \frac{\beta}{2} \frac{\|\zeta(w_t)\|^2}{\alpha^2 t}$$

for all $w_t \in H$.

PROOF. Since $\zeta(w)$ is β -Lipschitz continuous it holds¹³ that

¹³ *Ibid.*, Proposition Lemma 1.30.

$$f(v) - f(w) \leq \langle \zeta(w), v - w \rangle + \frac{\beta}{2} \|w - v\|^2$$

for all $v, w \in H$. Furthermore, $\langle \zeta(\check{w}), v - \check{w} \rangle = 0$, if $\check{w}$ is a minimizer of f in H by Fermat's rule. Combined with theorem 6.7 and lemma 5.10 this yields

$$\mathbb{E}[f(w_t) - f(\check{w})] \leq \frac{\beta}{2} \mathbb{E}[\|\check{w} - v\|^2] \leq \frac{\beta}{2} \frac{\|\zeta(w_t)\|^2}{\alpha^2 t}$$

which concludes the proof. $\square$

It follows that the rate of convergence of stochastic approximation on the strongly convex objective function f is $\mathcal{O}(1/t)$.

6.4 *Convex EXPoSE*

The previous results are independent of EXPoSE and can be used for general infinite-dimensional optimization problems. In this section we look at the EXPoSE model estimation from the perspective of such a stochastic optimization problem to find an ϵ -accurate *constant* time approximation of $\mu[\mathbb{P}_X]$. To achieve this we apply the bounds derived in the previous sections and show that there exists an optimal solution for our optimization problem. Moreover, we prove that this solution is unique. To achieve this goal we need to show that the objective function is strongly convex, and the derivative is β -Lipschitz and Bochner integrable.

OBVIOUSLY, if $\mu[\mathbb{P}_X] \in H \subset \mathcal{H}$, then the kernel mean map is the solution of the following optimization problem:

$$\begin{aligned} \min_{w \in H} g(w) &= \min_{w \in H} \|\mu[\mathbb{P}_X] - w\|^2 \\ &= \min_{w \in H} \langle w, w \rangle - 2\langle \mu[\mathbb{P}_X], w \rangle + \langle \mu[\mathbb{P}_X], \mu[\mathbb{P}_X] \rangle \\ &= \min_{w \in H} \frac{1}{2} \langle w, w \rangle - \langle \mu[\mathbb{P}_X], w \rangle. \end{aligned}$$

This is equivalent to the *stochastic optimization problem*, where we minimize over the expectation of the (stochastic) objective function f as

$$\min_{w \in H} \mathbb{E}[f(w, X)] = \min_{w \in H} \int_{\mathcal{X}} f(w, x) d\mathbb{P}_X(x),$$

where f is defined as

$$f(w, X) = \frac{1}{2} \langle w, w \rangle - \langle \phi(X), w \rangle.$$

For the remainder of this section we are concerned with this particular objective function.

6.4.1 *Existence & Uniqueness*

We saw that an infinite-dimensional optimization problem requires some special consideration. In the following we will show that a minimizer $\check{w}$ of the stochastic optimization problem of EXPoSE *exists* and is *unique*. As in the previous chapters, we assume that the input space $\mathcal{X}$ is a separable topological space and that the reproducing kernel is continuous as required by (A-1) and (A-2).

- (A-1) The input space $\mathcal{X}$ is a separable topological space.
- (A-2) The reproducing kernel k on $\mathcal{X}$ is continuous.

OUR FIRST CONCERN is the existence of $\check{w}$ and requires assumption (A-3) *i.e.* the kernel k is bounded in expectation such that

existence of $\check{w}$

$$\int_{\mathcal{X}} \sqrt{k(x, x)} d\mathbb{P}_X(x) \leq c$$

for some constant $c > 0$. In the following let $H \subset \mathcal{H}$ be a ball with radius c defined as

$$H = \{g \in \mathcal{H} \mid \|g\| \leq c\}$$

which is a closed bounded convex set. It is not difficult to show the existence of a minimizer of the objective function

$$\min_{w \in H} \mathbb{E}[f(w, X)] = \min_{w \in H} \int_{\mathcal{X}} f(w, x) d\mathbb{P}(x),$$

in H because we already know that the optimal solution is $\tilde{w} = \mu[\mathbb{P}_X]$. We also know that $\tilde{w}$ is an element of H since

$$\begin{aligned} \|\mu[\mathbb{P}_X]\| &= \left\| \int_{\mathcal{X}} \phi(x) d\mathbb{P}_X(x) \right\| \\ &\leq \int_{\mathcal{X}} \|\phi(x)\| d\mathbb{P}_X(x) \\ &\leq \int_{\mathcal{X}} \sqrt{k(x, x)} d\mathbb{P}_X(x) \\ &\leq c \end{aligned}$$

by assumption that k bounded in expectation (A-3).

THE UNIQUENESS of the minimizer $\tilde{w}$ requires f to have certain properties specified in the next corollary.

uniqueness of $\tilde{w}$

COROLLARY 6.9. Let H be a subset of $\mathcal{H}$. If $f: H \rightarrow \mathbb{R} \cup \{+\infty\}$ is proper, convex, coercive and lower-semicontinuous, then $\arg \min f$ is nonempty and weakly compact. If, moreover, f is strictly convex, then $\arg \min f$ is a singleton.¹⁴

¹⁴ Peypouquet, *Convex Optimization in Normed Spaces: Theory, Methods and Examples*, Corollary 2.19.

In the following we show that all these conditions hold. We start with two simple observations: First, by definition, all Hilbert spaces are *reflexive* since the embedding $x \mapsto (x^* \mapsto \langle x^*, x \rangle_*)$ into the bi-dual (Def. B.24) space is surjective. Second, the function f is *proper* as $\text{dom}(f) \neq \emptyset$ since $\mu[\mathbb{P}_X] \in \text{dom}(f)$. It remains to show that f is convex, coercive and lower-semicontinuous.

DEFINITION 6.10 (*weakly sequentially lower semicontinuous*). Let H be reflexive. A function $f: H \rightarrow \mathbb{R} \cup \{+\infty\}$ with the property

$$f(w) \leq \liminf_{t \rightarrow \infty} f(w_t)$$

for all sequences (w_t) which converge weakly to w i.e. $w_t \rightharpoonup w$ is called *weakly sequentially lower semicontinuous*. If the above property holds for all sequences (w_t) that converge strongly to w , $w_t \rightarrow w$, then f is said to be *lower semicontinuous*.

We now show that our objective function f is strongly convex and has continuous derivative which is subject to the subsequent proposition.

PROPOSITION 6.11. The objective $f(w, X) = \frac{1}{2} \langle w, w \rangle - \langle \phi(X), w \rangle$ is α -strongly convex and its stochastic gradient $\nabla f(w, X) = w - \phi(X)$ is β -Lipschitz with $\alpha = \beta = 1$.

PROOF. We know that f is α -strongly convex with $\alpha = 1$ since,

$$\begin{aligned} w \mapsto f(w, X) - \frac{\alpha}{2} \|w\|^2 &= \frac{1}{2} \langle w, w \rangle - \langle \phi(X), w \rangle - \frac{\alpha}{2} \|w\|^2 \\ &= -\langle \phi(X), w \rangle \end{aligned}$$

is convex in $w \in \mathcal{H}$. Furthermore, $Df(w, X): z \mapsto \langle w - \phi(X), z \rangle_\star \in \mathcal{H}^\star$ is the Fréchet derivative of f at $w \in \mathcal{H}$ since,

$$\lim_{h \rightarrow 0} \frac{\|f(w + h, X) - f(w, X) - \langle Df(w, X), h \rangle_\star\|}{\|h\|} = 0$$

using the notation $\langle \cdot, \cdot \rangle_\star$ for the dual map. By Riesz' representation theorem we can identify $Df(w, X): z \mapsto \langle w - \phi(X), z \rangle_\star$ with an element in $\mathcal{H}$ which we denote by $\nabla f(w, X) = w - \phi(X)$. This function is β -Lipschitz with constant $\beta = 1$ since

$$\begin{aligned} \|\nabla f(w, X) - \nabla f(v, X)\| &= \|w - \phi(X) - v + \phi(X)\| \\ &= \|w - v\| \end{aligned}$$

for all w and v in $\mathcal{H}$. □

We already showed that H is reflexive and f is proper. Since f strongly convex, it is strictly convex and coercive.¹⁵ Furthermore, f is continuous and thus also lower semicontinuous. Therefore, the remaining conditions of corollary 6.9 are also fulfilled and we can conclude that $\tilde{w}$ is the unique minimizer of f .

¹⁵ Peypouquet, *Convex Optimization in Normed Spaces: Theory, Methods and Examples*, Corollary 2.20.

6.4.2 Convergence Rates of the Kernel Mean Map

To derive exact rates of convergence for `EXPPOSE`, it remains to bound the stochastic gradient $\nabla f(w_t, X)$ in theorem 6.7 and theorem 6.8.¹⁶ The main contribution of this chapter is stated in the next theorem.

¹⁶ Since ∇f is a gradient, it is also subgradient.

THEOREM 6.12. Let k be a kernel such that (A-1) and (A-2) are fulfilled and $\mathcal{H}$ be the corresponding RKHS. Assume that the kernel k is bounded in expectation *i. e.*

$$\int_{\mathcal{X}} \sqrt{k(x, x)} d\mathbb{P}_X(x) \leq c$$

and let H be the ball in $\mathcal{H}$ with radius c , then

$$\begin{aligned} \mathbb{E}[\|w_t - \mu[\mathbb{P}_X]\|^2] &\leq \frac{4c^2}{t} \quad \text{and} \\ \mathbb{E}[f(w_t) - f(\mu[\mathbb{P}_X])] &\leq \frac{2c^2}{t} \end{aligned}$$

for all $t \in \mathbb{N}$.

PROOF. We first note that for w_t is an element of H for all $t \in \mathbb{N}$. This holds for $t = 1$ by definition of w_1 and for $t \geq 2$ the projection operator

Π_X guarantees that $w_t \in H$ which implies $\|w_t\| \leq c$. By definition of the EXPoSE stochastic gradient we get

$$\begin{aligned} \mathbb{E}[\|\nabla f(w_t, X)\|^2] &= \mathbb{E}[\|w_t - \phi(X)\|^2] \\ &\leq \|w_t\|^2 + \mathbb{E}[\|\phi(X)\|^2] + 2\|w_t\|\mathbb{E}[\|\phi(X)\|] \\ &\leq c^2 + \mathbb{E}[k(X, X)] + 2c\mathbb{E}[\sqrt{k(X, X)}] \\ &\leq 4c^2. \end{aligned}$$

This also shows that $\nabla f(w_t, X)$ is strongly integrable as required by theorem 6.7 and theorem 6.8 in which we plug in the upper bound of $\nabla f(w_t, X)$ to conclude the proof. $\square$

6.4.3 Convergence of EXPoSE

Since w_t converges strongly to $\check{w}$ as shown in theorem 6.12, we also have the weak convergence¹⁷

¹⁷ *Ibid.*

$$\lim_{t \rightarrow \infty} \langle u, w_t \rangle = \langle u, \check{w} \rangle, \quad \forall u \in H$$

and especially

$$\lim_{t \rightarrow \infty} \langle \phi(y), w_t \rangle = \langle \phi(y), \mu[\mathbb{P}_X] \rangle = \eta(y)$$

for all $y \in \mathcal{X}$ which justifies the use of w_t as a surrogate for $\mu[\mathbb{P}_X]$. The stochastic optimization procedure for EXPoSE is summarized in algorithm 6.3.

REQUIRE:

1: T : the number of iterations or ϵ : accuracy

ALGORITHM:

2: Set $w_1 \leftarrow 0$

3: FOR $t \leftarrow 1, 2, \dots, T$ DO

4: Sample x_t uniformly from $\mathbb{P}$

5: Set $\gamma_t \leftarrow \frac{1}{t}$

6: Set $f(w_t, X) \leftarrow w_t - \phi(x_t)$

7: Update $w_{t+1} \leftarrow w_t - \gamma_t \nabla f(w_t, X)$

8: Project $w_{t+1} \leftarrow w_{t+1} \cdot \max\{1, c\|w_{t+1}\|\}^{-1}$

9: RETURN w_{T+1}

Algorithm 6.3: A stochastic optimization procedure for EXPoSE.

We emphasize that the stochastic optimization approach presented here is relatively simple and requires only a few lines of code to implement. It also does not introduce additional parameters since the step size is known to be $\gamma_t = \frac{1}{t}$. Step sizes are crucial and difficult to determine in most optimization algorithms as they have a significant effect on the results. The bound of the kernel function is typically known and the number of iterations T is a trade-off between computing time and accuracy. Alternatively, given a desired accuracy ϵ , the number of iterations T can be calculated. Also, the projection operator Π_H in the last step takes a form which can efficiently be computed.

6.4.4 Regularization

We would like to mention that the reformulation of `EXPPOSE` as an optimization problem also introduces the opportunity to add constraints or enforce certain properties to the objective function. One approach is to define a general *regularizer* $\lambda\omega(w)$ on $\mathcal{H}$ replacing $\frac{1}{2}\langle w, w \rangle$ in the objective function as

$$\min_{w \in \mathcal{H}} \mathbb{E}[f(w, X)] = \min_{w \in \mathcal{H}} \int_{\mathcal{X}} \lambda\omega(w) - \langle \phi(X), w \rangle d\mathbb{P}(x),$$

with some regularization parameter $\lambda \geq 0$. An example would be to add a roughness penalty to the space of functions in $\mathcal{H}$ defining

$$\lambda\omega(w) = \lambda \langle D^2 w, D^2 w \rangle$$

where D denote the differential operator. However, the disadvantage of other objective functions is that these are in general not strongly convex yielding a slower convergence rate than the one derived in this chapter.

6.4.5 Other Optimization Routines

There are various alternative optimization techniques such as projected gradient descent¹⁸ or Nesterov's accelerated gradient descent¹⁹ which can also be applied in principle. However, a single gradient evaluation of these methods takes already $\mathcal{O}(n)$ time and hence, would be slower than the originally proposed `EXPPOSE` approach. Other stochastic gradient methods²⁰ can obtain a better convergence rate than $\mathcal{O}(1/t)$ for an objective function, composed of a sum of smooth functions. However, these approaches require multiple passes over the dataset and are therefore of no benefit.

¹⁸ Boyd et al., *Convex optimization*.

¹⁹ Nesterov, *A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$* .

²⁰ Roux et al., *A stochastic gradient method with an exponential convergence rate for finite training sets*.

6.5 Experimental Evaluation

In this section we present experimental results demonstrating the benefit of the proposed approach. This is challenging since the true data distribution $\mathbb{P}_X$ is unknown and it is therefore not possible to determine a closed-form solution of $\mu[\mathbb{P}_X]$. After all, we use the empirical kernel mean map $\mu[\mathbb{P}_n]$ as a surrogate for $\mu[\mathbb{P}_X]$ to evaluate the behavior of

$$\|w_t - \mu[\mathbb{P}_n]\| \approx \|w_t - \mu[\mathbb{P}_X]\|$$

as t increases. For sufficiently large sample sizes n we can expect $\mu[\mathbb{P}_n]$ to be a good proxy for $\mu[\mathbb{P}_X]$ by the law of large numbers. While it is theoretically possible to calculate $\|w_t - \mu[\mathbb{P}_n]\|$ explicitly this operation is of order $\mathcal{O}(n^2)$ and therefore intractable for large n . We face the dilemma that with a small sample size n we cannot expect $\mu[\mathbb{P}_n]$ to be a good proxy for $\mu[\mathbb{P}_X]$ and with a large sample size we cannot compute $\|w_t - \mu[\mathbb{P}_n]\|$. This requires us to avoid explicit feature maps ϕ and use their approximate counterpart $\hat{\phi}$ (§5.5) in the following experiment. We emphasize that the use of approximate feature maps has no impact on the previous theoretical analysis and all bounds hold in infinite-dimensional RKHSs and for the corresponding feature maps ϕ . The sole purpose of these approximations here are the experimental evaluations.

6.5.1 Setup & Datasets

We perform our analysis on the three largest datasets: KDDCUP, the Google Street View House Numbers (SVHN) and MNIST as described in the experimental evaluation of EXPOSE (§5.8.2). The bandwidth σ^2 of the squared exponential kernel used for these datasets are 5.6, 7.8 and 7.0 respectively, which we found to yield a reasonable anomaly detection performance. For MNIST and SVHN we use all instances of digit 1 as normal data and add 1% of randomly sampled images from the remaining digits as anomalies. A different normal/anomaly setup had no significant impact on the experimental results.

At each iteration of algorithm 6.3, we uniformly choose an instance from the remaining elements of the training set and the model w_t is updated according to the algorithm. Besides the convergence rate of the model, we will examine the anomaly detection scores

$$\begin{aligned}\eta(y) &= \langle \phi(y), \mu[\mathbb{P}_n] \rangle \quad \text{and} \\ \eta_t(y) &= \langle \phi(y), w_t \rangle\end{aligned}$$

calculated by the original EXPOSE predictor and the stochastic optimization approximation, respectively. For this purpose, we periodically evaluate η and η_t on 10 000 test instances which contain 50% normal and 50% anomalous observations. These test instances are not part of the training datasets and are dedicated to evaluate the classifier.

6.5.2 Evaluation

The experimental results with approximate feature maps are shown in Fig. 6.3. The first plot in each column contains traces of the objective function $f(w_t) - f(\check{w})$, where we estimated $\check{w}$ by $\mu[\mathbb{P}_n]$. We see that the stochastic optimization algorithm already reaches a reasonable low objective after a few thousand iterations. More important, we observe a similar effect in the second row when comparing $\|w_t - \check{w}\|$. w_t gets close to $\check{w}$ relatively fast, but it takes much more samples to estimate $\check{w}$ with a high accuracy. However, a high accuracy estimation is unnecessary to obtain good anomaly detection performance as shown in the following. We measure the anomaly detection rate by first plugging w_t and $\check{w}$ into the EXPOSE detectors η_t and η respectively. Thereafter, scores for all instances of the test dataset are calculated. The differences between these scores are shown in the third row. We see again that the stochastic optimization approximation η_t yields similar scores as η using the full model. The last row illustrates the development of the area under ROC²¹ as the optimization routine performs more iterations. *After only a few thousand iterations, η_t reaches the same predictive performance as the original EXPOSE predictor η .*

This confirms that a high accuracy approximation of $\mu[\mathbb{P}_X]$ does not necessarily lead to a better predictor. The key is, that for a given ϵ we can reach $\|w_t - \mu[\mathbb{P}_X]\| < \epsilon$ in a fixed number of iterations, *independent* of the dataset size n which reduced the computational complexity from $\mathcal{O}(n)$ to $\mathcal{O}(1)$ as claimed.

To see the performance gain in seconds, we stop the algorithm at a relatively high accuracy of $\|w_t - \mu[\mathbb{P}_X]\| \leq \epsilon = 0.005$ and compare the full model η with the approximation η_t (table 6.2). We observe that the

²¹ Hanley et al., *The meaning and use of the area under a receiver operating characteristic (ROC) curve.*

training time is significantly shorter while the predictive performance (the area under ROC) is still close to the one obtained by the full model. It is important to note that the computational complexity of η_t is constant. So even if the datasets grow in size, we would not need more training time to achieve the same accuracy.

	TIME [sec]			AUC	
	η	η_t		η	η_t
SVHN	437	32	■ –	0.890	0.890
MNIST	248	19	■ –	0.991	0.990
KDDCUP	961	81	■ –	0.981	0.981

Table 6.2: Comparison of the η and the stochastic approximation η_t .

6.6 Summary

In this chapter we view the `EXPPOSE` anomaly detection algorithm from the perspective of a stochastic optimization problem which enables us to find an ϵ -accurate approximation of the kernel mean map $\mu[\mathbb{P}_X]$ in *constant time*, independent of the training dataset size n .

To achieve this goal, we take existing stochastic optimization approaches and port them to infinite-dimensional problems. Moreover, we carefully choose an objective function f such that an optimal solution can be shown to *exist* and to be *unique*. The choice of the working subset H in $\mathcal{H}$ is of crucial importance to bound the gradient and to derive the step size automatically. It is also essential that f is chosen such that it is strongly convex and obeys a Lipschitz continuous gradient with the right constants.

As a result, the proposed approximation reduces the computational complexity of `EXPPOSE` and a sample estimation of the kernel mean map $\mu[\mathbb{P}_X]$ from the previous $\mathcal{O}(n)$ to $\mathcal{O}(1)$, whenever an ϵ -accurate estimation is sufficient. More precisely, we are able to determine the number of necessary stochastic optimization iterations T for a user defined error threshold ϵ such that $\mathbb{E}[\|w_T - \mu[\mathbb{P}_X]\|] < \epsilon$. The assumption is, that a very high accuracy estimation $\mu[\mathbb{P}_X]$ does not necessarily result in a better anomaly detection performance and hence there is no benefit in spending more computational resources than those unavoidable. This intuition is also confirmed experimentally on three large-scale datasets, where we reach the same anomaly detection performance long before all data is incorporated into the model.

We emphasize that, unlike other optimization routines, `EXPPOSE` does not have a regularization parameter. This is an important property as a regularization parameter is difficult to tune. An example is the Pegasos algorithm for support vector machines where “[...] the runtime to achieve a predetermined suboptimality threshold would increase in proportion to λ [the regularization parameter]. Very small values of λ (small amounts of regularization) result in rather long runtimes.”²² We expect our contribution to have significant implications for large-scale applications such as anomaly detection problems and other techniques, which are based on the kernel mean embedding.

²² Shalev-Shwartz et al., *Pegasos: Primal estimated sub-gradient solver for SVM*.

Figure 6.3: Evaluation of the stochastic optimization approach for EXPOSE. Each experiment was repeated 5 times represented by a dot. A darker color indicates a value closer to the mean of the 5 repetitions.

Online & Streaming Anomaly Detection

SYNOPSIS · In the last two chapters we introduced the `EXPOSE` anomaly detector and derived a constant-time learning technique based on stochastic optimization. In this chapter we develop the `EXPOSE` approach further to *online* and *streaming* anomaly detection. Here learning and prediction takes place simultaneously and new instances are classified based on previous observations from the stream. One of the key challenges we approach is the problem of evolving data streams which is also known as *concept drift*. We first present one incremental learning scheme which is lossless, *i.e.* the classifier trained incrementally coincides with the one obtained through batch learning. Subsequently, we propose two online learning rules to approach the problem of concept drift based on *windowing* and *gradual forgetting*. These approaches are generalized to learning with temporal kernel functions which explicitly models the concept of forgetting as part of the kernel function. We finally study the behavior of the previously proposed methods under *adversarial impact* where an attacker aims at the integrity of the anomaly detection system.

THE following content was partly published in

- M. Schneider, W. Ertel, and F. Ramos (2016a).
“Expected Similarity Estimation for Large-Scale Batch and Streaming Anomaly Detection”. In: *Machine Learning* 105:3, pp. 305–333
- M. Schneider, W. Ertel, and F. Ramos (2016b).
“Kernel Embeddings for Large-Scale Anomaly Detection”. In: *International Conference on Machine Learning (ICML 2016): Anomaly Detection Workshop*

7.1 Online Learning

Most classical machine learning algorithms first learn a predictor from a training dataset and then use this predictor to classify new instances. These algorithms make no predictions at the learning stage and are static afterwards, *i.e.* additional information from observations after the model is trained are not integrated to improve the classifier. The only way to change the model is to re-train the algorithm on a new dataset. They are hence referred to as *offline* or *batch* learning algorithms.

Furthermore, these offline techniques assume that the training dataset $z = (x_1, \dots, x_n)$ from the sample space $(\mathcal{Z}, \mathcal{X}, \mathbb{P}_Z)$ is permanently available and its elements are independent and identically distributed (i.i.d.), that is, $\mathbb{P}_Z = \bigotimes_{i=1}^n \mathbb{P}_X$. However, this assumption is often impractical in many situations and especially for *online* and *streaming* anomaly detection where observations arrive in a sequence $(x_1, x_2, x_3, \dots)$. A source of such data can be real-time surveillance systems, Internet traffic, sensor networks or financial transactions.

7.1.1 Data Streams

We define a *data stream* as a *temporally ordered* sequence of observations which is potentially *unbounded* in size. It is therefore impossible to store the entire stream or pass through it multiple times. This requires the development of new *online* learning algorithms which are able to continuously updated the classifier (model) as new observations are made without the constraint to store all past instances. This property is also reflected in the following excerpts where the term “hypothesis” refers in our terminology to the classifier or anomaly detector.

Online learning algorithms process each training instance once “on arrival” without the need for storage and reprocessing, and maintain a current hypothesis that reflects all the training instances seen so far.

— Oza, *Online bagging and boosting*

Online learning takes place in a sequence of consecutive rounds. On each round, the learner is given a question and is required to provide an answer to this question. [...] the learner may update the hypothesis after each round so as to be more accurate in later rounds.

— Shalev-Shwartz, *Online Learning: Theory, Algorithms, and Applications*

A crucial requirement in streaming anomaly detection is therefore the ability to make predictions as data arrives based on the information obtained from previous observations. Consequently, learning and prediction on streams has to take place simultaneously.

HOWEVER, online learning techniques should not be restricted to streaming applications, as they can also be used to process massive non-streaming datasets. Such datasets are distributed on large clusters where random access is too expensive and a sequential processing is therefore more realistic. We refer to this situation as *incremental learning*.

In contrast to conventional data streams, incremental learning assumes that there is no temporal ordering in the sequence and that the samples are still independently distributed. Hence incremental learning algorithms can be seen as a more memory-efficient version of their batch counterparts with the ability to include additional information at a later state without the need of a complete re-training. However, since incremental algorithms observe only one instance at a time, they are sometimes slower or inferior to their batch counterparts. Ideally, if both are trained on the same dataset the classifiers obtained through batch and incremental learning should coincide. This is then called a *lossless* incremental technique.

7.1.2 Concept Drift

A prominent characteristic of data streams is *non-stationarity* which means that our understanding of what is normal and anomalous changes over time. This is a property known as *concept drift*.¹ In environmental monitoring, for example the definition of “normal temperature” changes naturally with seasons. We can also expect that human behavior changes over time which requires us to redefine what anomalous actions or activities are. In system health monitoring we have to consider that the physical state of a machine can change over time *e.g.* due to abrasion or developing external forces. A good detector can react to such changing concepts by forgetting outdated information while incorporating new knowledge.

Consequently, the observations in non-stationary environments are no longer i.i.d. and the input space distribution changes over time. Mathematically this is modeled by the stochastic process $\{X_t \mid t \in \mathbb{N}\}$ and the corresponding sequence of distributions $\mathbb{P}_{X_t}$. Concept drift has varying manifestations as illustrated in Fig. 7.1.

¹ Gama et al., *A survey on concept drift adaptation*.

Figure 7.1: The different concept drift types as defined by Gama et al., Fig 2.

A *sudden* drift occurs if concepts are switched abruptly. This can happen, for example, in novelty detection. Once a novel instance is discovered, a second observation would not be novel again since it is already known. Concepts can also change *incrementally* with intermediate concepts *e.g.* temperature drift within a year. A *gradual* drift is different from an incremental one in the sense that there are no intermediate concepts, but the frequency of observations from each concept changes. We therefore observe essentially a shift in the sampling distribution and not the instances. The classical example of *reoccurring* concepts are seasons, where a concept drift happens, but old concepts are observed again after some time. In this case it would be advantageous to exploit the knowledge gained from past observations of the same concept. We will show that this can be implemented with so-called *temporal kernel functions*. The list of concept drift types presented here is not exhaustive and we can expect to observe a combination of various forms in real applications. However, this list demonstrates the variety of possibilities one might be confronted with.

Figure 7.2 illustrates the difference between incremental learning and model adaptation to concept drift. The eight plots represent EXPOSE predictions for the observations indicated by the black dots. Each row displays four snapshots with increasing time from left to right as data is observed from both clusters. We first sampled only from the left cluster and later only from the right. In the top row we incrementally build the model by adding more knowledge to it whereas in the bottom row the model evolves and slowly forgets outdated observations (the points from the left cluster).

Figure 7.2: An illustration between the difference of incremental learning and model adaptation to concept drift.

² Domingos et al., *Catching up with the Data: Research Issues in Mining Data Streams*.

IN summary, we identify the following requirements for algorithms operating on data streams similar to the list of Domingos et al.² The algorithm should

- Require only a small constant processing time per instance and use a fixed amount of memory, independent of the number of past observations. This is a direct consequence from the potential unboundedness of the data stream and the resulting impracticability to store the entire sequence.
- Build a model using at most a single pass over the data. Multiple passes over the same stream are often too expensive and generally not possible.
- Simultaneously learn and predict. This is necessary since we cannot wait until the stream is finished before we start the classification of arriving observations.
- Be able to deal with concept drift. It is necessary to adapt new concepts while forgetting outdated information. However, the compromise between robustness to noise or attacks and fast adaptation to changes should be controllable in an intuitive way.
- For streams without concept drift, incrementally learn a classifier which is equivalent to the one that would be obtained by an offline (batch) learning algorithm *i.e.* learn lossless. This can also be seen as an any-time algorithm which provides the best possible answer with the available information at hand.

There exist various incremental versions of common machine learning algorithms, however only a few are capable to perform anomaly detection. These techniques are not always lossless and often more difficult than their offline counterpart. An example of such an online anomaly detector is the incremental local outlier factor.³ Also nearest neighbor-based methods naturally fit into the incremental setting as new instances can simply be added to the dataset they are operating on. However, these techniques suffer from the same limitations as their batch counterparts. We will therefore consider algorithms which are especially designed for data stream learning in mind such as STORM, COD and HSTA (§3.3).

³ Pokrajac et al., *Incremental local outlier detection for data streams*.

7.2 *An Incremental EXPoSE Learning Approach*

We showed that EXPoSE can be trained in parallel and in distributed environments (§5.4). In this section we derive an incremental version of the EXPoSE anomaly detector which has no additional computational costs or parameters and is lossless. Given a stream $(x_1, x_2, x_3, \dots)$ of observations, the (empirical) EXPoSE classifier at time t is given by

$$\begin{aligned}\eta(y) &= \langle \phi(y), \mu[\mathbb{P}_t] \rangle \\ &= \left\langle \phi(y), \frac{1}{t} \sum_{i=1}^t \phi(x_i) \right\rangle\end{aligned}$$

where $\mu[\mathbb{P}_t]$ is the empirical kernel mean map calculated using the first t instances $(x_1, x_2, \dots, x_t)$ of the stream. Obviously, $\mu[\mathbb{P}_t]$ can be computed incrementally as

$$\begin{aligned}\mu[\mathbb{P}_t] &= \frac{1}{t} \sum_{i=1}^t \phi(x_i) \\ &= \mu[\mathbb{P}_{t-1}] + \frac{1}{t} (\phi(x_t) - \mu[\mathbb{P}_{t-1}])\end{aligned}\tag{7.1}$$

using the previous model $\mu[\mathbb{P}_{t-1}]$. The initial model is set to $\mu[\mathbb{P}_1] = \phi(x_1)$. Each model update can be performed in constant time and hence does neither increase the computational complexity nor the memory requirements of EXPoSE. Furthermore, this approach is clearly lossless.

7.3 *EXPoSE with Concept Drift Adaptation*

In this dissertation we are not concerned with the detection of concept drift, but we present different approaches which allow EXPoSE to adapt to changes in the environment. These techniques can either be used to perform a *blind*, or an *informed* adaptation. In the former we continuously “forget” information obtained from older observations without the knowledge if a concept drift took place or not. In contrast, we can perform informed adaptations by combining our proposed approaches with an external change detector and trigger a concept drift adaptation only if necessary.⁴

The most common ways to detect concept drift are to monitor the input space distribution or the classification performance and continuously statistically compare them to a baseline. An example of the former are the *cumulative sum* (CUSUM) sequential analysis⁵ and its variants. These methods assume that a concept drift occurred if the mean of the incoming data significantly changes. Alternatively one can compare the data distributions on different sliding windows using a statistical test based on Chernoff bounds.⁶ An approach which compares the classification accuracy of two models (*paired learners*) is presented by Bach et al.⁷ It uses a *baseline* model with is used to classify new instances and a *reactive* model trained on a shorter window of recent observations. A drift is detected based on the difference in accuracy between the baseline and reactive models.

In this section we derive efficient concept drift adaptation techniques for EXPoSE based on the most common approaches which either utilize *windowing* or *forgetting* mechanisms. In the following we denote the EXPoSE

⁴ Gama et al., *A survey on concept drift adaptation*.

⁵ Page, *Continuous Inspection Schemes*.

⁶ Kifer et al., *Detecting change in data streams*.

⁷ Bach et al., *Paired learners for concept drift*.

model at time t with w_t since the equality $w_t = \mu[\mathbb{P}_t]$ will not necessarily hold when concept drift adaptation is implemented.

7.3.1 Windowing

Windowing is a common technique which uses a buffer (the window) of $l \in \mathbb{N}$ previous observations. Whenever a new observation is added to the window, the oldest one is discarded. We can efficiently implement windowing for EXP_{OSE} given a data stream $(x_1, x_2, x_3 \dots)$ and the window size l . For $t < l$ we set $w_t = \frac{1}{t} \sum_{i=1}^t \phi(x_i)$ and use the incremental update

$$\begin{aligned} w_t &= \frac{1}{l} \sum_{i=t-l+1}^t \phi(x_i) \\ &= w_{t-1} + \frac{1}{l} \phi(x_t) - \frac{1}{l} \phi(x_{t-l}), \end{aligned}$$

whenever $t \geq l$. The computational complexity is still constant, however the memory consumption grows linearly with the window size. We use $w\text{-EXP}_{\text{OSE}}$ to refer to the sliding window implementation of concept drift adaptation.

The downside of a sliding window mechanism is the requirement to keep the past $l \in \mathbb{N}$ events in memory. Additionally, the sudden discard of a data point from the model can lead to abrupt changes in the classifier's predictions which is not always desirable. Another critical task is to select the correct window size. A shorter sliding window allows the algorithm to react faster to changes and requires less memory though the available data might not be representative to model the input space distribution or noise in the data has too much negative impact. On the other hand a wider window may take too long to adapt to concept drifts. The window size is therefore often dynamically adjusted⁸ or multiple competing windows are used.⁹

7.3.2 Gradual Forgetting (Decay)

The problems of sliding window approaches can be avoided if a forgetting mechanism is applied where the influence of older data gradually vanishes. Ideally a parameter controls the trade-off between fast adaptation to new observations and robustness against noise in the data. We can realize such a forgetting mechanism for EXP_{OSE} by replacing the factor $\frac{1}{t}$ in Eq. (7.1) by a constant $\gamma \in [0, 1)$ yielding

$$w_t = \begin{cases} \phi(x_t) & \text{for } t = 1 \\ \gamma \phi(x_t) + (1 - \gamma)w_{t-1}, & \text{for } t > 1 \end{cases}$$

where, with $\gamma = 0$, no new observations are integrated into the model. Obviously, this operation can be performed in constant time.

We can see that the concept drift adaptation on data streams using a forgetting mechanism can be implemented for EXP_{OSE} in constant time and memory. We could also apply more sophisticated decay rules making γ a function of t or x_t in the context of informed drift adaptation. We refer to this approach with $\gamma\text{-EXP}_{\text{OSE}}$.

Figure 7.3: Illustration of a sliding window of size $l = 4$ over a stream.

⁸ Widmer et al., *Learning in the presence of concept drift and hidden contexts*.

⁹ Lazarescu et al., *Using multiple windows to track concept drift*.

Figure 7.4: Gradual forgetting assigns a weight to each instance. Starting with a high weight for the most recent observations the weights gradually become smaller for past events and eventually vanish.

7.3.3 Predictions on Data Streams

So far we introduced three different approaches to learn the model for EXPOSE on data streams: one incremental learning approach and two evolving techniques. In order to make a prediction as a new observation is made we have to normalize the calculated score. This is necessary as a model update changes the score's magnitude. The same problem is not present in the batch version of EXPOSE since the model is static after the learning phase. To avoid this problem we can normalize the score dividing by $\|w_t\|^2$ which yields

$$\eta(y) = \frac{\langle \phi(y), w_t \rangle}{\|w_t\|^2}$$

as the EXPOSE classifier. We emphasize that the calculation of the normalization constant does not change the limiting behavior of runtime and memory requirements we derived earlier in this section since w_t can be accessed in constant time.

7.4 Temporal Kernel Functions

We saw that gradual forgetting assigns a weight to each instance where the most recent observations are weighted the most and less weight is assigned for events further in the past. Alternatively, we can explicitly add a temporal dimension $\mathcal{T}$ to the inputs and use a kernel which assigns less weight to instances that are temporally far away. Such a function is called *spatio-temporal kernel*. The new input space is then given by the product space $\mathcal{U} = \mathcal{X} \times \mathcal{T}$.

Although we normally have a good intuition about how to derive ordinary kernel functions, the same has shown to be difficult in the spatio-temporal case. This might be the reason most spatio-temporal kernels are constructed as a combination of two ordinary kernel functions. Many of them share the property that the kernel can be written as a pointwise product of two other kernel functions

$$\begin{aligned} k: \mathcal{U} \times \mathcal{U} &\rightarrow \mathbb{R} \\ k((x, t), (y, s)) &= k_x(x, y) \cdot k_t(t, s) \end{aligned}$$

where k_x is dedicated to model the dependence between instances in $\mathcal{X}$ and k_t is responsible for the temporal similarity. This type of kernel can be found in many textbooks about geostatistics¹⁰ under the name *intrinsic correlation model* and is known as *dot product* kernel in machine learning.¹¹ The result is a tensor product of Hilbert spaces.

Let $(\mathcal{H}_x, \langle \cdot, \cdot \rangle_x)$ and $(\mathcal{H}_t, \langle \cdot, \cdot \rangle_t)$ be the reproducing kernel Hilbert spaces with corresponding kernel and feature functions

$$\begin{aligned} k_x(x, y) &= \langle \phi(x), \phi(y) \rangle_x \quad \text{and} \\ k_t(t, s) &= \langle \varphi(t), \varphi(s) \rangle_t, \end{aligned}$$

respectively. Then

$$k_x(x, y) \cdot k_t(t, s) = \langle \phi(x) \otimes \varphi(t), \phi(y) \otimes \varphi(s) \rangle_{x \otimes t}$$

is the spatio-temporal kernel written as a tensor product of feature functions (lemma 4.18) in the Hilbert space $(\mathcal{H}_x \otimes \mathcal{H}_t, \langle \cdot, \cdot \rangle_{x \otimes t})$.

¹⁰ Wackernagel, *Multivariate geostatistics: an introduction with applications*.

¹¹ Rasmussen et al., *Gaussian Processes for Machine Learning*.

7.4.1 Spatio-Temporal EXPOSE

Assuming the temporal space $(\mathcal{T}, \mathcal{T})$ is measurable and there exists a random variable $T: (\Omega, \mathcal{F}) \rightarrow (\mathcal{T}, \mathcal{T})$, then the kernel embedding of the product measure $\mathbb{P}_X \otimes \mathbb{P}_T$ of the input space distribution $\mathbb{P}_X$ and the temporal distribution $\mathbb{P}_T$ is

$$\mu[\mathbb{P}_X \otimes \mathbb{P}_T] = \int_{\mathcal{U}} \phi \otimes \varphi d[\mathbb{P}_X \otimes \mathbb{P}_T]$$

where the notation for the product measure $\mathbb{P}_X \otimes \mathbb{P}_T$ and the tensor product $\phi \otimes \varphi$ is overloaded, but should be clear from the context. The EXPOSE anomaly detector is then

$$\eta((y, t)) = \langle \phi(y) \otimes \varphi(t), \mu[\mathbb{P}_X \otimes \mathbb{P}_T] \rangle$$

for all $(y, t) \in \mathcal{U}$. An advantage of the spatio-temporal approach is that the algorithm does not “forget” previously observed information since it can always be accessed by selecting the corresponding temporal variable $t \in \mathcal{T}$. Furthermore, we can use the same approach to tackle the problem of reoccurring concept drift. In a time series plot as the one in Fig. 7.5 (left) the instances are separated in space *and* time. The latter is not necessary desirable since a reoccurring concept is seen as dissimilar from its previous observation. If we simply ignore the dependence on time and project the data onto the temperature axis, we would not be able to distinguish between spring and autumn. In such scenarios it is better to use a *periodic* kernel function such as the *sinc kernel* (§4.5) which maps the data in a space where seasons are close to each other, independent of the year.

Figure 7.5: Schematic of reoccurring concept drift adaptation with a periodic kernel. The kernel function is used to map the time into a dimension where seasons are close to each other.

THERE IS NO best generic solution to concept drift adaptation on data streams. Depending on the drift type some approaches may work better than others. Additionally, we suggest to combine the approaches proposed in this chapter with an external drift detector and trigger a model adaptation only if necessary and in an informed way.¹²

7.5 Anomaly Detection under Adversarial Impact

It is clear that the objective of many anomaly detection applications is to prevent damage by identifying suspicious behavior. However, what

¹² Gama, *Knowledge discovery from data streams*.

if the anomaly detection system itself is under attack? We argued that a fundamental characteristic of streaming anomaly detection is the problem of non-stationary environments which requires us to perform concept drift adaptation. Such a situation can be exploited by an attacker who purposely manipulates the stream such that an anomalous observation is classified as normal which would otherwise be detected by the system. This is called *poisoning* or *causative integrity* attack.¹³

Previous works studied *online centroid anomaly detection* under adversarial impact where the input space distribution is modeled by a ball (hypersphere). Every instance within the hypersphere is classified as normal and those outside of the ball are labeled as anomalous. This model assumes that the radius of the ball is predefined and centered at the mean of normal data. The predictor is then

$$\eta(y) = \begin{cases} c_N & \text{if } \|y - \frac{1}{n} \sum_{i=1}^n x_i\| > \theta \\ c_A & \text{otherwise} \end{cases}$$

for radius $\theta > 0$. Nelson et al.¹⁴ analyzed this model in an incremental learning framework. An extension of their work to streaming anomaly detection under concept drift is provided by Kloft et al.¹⁵ Both limit their analysis to concept drift techniques which are based on a sliding window with a predefined buffer length.

A WEAKNESS of EXPOSE is that it was not designed with adversarial attacks in mind. This section analyzes the incremental and streaming versions of EXPOSE in this setting. On the contrary, batch learning methods are rarely subject to adversarial attacks as this would require control over the training set. Nevertheless the analysis performed for incremental learning can also be applied to the batch setting. We perform a worst-case investigation following the setup of Kloft et al.¹⁶ in which

- The attacker has complete knowledge about the learning algorithm. This is important since “one of the key principles of computer security, known as *Kerckhoff’s principle*, is that the robustness of any security instrument must not depend on keeping its operational functionality secret”.¹⁷
- The attacker has full control over the data stream and can inject arbitrary *attack points* to manipulate the anomaly detector.

¹³ Barreno et al., *Can machine learning be secure?*

¹⁴ Nelson et al., *Bounding an attack’s complexity for a simple learning model.*

¹⁵ Kloft et al., *Online anomaly detection under adversarial impact.*

¹⁶ Kloft et al., *Security analysis of online centroid anomaly detection.*

¹⁷ *Ibid.*

7.5.1 Poisoning Attacks against EXPoSE

Recall that the EXPOSE anomaly detector is defined as

$$\eta(y) = \langle \phi(y), w_t \rangle$$

where the model w_t is estimated from the observations of a data stream $(x_1, x_2, x_3, \dots)$ with undefined length. The goal of a poisoning attack is to manipulate the stream in a way such that the score $\eta(a)$ for an attack point $a \in \mathcal{X}$ is higher than a predefined threshold. We assume that the attacker cannot modify any previously observed instances, but is able to inject new observations into the stream. This manipulation must not be detected and therefore the attacker can only inject instances a_t which yield a sufficiently

high score within the aforementioned threshold. Assume that the kernel is bounded *i.e.* there exists a constant c such that $\|\phi(x)\| \leq c$ for all $x \in \mathcal{X}$. If the norm distance between two models w and $\tilde{w}$ is at most r , then

$$\begin{aligned} |\langle \phi(a), w \rangle - \langle \phi(y), \tilde{w} \rangle| &= |\langle \phi(a), w - \tilde{w} \rangle| \\ &\leq \|\phi(a)\| \cdot \|w - \tilde{w}\| \\ &\leq cr \end{aligned}$$

and therefore a shift of the `EXPPOSE` score is bounded by the shift of the model. Therefore, we can perform the analysis proposed in literature for online centroid anomaly detectors.

Without loss of generality we assume that an injected attack point $a_t \in \mathcal{X}$ must be within the ball with radius $\epsilon > 0$ centered at w_t *i.e.*

$$\|\phi(a_t) - w_t\| \leq \epsilon$$

to remain undetected. Under this constraint the attacker's objective is to shift the model w_t as far as possible using a greedy strategy.

WE START WITH the analysis of the incremental learning algorithm and assume that the attacker has full control over the data stream after n initial training points, written as $w_t = w_{n+i}$ with $i = t - n$.

THEOREM 7.1. A displacement of w_n of length $h > 0$ under the incremental learning rule of `EXPPOSE` requires at least

$$n \cdot \left(\exp\left(\frac{r}{\epsilon}\right) - 1 \right)$$

attack points.

Here we present an alternative simplified version of the proof provided by Nelson et al.¹⁸ with focus on `EXPPOSE`.

PROOF OF THEOREM 7.1. The incremental update rule of `EXPPOSE` after n instances is given by

$$w_t = \left(1 - \frac{1}{t}\right)w_{t-1} + \frac{1}{t}\phi(a_t)$$

and therefore we can bound the maximal possible shift after one iteration by

$$\begin{aligned} \|w_{t+1} - w_t\| &= \left\| \left(1 - \frac{1}{n+i}\right)w_t + \frac{1}{n+i}\phi(a_t) - w_t \right\| \\ &= \left\| \frac{1}{n+i}(\phi(a_t) - w_t) \right\| \\ &\leq \frac{1}{n+i}\epsilon, \end{aligned}$$

where we used the assumption that $\|\phi(a_t) - w_t\| \leq \epsilon$. Therefore, the maximal displacement which can be achieved by the attacker at time t is

$$\begin{aligned} \sum_{i=1}^t \frac{\epsilon}{n+i} &= \sum_{i=1}^{n+i} \frac{\epsilon}{i} - \sum_{i=1}^n \frac{\epsilon}{i} \\ &\leq \epsilon \ln(n+i) - \epsilon \ln(n) \\ &= \epsilon \ln\left(1 + \frac{i}{n}\right), \end{aligned}$$

where the upper bound can be obtained by bounding the harmonic series. The inversion of this inequality concludes the proof. $\square$

¹⁸ Nelson et al., *Bounding an attack's complexity for a simple learning model*.

That means an attacker needs an exponentially large number of attack points to poison our anomaly detector in the incremental learning context. In reality the situation might be even more difficult for an attacker without full control over the stream.

UNFORTUNATELY, when using a sliding window to perform concept drift adaptation, the results are not that optimistic.

THEOREM 7.2. A displacement of length $h > 0$ of the streaming anomaly detector with a finite sliding window update rule requires at least

$$l \cdot \frac{r}{\epsilon}$$

attack points, where $l \in \mathbb{N}$ is the window length.

PROOF. The window update rule of `EXPPOSE` using a buffer of length l is

$$w_t = w_{t-1} + \frac{1}{l}\phi(x_t) - \frac{1}{l}\phi(x_{t-l})$$

and therefore a model shift is bounded by

$$\begin{aligned} \|w_{t+1} - w_t\| &= \left\| w_t + \frac{1}{l}\phi(a_t) - \frac{1}{l}\phi(x_{t-l}) - w_t \right\| \\ &= \frac{1}{l} \|\phi(a_t) - \phi(x_{t-l})\| \\ &\leq \frac{\epsilon}{l} \end{aligned}$$

per update. □

In this case the number of attack points scales only linearly with the window size. Kloft et al.¹⁹ discuss additional update schemes, in which not the oldest instance is discarded from the window, but the nearest neighbor of the new observation or a randomly chosen instance is removed. However, we do not consider these update rules here which seem to be questionable for the purpose of concept drift adaptation.

¹⁹ Kloft et al., *Security analysis of online centroid anomaly detection*.

THIS SECTION DEMONSTRATED that anomaly detection under adversarial impact can be dangerous. The security analysis of the incremental `EXPPOSE` algorithm is optimistic in the sense that an attacker needs to inject an exponentially large number of attack points. However, this does not hold for concept drift adaptation based on sliding windows. Here the number of necessary attack points grows linearly with the window size and it is therefore necessary to consider other countermeasures as a protection against such attacks.

7.6 Experimental Evaluation

In this set of experiments we compare the streaming variants of `EXPPOSE` against `HSTA`, `STORM` and `COD`. All of these algorithms are blind methods as they adapt their models at regular intervals without knowing if a concept drift occurred or not.

THE evaluation of streaming algorithms is not as straightforward as the rating of batch learning techniques as there is no dedicated training phase after which the classifier can be evaluated. In this situation, there are two accepted techniques proposed in literature:

- Using a dedicated subset of the data (*holdout*) and evaluating the algorithm at regular time intervals. In this case, the holdout set must reflect the stream characteristics and therefore has to evolve with the stream to reflect concept drifts. Consequently, the holdout set is a data stream itself.
- Classify each instance as it is observed (*prequential*).²⁰ A performance metric can then be applied based on the prediction and the actual label of the instance. Since predictions are made on the stream directly, there are no special actions which have to be taken in case of concept drift. However, it is necessary that the stream contains anomalous instances, either naturally or artificially injected.

If possible, the holdout method is preferable since it is an unbiased risk estimator and we can use a balanced test set with the same number of normal instances and anomalies. This is a disadvantage of the prequential method since, by definition, an anomaly detection data stream contains only a few anomalies.

The imbalanced test set of the prequential method is problematic in the experiments in which STORM and COD are involved since these techniques assign a hard class label and not an anomaly score. Hence it is not possible to use the AUC as a performance measure which is insensitive to imbalanced classes. The *balanced accuracy* avoids this problem and uses

$$\frac{0.5 \cdot TP}{TP + FN} + \frac{0.5 \cdot TN}{TN + FP}$$

as a metric which compensates for the unequal class distribution.

²⁰ Dawid, *Present position and potential developments: Some personal views: Statistical theory: The prequential approach*.

Prequential originates from the terms predictive and sequential.

TP: true positives
TN: true negatives
FP: false positives
FN: false negatives

7.6.1 Datasets

There exist only a few non-synthetic datasets for anomaly detection with concept drift. Most of them are based on multi-class datasets, where each class represents a single concept. An example is to use the SVHN dataset and stream 9000 randomly sampled instances of the digits 1 to 9 in sequence, such that the 1000 instances of digit 1 appear first, then 1000 instances of digit 2 until digit 9 (see Fig. 7.6). This simulates a *sudden* concept drift. Every 25 time steps we calculate the accuracy using the holdout method for a dedicated random test set which comprises 500 instances of the normal class and 500 instances of anomalies. Here, the normal class is the digit which is streamed at time step t and all other classes are defined to be anomalies. Likewise, we proceed with the Satellite and Shuttle datasets.

Similarly, Ho²¹ propose the *three digit data stream* (TDDS) which contains four different concepts. Each concept consists of three digits of the USPS handwritten digits dataset.²² After all instances of concept 1 are processed, the stream switches to the second concept and so on until concept 4. We randomly inject 1% anomalies to each concept and use the prequential method for evaluation to calculate the balanced accuracy.

²¹ Ho, *A martingale framework for concept change detection in time-varying data streams*.

²² See Ho (*A martingale framework for concept change detection in time-varying data streams*) for a detailed description of the TDDS dataset.

Figure 7.6: The streaming SVHN experiment. A comparison of prediction accuracy under concept drift averaged over 5 repetitions.

ALL datasets presented so far contain one or more *sudden* (abrupt) concept drifts. Bifet et al.²³ propose a methodology to introduce a *smooth* (incremental) drift between two concepts. In this case each class represents a concept and their instances are sampled according to a Bernoulli distribution where the class probability smoothly changes from one class following a sigmoid function (Fig. 7.7). The concept drift occurs at t_0 and w defines the length of the drift interval. During this interval the instances of both concepts belong to the normal class. We apply this methodology on the USPS images to create the *smooth digit drift* (SDD) dataset. The stream starts with digit 1 and then smoothly changes to digit 2 at $t = 500$ with a drift interval of $w = 100$. The next drift to digit 3 occurs at $t = 1000$ and we repeat this procedure until digit 9. As before, we randomly add 1% anomalies to each concept and use the prequential method for evaluation. The characteristics of all datasets are summarized in table 7.1.

²³ Bifet et al., *New ensemble methods for evolving data streams*.

Figure 7.7: The sigmoid function used to introduce a smooth drift from one concept to another. We use a Bernoulli distribution, where p is the probability to sample an instance from concept 1 and $(1 - p)$ is the probability to sample from concept 2. The drift occurs at t_0 and w defines the duration during which both concepts are valid.

	# CONCEPTS	CONCEPT DRIFT TYPE	EVALUATION METHOD
SVHN	9	sudden	holdout
SATELLITE	3	sudden	holdout
SHUTTLE	2	sudden	holdout
TDDS	4	sudden	prequential
SDD	9	smooth	prequential

Table 7.1: Streaming datasets and their properties.

7.6.2 Evaluation

For experimental datasets used here, a sliding window of length 100 demonstrated to be an appropriate trade-off between drift adaptation and model accuracy. We therefore use this length for all algorithms and all datasets except γ -EXPOSE. A change of the window length affects W-EXPOSE, COD, STORM and HSTA in the same way. This is not unexpected as the window size determines the number of instances available to the algorithm. In each

experiment, the first 100 instances of a stream are used to configure the algorithm's parameters via cross-validation.

The predictive performance of all algorithms is relatively similar with an edge of the `EXPOSE` detectors. It can be observed that, as the stream changes from one digit to another, the accuracy suddenly drops which indicates that the current model is not valid anymore. After a short period of time, the model adapts, and the accuracy recovers. `EXPOSE` performs on average better than `COD`, `STORM` and `HSTA`. A possible interpretation of this result is the sound foundation in probability theory of our approach. The suboptimal performance of `HSTA` indicates that the random binary trees constructed by `HSTA` are not sufficiently fine-grained for this high-dimensional datasets. This interpretation is supported by the experiments with the low-dimensional Shuttle and Satellite data, where `HSTA` performs better.

	<code>w-EXPOSE</code>	<code>γ-EXPOSE</code>	<code>STORM</code>	<code>COD</code>	<code>HSTA</code>
SHUTTLE	0.88	0.88	0.75	0.74	<u>0.89</u>
SATELLITE	<u>0.89</u>	0.88	0.78	0.79	0.88
SVHN	0.71	<u>0.73</u>	0.68	0.68	0.66
TDDS	0.71	<u>0.71</u>	0.67	0.64	0.67
SDD	0.83	<u>0.85</u>	0.79	0.76	0.77
AVERAGE RANK	1.80	<u>1.70</u>	3.80	4.50	3.20

Table 7.2: Streaming anomaly detection performances (balanced accuracy).

THE average of all accuracies of the individual experiments can be found in table 7.2. Using the Nemenyi test for pairwise comparisons of *all* algorithms is too conservative and has little power confirmed by the critical difference diagram (Fig. 7.8). The only statistically significant difference at $p < 0.05$ is observed between `γ-EXPOSE` and `COD`.

Figure 7.8: Critical difference diagram of the stream anomaly detection performance. Algorithms which are significantly different (at $p < 0.05$) are *not* connected with a bar.

However, we can compare `w-EXPOSE` against all other window-based anomaly detectors using the Holm procedure. The ordered statistics for Holm's method and p values are listed in table 7.3. Here, we can reject the first and the second hypothesis since the corresponding p values are smaller than the adjusted α 's and conclude that `w-EXPOSE` performs significantly better than `COD` and `STORM`. However, the experimental data is not sufficient to confirm a difference between `w-EXPOSE` and `HSTA`.

	z	p	α
<code>COD</code>	3.31	0.001	0.017
<code>STORM</code>	2.45	0.014	0.025
<code>HSTA</code>	1.71	0.086	0.050

Table 7.3: The ordered statistics for Holm's method.

CHAPTER EIGHT

Concentration Inequalities

SYNOPSIS · In this chapter we develop probabilistic bounds for the empirical kernel embedding of distributions $\mu[\mathbb{P}_n]$. We study the behavior of $\mu[\mathbb{P}_n]$ in the setting of *sampling without replacement*. More precisely, we bound the error which is introduced if only a random subset of a much large dataset is used to estimate the kernel embedding. This can be necessary if the original dataset is simply too large to be processed or subsampling can reduce costs for various reasons. Serfling¹ provides concentration inequalities similar to Hoeffding’s famous bound,² but for sampling without replacement of real-valued random variables. Considering that the empirical kernel embedding resides in a separable Hilbert space we generalize Serfling’s inequality to such spaces. Furthermore, we show that our results also hold in Banach spaces which obey a certain smoothness property.

¹ Serfling, *Probability inequalities for the sum in sampling without replacement*.

² Hoeffding, *Probability inequalities for sums of bounded random variables*.

THE chapter content was partly published in

- M. Schneider (2016).

“Probability Inequalities for Kernel Embeddings in Sampling without Replacement”. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS 2016)*, pp. 66–74

8.1 Preliminaries

In the following we employ our standard statistical framework. Recall that given a realization $z = (x_1, \dots, x_n)$ from the samples space $(\mathcal{Z}, \mathcal{X}, \mathbb{P}_Z)$ the (empirical) **EXPose** anomaly detector is defined as

$$\begin{aligned} \eta(y) &= \langle \phi(y), \mu[\mathbb{P}_n] \rangle \\ &= \left\langle \phi(y), \frac{1}{n} \sum_{i=1}^n \phi(x_i) \right\rangle \end{aligned}$$

where $\mu[\mathbb{P}_n]$ is the embedding of the empirical measure after n draws. The law of large numbers implies that $\mu[\mathbb{P}_n]$ converges almost surely to $\mu[\mathbb{P}_X]$ for $n \rightarrow \infty$. The necessary requirement therefore is that $\mathbb{E}[\|\phi(X)\|] < \infty$ which follows from (A-3). We may ask how well $\mu[\mathbb{P}_n]$ approximates $\mu[\mathbb{P}_X]$ for any n . An answer to this question is given by the following *concentration inequality*.

(A-3) For kernel k and measure $\nu \in \mathcal{B}(\mathcal{X})$ it holds that $\int_{\mathcal{X}} \sqrt{k(x, x)} d\nu(x) < \infty$.

THEOREM 8.1. Let $\mathcal{H}$ be a separable RKHS such that all $\|\phi(x)\| \leq c$ almost surely (a.s.) for some constant $c \geq 0$. Then

$$\mathbb{P}(\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\| \geq \epsilon) \leq 2 \exp\left(-\frac{n\epsilon^2}{8c^2}\right)$$

for all $\epsilon > 0$.

This bounds the probability that the empirical kernel embedding $\mu[\mathbb{P}_n]$ differs more than ϵ from the actual embedding $\mu[\mathbb{P}_X]$ using n independent samples. This is (up to some constants) a Hilbert space version of Hoeffding's inequality. According to Roussas, concentration inequalities as the one above are of wide interest in various applications.

The problem of providing exponential bounds [...] is of paramount importance, both in Probability and Statistics. From a statistical viewpoint, such inequalities can be used, among other things, for the purpose of providing rates of convergence (both in the probability sense and almost surely) for estimates of various quantities.

— Roussas, *Exponential probability inequalities with some applications*

HOWEVER, there is a second family of inequalities which plays an important role in several applications. Consider a training set $(x_1, \dots, x_N)$ and assume that we can only process a random subset $(X_1, \dots, X_n)$ of size $n \leq N$. This can be required for various reasons *e.g.* the original dataset is simply too large to be processed or it is too expensive to collect all data. The central question is then “How much error does this subsampling introduce?” Obviously, $\mu[\mathbb{P}_n] \rightarrow \mu[\mathbb{P}_N]$ as $n \rightarrow N$, but we show that it is also possible to provide more precise probabilistic bounds in terms of a concentration inequality, *i.e.*

$$\mathbb{P}(\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_N]\| \geq \epsilon)$$

which is the probability that $\mu[\mathbb{P}_n]$ differs more than ϵ from $\mu[\mathbb{P}_N]$. The estimation of such quantities is well known in probability theory as *sampling without replacement* and formally defined as follows.

DEFINITION 8.2 (Sampling without Replacement). Let $(x_1, \dots, x_N)$ be a finite population of N elements. For any $n \in \mathbb{N}$ with $n \leq N$, the realization $(x_{I_1}, \dots, x_{I_n})$ of the random variables $(X_1, \dots, X_n)$ with

$$\mathbb{P}[(I_1, \dots, I_n) = (i_1, \dots, i_n)] = [N(N-1) \cdots (N-n+1)]^{-1}$$

is said to be *drawn without replacement* and $(X_1, \dots, X_n)$ is called *sample without replacement*.³

Sampling without replacement is itself subject of various application such as survey sampling,⁴ Markov chain Monte Carlo algorithms⁵ and computational learning theory⁶ to name a few.

³ Serfling, *Probability inequalities for the sum in sampling without replacement*, Section 1.

⁴ Kish, *Survey sampling*.

⁵ Bardenet et al., *Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach*.

⁶ Cannon et al., *Machine learning with data dependent hypothesis classes*.

8.1.1 Contributions

The main contributions of this chapter are two theorems which are concerned with probabilistic bounds in the setting of sampling without replacement. The first theorem provided a concentration inequality for the probability $\mathbb{P}(\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_N]\| \geq \epsilon)$ and is stated next.

THEOREM 8.3. Let $(X_1, \dots, X_n)$ be a random sample without replacement from the set $(x_1, \dots, x_N) \in \mathcal{X}^N$. Let k be a reproducing kernel with corresponding separable RKHS $\mathcal{H}$ such that all $\phi(x_1), \phi(x_2), \dots, \phi(x_N)$ are bounded almost surely by some constant $c \geq 0$. Then, for all $\epsilon > 0$

$$\mathbb{P}(\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_N]\| \geq \epsilon) \leq 2 \exp\left(-\frac{n\epsilon^2}{8c^2(1 - \frac{n-1}{N})}\right)$$

where

$$\mu[\mathbb{P}_n] = \frac{1}{n} \sum_{i=1}^n \phi(X_i) \quad \text{and} \quad \mu[\mathbb{P}_N] = \frac{1}{N} \sum_{i=1}^N \phi(x_i)$$

are empirical kernel mean maps for the measures $\mathbb{P}_n$ and $\mathbb{P}_N$, respectively.

The assumption that ϕ is essentially bounded is stronger than the assumption $\int_{\mathcal{X}} \sqrt{k(x, x)} d\mathbb{P}_X(x) < \infty$ (A-3) we used formerly, but necessary in what follows.

FURTHERMORE, we will show that theorem 8.3 is only a special case of a more general inequality for the sum $S_n = U_1 + \dots + U_n$ of independent random variables $U_1, \dots, U_n$ taking values in a separable Banach space.

THEOREM 8.4. Let $(\mathcal{B}, \|\cdot\|)$ be a $(2, \tau)$ -smooth⁷ separable Banach space and $(U_1, \dots, U_n)$ is drawn without replacement from $(u_1, \dots, u_N) \in \mathcal{B}^N$. We assume that all random variables U_i are essentially bounded, that is $\|U_i\| \leq h$ a.s. for some $h > 0$. Then for all $\epsilon > 0$

⁷ See Def. 8.8 for a characterization of $(2, \tau)$ -smooth.

$$\mathbb{P}\left(\max_{1 \leq k \leq n} \left\| \frac{N-n}{N-k} (S_k - k\bar{u}) \right\| \geq n\epsilon\right) \leq 2 \exp\left(-\frac{n\epsilon^2}{8\tau^2 h^2 (1 - \frac{n-1}{N})}\right),$$

where

$$\bar{u} = \frac{1}{N} \sum_{i=1}^N u_i \quad \text{and} \quad S_n = U_1 + \dots + U_n$$

are the mean of over all realizations and the sample sum, respectively.

THE next section reviews the concentration inequalities of Hoeffding and Serfling. Subsequently we introduce Pinelis theorem which plays an essential part in the proofs of theorem 8.3 and theorem 8.4.

8.2 Sampling Without Replacement

Historically, Hoeffding was among the first who derived bounds for the problem of sampling without replacement. His popular inequality⁸ provides an upper bound on the probability that the sum of random variables deviates from its expected value.

⁸ Hoeffding, *Probability inequalities for sums of bounded random variables*, Theorem 1.

THEOREM 8.5 (Hoeffding's Inequality). Let $Y_1, Y_2, \dots, Y_n$ be independent real-valued random variables with $a \leq Y_i \leq b$. Then

$$\mathbb{P}\left(\left|\frac{1}{n} \sum_{i=1}^n Y_i - \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n Y_i\right]\right| \geq \epsilon\right) \leq 2 \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$$

for any $\epsilon > 0$.

Hoeffding explicitly states that this inequality is also an upper bound for sampling without replacement from a finite set. The statement holds since

$$\mathbb{E} \left[f \left(\sum_{i=1}^n X_i \right) \right] \leq \mathbb{E} \left[f \left(\sum_{i=1}^n Y_i \right) \right]$$

given a continuous and convex function f , where $(X_1, X_2, \dots, X_n)$ is a sample *without* replacement and $(Y_1, Y_2, \dots, Y_n)$ is a sample *with* replacement.⁹

TAKING the finite size of the dataset into account, Serfling¹⁰ introduces a tighter bound for the sampling without replacement inequality. He replaces the factor n by $\frac{n}{(1 - (n-1)/N)}$ in the bound on the right-hand side of the equation in theorem 8.5 which leads to a significant improvement of the bound when n is large.

THEOREM 8.6 (Serfling's Inequality). Let $(X_1, X_2, \dots, X_n)$ be a random sample without replacement from the set $(x_1, \dots, x_N) \in \mathbb{R}^N$. Then

$$\mathbb{P} \left(\left| \frac{1}{n} \sum_{i=1}^n X_i - \frac{1}{N} \sum_{i=1}^N x_i \right| \geq \epsilon \right) \leq 2 \exp \left(- \frac{2n\epsilon^2}{(1 - \frac{n-1}{N})(b-a)^2} \right)$$

for all $\epsilon > 0$, where

$$a = \min_{1 \leq i \leq N} x_i \quad \text{and} \quad b = \max_{1 \leq i \leq N} x_i$$

are the minimum and maximum values in $(x_1, \dots, x_N)$ respectively.

WE SEE that Serfling already derived bounds in the case of sampling without replacement similar to the one we need to estimate the deviation of $\mu[\mathbb{P}_n]$ from $\mu[\mathbb{P}_N]$. However, his inequality holds for real-valued random variables. The random elements $U_i = \phi(X_i)$ used in the empirical kernel mean map¹¹ are elements of the reproducing kernel Hilbert space $\mathcal{H}$ of function which requires a more general theory of concentration inequalities.

THE remainder of this chapter is structured as follows. Even though $\mu[\mathbb{P}_n]$ is the empirical mean of Hilbert space valued random variables, we will derive more general bounds for Banach space valued martingales established in theorem 8.4 and then show that theorem 8.3 is only a special case thereof. In the next section we generalize Serfling's bound to random variables which take values in a separable Banach space instead of $\mathbb{R}$.

8.3 A Banach Space Inequality for Sampling without Replacement

For the deviation of the concentration inequality in theorem 8.4 we will exploit bounds for certain martingale structures suggested by Serfling. However, in contrast to Serfling we will not bound the moment generating function, but utilize Pinelis' theorem for Banach space valued martingales, which we state next together with some preliminaries.

⁹ Hoeffding, *Probability inequalities for sums of bounded random variables*, Theorem 4.

Figure 8.1: Comparison of probabilistic bounds for sampling without replacement of Hoeffding's and Serfling's inequality for $\epsilon = 0.01$, $b - a = 1$ and $N = 40\,000$.

¹⁰ Serfling, *Probability inequalities for the sum in sampling without replacement*.

¹¹ Recall that $\mu[\mathbb{P}_n] = \frac{1}{n} \sum_{i=1}^n \phi(X_i)$ and $\phi: \mathcal{X} \rightarrow \mathbb{R}$.

DEFINITION 8.7 (*Martingale*). The stochastic process $\{M_n\}$ is said to be a *martingale* with respect to the filtration $\{\mathcal{F}_n\}$ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ if

- M_n is integrable ($M_n \in \mathcal{L}^1$) and
- $\mathbb{E}[M_{n+1} \mid \mathcal{F}_n] = M_n$ a.s.

for all $n \in \mathbb{N}_0$.¹²

We assume that $M_0 = 0$ and that the martingale takes values in a separable Banach space $(\mathcal{B}, \|\cdot\|)$. The set of all such martingales is denoted by $\mathcal{M}(\mathcal{B})$. Pinelis' theorem which is introduced next makes a certain smoothness assumption on the Banach space which called (r, τ) -smooth.

DEFINITION 8.8 ((r, τ) -smooth Banach spaces). A Banach space $(\mathcal{B}, \|\cdot\|)$ is called (r, τ) -smooth for some $1 < r \leq 2$, if there exists a $\tau > 0$ such that

$$\|x + y\|^r + \|x - y\|^r \leq 2\|x\|^r + 2\tau^r\|y\|^r$$

for all $x, y \in \mathcal{B}$ and is simply called 2-smooth if it is $(2, \tau)$ -smooth for some $\tau > 0$.

The notion of 2-smooth Banach spaces is the analogue of the notion of a Banach space of type 2 in the case of martingale differences. For example any $\mathcal{L}^p$ space of $\mathbb{R}$ -valued functions is r -smooth for $r = \min(2, p)$. Then

$$\tau^2 = \begin{cases} 2 & \text{if } 1 \leq p < 2 \\ p - 1 & \text{if } p \geq 2 \end{cases}$$

provided the associated measure is σ -finite.¹³ We emphasize that any Hilbert space is $(2, 1)$ -smooth.

PINELIS proves an inequality for the probability that the martingale $\{M_n\}$ escapes a ball with radius ϵ (Fig. 8.2). The only requirement is that the martingale differences are bounded a.s. such that their countable sum is bounded a.s.

THEOREM 8.9 (*Pinelis' Inequality*). Suppose that $\{M_n\} \in \mathcal{M}(\mathcal{B})$, where $(\mathcal{B}, \|\cdot\|)$ is a $(2, \tau)$ -smooth separable Banach space and

$$\sum_{i=1}^{\infty} \text{ess sup} \|M_i - M_{i-1}\|^2 \leq b^2$$

for some $b > 0$. Then

$$\mathbb{P}(\sup \|M_n\| \geq \epsilon) \leq 2 \exp\left(-\frac{\epsilon^2}{2b^2\tau^2}\right)$$

for all $\epsilon > 0$.¹⁴

The challenge is to choose a martingale sequence such that Pinelis' theorem can be applied to the difference $\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_N]\|$ where b then yield the desired bound with a dependence on n .

¹² Kallenberg, *Foundations of modern probability*, Chapter 6.

¹³ Cuny, *A compact LIL for martingales in 2-smooth Banach spaces with applications*, Definition 2.2.

Figure 8.2: A martingale $\{M_n\}$ which escapes a ball with radius ϵ in the Banach space $(\mathcal{B}, \|\cdot\|)$.

¹⁴ Pinelis, *Optimum bounds for the distributions of martingales in Banach spaces*, Theorem 3.5, *An approach to inequalities for the distributions of infinite-dimensional martingales*, Theorem 3.

8.3.1 *A Serfling Type Inequality for Banach Spaces*

We start with the following stochastic process and show that it fulfills the martingale requirements and then bound its martingale differences. This is done in two consecutive steps.

LEMMA 8.10. Let $(\mathcal{B}, \|\cdot\|)$ be a $(2, \tau)$ -smooth separable Banach space and let $(U_1, U_2, \dots)$ be a sample without replacement from $(u_1, \dots, u_N) \in \mathcal{B}^N$. Assume that $\|u_i\| \leq h$ a.s. for all $1 \leq i \leq N$. Then the stochastic process $\{M_k\}$ defined by

$$M_k = \begin{cases} \frac{1}{N-k} \sum_{i=1}^k (U_i - \bar{u}) & \text{for } 1 \leq k < N \\ 0 & \text{else} \end{cases}$$

is a martingale and its differences are bounded by

$$\|M_k - M_{k-1}\| \leq \frac{2h}{N-k} \quad \text{a.s.}$$

for all $1 \leq k < N$.

PROOF OF LEMMA 8.10. Since U_k is uniformly drawn from the remaining $N - k + 1$ points, it holds that for $1 \leq k \leq N$

$$\begin{aligned} \mathbb{E}[U_k \mid U_1, \dots, U_{k-1}] &= \frac{\sum_{i=k}^N U_i}{N - k + 1} \\ &= \frac{N\bar{u} - \sum_{i=1}^{k-1} U_i}{N - k + 1} \\ &= \bar{u} - \frac{\sum_{i=1}^{k-1} U_i - (k-1)\bar{u}}{N - k + 1} \\ &= \bar{u} - M_{k-1}, \end{aligned} \tag{8.1}$$

where the expectation is taken with respect to the remaining elements $U_k, \dots, U_N$. By definition of the martingale in lemma 8.10 we have

$$\begin{aligned} M_k &= \frac{1}{N-k} \sum_{i=1}^k (U_i - \bar{u}) \\ &= \frac{1}{N-k} \sum_{i=1}^{k-1} (U_i - \bar{u}) + \frac{U_k - \bar{u}}{N-k} \\ &= \frac{N-k+1}{N-k} M_{k-1} + \frac{U_k - \bar{u}}{N-k} \end{aligned} \tag{8.2}$$

and hence

$$\begin{aligned} \mathbb{E}[M_k \mid M_1, \dots, M_{k-1}] &= \frac{N-k+1}{N-k} M_{k-1} + \frac{\mathbb{E}[U_k \mid M_1, \dots, M_{k-1}] - \bar{u}}{N-k} \\ &= \frac{N-k+1}{N-k} M_{k-1} + \frac{\bar{u} - M_{k-1} - \bar{u}}{N-k} \\ &= M_{k-1} \end{aligned}$$

where we used Eq. (8.1) in the second step. This shows that $\{M_k\}$ is a martingale. We can now bound the quantity $\|M_k - M_{k-1}\|$, but first

notice that

$$\begin{aligned}
\|U_k - \bar{u} + M_{k-1}\| &= \left\| U_k - \bar{u} + \frac{1}{N-k+1} \sum_{i=1}^{k-1} (U_i - \bar{u}) \right\| \\
&= \left\| U_k + \frac{1}{N-k+1} \sum_{i=1}^{k-1} U_i - \frac{1}{N-k+1} \sum_{i=1}^N u_i \right\| \\
&= \left\| U_k - \frac{1}{N-k+1} \sum_{i=k+1}^N u_i \right\| \\
&\leq 2h \quad \text{a.s.}
\end{aligned}$$

since U_k and $u_{k+1}, \dots, u_N$ are almost surely bounded by h . Equation (8.2) yields

$$M_k = M_{k-1} + \frac{U_k - \bar{u} + M_{k-1}}{N-k}$$

and hence it holds that

$$\begin{aligned}
\|M_k - M_{k-1}\| &= \frac{1}{N-k} \|U_k - \bar{u} + M_{k-1}\| \\
&\leq \frac{2h}{N-k}
\end{aligned}$$

for $1 \leq k < N$, which bounds the martingale differences. $\square$

WE follow a similar analysis as Serfling¹⁵ using the inequality

$$\sum_{r=k+1}^m \frac{1}{r^2} \leq \frac{m-k}{k(m+1)}, \quad 1 \leq k \leq m \in \mathbb{N}, \quad (8.3)$$

¹⁵ Serfling, *Probability inequalities for the sum in sampling without replacement*, Lemma 2.1.

and are now in a position to prove theorem 8.4.

PROOF OF THEOREM 8.4. Using the previous derivations we observe that for $1 \leq n < N$

$$\begin{aligned}
\frac{(N-n)^2}{4h^2} \sum_{k=1}^n \|M_k - M_{k-1}\|^2 &\leq \sum_{k=1}^n \frac{(N-n)^2}{(N-k)^2} \\
&= 1 + (N-n)^2 \sum_{k=N-n+1}^{N-1} k^{-2} \\
&\leq 1 + \frac{(N-n)^2(n-1)}{N(N-n)} \\
&= \frac{N + (N-n)(n-1)}{N} \\
&= n \left(1 - \frac{n-1}{N} \right)
\end{aligned}$$

where we used Eq. (8.3) in the second last step. Hence the bound on the martingale difference is

$$\sum_{k=1}^n \|M_k - M_{k-1}\|^2 \leq \frac{\alpha^2 n}{(N-n)^2}$$

with

$$\alpha^2 = 4h^2 \left(1 - \frac{n-1}{N}\right).$$

We now apply Pinelis' inequality (theorem 8.9) which yields

$$\begin{aligned} \mathbb{P}(\sup_k \|M_k\| \geq \lambda) &\leq 2 \exp\left(-\frac{\lambda^2}{2\tau^2 b^2}\right) \\ \mathbb{P}\left(\max_{1 \leq k \leq n} \left\| \frac{S_k - k\bar{u}}{N-k} \right\| \geq \lambda\right) &\leq 2 \exp\left(-\frac{\lambda^2 (N-n)^2}{2\tau^2 \alpha^2 n}\right) \end{aligned}$$

as we plug in the martingale $\{M_k\}$ and the bound $b^2 = \frac{\alpha^2 n}{(N-n)^2}$. Substituting $(N-n)\lambda = n\epsilon$ in the equation above yields

$$\begin{aligned} \mathbb{P}\left(\max_{1 \leq k \leq n} \left\| \frac{N-n}{N-k} (S_k - k\bar{u}) \right\| \geq n\epsilon\right) &\leq 2 \exp\left(-\frac{\epsilon^2}{2\tau^2 \alpha^2 n}\right) \\ &\leq 2 \exp\left(-\frac{n\epsilon^2}{8\tau^2 h^2 \left(1 - \frac{n-1}{N}\right)}\right), \end{aligned}$$

which concludes the proof. $\square$

We will now show that theorem 8.3 is a consequence of theorem 8.4 if an appropriate martingale structure is applied.

8.4 Kernel Embedding Concentration Inequalities

Using the results from the previous derivation we can now prove theorem 8.3.

PROOF. We first remind that if $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ is a Hilbert space, then it is always $(2, 1)$ -smooth. Let $U_1, U_2, \dots$ be independent random variables in the Hilbert space $\mathcal{H}$, bounded a.s. by $\|U_i\| \leq d$. From the elementary relation

$$\mathbb{P}\left(\left\| \sum_{i=1}^n (U_i - \bar{u}) \right\| \geq n\epsilon\right) \leq \mathbb{P}\left(\max_{1 \leq k \leq n} \left\| \frac{N-n}{N-k} \sum_{i=1}^k (U_i - \bar{u}) \right\| \geq n\epsilon\right)$$

it follows that

$$\mathbb{P}\left(\left\| \frac{1}{n} \sum_{i=1}^n U_i - \bar{u} \right\| \geq \epsilon\right) \leq 2 \exp\left(-\frac{n\epsilon^2}{8d^2 \left(1 - \frac{n-1}{N}\right)}\right)$$

by means of theorem 8.4. Substituting $U_i = \phi(X_i)$ and $u_i = \phi(u_i)$ yields

$$\mathbb{P}\left(\left\| \frac{1}{n} \sum_{i=1}^n \phi(X_i) - \frac{1}{N} \sum_{i=1}^N \phi(x_i) \right\| \geq \epsilon\right) \leq 2 \exp\left(-\frac{n\epsilon^2}{8d^2 \left(1 - \frac{n-1}{N}\right)}\right)$$

which concludes the proof for theorem 8.3. $\square$

FROM the previous discussion it is also easy to see that theorem 8.1 follows directly from Pinelis' theorem.

PROOF OF THEOREM 8.1. Obviously, $\phi(X_k) - \mu[\mathbb{P}_X]$ is a zero mean independent random variable for all $1 \leq k \leq n$ and a.s. bounded by $2d$. Consider the martingale difference sequence

$$M_k - M_{k-1} = \begin{cases} \phi(X_k) - \mu[\mathbb{P}_X] & \text{if } 1 \leq k \leq n \\ 0 & \text{otherwise} \end{cases}$$

with

$$\sum_{i=1}^{\infty} \text{ess sup} \|M_i - M_{i-1}\|^2 \leq 4d^2n$$

and we can apply theorem 8.9. $\square$

8.5 *An Alternative Approach for a Serfling-Type Inequality*

The use of Pinelis' theorem yields a simple and elegant proof of theorem 8.1 which even generalizes to concentration inequalities in Banach spaces. Alternatively we could approach Serfling's inequality directly. However, using elementary methods, the bound obtained that way would be too coarse. To illustrate the problem, consider a generalization of Serfling's bound to random variables in $\mathbb{R}^d$. Set $X_i = (X_i^1, X_i^2, \dots, X_i^d)$ and $x_i = (x_i^1, x_i^2, \dots, x_i^d)$ and assume that

$$\begin{aligned} a &= \min_{1 \leq i \leq N} \{x_i^1, x_i^2, \dots, x_i^d\} \quad \text{and} \\ b &= \max_{1 \leq i \leq N} \{x_i^1, x_i^2, \dots, x_i^d\} \end{aligned}$$

Then

$$\mathbb{P}\left(\left\|\frac{1}{n} \sum_{i=1}^n X_i - \frac{1}{N} \sum_{i=1}^N x_i\right\| \geq \epsilon\right) \leq 2d \exp\left(-\frac{2n\epsilon^2}{(1 - \frac{n-1}{N})d^2(b-a)^2}\right)$$

where we applied Serfling's inequality component wise. A comparison with theorem 8.1 reveals that this bound is of completely wrong order since it explicitly depends on the dimension d and therefore becomes useless in infinite dimensions.

However, Kallenberg and Sztencel¹⁶ demonstrate that it is sufficient to provide an inequality for a martingale in $\mathbb{R}^2$ which then generalizes to Hilbert spaces.

¹⁶ Kallenberg and Sztencel, *Some dimension-free features of vector-valued martingales*, Theorem 3.1.

THEOREM 8.11 (*Kallenberg & Sztencel*). Let $\{M_t\}$ be a martingale in $\mathbb{R}^d$ or in a Hilbert space. Then there exists some martingale $\{N_t\}$ in $\mathbb{R}^2$, defined on a suitable standard extension of the original filtered probability space, such that $\|M_t\| = \|N_t\|$ and $\|M_t - M_{t-1}\| = \|N_t - N_{t-1}\|$ almost surely.

It is therefore sufficient to provide a Serfling-type inequality for $\mathbb{R}^2$ such as

$$\mathbb{P}\left(\left\|\frac{1}{n} \sum_{i=1}^n X_i - \frac{1}{N} \sum_{i=1}^N x_i\right\| \geq \epsilon\right) \leq 4 \exp\left(-\frac{2n\epsilon^2}{(1 - \frac{n-1}{N})4(b-a)^2}\right)$$

which is closer to theorem 8.1.

8.6 *Related Work*

This is the first time a bound for the empirical kernel embedding in the case of sampling without replacement for is shown. However, other authors also provide inequalities for the convergence rate of the empirical kernel embedding *i.e.* a probabilistic bound for the error $\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\|$ which we bound by

$$\mathbb{P}(\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\| \geq \epsilon) \leq 2 \exp\left(-\frac{n\epsilon^2}{8c^2}\right)$$

in theorem 8.1. In contrast to the martingale approach presented here, most related derivations are based on McDiarmid's concentration theorem.¹⁷

¹⁷ McDiarmid, *On the method of bounded differences*, Lemma 1.2.

THEOREM 8.12 (McDiarmid). Let $X_1, X_2, \dots, X_n$ be independent random variables taking values in a set A_i for each $1 \leq i \leq n$. Suppose that the (measurable) function $f: \times_{i=1}^n A_i \rightarrow \mathbb{R}$ satisfies

$$|f(u) - f(v)| \leq b_i$$

whenever the vectors u and v differ only in the i th coordinate. Define $Y = f(X_1, X_2, \dots, X_n)$, then for any $\epsilon > 0$,

$$\mathbb{P}(|Y - \mathbb{E}[Y]| \geq \epsilon) \leq 2 \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n b_i^2}\right).$$

The theorem allows us to upper bound the variation of a function f based on the maximal deviation in each coordinate. The use of

$$f(X_1, X_2, \dots, X_n) = \frac{1}{n} \sum_{i=1}^n X_i$$

yields Hoeffding's inequality as a simple special case of McDiarmid's inequality.

To bound the empirical kernel embedding, Shawe-Taylor et al.¹⁸ suggest to apply McDiarmid's inequality to

¹⁸ Shawe-Taylor et al., *Kernel methods for pattern analysis*, Chapter 4.

$$f(X_1, X_2, \dots, X_n) = \|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\|$$

which then provides the bound

$$\mathbb{P}(\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\| - \mathbb{E}[\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\|] \geq \epsilon) \leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n b_i^2}\right)$$

with $b_i = 2c/n$ whenever $\sup_{x \in \mathcal{X}} \sqrt{k(x, x)} \leq c$. It then remains to estimate $\mathbb{E}[\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\|]$. Shawe-Taylor et al. provide the suboptimal concentration inequality

$$\mathbb{P}(\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\| \geq \epsilon) \leq 2 \exp\left(-\frac{(\epsilon\sqrt{n} - 2c)^2}{2c^2}\right)$$

using the bound $\mathbb{E}[\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\|] \leq 2c/\sqrt{n}$. A tighter probabilistic inequality can be derived in terms of the *Rademacher average*. It is defined in terms of the following sequence which can be seen as repeated coin flips.

DEFINITION 8.13 (*Rademacher sequence*). Let $(\Theta, \mathcal{C}, \nu)$ be a probability space and $\sigma_i: \Theta \rightarrow \{-1, +1\}$ be independent random variables with measure $\nu(\sigma_i = 1) = \nu(\sigma_i = -1) = \frac{1}{2}$ for all $i = 1, \dots, n$. Then $\sigma_1, \dots, \sigma_n$ is called a *Rademacher sequence* with respect to ν .¹⁹

¹⁹ Steinwart and Christmann, *Support vector machines*, Definition 7.8.

DEFINITION 8.14 (*Rademacher Average*). Let $\mathcal{G}$ be a class of function mapping from $\mathcal{X}$ to $\mathbb{R}$ and $X_1, X_2, \dots, X_n$ be independent random variables distributed according to $\mathbb{P}_X$. The (empirical) *Rademacher average* of $\mathcal{G}$ is the random variable

$$\hat{\mathcal{R}}_n(\mathcal{G}) = \int \sup_{g \in \mathcal{G}} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i g(X_i) \right| d\nu$$

where $\sigma_1, \dots, \sigma_n$ is a Rademacher sequence with respect to some distribution ν . Its expectation $\mathcal{R}_n(\mathcal{G}) = \mathbb{E}[\hat{\mathcal{R}}_n(\mathcal{G})]$ is sometimes called *Rademacher complexity* of $\mathcal{G}$.²⁰

²⁰ Bartlett et al., *Rademacher and Gaussian Complexities: Risk Bounds and Structural Results*, Definition 2.

The Rademacher average can now be used to bound the empirical kernel embedding as shown by Smola et al.²¹

²¹ Smola et al., *A Hilbert space embedding for distributions*, Theorem 2.

THEOREM 8.15. Assume that $\|f\|_\infty \leq b$ for all $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}} \leq 1$. Then

$$\mathbb{P}(\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\| \geq \epsilon) \leq \exp\left(-\frac{n(\epsilon - 2\mathcal{R}_n(\mathcal{H}))^2}{b^2}\right)$$

for all $\epsilon > 0$.

In this case it remains to bound the Rademacher average. “[The inequality] tells us that we do not need to have access to actual distributions in order to compute $\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_X]\|$ approximately — as long as $\mathcal{R}_n(\mathcal{H}) = \mathcal{O}(n^{-\frac{1}{2}})$, a finite sample from the distributions will yield error of $\mathcal{O}(n^{-\frac{1}{2}})$.”²²

²² *Ibid.*

8.7 Empirical Evaluation

In this section we empirically evaluate the deviation between the kernel embeddings $\mu[\mathbb{P}_n]$ and $\mu[\mathbb{P}_N]$ as $n \rightarrow N$. We can calculate the norm difference $\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_N]\|$ exactly since

$$\begin{aligned} \|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_N]\|^2 &= \langle \mu[\mathbb{P}_n], \mu[\mathbb{P}_n] \rangle - 2\langle \mu[\mathbb{P}_n], \mu[\mathbb{P}_N] \rangle + \langle \mu[\mathbb{P}_N], \mu[\mathbb{P}_N] \rangle \\ &= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n k(X_i, X_j) + \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N k(x_i, x_j) \\ &\quad - \frac{2}{nN} \sum_{i=1}^n \sum_{j=1}^N k(X_i, x_j) \end{aligned}$$

holds and we never have to represent ϕ explicitly. As evaluation datasets we choose $N = 1000$ random instances of the MNIST,²³ USPS and ImageNet²⁴ benchmarks. The error $\|\mu[\mathbb{P}_n] - \mu[\mathbb{P}_N]\|$ for $1 \leq n \leq N$ is evaluated using

²³ LeCun et al., *Gradient-based learning applied to document recognition*.

²⁴ Russakovsky et al., *ImageNet Large Scale Visual Recognition Challenge*.

the scaled squared exponential and Matérn kernel functions which are defined as

$$\begin{aligned} k(x, y) &= \alpha \exp\left(-\frac{\|x - y\|_2^2}{2\sigma^2}\right) \\ k(x, y) &= \alpha \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|x - y\|_2}{\sigma}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}\|x - y\|_2}{\sigma}\right) \end{aligned}$$

respectively (§4.5). In the experiments we set the kernel parameters according to table 8.1 which yields kernel values with a sufficient variance to evaluate the probabilistic bounds. For both kernels, we vary the scaling factor α such that $\|\phi(x)\| \leq c$, for $c = 1, 3, 5$ to gain insight how the convergence rate is influenced by the feature function. A variation of the bandwidth σ^2 does not have any influence on the bound c and is hence not varied.

The results of ten independent experiments are illustrated in Fig. 8.3, where we report the mean $\pm 2 \times$ variance. As reflected in theorem 8.3, the convergence rate becomes slower as $\|\phi(x)\|$ grows and we also observe an increased variance. The influence of this bound dominates over the influence of the kernel characteristics and the dataset properties. This is reflected in the shape of the curves which are very similar on all six experimental settings.

IN CONCLUSION, this chapter analyzed the behavior of the empirical kernel mean map in terms of its convergence rates in the setting of *sampling without replacement*. The derived concentration inequalities are a generalization of Serfling’s work²⁵ (which has numerous applications in probability, statistics and machine learning) to random variables in separable Banach spaces. Our inequalities do not use any knowledge about the underlying distribution other than the requirement of bounded martingale differences and are therefore applicable to a wide class of problems.

PARAMETERS	
MNIST	$\nu = \frac{3}{2}, \sigma^2 = 10$
USPS	$\nu = \frac{3}{2}, \sigma^2 = 10$
IMAGENET	$\nu = \frac{3}{2}, \sigma^2 = 100$

Table 8.1: Parameter sets used in the experimental evaluation.

²⁵ Serfling, *Probability inequalities for the sum in sampling without replacement*.

Figure 8.3: The deviation between $\mu[\mathbb{P}_n]$ and $\mu[\mathbb{P}_N]$ as $n \rightarrow N$ for different datasets and different kernels. Each kernel is scaled such that $\|\phi(x)\| \leq d$, for $d = 1, 3, 5$. We repeated each experiment several times and plot the mean (solid line) and 95% CIs (shaded region). Each row shows a different dataset (MNIST, USPS, ImageNet) and each column a different kernel (Matérn and squared exponential).

Distance-Based Feature Maps

SYNOPSIS · Chapter 5 reviewed random kitchen sinks (§5.5.1) and Nyström methods (§5.5.3) as two common approximations which allow us to represent a kernel feature map explicitly as a real-valued vector. The contribution of this chapter is an approximate feature map for kernel functions which are based on a distance metric and is called *distance-based feature map*.

THE approximate feature maps introduced in this chapter were partly published in

- M. Schneider, W. Ertel, and G. Palm (2015b).
“Kernel Feature Maps from Arbitrary Distance Metrics”. In: *KI 2015: Advances in Artificial Intelligence*. Ed. by S. Hölldobler et al. Springer International Publishing, pp. 137–150

9.1 Related Work

Recall that the idea behind approximate feature maps is to find a function $\hat{\phi}$ for a given kernel k such that

$$k(x, y) \approx \langle \hat{\phi}(x), \hat{\phi}(y) \rangle$$

and $\hat{\phi}: \mathcal{X} \rightarrow \mathbb{R}^r$ for some $r \in \mathbb{N}$. Random kitchen sinks and Nyström’s technique are the most common feature map approximations and had been reviewed in chapter 5. The success of these methods gained attention lead to further development in several directions. Examples include an extension of the random kitchen sink approach generalized to the class of non-translation-invariant histogram-based kernels by Li et al.¹ *i.e.* the exponentiated χ^2 kernel²

$$k(x, y) = \exp\left(-\sum_{i=1}^d \frac{(x_i - y_i)^2}{x_i + y_i}\right)$$

on $\mathbb{R}^d$. Likewise, Vedaldi et al. derive approximate feature maps for additive homogeneous kernels of the form

$$k(x, y) = \sum_{i=1}^m k_i(x_i, y_i),$$

where $k_1 \dots, k_m$ are themselves positive definite kernel functions. Kar et al.³ are presenting an approximation for positive definite dot product kernels which are of the form $k(x, y) = f(\langle x, y \rangle)$ for some real-valued function $f: \mathbb{R} \rightarrow \mathbb{R}$.

¹ Li et al., *Random Fourier approximations for skewed multiplicative histogram kernels*.

² Chapelle et al., *Support vector machines for histogram-based image classification*.

³ Kar et al., *Random feature maps for dot product kernels*.

MULTIPOLE EXPANSIONS can serve as an alternative for problem sets with a relatively low dimensionality as argued by Viet Le et al.

Fast multipole expansions [...] offer one avenue for efficient function expansions whenever the dimensionality of the underlying space is relatively modest. However, for high dimensions they become computationally intractable in terms of space partitioning, due to the curse of dimensionality. Moreover, they are typically tuned for localized basis functions, specifically the Gaussian RBF kernel.⁴

⁴ Viet Le et al., *Fastfood: Approximate Kernel Expansions in Loglinear Time*.

We refer to the literature and the references therein for a detailed description of each technique reviewed here.

9.2 Proposed Approach

The new approach introduced in this chapter is based on Schoenberg's theorem about conditionally negative definite functions.

DEFINITION 9.1 (Conditionally Negative Definite). Let $\mathcal{X}$ be a topological space. A non-negative function $g: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be *conditionally negative definite* if

$$\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j g(x_i, x_j) \leq 0$$

for any $n \geq 1$ and all choices $\alpha_1, \dots, \alpha_n \in \mathbb{R}$, $x_1, \dots, x_n \in \mathcal{X}$ such that $\sum_{i=1}^n \alpha_i = 0$.

The following theorem is due to Schoenberg⁵ and provides a connection between conditionally negative definite functions and metrics.

⁵ Schoenberg, *Metric spaces and positive definite functions*.

THEOREM 9.2 (Schoenberg). Let $g: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a symmetric function on a measurable space $(\mathcal{X}, \mathcal{X})$. Then g is conditionally negative definite if and only if $\tau = \sqrt{g}$ is a metric and there is an isometry which embeds the metric space $(\mathcal{X}, \tau)$ into a Hilbert space. Moreover, the function

$$k(x, y) = \exp(-\sigma g(x, y))$$

is positive definite for all $\sigma > 0$.⁶

⁶ Rao, *Convexity properties of entropy functions and analysis of diversity*, Theorem 4.1.

We see that under certain assumptions we can create a reproducing kernel from a metric. Our goal is now to create approximate feature maps for this type of kernel functions. Notice that if the metric is induced by the Euclidean norm, the squared exponential kernel (§4.5) is recovered. Assume we can find a mapping $\varphi: \mathcal{X} \rightarrow \mathbb{R}^r$ with $r \in \mathbb{N}$ such that $\tau(x, y)^2 \approx \|\varphi(x) - \varphi(y)\|_2^2$, then we can create an approximate feature map

$$\langle \hat{\phi}(x), \hat{\phi}(y) \rangle \approx \exp(-\sigma \tau(x, y)^2) \approx \exp(-\sigma \|\varphi(x) - \varphi(y)\|_2^2)$$

using the random kitchen sink approach. However, the challenge is to design the projection φ in a computationally and memory-efficient way. This disqualifies multidimensional scaling⁷ since it requires the calculation of all the eigenpairs of the distance matrix which takes cubic time.

⁷ Torgerson, *Multidimensional scaling*.

THE *FastMap*⁸ algorithm is especially designed to overcome the computational limitations of multidimensional scaling. It was originally proposed as an approach to map points onto a low-dimensional manifold such that dissimilarities are approximately preserved. FastMap creates an r -dimensional embedding as it projects an object $x \in \mathcal{X}$ onto a line defined by two *pivot points* $p, q \in \mathcal{X}$. This projection can be derived from the cosine law and is given by

$$L_{p,q}(x) = \frac{\tau(x,p)^2 + \tau(p,q)^2 - \tau(x,q)^2}{2\tau(p,q)}$$

as illustrated in Fig. 9.1. The original proposed algorithm iteratively projects all points in a set onto r different lines, where the pivot points are chosen in a way such that p and q are the most dissimilar objects: First q is randomly selected from the set of all data points. The point that is farthest apart from q is selected as p . In the last step q is updated to be point farthest apart from p . Thus, the complexity of FastMap is $\mathcal{O}(rn)$ while requiring $\mathcal{O}(r)$ storage for the pivots. We propose an approach based on the FastMap embedding with some modifications. Notice that for large datasets it is not feasible to iterate r times over all n data points to find the pivot elements. We therefore suggest to sample r pivot pairs from the training set $z = (x_1, \dots, x_n)$, *i.e.*

$$s = \{(p_1, q_1), \dots, (p_r, q_r)\} \subset z \times z \quad \text{with } p_i \neq q_i$$

and use these to calculate the r line projections which reduces the computational complexity to $\mathcal{O}(r)$. Based on the FastMap projections we create the mapping

$$\begin{aligned} \varphi: \mathcal{X} &\rightarrow \mathbb{R}^r \\ \varphi(x) &= (L_{p_1, q_1}(x), \dots, L_{p_r, q_r}(x)) \end{aligned}$$

which we will refer to as the *projection map*. The projection map takes an element $x \in \mathcal{X}$ and applies all r line projections with the pivot elements from s (Fig. 9.2). Since FastMap attempts to preserve dissimilarities, we know that $\|\varphi(x) - \varphi(y)\|_2$ is approximately proportional to $\tau(x, y)$. We are now in the position to apply the random kitchen sink approximation to create explicit feature maps $\hat{\phi}$ as

$$\langle \hat{\phi}(x), \hat{\phi}(y) \rangle \approx \exp(-\sigma \tau(x, y)^2).$$

The approximation accuracy depends on the number of line projections r which is independent of data dimension and the sample size which makes it applicable to large-scale applications. However, the computational complexity of our proposed approach depends on the computational complexity of the distance function through $L_{p,q}$. If the distance function cannot be evaluated efficiently, the approximation will be accordingly slower.

9.3 Experimental Evaluation

In the empirical evaluation we use three different distance functions on two different datasets to create approximate feature maps using the proposed distance-based feature map approach. The number of kernel expansions for

⁸ Faloutsos et al., *FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*.

Figure 9.1: The pseudo line projection of x onto the line defined by the pivots p and q .

Figure 9.2: Sketch of the map φ . The points x_i are projected onto the lines spanned by (p_1, q_1) and (p_2, q_2) . The position on these lines (calculated with $L_{p_1, q_1}(x)$ and $L_{p_2, q_2}(x)$) define their coordinates in the new space on the right.

the random kitchen sink approximation is set to 10 000 in all experiments. The number of random projections (pivot pairs) is varied to evaluate its effect on the approximation quality and the runtime. The first experiment is conducted on the discrete Fréchet distance as described in the following.

THE *Fréchet distance* can be used as a distance measure between two curves in a metric space. A discrete variation of this measure uses in the following experiment can be found in the work of Eiter et al.⁹ An intuitive description of the Fréchet distance is the minimal distance required to connect the points on two curves. It is often illustrated as a dog on one curve and its handler on another curve. Both are connected through a leash and can only go forward, but vary their speed. The Fréchet distance is the minimal length of any leash necessary such that both the dog and its handler can move the curves from the starting points to the endpoints.

We build an explicit feature map for a kernel-based on the discrete Fréchet distance and evaluate it empirically on the *hurricane dataset*. This dataset contains 7057 records of hurricane tracks collected by the National Oceanic and Atmospheric Administration (NOAA). We use only the longitude and latitude coordinates which describe the track of each hurricane (Fig. 9.3). As the tracks are not necessarily of the same length, it is impossible to apply, for example the Euclidean distance.

⁹ Eiter et al., *Computing discrete Fréchet distance*.

Figure 9.3: Random samples from the hurricane dataset.

Figure 9.4: Distance-based feature maps using the *Fréchet distance*. Left: Dependence between the mean squared error (MSE) and the number of random projections. Middle: Runtime dependence on the number of random projections. Right: Scatter plot between the approximation and the true kernel value.

The results of this experiment are illustrated in Fig. 9.4. The plot on the left shows the mean squared error (MSE) between $\langle \hat{\phi}(x), \hat{\phi}(y) \rangle$ and $\exp(-\sigma\tau(x, y))$, depending on the number of random projections. We report the result of each of the 10 repetitions and indicate their mean by the solid line. The approximation error becomes smaller as more random projections (pivots) are used. Moreover, we see that, since the pivot pairs are chosen at random, there is more variance in the approximation error if fewer pairs are used. The second plot in the middle shows the time needed to calculate the kernel value. This confirms the linear dependence of the runtime on the number of pivot pairs as evident from the definition of $L_{p,q}$. The difference between the approximation $\langle \hat{\phi}(x), \hat{\phi}(y) \rangle$ and the target kernel value $\exp(-\sigma\tau(x, y))$ is illustrated in the scatter plot on the right where we used 1000 random samples from the hurricane dataset as inputs for these functions.

THE *Hellinger distance* is a measure for the similarity between two distributions. It is closely related to the *Bhattacharyya distance* which we review in the next experiment. We look at the Hellinger distance first, since unlike the Bhattacharyya distance, it obeys the triangle inequality and is a proper metric. For two probability distributions $\mathbb{P}$ and $\mathbb{Q}$ the Bhattacharyya coefficient β is defined as

$$\beta(\mathbb{P}, \mathbb{Q}) = \int \sqrt{\mathbb{P}(x)\mathbb{Q}(x)} dx$$

for continuous probability distributions and

$$\beta(\mathbb{P}, \mathbb{Q}) = \sum_x \sqrt{\mathbb{P}(x)\mathbb{Q}(x)}$$

if $\mathbb{P}$ and $\mathbb{Q}$ are discrete. The Hellinger distance is then obtained as

$$\tau(\mathbb{P}, \mathbb{Q}) = \sqrt{1 - \beta(\mathbb{P}, \mathbb{Q})}$$

from the Bhattacharyya coefficient.

We apply this distance measure on the *Eight Scene Categories* dataset¹⁰ which contains 8 outdoor scene categories: *coast*, *mountain*, *forest*, *open country*, *street*, *inside city*, *tall buildings* and *highways* (Fig. 9.5). There are a total of 2600 RGB color images with 256×256 pixels. Image features are created from color histograms and local binary pattern histograms¹¹ which are then used as arguments for the Hellinger distance.

Figure 9.5: Random samples from the Eight scene categories dataset.

¹⁰ Ojala et al., *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*.

¹¹ *Ibid.*

Figure 9.6: Distance-based feature maps using *Hellinger's distance*. See Fig. 9.4 for a detailed description.

The experimental results are shown in Fig. 9.6. Even for only a few pivots the mean squared error is relatively low (an order of magnitude lower than in the experiment which evaluates the Fréchet distance). The computational time is also much lower since the Hellinger distance is fast to compute. The low mean squared error is also reflected in the scatter plot where all points are very close to on the diagonal.

THE *Bhattacharyya distance* is an example of a non-metric distance measure and hence we cannot expect to obtain a good feature map approximation using distance-based feature maps for the following reasons: The input space $(\mathcal{X}, \tau)$ is required to be a metric space by Schoenberg's theorem and additionally, the FastMap pseudo line projections are only theoretically

justified if this assumption holds. Nevertheless, in this experiment we show the effects if this requirement is violated and $(\mathcal{X}, \tau)$ is not a metric space.

The Bhattacharyya distance is used to measure the similarity of two probability distributions and is closely related to Hellinger's distance. For two probability distributions $\mathbb{P}$ and $\mathbb{Q}$ the Bhattacharyya distance is defined as

$$\tau(\mathbb{P}, \mathbb{Q}) = -\ln(\beta(\mathbb{P}, \mathbb{Q}))$$

where β is the Bhattacharyya coefficient as defined before. Similar to the previous experiment we use color histograms and local binary pattern histograms as image features on the Eight Scene Categories dataset.

The results are illustrated in Fig. 9.7. In contrast to Hellinger distance we observe a very high mean squared error which does not improve (unlike in the other experiments) with additional random projections. The high variance indicate that the choice of pivot pairs has a significant influence on the accuracy. We interpret this as an effect of the missing triangle inequality which invalidates the proposed pseudo line projections. This is also reflected in the scatter plot where the data is spread from the diagonal. Hence, the proposed distance-based feature maps should only be used of metric input spaces.

Figure 9.7: Distance-based feature maps using the *Bhattacharyya distance*. See Fig. 9.4 for a detailed description.

Part III

EPILOGUE

CHAPTER TEN

Conclusion

SYNOPSIS · This final chapter discusses the main conclusions and contributions of the thesis and gives directions for future work.

10.1 Contributions

The literature review discussed several successful anomaly detection techniques and identified their weaknesses when applied to large-scale datasets. In essence, most of these algorithms can handle a large number instances *or* high-dimensional data, but seldom both without loss in their predictive performance.

THE core of this dissertation is a new anomaly detection algorithm called **EXPOSE** which is especially designed to approach large-scale problem sets. The use of a kernel function enables us to apply **EXPOSE** to a wide range of different input domains where the kernel acts as a similarity measure. The assumption made by **EXPOSE** is that normal instances are similar to each other while anomalies differ from most observations of the inputs space distribution. We avoid the difficulties that arise when directly manipulating high-dimensional probability measures by means of a *kernel embedding of distributions*. This embedding maps the measure into a reproducing kernel Hilbert space and, as a result, transfers the anomaly detection problem into this new space.

A significant part of chapter 5 is dedicated to show that the **EXPOSE** anomaly detector can be formulated as an inner product in such a Hilbert space which permits **EXPOSE** to make predictions in *constant* time. Furthermore, we showed that the **EXPOSE** model can be estimated from a set of samples whenever the input space distribution is unknown, which is the default in anomaly detection. This estimate takes *linear* time and requires *constant* memory. Notably, the sample estimator takes a form which can be processed in parallel and on distributed environments without additional computational costs. The proposed approach is, despite these favorable properties, relatively simple and requires no parameters besides the kernel function. The experimental evaluation revealed that, while **EXPOSE** can compete with established anomaly detection algorithms, it is several orders of magnitude faster on large-scale, high-dimensional datasets than approaches with the same predictive performance. This result can mainly be contributed to three properties: First, its simplicity makes the proposed approach *configurable* which means that more computational resources can be dedicated to find an appropriate kernel function and its parameters.

Second, the success of kernel-based techniques in the last decades indicate that a Hilbert space feature map yields an adequate representation which is powerful enough for many machine learning problems, especially in high dimensions. Third, with roots in probability, `EXPose` takes the benefits of statistical approaches without the need to choose a parametric representation of the input space distribution and hence avoids the problems which accompany direct density estimations.

`WE` showed that `EXPose`'s empirical estimator causes the computational complexity to grow linearly with the sample size. This seems unavoidable if the information of all observations is taken into account to build a good model. Chapter 6 gives an affirmative answers the question “*Is it possible to become completely independent of the dataset size?*” if a high accurate estimation of `EXPose`'s model does not improve its predictive performance. We employ stochastic convex optimization theory to provide theoretical guarantees for the new algorithm with can determine the number of necessary optimization iterations for a user defined approximation accuracy *independent* of the sample size. The accompanied empirical findings indicate that our initial assumption holds for the evaluated datasets since the approximate solution achieves the same predictive performance as the original `EXPose` anomaly detector in a fraction of the time. In summary, neither learning nor prediction nor the memory consumption of `EXPose` is constrained by the dataset size.

`THE` core of `EXPose` is further developed to *online* and *streaming anomaly detection* algorithms. We argued that learning on data streams differ from incremental learning as the former requires the ability to deal with concept drift. We proposed several modifications to `EXPose` ranging from simple windowing and forgetting mechanisms to more sophisticated approaches based on temporal kernel functions which directly incorporate the concept drift adaptation into the kernel. Generally, there is no best generic solution since, depending on the drift type, some approaches may work better than others. It is therefore advantageous to have different options available. The empirical evaluation confirmed that `EXPose` can also compete with established anomaly detection algorithms on data streams under concept drift with different characteristics. Although these results are promising we recommend to combine the presented techniques with a concept drift detector to perform an (informed) model adaptations whenever necessary.

A further analysis revealed the weakness that `EXPose` is vulnerable to *adversarial attacks*. Although this is not problematic in the incremental learning model as an attacker would need to induce an exponentially large number of attack points, the same does not hold for concept drift adaptation based on a sliding window. In this case the number of necessary attack points grows linearly with the window size. It is therefore necessary to consider other countermeasures as a protection against such attacks.

`ALL` contributions of this thesis are accompanied by theoretical analyses. For this reason, chapter 8 contributes two special probabilistic bounds in form of concentration inequalities in the setting of *sampling without replacement*. The first inequality bounds the error made by the empirical

kernel mean map on random subsets of the data. The second bound is a generalization of Serfling’s inequality to sums of random variables in separable Banach spaces. This is a very general probabilistic bound and we expect it to be of much wider interest than the application in this work.

BESIDES THESE CONTRIBUTIONS much remains to be done. The choice of kernel is critical for the predictive performance of `EXPPOSE` as for kernel-based techniques in general. Although the squared exponential kernel is a viable default considering that it is characteristic and universal, there is still no automated way to select the *best* kernel function for a specific application.

The success of `LOF` indicates that input spaces with varying density regions exist and that anomaly detection should be performed *localized* in a neighborhood of the query point. It is unclear if `EXPPOSE` can be further improved by examining a similar approach. Moreover, our theoretical understanding of densities in high-dimensional spaces is still limited and we need more insight on why kernel methods work in these situations.

Most approaches can *detect* anomalous observations, but are not able to explain *why* these instances were classified as anomalies. This information would certainly be beneficial from a theoretical and practical point of view.

Online and streaming anomaly detection would benefit from a *regret decision analysis*. This could provide more guidance on which concept drift adaptation technique is more appropriate for a certain problem class and may lead to new algorithms. Additionally, we would be able to contribute regret bounds under adversarial attacks.

The concentration inequalities derived in chapter 8 require an *essentially bounded* kernel. It would be more consistent with other contributions of this dissertation if this assumption could be weakened to a kernel which is bounded *in expectation*. Additionally, we are optimistic that a Bernstein concentration bound (a probabilistic inequality which takes the variance of the random variable into account) can be shown to hold in separable Banach spaces as well.

IN THE BEGINNING we argued that a good anomaly detector should aim for *scalability*, *robustness* and *configurability*. Taken together, the contributions of this thesis showed that `EXPPOSE` achieves these aims and makes a significant step towards a general-purpose anomaly detector which is both fast *and* powerful.

Part IV

APPENDIX

CHAPTER A

Probability Theory

The history demonstrates that the notions of randomness and probabilities are difficult — very difficult — to apprehend, and that the notions are fairly new and alien to humans. (I don't think that mankind ever was meant to apprehend randomness and probabilities.) Second, the history and a wealth of examples [...] demonstrate that our intuition about probability is poor and often takes the wrong track. Thus, when you evaluate probabilities trust your computations and doubt your intuition. For this reason the interpretation of probabilities presents a difficult and fundamental problem, and the past — from the emerge of probabilities in 1550 until today — contains numerous examples of misinterpretations and miscalculations.

— Hoffmann-Jørgensen, *Probability with a view toward statistics*

DEFINITION A.1 (Measurable Space). A measurable space is a tuple $(\Omega, \mathcal{F})$, where Ω is a nonempty set and $\mathcal{F}$ is a σ -algebra of its subsets.

DEFINITION A.2 (Borel σ -algebra). Let (X, τ) be a topological space. The Borel σ -algebra $\mathbb{B}(X)$ is the smallest σ -algebra that contains the open sets of the topology. $(X, \mathbb{B}(X))$ is called Borel measurable space.

DEFINITION A.3 (Products of Measurable Spaces). Consider the measurable spaces $(\Omega_1, \mathcal{F}_1), \dots, (\Omega_n, \mathcal{F}_n)$. Their product $(\Omega, \mathcal{F})$ is defined as

$$\Omega = \prod_{i=1}^n \Omega_i \quad \text{and} \quad \mathcal{F} = \bigotimes_{i=1}^n \mathcal{F}_i,$$

where

$$\bigotimes_{i=1}^n \mathcal{F}_i = \sigma\left(\prod_{i=1}^n \mathcal{F}_i\right)$$

is the smallest σ -algebra containing $\prod_{i=1}^n \mathcal{F}_i$.

DEFINITION A.4 (Measurability). Let $(\Omega, \mathcal{F})$ and $(\Omega', \mathcal{F}')$ be two measurable spaces. The mapping $\mu: \Omega \rightarrow \Omega'$ is called $(\mathcal{F}, \mathcal{F}')$ -measurable iff

$$\mu^{-1}(A') = \{\omega \in \Omega \mid \mu(\omega) \in A'\} \in \mathcal{F} \quad (A' \in \mathcal{F}').$$

Furthermore, we will use the common notation $\mu: (\Omega, \mathcal{F}) \rightarrow (\Omega', \mathcal{F}')$ to imply that μ is $(\mathcal{F}, \mathcal{F}')$ -measurable.

Figure A.1: Bayes' pool table. Bayes et al., *An essay toward solving a problem in the doctrine of chances*.

Let Ω be a non-empty set and let $\mathcal{C}$ be a collection of subsets of Ω . Then $\sigma(\mathcal{C})$ is the smallest σ -algebra that contains $\mathcal{C}$.

We call μ

- a *probability measure*, if $\mu(\Omega) = 1$,
- a *finite measure*, if $\mu(\Omega) < \infty$,
- a *σ -finite measure*, if there exists a sequence $\{A_n\}_{n \in \mathbb{N}}$ in Ω such that $\bigcup_n A_n = \Omega$ and $\mu(A_n) < \infty$.

DEFINITION A.5 (*Measure Space*). A *measure space* is defined as the triple $(\Omega, \mathcal{F}, \mu)$, where $(\Omega, \mathcal{F})$ is measurable space and μ is a measure on Ω .

DEFINITION A.6 ($\mathcal{L}^p$ spaces). Let $(\mathcal{X}, \mathcal{F}, \mu)$ be a measure space. For any $p \in [1, \infty)$ we denote the set of all measurable functions $f: \mathcal{X} \rightarrow \mathbb{R}$ which satisfy

$$\int_{\mathcal{X}} |f(x)|^p d\mu(x) < \infty$$

by $\mathcal{L}^p$ or $\mathcal{L}^p(\mathcal{X})$ or $\mathcal{L}^p(\mu)$, depending on the context. Moreover, the norm with respect to the $\mathcal{L}^p$ space is

$$\|f\|_{\mathcal{L}^p} = \left(\int_{\mathcal{X}} |f(x)|^p d\mu(x) \right)^{\frac{1}{p}}$$

for all $f \in \mathcal{L}^p$.

DEFINITION A.7 (*Null Set, Almost Everywhere*). Let $(\Omega, \mathcal{F}, \mu)$ be a measure space. $N \in \mathcal{F}$ is called *null set* if $\mu(N) = 0$. Two functions f, g are said to be equal *almost everywhere* (written as $f = g$, *a. e.*) if there exists a null set N such that $f(x) = g(x)$ for all $x \in N^c$.

DEFINITION A.8 (*Equivalence Class of Functions*). We say that two functions $f, g \in \mathcal{L}^p(\mu)$ are equivalent, written as $f \sim g$, if $f(x) = g(x)$ μ -a.e. for all $x \in \mathcal{X}$. We denote the set of *equivalence classes* by

$$L^p(\mu) = \{[f] \mid f \in \mathcal{L}^p(\mu)\}$$

where $[f] = \{g \in \mathcal{L}^p(\mu) \mid f \sim g\}$.

DEFINITION A.9 (*Support of a Measure*). For any measure μ on a topological space S , the support $\text{supp } \mu$ is defined¹ as the smallest closed set $A \subset S$ with $\mu(A^c) = 0$.

¹ Kallenberg, *Foundations of modern probability*, page 9.

DEFINITION A.10 (*Probability Space*). A *probability space* is a measure space $(\Omega, \mathcal{F}, \mathbb{P})$, where $\mathbb{P}$ is a probability measure. Instead of a.e. we will use *a.s.* which is short for *almost surely*.

DEFINITION A.11 (*Random Element*). Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(\Omega', \mathcal{F}')$ a measure space. We call the measurable mapping $X: \Omega \rightarrow \Omega'$ a *random element* or *random variable* (of Ω).

DEFINITION A.12 (*Probability Distribution*). Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $X: (\Omega, \mathcal{F}) \rightarrow (\Omega', \mathcal{F}')$ be a random variable. The image of the measure $\mathbb{P}$ under X , denoted by $\mathbb{P}_X$, is called a (probability) *distribution* of X and is defined as

$$\mathbb{P}_X(A') = \mathbb{P}[X^{-1}(A')] = \mathbb{P}[X \in A'] \quad (A' \in \mathcal{F}').$$

We will sometimes just write $\mathbb{P}(X)$ if we talk about the distribution of X .

DEFINITION A.13 (*Indicator Function*). Let A be a subset of Ω . The function $1_A: \Omega \rightarrow \mathbb{R}$ with

$$\mathbb{I}_A(\omega) = \begin{cases} 1 & (\omega \in A) \\ 0 & (\omega \notin A) \end{cases}$$

is called *indicator function*.

DEFINITION A.14 (*Dirac Measure*). Let $\omega \in \Omega$. The measure δ_ω with

$$\delta_\omega(A) = \mathbb{I}_A(\omega)$$

for any subset A of Ω is called *Dirac measure*.

DEFINITION A.15 (*Lebesgue Integral*). Let $(\Omega, \mathcal{F}, \mu)$ be a measurable space. We say the *Lebesgue integral* of the measurable function $f: \Omega \rightarrow \mathbb{R}$ exists or is defined if

$$\int_{\Omega} f^+ d\mu < \infty \quad \text{and} \quad \int_{\Omega} f^- d\mu < \infty,$$

where

$$f^+ = \max(f, 0) \quad \text{and} \quad f^- = -\min(f, 0)$$

are the positive and negative parts of the function. If the integral of f exists, it is defined as

$$\int_{\Omega} f d\mu = \int_{\Omega} f^+ d\mu - \int_{\Omega} f^- d\mu$$

and f is called *integrable*.

DEFINITION A.16 (*Integral Measure*). For a function $f \in \mathcal{L}^1([0, \infty])$, we define the measure $\nu: \mathcal{F} \rightarrow [0, \infty]$ by

$$\nu(A) = \int_A f d\mu$$

and say ν is *absolute continuous* with respect to μ if $\nu(A) = 0$, whenever $\mu(A) = 0$, $A \in \mathcal{F}$ and write $\nu \ll \mu$.

COROLLARY A.17 (*Radon-Nikodym*). Let μ and ν be σ -finite measures on $(\Omega, \mathcal{F})$ with $\nu \ll \mu$. Then there exists a μ -a. e. unique measurable function $f: \Omega \rightarrow \mathbb{R}_+$ such that

$$\nu(A) = \int_A f d\mu \quad (A \in \mathcal{F}).$$

which is called *Radon-Nikodym derivative* of ν with respect to μ denoted by $f = \frac{d\nu}{d\mu}$.

DEFINITION A.18 (*Density*). Let $(\Omega, \mathcal{F}) = (\mathbb{R}^n, \mathbb{B}(\mathbb{R}^n))$, ν a σ -finite measure on $(\Omega, \mathcal{F})$ and λ the Borel-Lebesgue measure. If

$$f = \frac{d\nu}{d\lambda^n}$$

then we call f the *Lebesgue density* of ν .

DEFINITION A.19 (*Expectation*). Let X be a random variable on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The *expectation* of X is defined as

$$\int X \, d\mathbb{P} = \int_{\Omega} X(\omega) \, \mathbb{P}[d\omega] = \int_{\Omega} X(\omega) \, d\mathbb{P}(\omega)$$

and denoted by $\mathbb{E}[X]$.

DEFINITION A.20 (*Conditional Expectation*). Let $X \in \mathcal{L}^1$ be a random variable on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and $\mathcal{G}$ be a sub- σ -algebra of $\mathcal{F}$. A $\mathcal{G}$ -measurable random variable $\xi \in \mathcal{L}^1$ is called the *conditional expectation* of X with respect to $\mathcal{G}$ iff

$$\mathbb{E}[\xi \mathbb{I}_A] = \mathbb{E}[X \mathbb{I}_A] \quad \text{for all } A \in \mathcal{G}$$

and denoted with $\mathbb{E}[X \mid \mathcal{G}]$.

DEFINITION A.21 (*Empirical Measure*). Let $(\mathcal{Z}, \mathcal{Z}, \mathbb{P}_Z)$ be the Cartesian product of the probability space $(\mathcal{X}, \mathcal{X}, \mathbb{P}_X)$ and $X_1, X_2, \dots$, be independent random variables taking values in $(\mathcal{X}, \mathcal{X})$ with $\mathbb{P}_{X_i} = \mathbb{P}_X$. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be the underlying source of events of $(\mathcal{Z}, \mathcal{Z}, \mathbb{P}_Z)$ as $Z: \Omega \rightarrow \mathcal{Z}$. The *empirical measures* $\mathbb{P}_n$ are defined by

$$\mathbb{P}_n(S)(\omega) = \frac{1}{n} \sum_{i=1}^n \delta_{X_i(\omega)}(S), \quad S \in \mathcal{X}, \quad \omega \in \Omega$$

and for any bounded, continuous, real-valued function f on $\mathcal{X}$

$$\int f \, d\mathbb{P}_n = \frac{1}{n} \sum_{i=1}^n f(X_i)$$

converges to $\int f \, d\mathbb{P}_X$ a.s. as $n \rightarrow \infty$.

DEFINITION A.22 (*Stochastic Process*). A (discrete-time) *stochastic process* on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a sequence $\{X_n \mid n \in \mathbb{N}_0\}$ of random elements.

DEFINITION A.23 (*Limit Inferior, Limit Superior*). Let $\{A_n \mid n \in \mathbb{N}\}$ be a sequence of sets in Ω . The *limit inferior* and *limit superior* are defined as

$$\begin{aligned} \liminf_{n \rightarrow \infty} A_n &= \sup_n \left(\inf_{k \geq n} A_k \right) \quad \text{and} \\ \limsup_{n \rightarrow \infty} A_n &= \inf_n \left(\sup_{k \geq n} A_k \right) \end{aligned}$$

respectively.

CHAPTER B

Functional Analysis

In this chapter we summarize several results and notions from *functional analysis* used in this work. For a detailed description and derivations we refer to the book of Meise et al.¹

¹ Meise et al., *Introduction to functional analysis*.

Figure B.1: Relation between topological spaces. The elements in the purple set are complete, *i.e.*, every Cauchy sequence is convergent. Completeness is a property of the metric and not of the topology, hence only metric spaces can be complete.

B.1 Metric Spaces & Topologies

DEFINITION B.1 (*Topology, Topological Space*). Let X be a nonempty collection and τ a set of subsets of X . τ is called *topology* on X and (X, τ) is called *topological space* if

- $\emptyset \in \tau$ and $X \in \tau$,
- the union of any number of elements of τ is an element of τ ,
- the intersection of a finite number of elements of τ is an element of τ .

We call a member of τ an *open set* in X . The complement of an open set is a *closed set*.

DEFINITION B.2 (*Metric Space*). A metric is a distance function between elements in a set. Let X be a set. The map $d: X \times X \rightarrow \mathbb{R}$ is called *metric* if for all $x, y, z \in X$ the following conditions hold:

$$\begin{aligned} d(x, y) &\geq 0 && \text{(non-negativity)} \\ d(x, y) &= 0 \text{ if and only if } x = y && \text{(identity of indiscernibles)} \\ d(x, y) &= d(y, x) && \text{(symmetry)} \\ d(x, z) &\leq d(x, y) + d(y, z) && \text{(triangle inequality).} \end{aligned}$$

The tuple (X, d) is called *metric space*.

DEFINITION B.3 (*Interior, Closure, Boundary*). Consider the topological space (X, τ) and let U be any subset of X .

- $\overset{\circ}{U}$ is the largest open set included in U and called the *interior* of U ,
- $\overline{U}$ is the smallest closed set including U , called *closure* of U ,
- $\partial U = \overline{U} \setminus \overset{\circ}{U}$ is the *boundary* of U .

DEFINITION B.4 (*Neighborhood*). A *neighborhood* of a point x is a set X containing x in its interior.

DEFINITION B.5 (*Open Ball*). Let (X, d) be a metric space. For $p \in X$ and $\epsilon > 0$

$$B_\epsilon(p) = B(p, \epsilon) = \{x \in X \mid d(x, p) < \epsilon\}$$

is the *open ball* about p with radius ϵ .

DEFINITION B.6 (*Dense Set*). A subset U of a topological space X is *dense* (in X) if $\overline{U} = X$.

DEFINITION B.7 (*Separable Space*). A topological space is *separable* if it includes a countable dense subset.

DEFINITION B.8 (*Open Cover*). Let (X, τ) be a topological space and $M \subset X$. A nonempty collection $\{U_\alpha\}_{\alpha \in A}$ of open sets with

$$M \subset \bigcup_{\alpha \in A} U_\alpha$$

is called an *open cover* of M .

DEFINITION B.9 (*Compactness*). Let (X, τ) be a topological space. $M \subset X$ is called *compact* if each of its open covers has a *finite subcover*. Equivalently for every open cover $\{U_\alpha\}_{\alpha \in A}$ there is a finite subset J of A such that $\{U_i\}_{i \in J}$ is an open cover of M .

DEFINITION B.10 (*Inclusion*). Let A be a subset of B . Then

$$\iota_A: A \rightarrow B, \quad \iota_A(x) = x$$

is called *inclusion* and sometimes written as $A \hookrightarrow B$.

DEFINITION B.11 (*Limit, Convergence*). Let $(x_n)_{n \in \mathbb{N}}$ be a sequence in a topological space (X, τ) . The element x_0 is called the *limit* of (x_n) if for every neighborhood U of x_0 there exists a $n_0 \in \mathbb{N}$ such that $x_n \in U$ for all $n \geq n_0$. If a sequence has a limit, it is called *convergent* and we write

$$\lim_{n \rightarrow \infty} x_n = x_0 \quad \text{or} \\ x_n \rightarrow x_0 \text{ as } n \rightarrow \infty.$$

A sequence is called *divergent* if it has no limit.

DEFINITION B.12 (*Cauchy Sequence*). A *Cauchy sequence* in a metric space (X, d) is a sequence (x_n) if for every $\epsilon > 0$ there exists $n_0 \in \mathbb{N}$ such that $d(x_n, x_m) < \epsilon$ for all $m, n \geq n_0$.

DEFINITION B.13 (*Complete Metric Space*). A metric space is called *complete* if every Cauchy sequence is convergent.

DEFINITION B.14 (*Bounded Set*). A subset M of a metric space (X, d) is called *bounded* if there exists a $x \in X$ and an $\epsilon > 0$ such that for all $M \subset B(x, \epsilon)$.

DEFINITION B.15 (*Total Boundedness*). Let (X, d) be a metric space. A set $M \subset X$ is *totally bounded* if for every $\epsilon > 0$ there exist finitely many elements $a_1, \dots, a_n \in M$, such that the union of open balls $\{B(a_i, \epsilon)\}_{i=1, \dots, n}$ is an open cover of M .

DEFINITION B.16 (*Continuous Function*). Let X and Y be two topological spaces. A map $f: X \rightarrow Y$ is called continuous at $x \in X$ if for every neighborhood $V \subseteq Y$ of $f(x)$ there exists a neighborhood $U \subseteq X$ of x such that $f(U) \subseteq V$. The function $f: X \rightarrow Y$ is called *continuous* if it is continuous at all $x \in X$. We define $C(X, Y) = \{f: X \rightarrow Y \mid f \text{ continuous}\}$.

B.2 Banach Spaces

DEFINITION B.17 (*Norm, Normed Space*). Let $\mathcal{F}$ be a vector space over $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. A map $\|\cdot\|_{\mathcal{F}}: \mathcal{F} \rightarrow [0, \infty)$, $x \mapsto \|x\|_{\mathcal{F}}$ is called *norm* on $\mathcal{F}$ if for all $x, y \in \mathcal{F}$ and $\alpha \in \mathbb{K}$

$$\begin{aligned} \|x\|_{\mathcal{F}} &\geq 0, & (\text{non-negativity}) \\ \|x\|_{\mathcal{F}} &= 0 \text{ iff } x = 0, & (\text{identity of indiscernibles}) \\ \|\alpha x\|_{\mathcal{F}} &= |\alpha| \|x\|_{\mathcal{F}}, & (\text{positive homogeneity}) \\ \|x + y\|_{\mathcal{F}} &\leq \|x\|_{\mathcal{F}} + \|y\|_{\mathcal{F}}. & (\text{triangle inequality}) \end{aligned}$$

We call $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$ or simply $(\mathcal{F}, \|\cdot\|)$ a *normed space*.

DEFINITION B.18 (*Induced Metric*). Let $(\mathcal{F}, \|\cdot\|)$ be a normed space. Then

$$d(x, y) = \|x - y\| \quad \forall x, y \in \mathcal{F}$$

is the metric induced by the norm $\|\cdot\|_{\mathcal{F}}$. Hence $(\mathcal{F}, d)$ is a metric space.

DEFINITION B.19 (*Banach Space*). A *Banach space* is a normed space which is complete with respect to the metric induced by its norm, *i.e.*, it contains the limits of all its Cauchy sequences.

DEFINITION B.20 (*Supremum Norm*). The vector space $(\mathcal{B}, \|\cdot\|_\infty)$ of functions $f: X \rightarrow \mathbb{R}$ with

$$\|f\|_\infty = \sup_{x \in X} |f(x)|$$

is a Banach space.

B.3 Bounded Linear Operators

DEFINITION B.21 (*Linear Operator*). Consider the normed spaces $(\mathcal{F}, \|\cdot\|)$ and $(\mathcal{G}, \|\cdot\|)$ over $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. A function $A: \mathcal{F} \rightarrow \mathcal{G}$ is called a *linear operator* if it satisfies

$$\begin{aligned} A(\alpha f) &= \alpha(Af) & \forall \alpha \in \mathbb{K}, f \in \mathcal{F}, & \quad (\text{homogeneity}) \\ A(f + g) &= Af + Ag & \forall f, g \in \mathcal{F}. & \quad (\text{additivity}) \end{aligned}$$

We define that

- $\text{Hom}(\mathcal{F}, \mathcal{G})$ denotes the set of all linear operators from $\mathcal{F}$ to $\mathcal{G}$,
- $\mathcal{L}(\mathcal{F}, \mathcal{G}) = \{A \in \text{Hom}(\mathcal{F}, \mathcal{G}) \mid A \text{ continuous}\}$ and $\mathcal{L}(\mathcal{F}) = \mathcal{L}(\mathcal{F}, \mathcal{F})$.

DEFINITION B.22 (*Bounded Operator*). Let $(\mathcal{F}, \|\cdot\|)$ and $(\mathcal{G}, \|\cdot\|)$ be two normed spaces. A linear operator $A: \mathcal{F} \rightarrow \mathcal{G}$ is said to be *bounded* if A maps every bounded subset of $\mathcal{F}$ onto a bounded subset of $\mathcal{G}$. Equivalently, for every $f \in \mathcal{F}$ there exists an $\lambda > 0$ such that $\|Af\|_{\mathcal{G}} \leq \lambda \|f\|_{\mathcal{F}}$.

B.4 Inner Product Spaces

DEFINITION B.23 (*Inner Product, Inner Product Space*). Let $\mathcal{F}$ be a vector space over the field $\mathbb{K}$. The map $\langle \cdot, \cdot \rangle_{\mathcal{F}}: \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{K}$ is called *inner product* on $\mathcal{F}$ if for all vectors $x, y, z \in \mathcal{F}$ and $\alpha, \beta \in \mathbb{K}$ the following conditions hold:

$$\begin{aligned} \langle x, y \rangle_{\mathcal{F}} &= \overline{\langle y, x \rangle_{\mathcal{F}}}, & (\text{conjugate symmetry}) \\ \langle \alpha x + \beta y, z \rangle_{\mathcal{F}} &= \alpha \langle x, z \rangle_{\mathcal{F}} + \beta \langle y, z \rangle_{\mathcal{F}}, & (\text{linearity in arguments}) \\ \langle x, x \rangle_{\mathcal{F}} &\geq 0 \text{ and } \langle x, x \rangle_{\mathcal{F}} = 0 \Rightarrow x = 0. & (\text{positive-definiteness}) \end{aligned}$$

We call the tuple $(\mathcal{F}, \langle \cdot, \cdot \rangle_{\mathcal{F}})$ or simply $(\mathcal{F}, \langle \cdot, \cdot \rangle)$ an *inner product space*.

DEFINITION B.24 (*Dual Space, Bi-Dual Space*). Let $(\mathcal{F}, \|\cdot\|)$ be a normed space over $\mathbb{K}$. The space of *continuous linear functionals*

$$\mathcal{F}^* = \mathcal{L}(\mathcal{F}, \mathbb{K})$$

is called the *dual space* of $\mathcal{F}$. Let $f \in \mathcal{F}^*$ and $x \in \mathcal{F}$, then the evaluation of f at x is called the *dual map* or *dual pairing* and denoted by $\langle f, x \rangle_* = f(x)$. Every Banach space is isometrically embedded into its *bi-dual space* $\mathcal{F}^{**} = (\mathcal{F}^*)^*$.

CHAPTER C

Convex Analysis

DEFINITION C.1 (*Convex Set, Convex Function*). Let $\mathcal{F}$ be a topological vector space over $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. A subset $\mathcal{X}$ in $\mathcal{F}$ is called *convex* if $\forall x, y \in \mathcal{X}$ and $\lambda \in [0, 1]$

$$\lambda x + (1 - \lambda)y \in \mathcal{X},$$

which means $\mathcal{X}$ contains all its line segments. A function $f: \mathcal{X} \rightarrow \mathbb{R}$ is called *convex* if

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

for $\forall x, y \in \mathcal{X}$ and $\lambda \in [0, 1]$.

DEFINITION C.2 (*Effective Domain*). The *effective domain* of a function $f: \mathcal{X} \rightarrow \mathbb{R} \cup \{\pm\infty\}$ is defined as

$$\text{dom}(f) = \{x \in \mathcal{X} \mid f(x) < +\infty\}$$

where $\mathcal{X}$ is a normed vector space.

DEFINITION C.3 (*Weak Convergence*). Let $(\mathcal{F}, \|\cdot\|)$ be a normed space over $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ and (u_n) be a sequence in $\mathcal{F}$. If and only if for all elements $f \in \mathcal{F}^*$

$$\langle f, u_n \rangle_{\star} \rightarrow \langle f, u \rangle_{\star} \quad \text{as } n \rightarrow \infty$$

we say that (u_n) *converges weakly* to u and write $u_n \rightharpoonup u$.

PROPOSITION C.4. Strong convergence implies weak convergence. The converse is only true in finite-dimensional spaces.¹

DEFINITION C.5 (*Coercive*). Let $(\mathcal{F}, \|\cdot\|)$ be a real normed space. The function $f: M \subseteq \mathcal{F} \rightarrow \mathbb{R}$ is called *coercive* on M iff

$$\frac{f(u)}{\|u\|} \rightarrow +\infty$$

as $\|u\| \rightarrow \infty$.

DEFINITION C.6 (*Weakly Sequentially Continuous*). Let $(\mathcal{F}, \|\cdot\|)$ be a real normed space. Iff for each $u \in M \subseteq \mathcal{F}$ and each sequence (u_n) in M

$$u_n \rightharpoonup u \implies f(u_n) \rightarrow f(u) \quad \text{as } \|u\| \rightarrow \infty$$

then f is called *weakly sequentially continuous* on M .

DEFINITION C.7 (*Weakly Closed*). Let (u_n) be a sequence in a subset M of a real normed space $(\mathcal{F}, \|\cdot\|)$. If $(u_n) \rightharpoonup u$ implies that $u \in M$ then M is said to be *weakly closed*.

¹ Zeidler, *Applied Functional Analysis*, Page 50, Proposition 3.

Bibliography

- Agarwal, A., P. Bartlett, P. Ravikumar, and M. Wainwright (2011).
“Information-Theoretic Lower Bounds on the Oracle Complexity of
Convex Optimization Convex optimization”. In: *Advances in Neural
Information Processing Systems*, pp. 1–9 (cited on pages 88, 90).
- Aggarwal, C. C. (2013).
Outlier analysis. Springer (cited on page 13).
- Aleskerov, E., B. Freisleben, and B. Rao (1997).
“Cardwatch: A neural network based database mining system for credit
card fraud detection”. In: *Computational Intelligence for Financial
Engineering* (cited on page 5).
- Aliprantis, C. D. and K. Border (2006).
Infinite dimensional analysis: a hitchhiker’s guide. Springer Science &
Business Media (cited on pages 65, 66).
- Angiulli, F. and F. Fasseti (2010).
“Distance-based outlier queries in data streams: the novel task and
algorithms”. In: *Data Mining and Knowledge Discovery* 20.2, pp. 290–
324 (cited on page 33).
- Arthur, D. and S. Vassilvitskii (2007).
“k-means++: The advantages of careful seeding”. In: *Proceedings of
the eighteenth annual ACM-SIAM symposium on Discrete algorithms*.
Society for Industrial and Applied Mathematics, pp. 1027–1035 (cited
on page 45).
- Bach, S. H. and M. A. Maloof (2008).
“Paired learners for concept drift”. In: *Data Mining, 2008. ICDM’08.
Eighth IEEE International Conference on*. IEEE, pp. 23–32 (cited on
page 107).
- Bardenet, R., A. Doucet, and C. Holmes (2014).
“Towards scaling up Markov chain Monte Carlo: an adaptive subsam-
pling approach”. In: *Proceedings of the 31st International Conference on
Machine Learning*, pp. 405–413 (cited on page 118).
- Barreno, M. et al. (2006).
“Can machine learning be secure?” In: *Proceedings of the 2006 ACM Sym-
posium on Information, computer and communications security*. ACM,
pp. 16–25 (cited on page 111).

- Bartlett, P. L. and S. Mendelson (2002).
 “Rademacher and Gaussian Complexities: Risk Bounds and Structural Results”. In: *Journal of Machine Learning Research* 3, pp. 463–482 (cited on page 127).
- Basharat, A., A. Gritai, and M. Shah (2008).
 “Learning object motion patterns for anomaly detection and improved object detection”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, pp. 1–8 (cited on page 28).
- Bayes, T. et al. (1984).
 “An essay toward solving a problem in the doctrine of chances”. In: *Philosophical Transactions of the Royal Society of London* 53 (cited on page 145).
- Bellman, R. E. (1961).
Adaptive control processes: A guided tour. Princeton University Press (cited on page 31).
- Bentley, J. L. (1975).
 “Multidimensional binary search trees used for associative searching”. In: *Communications of the ACM* 18.9, pp. 509–517 (cited on page 38).
- Berlinet, A. and C. Thomas-Agnan (2004).
Reproducing Kernel Hilbert Spaces in Probability and Statistics. 1st ed. Springer Science & Business Media, p. 355 (cited on pages 53, 54, 58, 59, 62, 67).
- Bifet, A. et al. (2009).
 “New ensemble methods for evolving data streams”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 139–148 (cited on page 115).
- Bochner, S. (1932).
Vorlesungen über Fouriersche Integrale. Akademische Verlagsgesellschaft (cited on page 56).
- Bousquet, O. and L. Bottou (2008).
 “The tradeoffs of large scale learning”. In: *Advances in neural information processing systems*, pp. 161–168 (cited on pages 87, 88).
- Boyd, S. and L. Vandenberghe (2004).
Convex optimization. Cambridge University Press, pp. 1–730 (cited on pages 87, 91, 98).
- Breunig, M. M., H.-P. Kriegel, R. T. Ng, and J. Sander (2000).
 “LOF: identifying density-based local outliers”. In: *ACM Sigmod Record*. Vol. 29. 2. ACM, pp. 93–104 (cited on pages 34, 37).
- Cannon, A., J. M. Ettinger, D. Hush, and C. Scovel (2002).
 “Machine learning with data dependent hypothesis classes”. In: *The Journal of Machine Learning Research* 2, pp. 335–358 (cited on page 118).

- Chandola, V., A. Banerjee, and V. Kumar (2009).
 “Anomaly detection: A survey”. In: *ACM Computing Surveys (CSUR)* 41.3, pp. 1–58 (cited on pages 5, 13).
- Chang, C.-C. and C.-J. Lin (2011).
 “LIBSVM: A library for support vector machines”. In: *ACM Transactions on Intelligent Systems and Technology* 2.3, pp. 1–27 (cited on page 81).
- Chapelle, O., P. Haffner, and V. N. Vapnik (1999).
 “Support vector machines for histogram-based image classification”. In: *IEEE Transactions on Neural Networks* 10.5, pp. 1055–1064 (cited on page 131).
- Chawla, N. V., N. Japkowicz, A. Kotcz, and P. Drive (2004).
 “Editorial: special issue on learning from imbalanced data sets”. In: *ACM Sigkdd Explorations Newsletter* 6.1, pp. 1–6 (cited on page 14).
- Chentsov, N. N. (1962).
 “Evaluation of an unknown distribution density from observations”. In: *Doklady* 147.1, p. 45 (cited on page 75).
- Chwialkowski, K. and A. Gretton (2014).
 “A Kernel Independence Test for Random Processes”. In: *Proceedings of 31st International Conference on Machine Learning* (cited on page 62).
- Cortes, C. and V. Vapnik (1995).
 “Support-vector networks”. In: *Machine learning* 20.3, pp. 273–297 (cited on page 54).
- Cox, D. R. (1958).
 “The regression analysis of binary sequences”. In: *Journal of the Royal Statistical Society*, pp. 215–242 (cited on page 54).
- Cuny, C. (2015).
 “A compact LIL for martingales in 2-smooth Banach spaces with applications”. In: *Bernoulli* 21.1, pp. 374–400 (cited on page 121).
- Davis, J. and M. Goadrich (2006).
 “The relationship between Precision-Recall and ROC curves”. In: *Proceedings of the 23rd international conference on Machine learning*. ACM, pp. 233–240 (cited on page 21).
- Dawid, A. P. (1984).
 “Present position and potential developments: Some personal views: Statistical theory: The prequential approach”. In: *Journal of the Royal Statistical Society. Series A (General)*, pp. 278–292 (cited on page 114).
- Dean, J. and S. Ghemawat (2008).
 “MapReduce: simplified data processing on large clusters”. In: *Communications of the ACM* 51.1, pp. 107–113 (cited on page 68).

- Dempster, A. P., N. M. Laird, and D. B. Rubin (1977).
 “Maximum likelihood from incomplete data via the EM algorithm”.
 In: *Journal of the royal statistical society. Series B (methodological)* 39.1,
 pp. 1–38 (cited on page 28).
- Demšar, J. (2006).
 “Statistical comparisons of classifiers over multiple data sets”. In: *The Journal of Machine Learning Research* 7, pp. 1–30 (cited on pages 22, 23).
- Domingos, P. and G. Hulten (2001).
 “Catching up with the Data: Research Issues in Mining Data Streams”.
 In: *DMKD* (cited on page 106).
- Dunn, O. J. (1959).
 “Estimation of the Medians for Dependent Variables”. In: *Ann. Math. Statist.* 30.1, pp. 192–197 (cited on page 23).
- Dunning, T. and E. Friedman (2014).
Practical Machine Learning: A New Look at Anomaly Detection.
 O’Reilly Media, p. 66 (cited on page 3).
- Edgeworth, F. Y. (1887).
 “On discordant observations”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 23.143, pp. 364–375 (cited on page 5).
- Efromovich, S. (2008).
Nonparametric curve estimation: methods, theory, and applications.
 Springer-Verlag New York, p. 414 (cited on pages 75, 76).
- Eiter, T. and H. Mannila (1994).
 “Computing discrete Fréchet distance”. In: *Rapport technique num. CD-TR 94*, p. 64 (cited on page 134).
- Endler, D. (1998).
 “Intrusion detection. applying machine learning to Solaris audit data”.
 In: *Computer Security Applications Conference, 1998. Proceedings. 14th Annual*. IEEE, pp. 268–279 (cited on page 30).
- Eto, M. R. (n.d.).
The Concept of Depth In Statistics. Tech. rep. (cited on page 31).
- Faloutsos, C. and K.-I. Lin (1995).
 “FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets”. In: *ACM SIGMOD Record* 24.2, p. 163 (cited on page 133).
- Fawcett, T. (2006).
 “An introduction to ROC analysis”. In: *Pattern recognition letters* 27.8,
 pp. 861–874 (cited on pages 20, 21).

- Fino, B. J. and V. R. Algazi (1976).
 “Unified matrix treatment of the fast Walsh-Hadamard transform”. In:
IEEE Transactions on Computers 11, pp. 1142–1146 (cited on page 73).
- Friedman, J. H., W. Stuetzle, and A. Schroeder (1984).
 “Projection Pursuit Density Estimation”. In: *Journal of the American Statistical Association* 79.387, p. 599 (cited on page 31).
- Friedman, M. (1937).
 “The use of ranks to avoid the assumption of normality implicit in the analysis of variance”. In: *Journal of the American Statistical Association* 32.200, pp. 675–701 (cited on page 22).
- Fukumizu, K., A. Gretton, B. Schölkopf, and B. K. Sriperumbudur (2009).
 “Characteristic kernels on groups and semigroups”. In: *Advances in Neural Information Processing Systems*, pp. 473–480 (cited on page 64).
- Fukumizu, K., L. Song, and A. Gretton (2013).
 “Kernel Bayes’ rule: Bayesian inference with positive definite kernels”. In: *The Journal of Machine Learning Research* 14, pp. 3753–3783 (cited on page 62).
- Gama, J. (2010).
Knowledge discovery from data streams. CRC Press (cited on page 110).
- Gama, J. et al. (2014).
 “A survey on concept drift adaptation”. In: *ACM Computing Surveys (CSUR)* 46.4, p. 44 (cited on pages 7, 19, 105, 107).
- Ghanem, R. G. and P. D. Spanos (2003).
Stochastic finite elements: a spectral approach. Courier Corporation (cited on page 55).
- Goodman, J. E. and J. O’Rourke (2004).
Handbook of discrete and computational geometry. 2nd ed. CRC press (cited on pages 6, 38).
- Gretton, A. et al. (2006).
 “A kernel method for the two-sample-problem”. In: *Advances in neural information processing systems* (cited on pages 62, 63).
- Gretton, A. et al. (2012).
 “A kernel two-sample test”. In: *The Journal of Machine Learning Research* 13.1, pp. 723–773 (cited on page 62).
- Hanley, J. A. and B. J. McNeil (1982).
 “The meaning and use of the area under a receiver operating characteristic (ROC) curve”. In: *Radiology* 143.1, pp. 29–36 (cited on page 99).
- Hardy, R. L. (1971).
 “Multiquadric equations of topography and other irregular surfaces”. In: *Journal of geophysical research* 76.8, pp. 1905–1915 (cited on page 58).

- Hawkins, D. M. (1980).
Identification of Outliers. Springer, p. 200.
- Ho, S.-S. (2005).
 “A martingale framework for concept change detection in time-varying data streams”. In: *Proceedings of the 22nd international conference on Machine learning*. ACM, pp. 321–327 (cited on page 114).
- Hodge, V. J. and J. Austin (2004).
 “A survey of outlier detection methodologies”. In: *Artificial Intelligence Review* 22.2, pp. 85–126 (cited on page 13).
- Hoeffding, W. (1963).
 “Probability inequalities for sums of bounded random variables”. In: *Journal of the American statistical association* 58.301, pp. 13–30 (cited on pages 68, 117, 119, 120).
- Hoffmann-Jørgensen, J. (1994).
Probability with a view toward statistics. Chapman & Hall.
- Holm, S. (1979).
 “A simple sequentially rejective multiple test procedure”. In: *Scandinavian journal of statistics* 6.2, pp. 65–70 (cited on page 23).
- Iman, R. L. and J. M. Davenport (1980).
 “Approximations of the critical region of the Friedman statistic”. In: *Communications in Statistics-Theory and Methods* 9.6, pp. 571–595 (cited on page 22).
- Izenman, A. J. (1991).
 “Recent developments in nonparametric density estimation”. In: *Journal of the American Statistical Association* 86.413, pp. 205–224 (cited on page 76).
- Jacobs, A. (2009).
 “The pathologies of big data”. In: *Communications of the ACM* 52.8, pp. 36–44 (cited on page 6).
- Janssens, J. H. M., F. Huszár, E. O. Postma, and H. J. van den Herik (2012).
Stochastic outlier selection. Tech. rep. Technical report TiCC TR 2012-001, Tilburg University, Tilburg Center for Cognition and Communication, Tilburg, The Netherlands (cited on pages 38, 88).
- Janssens, J., E. Postma, and J. van den Herik (2013).
 “Density-Based Anomaly Detection in the Maritime Domain”. In: *Situation Awareness with Systems of Systems*. Springer, pp. 119–131 (cited on page 37).
- Kallenberg, O. (2006).
Foundations of modern probability. Springer Science & Business Media (cited on pages 15, 54, 121, 146).

- Kallenberg, O. and R. Sztencel (1991).
 “Some dimension-free features of vector-valued martingales”. In: *Probability theory and related fields* 88.2, pp. 215–247 (cited on page 125).
- Kar, P. and H. Karnick (2012).
 “Random feature maps for dot product kernels”. In: *International Conference on Artificial Intelligence and Statistics*, pp. 583–591 (cited on page 131).
- Kaufman, L. and P. J. Rousseeuw (2009).
Finding groups in data: An introduction to cluster analysis. Vol. 344. John Wiley & Sons (cited on page 45).
- Kent, S. M. (1994).
 “Sloan digital sky survey”. In: *Science with Astronomical Near-Infrared Sky Surveys*. Springer, pp. 27–30 (cited on page 5).
- Kifer, D., S. Ben-David, and J. Gehrke (2004).
 “Detecting change in data streams”. In: *Proceedings of the Thirtieth international conference on Very large data bases*. VLDB Endowment. Elsevier {BV}, pp. 180–191 (cited on page 107).
- Kish, L. (1965).
Survey sampling. New York: John Wiley and Sons (cited on page 118).
- Kloft, M. and P. Laskov (2010).
 “Online anomaly detection under adversarial impact”. In: *International Conference on Artificial Intelligence and Statistics*, pp. 405–412 (cited on page 111).
- Kloft, M. and P. Laskov (2012).
 “Security analysis of online centroid anomaly detection”. In: *The Journal of Machine Learning Research* 13.1, pp. 3681–3724 (cited on pages 111, 113).
- Knorr, E. M. and R. T. Ng (1998).
 “Algorithms for mining distance-based outliers in large datasets”. In: *Proceedings of the International Conference on Very Large Data Bases*. Citeseer, pp. 392–403 (cited on page 33).
- Knorr, E. M., R. T. Ng, and V. Tucakov (2000).
 “Distance-based outliers: algorithms and applications”. In: *The VLDB Journal—The International Journal on Very Large Data Bases* 8.3-4, pp. 237–253 (cited on page 33).
- Kontaki, M. et al. (2011).
 “Continuous monitoring of distance-based outliers over data streams”. In: *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*. IEEE, pp. 135–146 (cited on page 33).

- Kriegel, H.-P. and A. Zimek (2008).
 “Angle-based outlier detection in high-dimensional data”. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 444–452 (cited on page 43).
- Kumar, V. (2005).
 “Parallel and distributed computing for cybersecurity”. In: *IEEE Distributed Systems Online* 6.10, p. 1 (cited on page 5).
- Lazarescu, M. M., S. Venkatesh, and H. H. Bui (2004).
 “Using multiple windows to track concept drift”. In: *Intelligent data analysis* 8.1, pp. 29–59 (cited on page 108).
- Le, Q., T. Sarlós, and A. J. Smola (2013).
 “Fastfood: approximating kernel expansions in loglinear time”. In: *Proceedings of the international conference on machine learning* (cited on page 71).
- LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998).
 “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324 (cited on page 127).
- Li, F., C. Ionescu, and C. Sminchisescu (2010).
 “Random Fourier approximations for skewed multiplicative histogram kernels”. In: *Pattern Recognition*. Springer, pp. 262–271 (cited on page 131).
- Lichman, M. (2013).
UCI Machine Learning Repository (cited on page 80).
- Lindsay, B. G. (1983).
 “The geometry of mixture likelihoods: a general theory”. In: *The annals of statistics* 11.1, pp. 86–94 (cited on page 28).
- Liu, F. T., K. M. Ting, and Z.-H. Zhou (2012).
 “Isolation-based anomaly detection”. In: *ACM Transactions on Knowledge Discovery from Data* 6.1, p. 3 (cited on pages 41, 80).
- Lloyd, S. (1982).
 “Least squares quantization in PCM”. In: *Information Theory, IEEE Transactions on* 28.2, pp. 129–137 (cited on page 44).
- MacQueen, J. (1967).
 “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, pp. 281–297 (cited on page 44).
- Matérn, B. (2013).
Spatial variation. Vol. 36. Springer Science & Business Media (cited on page 57).

McAfee (2014).

“Net Losses: Estimating the Global Cost of Cybercrime”. In: *McAfee, Centre for Strategic & International Studies* (cited on page 5).

McDiarmid, C. (1989).

“On the method of bounded differences”. In: *Surveys in combinatorics* 14.1, pp. 148–188 (cited on page 126).

McGill, R., J. W. Tukey, and W. A. Larsen (1978).

“Variations of box plots”. In: *The American Statistician* 32.1, pp. 12–16 (cited on page 27).

Meise, R. and D. Vogt (1997).

Introduction to functional analysis. Oxford University Press (cited on page 149).

Mercer, J. (1909).

“Functions of positive and negative type, and their connection with the theory of integral equations”. In: *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, pp. 415–446 (cited on page 55).

Mosler, K. (2013).

“Depth Statistics”. In: *Robustness and Complex Data Structures: Festschrift in Honour of Ursula Gather*. Ed. by C. Becker, R. Fried, and S. Kuhnt. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. Depth Stat, pp. 17–34 (cited on page 31).

Nadaraya, È. A. (1965).

“On non-parametric estimates of density functions and regression curves”. In: *Theory of Probability & Its Applications* 10.1, pp. 186–190 (cited on page 30).

Nelson, B. and A. D. Joseph (2006).

“Bounding an attack’s complexity for a simple learning model”. In: *Proc. of the First Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML), Saint-Malo, France*. Citeseer (cited on pages 111, 112).

Nemenyi, P. (1963).

“Distribution-free multiple comparisons”. PhD thesis. Princeton University. (cited on page 23).

Nemirovski, A., A. Juditsky, G. Lan, and A. Shapiro (2009).

“Robust stochastic approximation approach to stochastic programming”. In: *SIAM Journal on Optimization* 19.4, pp. 1574–1609 (cited on pages 90, 91).

Nesterov, Y. (2014).

“Subgradient methods for huge-scale optimization problems”. In: *Mathematical Programming* 146.1-2, pp. 275–297 (cited on page 88).

Nesterov, Y. (1983).

“A method of solving a convex programming problem with convergence rate $O(1/k^2)$ ”. In: *Soviet Mathematics Doklady* 27.2, pp. 372–376 (cited on page 98).

Nesterov, Y. (2004).

Introductory lectures on convex optimization. Vol. 87. Springer Science & Business Media (cited on page 91).

Netzer, Y. et al. (2011).

“Reading digits in natural images with unsupervised feature learning”. In: *NIPS workshop on deep learning and unsupervised feature learning*. Vol. 2011, p. 4 (cited on page 81).

Ojala, T., M. Pietikainen, and T. Maenpää (2002).

“Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.7, pp. 971–987 (cited on page 135).

Owhadi, H. and C. Scovel (2015).

“Separability of reproducing kernel Hilbert spaces”. In: *arXiv preprint arXiv:1506.04288*. arXiv: 1506.04288 (cited on page 53).

Oza, N. C. (2005).

“Online bagging and boosting”. In: *IEEE International Conference on Systems, Man and Cybernetics*. Vol. 3. Institute of Electrical & Electronics Engineers (IEEE), pp. 2340–2345.

Page, E. S. (1954).

“Continuous Inspection Schemes”. In: *Biometrika* 41.1/2, p. 100 (cited on page 107).

Parzen, E. (1962).

“On estimation of a probability density function and mode”. In: *The annals of mathematical statistics*, pp. 1065–1076 (cited on page 29).

Pedersen, G. K. (2012).

Analysis now. Vol. 118. Springer Science & Business Media (cited on pages 51, 52).

Peypouquet, J. (2015).

Convex Optimization in Normed Spaces: Theory, Methods and Examples. Springer (cited on pages 92, 93, 95–97).

Pinelis, I. (1992).

“An approach to inequalities for the distributions of infinite-dimensional martingales”. In: *Probability in Banach Spaces, 8: Proceedings of the Eighth International Conference*. Springer, pp. 128–134 (cited on page 121).

Pinelis, I. (1994).

“Optimum bounds for the distributions of martingales in Banach spaces”. In: *The Annals of Probability*, pp. 1679–1706 (cited on page 121).

Pokrajac, D., A. Lazarevic, and L. J. Latecki (2007).

“Incremental local outlier detection for data streams”. In: *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 504–515 (cited on page 106).

Prastawa, M., E. Bullitt, S. Ho, and G. Gerig (2004).

“A brain tumor segmentation framework based on outlier detection”. In: *Medical image analysis* 8.3, pp. 275–283 (cited on page 5).

Rahimi, A. and B. Recht (2007).

“Random features for large-scale kernel machines”. In: *Advances in neural information processing systems*, pp. 1177–1184 (cited on pages 70, 71).

Rahimi, A. and B. Recht (2008).

“Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning”. In: *Advances in neural information processing systems*, pp. 1313–1320 (cited on page 70).

Ramaswamy, S., R. Rastogi, and K. Shim (2000).

“Efficient algorithms for mining outliers from large data sets”. In: *ACM SIGMOD Record*. Vol. 29. 2. ACM, pp. 427–438 (cited on page 33).

Rao, C. R. (1984).

“Convexity properties of entropy functions and analysis of diversity”. In: *Lecture Notes-Monograph Series*, pp. 68–77 (cited on page 132).

Rasmussen, C. E. and C. K. I. Williams (2005).

Gaussian Processes for Machine Learning. The MIT Press (cited on page 109).

Redner, R. A. and H. F. Walker (1984).

“Mixture densities, maximum likelihood and the EM algorithm”. In: *SIAM review* 26.2, pp. 195–239 (cited on page 28).

Reinsel, D. and J. Gantz (2012).

“The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East”. In: *Bigger Digital Shadows, and Biggest Growth in the Far East* 12 (cited on page 6).

Ridder, D. de, D. M. J. Tax, and R. P. W. Duin (1998).

“An experimental comparison of one-class classification methods”. In: *Proceedings of the 4th Annual Conference of the Advanced School for Computing and Imaging, Delft* (cited on page 34).

Robbins, H. and S. Monro (1951).

“A stochastic approximation method”. In: *The annals of mathematical statistics*, pp. 400–407 (cited on page 89).

Roberts, S. J. (2000).

“Extreme value statistics for novelty detection in biomedical data processing”. In: *Science, Measurement and Technology, IEE Proceedings*. Vol. 147. 6. IET, pp. 363–367 (cited on page 28).

Rosenblatt, F. (1958).

“The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6, p. 386 (cited on page 54).

Rosenblatt, M. (1956).

“Remarks on some nonparametric estimates of a density function”. In: *The Annals of Mathematical Statistics* 27.3, pp. 832–837 (cited on page 29).

Roth, V. (2004).

“Outlier detection with one-class kernel Fisher discriminants”. In: *Advances in Neural Information Processing Systems*, pp. 1169–1176 (cited on page 39).

Roussas, G. G. (1996).

“Exponential probability inequalities with some applications”. In: *Lecture Notes-Monograph Series*, pp. 303–319 (cited on page 118).

Roux, N. L., M. Schmidt, and F. R. Bach (2012).

“A stochastic gradient method with an exponential convergence rate for finite training sets”. In: *Advances in Neural Information Processing Systems*, pp. 2663–2671 (cited on pages 88, 98).

Russakovsky, O. et al. (2015).

“ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision*, pp. 1–42 (cited on page 127).

Schneider, M. (2016).

“Probability Inequalities for Kernel Embeddings in Sampling without Replacement”. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS 2016)*, pp. 66–74 (cited on pages 9, 117).

Schneider, M., W. Ertel, and G. Palm (2015a).

“Expected Similarity Estimation for Large-Scale Anomaly Detection”. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2015)*. IEEE, pp. 1–8 (cited on pages 9, 61).

Schneider, M., W. Ertel, and G. Palm (2015b).

“Kernel Feature Maps from Arbitrary Distance Metrics”. In: *KI 2015: Advances in Artificial Intelligence*. Ed. by S. Hölldobler, M. Krötzsch, R. Peñaloza, and S. Rudolph. Springer International Publishing, pp. 137–150 (cited on pages 9, 131).

- Schneider, M., W. Ertel, and G. Palm (2016).
 “Constant Time EXpected Similarity Estimation for Large-Scale Anomaly Detection”. In: *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*. IOS Press, pp. 12–20 (cited on pages 9, 87).
- Schneider, M., W. Ertel, and F. Ramos (2016a).
 “Expected Similarity Estimation for Large-Scale Batch and Streaming Anomaly Detection”. In: *Machine Learning* 105.3, pp. 305–333 (cited on pages 9, 61, 103).
- Schneider, M., W. Ertel, and F. Ramos (2016b).
 “Kernel Embeddings for Large-Scale Anomaly Detection”. In: *International Conference on Machine Learning (ICML 2016): Anomaly Detection Workshop* (cited on pages 9, 103).
- Schoenberg, I. J. (1938).
 “Metric spaces and positive definite functions”. In: *Transactions of the American Mathematical Society* 44.3, pp. 522–536 (cited on page 132).
- Schölkopf, B., J. C. Platt, et al. (2001).
 “Estimating the support of a high-dimensional distribution”. In: *Neural computation* 13.7, pp. 1443–1471 (cited on pages 39, 80).
- Schölkopf, B. and A. J. Smola (2001).
Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press (cited on pages 39, 40).
- Schölkopf, B., A. J. Smola, R. C. Williamson, and P. L. Bartlett (2000).
 “New support vector algorithms”. In: *Neural computation* 12.5, pp. 1207–1245 (cited on page 40).
- Scott, C. (2012).
 “The equivalence between the one-class and paired support vector machines for nonseparable data”. In: *Tech. Rep. CSPL-412*. Communications and Signal Processing Laboratory, University of Michigan Ann Arbor, MI (cited on page 40).
- Sejdinovic, D., B. Sriperumbudur, A. Gretton, and K. Fukumizu (2013).
 “Equivalence of distance-based and RKHS-based statistics in hypothesis testing”. In: *The Annals of Statistics* 41.5, pp. 2263–2291 (cited on page 63).
- Serfling, R. J. (1974).
 “Probability inequalities for the sum in sampling without replacement”. In: *The Annals of Statistics* 2, pp. 39–48 (cited on pages 117, 118, 120, 123, 128).
- Shalev-Shwartz, S. (2007).
 “Online Learning: Theory, Algorithms, and Applications”. PhD thesis. Hebrew University, p. 162.

- Shalev-Shwartz, S., Y. Singer, N. Srebro, and A. Cotter (2011).
 “Pegasos: Primal estimated sub-gradient solver for SVM”. In: *Mathematical Programming*. Vol. 127. 1, pp. 3–30 (cited on page 100).
- Shannon, C. E. (2001).
 “A mathematical theory of communication”. In: *ACM SIGMOBILE Mobile Computing and Communications Review* 5.1, pp. 3–55 (cited on page 38).
- Shawe-Taylor, J. and N. Cristianini (2003).
Estimating the moments of a random vector with applications. Tech. rep. Department of Computer Science, Royal Holloway, University of London (cited on page 77).
- Shawe-Taylor, J. and N. Cristianini (2004).
Kernel methods for pattern analysis. Cambridge university press (cited on pages 77, 126).
- Smola, A. J., A. Gretton, L. Song, and B. Schölkopf (2007).
 “A Hilbert space embedding for distributions”. In: *Algorithmic Learning Theory*. Springer, pp. 13–31 (cited on pages 62, 67, 127).
- Song, L., B. Boots, et al. (2010).
 “Hilbert space embeddings of hidden Markov models”. In: *Proceedings of the 27th international conference on machine learning*, pp. 991–998 (cited on page 62).
- Song, L., A. Gretton, et al. (2011).
 “Kernel belief propagation”. In: *International Conference on Artificial Intelligence and Statistics*, pp. 707–715 (cited on page 62).
- Song, L., J. Huang, A. J. Smola, and K. Fukumizu (2009).
 “Hilbert space embeddings of conditional distributions with applications to dynamical systems”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, pp. 961–968 (cited on page 62).
- Song, L., A. J. Smola, et al. (2012).
 “Feature selection via dependence maximization”. In: *The Journal of Machine Learning Research* 13.1, pp. 1393–1434 (cited on page 62).
- Spence, C., L. Parra, and P. Sajda (2001).
 “Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model”. In: *Mathematical Methods in Biomedical Image Analysis, 2001. MMBIA 2001. IEEE Workshop on*. IEEE, pp. 3–10 (cited on page 5).
- Sriperumbudur, B. K., A. Gretton, et al. (2008).
 “Injective Hilbert space embeddings of probability measures”. In: *Proceedings of the 21st Annual Conference on Learning Theory*, pp. 111–122 (cited on pages 62, 64).

- Sriperumbudur, B. K. (2010).
 “Reproducing kernel space embeddings and metrics on probability measures”. PhD thesis. University of California, San Diego, p. 192 (cited on pages 56, 64).
- Sriperumbudur, B. K. and A. Gretton (2010).
 “Hilbert space embeddings and metrics on probability measures”. In: *The Journal of Machine Learning Research* 11, pp. 1517–1561 (cited on page 66).
- Sriperumbudur, B. K. and Z. Szabó (2015).
 “Optimal rates for random Fourier features”. In: *Advances in Neural Information Processing Systems*, pp. 1144–1152 (cited on pages 71, 73).
- Steinwart, I. (2003).
 “Sparseness of support vector machines”. In: *The Journal of Machine Learning Research* 4, pp. 1071–1105 (cited on pages 6, 41).
- Steinwart, I. and A. Christmann (2008).
Support vector machines. Springer (cited on pages 53–55, 59, 63, 65, 127).
- Tan, S. C., K. M. Ting, and T. F. Liu (2011).
 “Fast anomaly detection for streaming data”. In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. Vol. 22. 1. Citeseer, p. 1511 (cited on page 42).
- Tax, D. M. J. (2001).
 “One-class classification”. PhD thesis. Technische Universiteit Delft (cited on page 29).
- Tax, D. M. J. and R. P. W. Duin (2004).
 “Support vector data description”. In: *Machine learning* 54.1, pp. 45–66 (cited on pages 39, 80).
- Torgerson, W. S. (1952).
 “Multidimensional scaling”. In: *Psychometrika* 17.4, pp. 401–419 (cited on pages 47, 64, 132).
- Tufte, E. R. and P. R. Graves-Morris (1983).
The visual display of quantitative information. Vol. 2. 9. Graphics press Cheshire, CT (cited on pages 26, 27).
- Tukey, J. W. (1975).
 “Mathematics and the picturing of data”. In: *Proceedings of the international congress of mathematicians*. Vol. 2, pp. 523–531 (cited on page 31).
- Vapnik, V. N. (1995).
The Nature of Statistical Learning Theory. Springer Science & Business Media (cited on page 17).

- Vedaldi, A. and A. Zisserman (2012).
 “Efficient Additive Kernels via Explicit Feature Maps”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.3, pp. 480–492 (cited on page 131).
- Viet Le, Q., T. Sarlos, and A. J. Smola (2014).
 “Fastfood: Approximate Kernel Expansions in Loglinear Time”. In: *ArXiv e-prints*. arXiv: 1408.3060 [cs.LG] (cited on page 132).
- Vondrick, C., A. Khosla, T. Malisiewicz, and A. Torralba (2013).
 “Hoggles: Visualizing object detection features”. In: *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, pp. 1–8 (cited on page 81).
- Wackernagel, H. (2013).
Multivariate geostatistics: an introduction with applications. Springer Science & Business Media (cited on page 109).
- Webb, G. I. (2000).
 “Multiboosting: A technique for combining boosting and wagging”. In: *Machine learning* 40.2, pp. 159–196 (cited on pages 21, 22).
- Widmer, G. and M. Kubat (1996).
 “Learning in the presence of concept drift and hidden contexts”. In: *Machine learning* 23.1, pp. 69–101 (cited on page 108).
- Wied, D. and R. Weißbach (2012).
 “Consistency of the kernel density estimator: A survey”. In: *Statistical Papers* 53.1, pp. 1–21 (cited on page 31).
- Williams, C. and M. Seeger (2001).
 “Using the Nyström method to speed up kernel machines”. In: *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*. EPFL-CONF-161322. MIT Press, pp. 682–688 (cited on page 73).
- Wu, K. et al. (2014).
 “RS-Forest: A Rapid Density Estimator for Streaming Anomaly Detection”. In: *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, pp. 600–609 (cited on page 43).
- Yu, H., J. Yang, and J. Han (2003).
 “Classifying large data sets using SVMs with hierarchical clusters”. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 306–315 (cited on page 81).
- Zeidler, E. (1995).
Applied Functional Analysis. Springer New York, p. 417 (cited on pages 91, 153).

Zhang, B. et al. (2011).

“A probabilistic fault detection approach: application to bearing fault detection”. In: *IEEE Transactions on Industrial Electronics* 58.5, pp. 2011–2018 (cited on page 5).

Zuo, Y. and R. Serfling (2000).

“General notions of statistical depth function”. In: *Annals of statistics* 28.2, pp. 461–482 (cited on page 31).

Index

- (τ, τ) -smooth, 121
- $\mathcal{L}^p$, 146
- α -strongly convex, 90
- ϵ -accurate, 88
- σ -finite measure, 146

- a. e., 146
- a. s., 146
- ABOD, 43
- absolute continuous, 147
- accuracy, 20
- almost everywhere, 146
- almost surely, 146
- anomaly, 3
- anomaly score, 16
- approximate feature map, 70
- area under the PR curve, 21
- area under the ROC curve, 21
- Aronszajn map, 53
- attack point, 111

- balanced accuracy, 114
- Banach space, 151
- batch learning, 18
- Bhattacharyya distance, 136
- bi-dual space, 152
- Bochner integrable, 65
- Borel measurable space, 145
- boundary, 150
- bounded, 151

- Cauchy sequence, 151
- causative integrity, 111
- centre of mass, 77
- characteristic, 64
- characteristic length-scale, 57
- classifier, 16
- closed set, 149
- closure, 150
- COD, 33
- coercive, 153
- compact, 150
- complete, 151
- concept drift, 19, 105
- conditional expectation, 148
- conditionally negative definite, 132
- conical sum, 58
- continuous, 151
- continuous linear functionals, 152
- continuous outlier detection, 33
- convergent, 151
- convex, 153
- critical difference, 23
- critical values, 23

- data depth, 31
- data stream, 104
- decision space, 15
- decision-rule, 16
- dense set, 150
- depth function, 32
- Dirac measure, 147
- distance-based outlier, 33
- distance-based outlier detection, 33
- distribution, 146
- divergent, 151
- dominated, 27
- drawn without replacement, 118
- dual map, 152
- dual pairing, 152
- dual space, 152

- effective domain, 153
- empirical kernel embedding, 67
- empirical measures, 148
- empirical risk function, 18
- equivalence class, 146
- evaluation functional, 52
- expectation, 148

- F-ABOD, 43
- F-score, 21
- false negatives, 16
- false positives, 16
- false-negative, 21
- false-positive, 20
- feature function, 54
- feature map, 53
- finite measure, 146
- finite subcover, 150

- Fréchet distance, 134
- Gaussian mixture model, 28
- Hellinger distance, 135
- Hilbert space, 51
- Hoeffding's inequality, 119
- holdout set, 114
- HSTA, 42
- hypothesis space, 16
- i.i.d., 104
- identifiable, 16
- iForest, 41
- IMQ kernel, 58
- inclusion, 150
- incremental learning, 19
- incremental learning, 104
- indicator function, 147
- inner product, 152
- inner product functional, 52
- inner product space, 152
- input space, 15
- integrable, 147
- integral operator, 55
- interior, 150
- inverse multiquadric, 58
- IQR, 27
- k-means, 44
- k-nearest neighbors, 33
- k-nearest neighbors distance, 32
- k-reachability distance, 34
- kernel, 53
- kernel Bayes' rule, 62
- kernel density estimator, 29
- kernel embedding, 62
- kernel embedding of distributions, 62
- kernel mean map, 62
- label space, 15
- Laplacian kernel, 57
- learning, 17
- Lebesgue density, 147
- Lebesgue integral, 147
- likelihood function, 27
- limit, 151
- limit inferior, 148
- limit superior, 148
- linear operator, 152
- local outlier factor, 35
- loss function, 17
- lossless, 104
- lower semicontinuous, 95
- martingale, 121
- Matérn kernel, 57
- maximum likelihood, 27
- maximum likelihood estimator, 27
- maximum mean discrepancy, 63
- McDiarmid's inequality, 126
- MDS, 47
- measurable, 145
- measurable space, 145
- measure space, 146
- Mercer representation, 55
- metric, 150
- metric space, 150
- minimizing sequence, 90
- mixture model, 28
- MMD, 63
- multi-class dataset, 80
- multidimensional scaling, 47
- nearest neighbor data description, 34
- nearest neighbors, 33
- nearest neighbors distance, 33
- neighborhood, 150
- neighborhood density, 35
- non-expanding, 89
- non-parametric, 16
- non-stationary, 19
- norm, 151
- normal score, 16
- normed space, 151
- null set, 146
- Nyström feature map, 74
- offline learning, 18
- one-class support vector machine, 6, 39
- online, 19
- online centroid anomaly detection, 111
- open ball, 150
- open cover, 150
- open set, 149
- Ornstein-Uhlenbeck kernel, 57
- outlier selection, 14
- output space, 15
- parameter space, 16
- parameterized, 16
- parametric hypothesis space, 16
- Parzen density estimator, 29
- Pettis integral, 66
- Pinelis' inequality, 121
- poisoning, 111
- positive definite, 52
- precision, 21
- precision-recall, 21
- prediction space, 15
- predictor, 16
- prequential set, 114
- probability measure, 146
- probability space, 146

- projection, 89
- proper, 95
- Rademacher average, 127
- Rademacher complexity, 127
- Rademacher sequence, 127
- Radon-Nikodym derivative, 147
- random element, 146
- random kitchen sinks, 70
- random variable, 146
- recall, 21
- reflexive, 95
- regularizer, 98
- reproducing kernel, 53
- reproducing kernel Hilbert space, 52
- reproducing property, 53
- risk function, 17
- RKHS, 52
- RKS, 70
- sample space, 17
- sample without replacement, 118
- sampling without replacement, 118
- SE kernel, 56
- semi-supervised anomaly detection, 14
- separable space, 150
- sequentially compact, 91
- Serfling's Inequality, 120
- sinc kernel, 57
- slack variables, 39
- spatio-temporal kernel, 109
- square summable sequences, 51
- squared exponential kernel, 56
- statistical depth function, 32
- statistical model, 16
- stochastic outlier selection, 37
- stochastic process, 148
- stochastic subgradient, 89
- STORM, 33
- stream outlier miner, 33
- streaming anomaly detection, 19, 20
- streaming half-space-trees, 42
- strictly positive definite, 53
- strong integrable, 65
- subdifferential, 89
- subgradient, 89
- supervised anomaly detection, 13
- support of measure, 146
- support vector data description, 39
- support vector machine, 39
- tensor product, 58
- test set, 20
- topological space, 149
- topology, 149
- totally bounded, 151
- training, 17
- training set, 17
- translation invariant, 56
- true-negative, 21
- true-positive, 20
- universal, 65
- unsupervised anomaly detection, 14
- weak convergence, 91
- weak integral, 66
- weakly closed, 153
- weakly integrable, 66
- weakly sequentially continuous, 153
- weakly sequentially lower semicontinuous, 95

List of Figures

2.1	Classification vs. anomaly detection	14
2.2	Source, input & label spaces	15
2.3	Source, input, prediction & label space	16
2.4	Sample & hypothesis space	17
2.5	Taxonomy online learning	19
2.6	The four possible classification outcomes	21
2.7	Receiver operating characteristic curve	21
2.8	Nemenyi test	23
3.1	Distribution of normal data	26
3.2	Normal distribution	26
3.3	A box plot	27
3.4	GMM density estimation	29
3.5	KDE density estimation	30
3.6	The histogram estimator	30
3.7	Tukey's half-space depth	32
3.8	Nearest neighbors distance	32
3.9	Sliding window	33
3.10	NNDD density estimation	34
3.11	LOF density estimation	35
3.12	LOF algorithm	36
3.13	Stochastic outlier selection	38
3.14	OC-SVM schematic	40
3.15	OC-SVM example	41
3.16	iForest example	42
3.17	Angle-based outlier detection	44
3.18	k-means anomaly detection	45
3.19	Visualization of the feature map embedding	47
3.20	Empirical EXPOSE estimator	48
3.21	Runtime vs. rank (batch)	49
4.1	Integral operator factorization	55
4.2	Mercer representation	56
4.3	Kernel Functions	57
5.1	Kernel embedding	64
5.2	EXPOSE estimator	68
5.3	Distributed EXPOSE	69
5.4	Random kitchen sinks	72
5.5	Centre of mass	78

5.6	Point Difficulty	79
5.7	Anomaly Frequency	79
5.8	Clustered Data	79
5.9	Runtime vs. rank (batch)	82
5.10	Log runtime vs. rank (batch)	83
5.11	Critical difference diagram (batch)	83
6.1	Gradient descent	88
6.2	Subgradients	89
6.3	Evaluation of stochastic EXPose	101
7.1	Concept drift types	105
7.2	Incremental learning & concept drift	106
7.3	Sliding window	108
7.4	Gradual forgetting	108
7.5	Periodic kernel & reoccurring drift	110
7.6	Streaming SVHN experiment	115
7.7	Sigmoid function	115
7.8	Nemenyi test stream	116
8.1	Comparison of Hoeffding's and Serfling's inequality	120
8.2	A martingale	121
8.3	Deviation between $\mu[\mathbb{P}_n]$ and $\mu[\mathbb{P}_N]$	129
9.1	Pseudo line projections	133
9.2	Distance-based feature maps	133
9.3	Hurricane dataset	134
9.4	DBFM with Fréchet distance	134
9.5	8 scene categories dataset	135
9.6	DBFM with Hellinger's distance	135
9.7	DBFM with the Bhattacharyya distance	136
A.1	Bayes' pool table	145
B.1	Topological Spaces	149

List of Tables

2.1	Critical values for the Nemenyi and Bonferroni-Dunn tests	23
5.1	Batch dataset properties	81
5.2	Batch anomaly detection (AUC)	84
5.3	Batch anomaly detection (time)	85
6.1	Problem sizes, complexity and operations	88
6.2	Experimental comparison of stochastic optimization	100
7.1	Streaming datasets and their properties	115
7.2	Streaming anomaly detection performances	116
7.3	Holm's test (online)	116
8.1	Experimental setup for concentration inequalities	128

List of Algorithms

5.1	Random kitchen sinks	71
5.2	Nyström feature map	74
6.3	Stochastic optimization	97