University College London
Centre for Computational Statistics and Machine Learning

M.Sc. Machine Learning

Adaptive Large-Scale Kernel Two-Sample Testing

Heiko Strathmann

Supervisors:
Arthur Gretton
Massimiliano Pontil

September 2012

Acknowledgements

Most results of this report are based on and included in [Gretton et al., 2012c], which was accepted at NIPS 2012. I would like to express my special thanks to the co-authors, in particular to Arthur Gretton, Bharath Sriperumbudur, Dino Sejdinovic, and Massimiliano Pontil for guidance, discussions and making this work possible in general. In particular, thanks to Arthur and Massimiliano for taking the time to supervise my dissertation.

I am grateful to the *Studienstiftung des deutschen Volkes* for supporting my studies and for making my stay at the UCL possible.

Thank you, parents. This work is dedicated to you.

Thank you, Debora.

Abstract

This Master's dissertation deals with the problem of *two-sample testing*: given sets of samples from two probability distributions p and q respectively, this is the problem of deciding whether $p \neq q$ with a high confidence. The problem is approached with the *Maximum Mean Discrepancy (MMD)*, a metric on mean embeddings of probability distributions in *reproducing kernel Hilbert spaces (RKHS)* that can be used to construct two-sample tests, [Gretton et al., 2012a]. It can be seen as a kernel-based generalisation of the *t-test*. In particular, large-scale test versions and the problem of choosing a kernel for these are investigated. As is the case for any kernel-based method, kernel choice is crucial for performance.

There are three main contributions to this field: the first one is providing a detailed description of a *large-scale* test that can be computed in an *on-line* way, as briefly introduced in [Gretton et al., 2012a]. This work argues and shows in experiments that the test outperforms state-of-the-art quadratic time methods when problems need too many samples to fit into computer memory. The same is true if there is infinite data but only limited computation time available.

Second, a novel criterion for kernel selection for the above test that operates in such way that the test's type II error is minimised. It is argued that the criterion is well behaved for extreme kernel choices and that it yields *optimal* choice. Experimental results show that it performs better than, or at least comparable to, state-of-the-art methods in practice.

Third, a generalisation of kernel two-sample tests to use a *multiple kernel learning (MKL-)*style non-negative linear combinations of a finite number of baseline kernels is proposed. Using convex optimisation, kernel weights are selected in-line with the above described new criterion. Along with a formal description, experiments are devised to motivate usefulness of the approach.

In addition, a set of new benchmarks for two-sample testing is described. These synthetic datasets are difficult in the way that distinguishing characteristics are hidden at a different length-scale than in the overall data.

Theoretical and experimental results show that the methods described in this work allow to perform two-sample testing in two yet non-processable domains: large-scale data and MKL-style kernel selection. In addition they outperform state-of-the-art methods in many existing cases, in particular on datasets which represent hard problems for two-sample testing or when infinite data is available.

Contents

1. Introduction

This section gives an introduction for this work. Section 1.1 gives a brief and non-formal overview of the subject of analysis in this work: kernel based two-sample-testing. Single aspects are motivated, i.e. why two-sample testing is useful, why kernels give an advantage, and why there is a problem of selecting these. A description of state-of-the-art methods and their possible drawbacks is given. Afterwards, approaches for overcoming these drawbacks and emerging questions that are answered in this work are formulated. Finally, section 1.2 briefly outlines contents of each chapter and provides a guide how to read this work.

1.1. Problem Description & Outline

This work deals with the problem of statistical *two-sample testing*: distinguishing two probability distributions on the base of drawn samples.

> Given two probability distributions p and q and two sets of i.i.d.[1] samples $X := \{x_1, ..., x_m\}$ and $Y := \{y_1, ..., y_n\}$ drawn from p and q respectively, is it possible to decide $p \neq q$ with a high confidence?

In particular, two-sample tests are constructed using the so-called *Maximum Mean Discrepancy*, [Gretton et al., 2012a], which can be seen as a generalisation of a t-test in *reproducing kernel Hilbert spaces (RKHS)*: it is the distance of the *mean-embeddings* of two distributions in such spaces. Since the method is based on *kernels*, the problem of choosing those appropriately arises.

> How to select a kernel in such way that a two-sample test on the base of that kernel has a low probability of giving a wrong answer?

This work analyses methods that *adapt* to data in the sense of selecting kernels whose usage leads to less probable wrong test answers. The problem is discussed in context of *large-scale* two-sample tests, i.e. tests whose computational costs are linear in terms of the number of used samples.

1.1.1. Importance of Two-Sample Testing

The importance of two-sample testing can most easily be seen when looking at practical examples. In machine learning, usually the assumption is that data is i.i.d. sampled, i.e.

[1] i.i.d. stand for *independent and identically distributed*, which means that samples are independently sampled from identical random variables.

all samples come from the *same* distribution(s). If this condition is not met, it may have a negative impact on results and therefore has to be considered when designing algorithms and experiments. Two-sample tests address exactly this question. The following illustrative cases are taken from a list in [Sugiyama et al., 2011, Section 1.1].

When experimental data is collected during multiple processes, it is often desirable to treat all samples as one big set instead of processing them individually. This for example happens when biological data is collected from different laboratories. [Borgwardt et al., 2006, Section 1] motivate that a large dataset constructed by joining multiple single ones might support conclusions which are impossible to infer using these small parts. Two-sample testing allows one to decide whether individual samples may be merged or should be treated individually.

When algorithms work on non-stationary processes, two-sample testing helps in deciding whether some kind of data adoption scheme has to be carried out between sets of samples. This might, for example occur, in different sessions of brain-computer interface experiments [Sugiyama et al., 2007]. Two-sample testing can be used to confirm whether underlying distributions have significant differences. When they have not, data-intensive non-stationary adaptation, as in the above paper, can be avoided. This increases stability.

The idea of multi-task learning, see [Caruana, 1997], is to solve multiple related learning tasks jointly. This might give better solutions than solving each task individually. However, when individual tasks are not related at all, multi-task learning usually leads to worse results. Two-sample testing allows to decide whether two tasks are related by looking at their datasets.

1.1.2. Usefulness of Kernels

Kernel functions are a widely used tool for generalising machine learning algorithms into high-dimensional or even infinite spaces. Many linear algorithms may be "kernelised", which means that computations are implicitly performed in such high-dimensional spaces using a kernel. The elegance and usefulness here is the fact that these spaces are not touched explicitly, i.e. neither elements nor operations in them are actually stored or performed. Despite this, results take advantage of the richness of the underlying spaces.

Famous examples include linear ridge regression (see [Shawe-Taylor and Cristianini, 2004, Section 2]) which is fitting data with a linear function and a complexity constraint on the descriptor. Its kernel counterpart implicitly fits a linear function in a high-dimensional space, however, the resulting predictor corresponds to a non-linear function in the original space – more complex data can be described. Similar approaches have been applied to principal component analysis [Schölkopf and Smola, 2001, Section 14.2], support vector machines [Shawe-Taylor and Cristianini, 2000], and Bayesian linear regression [Rasmussen and Williams, 2006].

Another advantage of kernel methods is that the input domain of data does not have to be numerical. There exist valid kernels on strings, graphs, trees, and other structured data. Therefore, numerical algorithms such as SVM-classification can be performed on non-numerical data directly.

Two-sample testing also takes advantages from using kernels. Classical methods such as the t-test do perform well in high dimensional spaces and are restricted to numerical data – they cannot for example not be applied to text data directly. A recently proposed method for kernel based two-sample testing is based around the maximum mean discrepancy (MMD), [Gretton et al., 2012a]. The "kernelisation", i.e. non-linear generalisation to arbitrarily complex spaces as described above, happens here via embedding the mean of probability distributions into such spaces and then computing the distance of these embeddings. This ultimately results in a metric on probability distributions.

Using this metric, two-sample tests may be constructed in such a way that, in theory and using infinite samples, *any* two probability distributions can be distinguished. Another advantage is again that two-sample tests may be performed on non-numerical data such as text directly. The method has been successfully used with neurological data [Gretton et al., 2012a] and graphs [Borgwardt et al., 2006]. It outperforms existing methods in various ways. Thus, it is one of the state-of-the-art methods for two-sample testing on complex data.

1.1.3. Importance of Kernel Selection

Regardless of details of any kernel algorithm, performance is dependent on the particular utilised kernel. This is a drawback of using kernels since there is no general method for selecting these effectively. As wrong choices lead to weak results, kernel *selection* is paramount. Optimally, one wants to use the one kernel that gives best possible results. This involves different types of choices:

1. The domain of the kernel, i.e. the set it is defined on or the *format* that its input has. This choice is usually dependent on used data. Since there exist kernels for different kinds of data (numerical, strings, graphs) this highlights one advantage of kernel methods in general: their domain is flexible. However, this dissertation utilises numerical data only and uses the most common kernels for such – the Gaussian kernel, [Hsu et al., 2003].

2. The next choice involves selecting more fine-grained attributes or parameters of kernel families. An example is the bandwidth of a Gaussian kernel or the sliding window size of a string kernel. Selecting these parameters is a more challenging problem and usually cannot be done a-priori since they mainly depend on used data. Popular approaches for dealing with parameter selection include cross-validation in the context of SVM-classification [Hsu et al., 2003] or gradient based approaches on error-functions in the context of Gaussian process regression [Rasmussen and Williams, 2006, Chapter 5]. One main focus of this work is how to select such kernel parameters for two-sample-testing.

3. Since combinations (products/sums) of kernels are valid kernels, choosing appropriate weights for these combinations is also a form of kernel selection. Combining kernels becomes interesting when two different domains should be included in a method or different dimensions should have different parameters. This approach

has intensively studied for classification problems under the term *multiple kernel learning (MKL)* (see e.g. [Argyriou et al., 2006], Rakotomamonjy et al. [2008], or Sonnenburg et al. [2006]). Usually, a finite number of base kernels is chosen a-priori and then used to construct a non-negative linear combination. This work describes how to select weights of such combinations in the context of two-sample testing.

1.1.4. Main Literature, State of the Art Methods and Their Drawbacks

Since kernel two-sample testing is a relatively new method, the problem of selecting kernels has not yet been fully explored. In fact, most methods in literature use a heuristic method to determine the length scale of a Gaussian kernel. See [Gretton et al., 2012a, Appendix C]. This method chooses the bandwidth parameter in such way that it reflects the dominant scaling in used data – the median. It is a reasonable choice at first glance, as experiments in literature suggest. However, it is shown in this work that the method fails whenever distinguishing characteristics of distributions are hidden at different length-scales. The median in this sense is not adaptive at all: in fact, the method is almost equal to normalising data and then using a unit sized kernel. Another drawback is its unsuitability for selecting weights of kernel combinations.

An adaptive method has been proposed in [Sriperumbudur et al., 2009]. It is a heuristic based on maximising the used test statistic. While being a first step towards more advanced methods for kernel selection, it remains suboptimal as will be seen later on. Moreover, it has not yet been examined in context of combined kernels.

Another method that was suggested recently is related to the above method: minimising expected risk of a MMD-related binary classifier attached with a linear loss as described in [Sugiyama et al., 2011]. The approach uses cross-validation in order to minimise mentioned risk while preventing overfitting. It is justified by the fact that the MMD can be interpreted as such a binary classifier with linear loss. Minimising the latter corresponds to maximising the MMD as was shown in [Sriperumbudur et al., 2009]. Differences between the approaches include a protection from overfitting which is inherited from cross-validation. In addition, cross-validation potentially increases numerical stability since multiple folds are averaged. However, minimising the linear loss has quadratic time costs for every evaluated kernel. In context of linear time two-sample tests, the results would have to be much better to justify such costs. Thus far, the method has only been compared to the median approach described above. This work will fill the gap and compare it against [Sriperumbudur et al., 2009]. Note that the method is not suitable for selecting weights of combined kernels since every weight combination would have to be tried; this is computationally infeasible.

To summarise, state-of-the-art methods have neither been properly compared against each other, nor does any of the methods offer any guarantees regarding performance of resulting two-sample tests. In addition, none of the methods has been investigated in context of weight selection for combined kernels.

1.1.5. An Approach and Emerging Questions

This work fills previously mentioned gaps by describing and evaluating the following points.

- New methods for kernel selection that offer theoretical guarantees. These are experimentally compared against existing ones. Existing ones will be compared in a unified way.
- Generalisation of these new methods to weight selection of combined kernels. This approach will be motivated and experimentally compared against using single kernels.
- Linear-time large-Scale two-sample testing. Such on-line tests will be motivated and evaluated as their quadratic time counterparts in cases where available data is infinite.

New Criteria for Kernel Selection In order to perform kernel selection, a criterion that acts as a performance measure is needed. In other words, for every possible kernel choice it must be possible to compute a number that reflects accuracy or performance of underlying methods. This work proposes two novel such criteria in context of linear time two-sample-testing with the maximum mean discrepancy. Natural questions to new criteria are:

1. Do they offer statistical guarantees?
2. Are they better than existing approaches?

The term *better* will be formalised later. To answer these question, new methods are investigated theoretically. In addition they will be empirically tested against state-of-the-art approaches in several different contexts.

Combined Kernel for Two-Sample Tests Above questions will initially be examined in context of selecting single kernels' parameters. In addition, one of the new criteria is also useful for the problem of selecting weights of combined kernels. Since kernel methods for statistical testing are a relatively new field, such combinations have not yet been examined. Questions that arise include:

1. Is it possible to efficiently select kernel weights using the novel criterion?
2. In what kind of situations does usage of kernel combinations give an advantage over using single kernels?

In order to answer these questions, an optimisation problem whose solution corresponds to optimal kernel weights will be derived. It will be tested in a context where selecting weights corresponds to selecting appropriate features in the sense that they are important for two-sample testing. Examples where single kernels give bad results will be constructed.

1.1.6. Large Scale Two-Sample Tests

There exist two different ways of constructing a two-sample test based on the maximum mean discrepancy. Both are described in [Gretton et al., 2012a, Section 2 & 6]. The natural choice has quadratic time costs and de-facto has to store all used data in the form of kernel matrices in memory. It can be used for cases where data is finite and one want to get the best possible result using all data. All mentioned kernel selection methods in literature were designed for this quadratic time test. However, there also exists a linear time statistic which does *not* have store all used data in memory. It is more suited for *streaming data* cases, where huge amounts of data (that cannot be stored in memory) have to be processed. Using this statistic, this work will describe how to construct a linear time MMD-based two-sample test that is able to deal with previously un-processable amounts of data. It will be argued that the approach is easily parallelisable and therefore scales up with growing cluster computers.

The linear time test has appealing properties that can be exploited to construct a new criterion for *optimal* kernel choice in the sense that the test's error is minimised (details follow). In addition, in [Gretton et al., 2012a, 6], it is briefly mentioned that the test performs better than its quadratic time counterpart when computational time is fixed and data is infinite. This property will be investigated in a greater detail in this work. The following questions arise.

1. What is a problem hard enough that it is not solvable using the quadratic test due to memory constraints? How does the linear test perform here?

2. Given a fixed computational time and unlimited data, which test reaches better performance?

Both tests will be compared against each other in context of the described case of infinite available data.

1.1.7. Providing an Overview

Another goal of this work is of a non-methodological nature: providing an overview of the field of kernel two-sample-testing using the maximum mean discrepancy. Over the past years, since the theory was developed, a significant amount of papers has been published in different sources. This work tries to collect most important ideas and concepts in order to provide a unified guideline for people who are willing to use the contained methods.

Since theoretical fundamentals around reproducing kernel Hilbert spaces might not be widely known, they are given at a greater detail than necessary for performing experiments – including selected proofs. All this is done with the hope that this document will be useful for future students working on the field.

1.2. Chapter Overview

This section provides short descriptions of all chapters to provide an overview. In principal, the order of reading should be chronologically. Technical parts of some chapter have

been put into the appendix. Each chapter starts with a brief overview of its contents along with references to individual sections. In addition, referenced literature is listed as well as the author's original contributions to the chapter's topic. Each chapter ends with a brief summary of most important points.

Chapter 2 – Theoretical Preliminaries: Kernel Theory for Two-Sample Testing describes theoretical fundamentals that are needed in order to understand kernel two-sample testing based on the MMD. Statistical hypothesis tests are introduced, basic theory regarding RKHS and kernels is given along with a description of the mainly used Gaussian kernel, and the MMD is introduced in context of RKHS. Finally, utilised estimates of the MMD that will be used to construct two-sample tests are described along with their distributions.

While this introduction is kept short, understanding where the MMD comes from requires a significant amount of theoretical concepts around RKHS. In order to build an intuition for these concepts, selected proofs are given where illustrative. However, statistical testing theory, MMD definition, and its estimates are most fundamental for understanding experimental results of this work.

Chapter 3 – Methods for Test Construction & Kernel Choice starts by describing how to construct two-sample tests using estimates of the MMD. A general method is given along with specialised ones for linear and quadratic time MMD estimate. In particular, an on-line method for computing large-scale tests using the linear time MMD estimate is described.

The chapter continues by introducing existing methods for kernel selection for MMD-based two-sample tests and describes downsides of these. Finally, a novel criterion for kernel selection for linear time tests is suggested along with theoretical arguments why it yields optimal kernel choice.

Chapter 4 – Combined Kernels for Two-Sample Testing begins with a brief formal introduction on combined kernels. The concept is then applied to the MMD for which an expression in terms of kernel weights is derived. Such expressions are also provided for linear time estimates of the MMD. Finally, a method for selecting optimal kernel weights w.r.t. the described new criterion is described. The latter is based on convex optimisation. A similar approach for the state-of-the-art method for kernel selection is outlined at last.

Chapter 5 – Experimental Results contains all experimental evaluations of this work. It starts by describing all used datasets and stating why these are used and what to expect from results on these. A small overview of most important points is followed by a detailed description. Next, convergence of the linear time threshold construction that is largely used in this work is examined in order to justify its usage. Then, experiments on and analysis of all datasets, paired with all described kernel selection methods, follow.

Finally, experiments that motivate using a linear time test instead of a quadratic one are described.

Chapter 6 – Outlook points out directions for further research. The main part is a suggestion for a criterion for kernel selection for quadratic time MMD which is similar to the one for the linear time MMD. Necessary expressions of null and alternative distributions are stated, as well as quadratic time estimates for them. Initial experiments are reported. Another point is a cross-validation style averaging of kernel selection criteria in order to make them more robust.

Chapter 7 – Summary finally summarises main results of this work.

Appendix A – Proofs Omitted in the Main Text contains technical mathematical derivations that are taken off the main texts for reasons of readability.

Appendix B – Open-Source Implementation: SHOGUN briefly introduces a public available implementation of most methods of this work.

2. Theoretical Preliminaries: Kernel Theory for Two-Sample Testing

Chapter Overview This chapter provides theoretical fundamentals necessary to understand statistical hypothesis testing. Kernel based tests using the Maximum Mean Discrepancy (MMD), and a list of MMD estimates which are the main tool utilised in this work are established. Notation of the MMD requires a significant amount of theory around reproducing kernel Hilbert spaces (RKHS). In order to provide some intuition on formal aspects of RKHS and MMD, a selection of important results to establish these are given. Selected proofs are sketched where instructive.

Section 2.1 defines all necessary terms in context of two-sample testing and points out how tests can be constructed using p-values of null distributions or thresholds. Section 2.2 defines the MMD on arbitrary classes of functions, not yet specialised on RKHS. The latter are introduced in section 2.3 which contains basic notions of positive definite kernels, RKHS, and the Gaussian kernel. In section 2.4, the MMD is established in the context of kernels by introducing mean embeddings of probability distributions into RKHS. This leads to the important result that the MMD can distinguish any two probability distributions. Section 2.4.3 introduces MMD estimates as mainly tools utilised of this work. At last, section 2.5 provides a summary of most important contents of this chapter.

Sections 2.1, 2.2, and 2.4.3 are the most important as these provide required groundwork to understand experimental results. The other sections in this chapter are less important but will provide better understanding of how and why MMD works.

Literature & Contributions This chapter mainly summarises results from books and a few papers. It does not contain any original work except for a unified presentation of concepts. For statistical testing, the books [Berger and Casella, 2002] and [Dudley, 2002] are the main references – although some definitions are taken from the [Gretton et al., 2012a]. Kernel theory is mainly based on the books [Shawe-Taylor and Cristianini, 2004] and [Schölkopf, 1997]. RKHS theory comes from [Steinwart and Christmann, 2008] and [Berlinet and Thomas-Agnan, 2004]. Theory on functional analysis can be found in [Reed and Simon, 1980]. MMD notion is mainly taken from [Gretton et al., 2012a] (in particular MMD estimates). Some other MMD related results are from [Sriperumbudur et al., 2010] and [Borgwardt et al., 2006].

2.1. Statistical Hypothesis Testing

Statistical hypothesis testing deals with the problem of falsifying a previously defined hypothesis. In this work, a special instance called the *two-sample-test* is of interest. Informally, it tries to answer the following problem: given two sets of observations, is it possible to tell whether these come from different sources? A more formal problem definition is given now.

Problem 1. *Let x and y be random variables defined on a topological space $\mathcal{X}$, with respective Borel probability measures p and q. Given observations $X := \{x_1, ..., x_m\}$ and $Y := \{y_1, ..., y_n\}$, independently and identically distributed (i.i.d.) from p and q, respectively, can one decide whether $p \neq q$?*

In practice, one addresses the problem whether there can be found a *statistically significant* difference between the distributions p and q. To this end, some important definitions from [Berger and Casella, 2002] are given to formalise the problem.

A *hypothesis* is a statement about a population parameter. Based on samples from the population, the goal of a hypothesis test is to decide which one of two complementary hypothesises is true. The two complementary hypothesises are called *null hypothesis* and *alternative hypothesis*, denoted by H_0 and H_A respectively.

Using these terms, it is possible to define a test.

Definition 1. *(Two-sample test) Given i.i.d. samples $X \sim p$ of size m and $Y \sim q$ of size n, a statistical test $\mathcal{T} : \mathcal{X}^m \times \mathcal{X}^n \to \{0, 1\}$ distinguishes between the null hypothesis $H_0 : p = q$ and the alternative hypothesis $H_A : p \neq q$. It returns one on rejection of H_0.*

This is done via comparing a *test-statistic*, which is a number that is computed using the samples X, Y, and to compare it against a *threshold*. If the statistic is larger than the threshold, the test rejects H_0 (and accepts H_A).[1]

Since the test statistic is computed on a finite number of samples, it is possible that the test gives an incorrect answer. There are two types of errors.

Definition 2. *(Different error types)*

- *A* type I error *is made when $H_0 : p = q$ is wrongly rejected. That is, the tests says that the samples are from different distributions when they are not.*
- *A* type II error *is made when $H_0 : p = q$ is wrongly accepted. That is, the tests says that the samples are from same distributions when they not.*

It is intuitively clear that a good test has a low type II error since one is usually interested in finding differences between samples. A test that always rejects $H_0 : p = q$ has zero type II error – however, it may have a large type I error. Therefore, one needs to control type I error while trying to keep type II error small. In order to do this, one

[1] On a philosophical level, there is a difference between *rejecting* H_0 and *accepting* H_A. However, following [Berger and Casella, 2002], this distinction is not made. In context of this work, it is not important since the main approach is to minimise type II error for given type I error.

defines a *level* α of the test which is an upper bound on the probability for a type I error. This is a predefined parameter of the two-sample test which determines the choice of the mentioned test threshold to which the test statistic is compared. A test is said to be *consistent* if for a given fixed upper bound α one the type I error, it reaches zero type II error in the infinite sample limit. Statistical significance can then be expressed in terms of the level α.

2.1.1. Null Distribution, Threshold, and P-value

Any test statistic is dependent on finite sample data. Therefore, for different data, they will lead to different values. In fact, computing an estimate is equivalent to sampling from a distribution: the distribution of test statistics, given that the null hypothesis $H_0 : p = q$ is true, is called the *null distribution*. When $H_A : p \neq q$ is true, samples come from the *alternative distribution*.

The null distribution is an important tool to build a two-sample test: when it is known, it is easy to see where the actual sample lies. If the sample lies above a high, say 95% quantile, it is very unlikely (maximum $\alpha = 0.05$) that the sample in fact comes from the null distribution. It is clear how a test can be constructed: the null distribution has to be somehow approximated, a test statistic has to be computed on base of the provided data. Then, the null hypothesis is rejected if the statistic sample lies above a certain threshold. Intuitively, this threshold defines an upper bound on the probability that the null hypothesis $H_0 : p = q$ is rejected when its true – the type I error as described in definition 2. Figure 2.1 illustrates thresholds and different error types for the case when both null and alternative distribution are normal.

Closely related is the *p-value* of the test statistic: it is the $(1 - p)$-quantile where the statistic lies in the null distribution. It therefore also defines an upper bound on the type I error. Since it is non-binary, it is slightly more informative than just a threshold for a fixed p-value. However, there is no difference between defining a threshold for a given p-value and rejecting if the statistic's p-value is smaller than a one.

2.2. The Maximum Mean Discrepancy

The test statistic used in this work will be an empirical estimate of a metric called *maximum mean discrepancy*. This section describes where this metric comes from.

In order to solve problem 1, it is desirable to have a criterion than takes a unique value if and only if $p = q$. In this work, the expected value of functions will be used as a criterion. If there is no ambiguity, the following notation for expected values w.r.t. distributions p and q is used: $\mathbf{E}_x[f(x)] := \mathbf{E}_{x\sim p}[f(x)]$ and $\mathbf{E}_y[f(y)] := \mathbf{E}_{y\sim p}[f(y)]$. Lemma 1, relates expected values of bounded continuous functions and equality of probability distributions in the following way.

Lemma 1. *Let $(\mathcal{X}, d)$ be a metric space, and let p and q be two Borel probability measures defined on $\mathcal{X}$. Then $p = q$ if and only if $\boldsymbol{E}_x[f(x)] = \boldsymbol{E}_y[f(y)]$ for all $f \in \mathcal{C}(\mathcal{X})$, where $\mathcal{C}(\mathcal{X})$ is the space of bounded continuous functions on $\mathcal{X}$.*

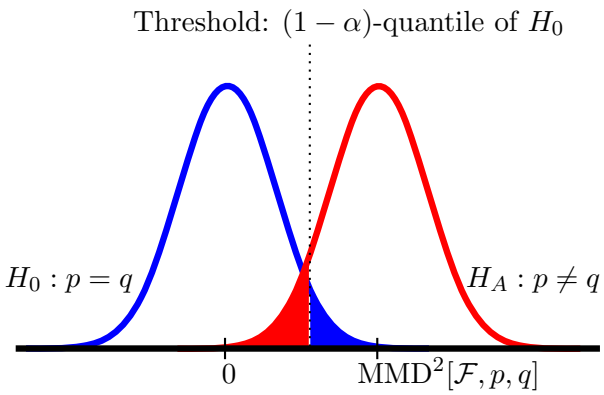


Figure 2.1.: Depiction of two-sample testing. Statistics that are generated by $H_0 : p = q$ come from the null distribution (red). Those generated by $H_A : p \neq q$ come from the alternative distribution (blue). The $(1-\alpha)$-quantile of the null distribution defines a threshold (dotted line) for testing: if a statistic lies above that threshold, the probability that it was generated by $H_0 : p = q$ is less then α. A type I error occurs when $H_0 : p = q$ is wrongly rejected: this happens when samples from the null distribution lie above the threshold (blue area). A type II error occurs when $H_0 : p = q$ is wrongly accepted: this happens when a sample from the alternative distribution lies below the threshold (red area).

The space $\mathcal{C}(\mathcal{X})$ in principle allows to distinguish $p = q$, however, in practice it is too rich when using only finite samples – too many samples would be needed. The *maximum mean discrepancy* restricts the space of functions to choose from and corresponds to the largest distance of function means in that space.

Definition 3. *(Maximum Mean Discrepancy, MMD) Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$, and p, q, x, y, X, Y be defined as above. Define the maximum mean discrepancy as*

$$\mathrm{MMD}[\mathcal{F}, p, q] := \sup_{f \in \mathcal{F}} \left(\boldsymbol{E}_x[f(x)] - \boldsymbol{E}_y[f(y)] \right) \tag{2.1}$$

In order to compute a test statistic, a simple estimator of the MMD can straightforwardly be obtained by replacing the population expressions by empirical expectations based on the samples X and Y

$$\mathrm{MMD}_b[\mathcal{F}, p, q] := \sup_{f \in \mathcal{F}} \left(\frac{1}{m} \sum_{i=1}^{m} f(x_i) - \frac{1}{n} \sum_{i=1}^{n} f(y_i) \right)$$

Clearly, a simpler class $\mathcal{F}$ will require less data to construct a test. On the other hand, $\mathcal{F}$ must be rich enough to uniquely identify $p = q$. The choice of $\mathcal{F}$ in this work will be a unit ball in a *reproducing kernel Hilbert space*.

2.3. Kernel theory

The maximum mean discrepancy in expression 2.1 will be used along with functions that are elements of reproducing kernel Hilbert spaces. This will allow to compute estimates using kernel functions. In this section, basic definitions and results that are needed to establish and understand this goal are provided. All these results can be found in different sources in literature. This section tries to comprehend all necessary results in one go in context of two sample testing with the maximum mean discrepancy. The section may be skipped on a first reading since experimental results may be understood without it.

2.3.1. Feature Maps and Kernels

One central definition of this work is the following.

Definition 4. *(Kernel) Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called kernel if there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that for all $x, y \in \mathcal{X}$*

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product[2] in $\mathcal{H}$.

Note that this definition does make almost no assumptions on the structure of $\mathcal{X}$ and that kernel functions are symmetric by definition of the inner product. Also note that kernels are very simple objects – the inner product of two points mapped to a Hilbert space.

Given a kernel, one can compute the so called *kernel matrix* or *gram matrix*. The will frequently be used in this work.

Definition 5. *(Kernel matrix or Gram matrix) Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel and $X = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ a set of samples from $\mathcal{X}$. The gram matrix as*

$$K_{ij} := k(\boldsymbol{x}_i, \boldsymbol{x}_j) \qquad (1 \leq i, j \leq m)$$

is called gram matrix or kernel matrix w.r.t. X.

Kernel matrices are symmetric by definition of the kernel. In addition all kernel matrices are positive definite, which is a very useful property that will be used throughout this work. In the following, positive definiteness is defined.

[2] An inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ in the space $\mathcal{H}$ satisfies for any $x, y, z \in \mathcal{H}$ and $\alpha \in \mathbb{R}$

$\langle x + y, z \rangle_{\mathcal{H}} = \langle x, z \rangle_{\mathcal{H}} + \langle y, z \rangle_{\mathcal{H}}$ $\qquad$ $\langle \alpha x, y \rangle_{\mathcal{H}} = \alpha \langle x, y \rangle_{\mathcal{H}}$

$\langle x, y \rangle_{\mathcal{H}} = \langle y, x \rangle_{\mathcal{H}}$ $\qquad$ $\langle x, x \rangle_{\mathcal{H}} \geq 0$ and equal only if $x = \mathbf{0}$

Definition 6. *(Positive definite functions and matrices) A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite if for all $\boldsymbol{a} = (a_1, ..., a_m) \in \mathbb{R}^m$ and all $\boldsymbol{x}_1, ..., \boldsymbol{x}_m \in \mathcal{X}$, it holds*

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$$

As mentioned, all kernel matrices are positive definite. The proof is simple and instructive and therefore given here.

Lemma 2. *(Kernel matrices are positive definite) Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel with corresponding Hilbert space $\mathcal{H}$ and corresponding feature map $\phi : \mathcal{X} \to \mathcal{H}$. Let $\boldsymbol{a} = (a_1, ..., a_m) \in \mathbb{R}^m$ and $\boldsymbol{x}_1, ..., \boldsymbol{x}_m \in \mathcal{X}$. Then the kernel matrix K corresponding to k is positive definite.*

Proof.

$$\mathbf{a}^T K \mathbf{a} = \sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \sum_{i=1}^{m} a_i \phi(\mathbf{x}_i), \sum_{j=1}^{m} a_j \phi(\mathbf{x}_j), \right\rangle_{\mathcal{H}} = \left\| \sum_{i=1}^{m} a_i \phi(\mathbf{x}_i) \right\|_{\mathcal{H}}^2 \geq 0$$

where the first equality uses the definition of the kernel matrix, the second equality uses the definition of kernel and inner product and the third equality uses the definition of the norm which is by definition non-negative. □

The other direction is also true: Every positive definite function is the inner product of features $\phi(\mathbf{x})$ in a Hilbert space $\mathcal{H}$, however the space $\mathcal{H}$ and the feature map ϕ may be unknown. This is an extremely useful property since it allows to use *any* positive definite function as a kernel – even when the underlying feature mapping and Hilbert space are unknown. A rather technical proof can be found in [Steinwart and Christmann, 2008, Theorem 4.16].

2.3.2. Reproducing Kernel Hilbert Spaces

Using all the above constructs, it is now possible to introduce the notion of *reproducing kernel Hilbert spaces (RKHS)*. These are, informally speaking, spaces of *functions* from $\mathcal{X}$ to $\mathcal{H}$ – including feature maps $\phi : \mathcal{X} \to \mathbb{R}$. Throughout the notation $f(\cdot) \in \mathcal{H}$ denotes the function $f : \mathcal{X} \to \mathcal{R}$ itself (i.e. the element of the space $\mathcal{H}$; $f \neq f(x)$) and $k(\cdot, x) \in \mathcal{H}$ denotes the kernel k with one argument, i.e. x, fixed and the other argument variable. Since kernels are symmetric $k(\cdot, x) = k(x, \cdot)$.

In the following, two definitions of RKHS are given, along with a proof that these are identical. A unique connection between positive definite kernels and RKHS is described.

Definition 7. *(Reproducing Kernel Hilbert Space via reproducing property) Let $\mathcal{H}$ be a Hilbert space of real value functions on the non-empty set $\mathcal{X}$ of the form $f : \mathcal{X} \to \mathbb{R}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a* reproducing kernel *of $\mathcal{H}$ if for all $\boldsymbol{x} \in \mathcal{X}$ and all*

$f \in \mathcal{H}$, it holds

$$k(\cdot, \boldsymbol{x}) \in \mathcal{H}$$
$$\langle f, k(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}} = f(\boldsymbol{x}) \qquad \textit{(the reproducing property)}$$

Note that it follows that $\langle k(\mathbf{x}, \cdot), k(\cdot, \mathbf{y}) \rangle_{\mathcal{H}} = k(\mathbf{x}, \mathbf{y})$ and therefore $\phi(\mathbf{x}) = k(\mathbf{x}, \cdot)$, which is called the *canonical feature map* of $\mathcal{H}$.

To define an RKHS in an identical way, the following definition is needed.

Definition 8. *(Evaluation operator) Let $f \in \mathcal{H}, f : \mathcal{X} \to \mathcal{R}$, and $x \in \mathcal{X}$. The operator $\delta_x : \mathcal{H} \to \mathbb{R}$ evaluates f at the point x, i.e.*

$$\delta_x f = f(x)$$

The next definition is identical to definition 7 but uses the evaluation operator.

Definition 9. *(Reproducing Kernel Hilbert Space via bounded operator) Let $\mathcal{H}$ be a Hilbert space, $f : \mathcal{X} \to \mathbb{R}$ an element therein, and $\delta_x : \mathcal{H} \to \mathbb{R}$ be an evaluation operator at $x \in \mathcal{X}$. $\mathcal{H}$ is a RKHS if the δ_x is bounded, i.e. there exists a $\lambda_x \geq 0$ such that*

$$|f(x)| = |\delta_x f| \leq \lambda_x ||f||_{\mathcal{H}}$$

This implies

$$|f(x) - g(x)| = |\delta_x (f - g)| \leq \lambda_x ||f - g||_{\mathcal{H}}$$

which is an interesting property: If two functions have the same norm in a RKHS (right hand side of equation is zero), they agree at every point (left hand side also has to be zero). As already mentioned, definitions 7 and 9 are identical. This is now shown.

Theorem 1. *(Equivalence of reproducing kernel and bounded operator) A Hilbert space $\mathcal{H}$ has a reproducing kernel if and only if its evaluation operators are bounded.*

Proof. For illustration, only one direction is proofed here. For the other direction, see for example [Steinwart and Christmann, 2008, Theorem 4.20]. Let H be a RKHS with a reproducing kernel k, i.e. $\langle f, k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$ for all $f \in \mathcal{H}$. Then its evaluation operators δ_x are bounded as

$$\begin{aligned}
|\delta_x f| &= f(x) && \text{(Evaluation operator)} \\
&= |\langle f, k(\cdot, x) \rangle_{\mathcal{H}}| && \text{(Reproducing property)} \\
&\leq ||k(\cdot, x)||_{\mathcal{H}} ||f||_{\mathcal{H}} && \text{(Cauchy-Schwarz)} \\
&= \langle k(\cdot, x), k(\cdot, x) \rangle_{\mathcal{H}}^{1/2} ||f||_{\mathcal{H}} && \text{(Norm definition)} \\
&= \sqrt{k(x, x)} ||f||_{\mathcal{H}} && \text{(Kernel definition)}
\end{aligned}$$

□

Finally, the following theorem establishes connection between the introduced notations of positive definite kernels and RKHS. It is very important since it guarantees the unique existence of a RKHS for every kernel there is. The proof is skipped here since it requires even more tools than already described.

Theorem 2. *(Moore-Aronszajn) For every positive definite kernel k, there exists only one Hilbert space with k as reproducing kernel.*

2.3.3. A Swiss-Army Knife: the Gaussian Kernel

One popular kernel for numerical data is the *Gaussian* (RBF-) kernel. It has been successfully applied to countless examples and is used throughout this work.

Definition 10. *(Gaussian Kernel) Given two m dimensional real vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} \subseteq \mathbb{R}^m$, the* Gaussian kernel *is defined as $k_\sigma : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with*

$$k_\sigma(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{y}||_2^2}{2\sigma^2}\right)$$

where the kernel (band-)width *σ is a parameter.*

Parameter σ determines the length scale the kernel focusses on. It is clear that selection of the bandwidth is very important for any method that uses this kernel – it corresponds to the question which scaling of data contains relevant information.

2.4. The MMD in Reproducing Kernel Hilbert Spaces

In the following section, connections between the MMD and RKHS are drawn. This section, contains mostly theoretical results which are not important in order to understand experimental results of this work. Therefore, this section may be skipped in a first reading.

In definition 2.1, the MMD is defined using *expectations of functions* that belong to a certain class. This function class is now chosen to lie within a RKHS; the functions are elements of this RKHS. Probability distributions may be embedded into a RKHS using the notation of *mean embeddings*. These are the expected values of embeddings of the probability distributions in a RKHS. The distance of these mean embeddings in the RKHS is the basis of the maximum mean discrepancy.

2.4.1. Mean Embeddings

Definition 11. *(Mean embedding) Let $\mathcal{H}$ be a Hilbert space and $f \in \mathcal{F}$, let p be probability measure. The* mean embedding *$\mu_p \in \mathcal{H}$ satisfies*

$$\boldsymbol{E}_x f = \langle f, \mu_p \rangle$$

For details on mean embeddings, see [Berlinet and Thomas-Agnan, 2004, Chapter 4]. The first concern is: if and under which conditions does a mean embedding exist?

Theorem 3. *(Riesz represantion) In a Hilbert space $\mathcal{H}$, the bounded linear operator $A : \mathcal{H} \to \mathbb{R}$, can be written as $\langle \cdot, g_A \rangle_{\mathcal{H}}$ for some $g_A \in \mathcal{H}$. That is, for any element $f \in \mathcal{H}$, there exists $g_A \in \mathcal{H}$ such that*

$$Af = \langle f, g_A \rangle_{\mathcal{H}}$$

Lemma 3. *[Sriperumbudur et al., 2010, Theorem 1] Let $\mathcal{H}$ be a RKHS with kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. If $k(\cdot, \cdot)$ is measurable and for any $x \in \mathcal{X}$*

$$\boldsymbol{E}_x \left[\sqrt{k(x, x)} \right] < \infty$$

then the mean embedding lies in $\mathcal{H}$, i.e. $\mu_p \in \mathcal{H}$.

Proof. The (linear) expectation operator $T_p f := \mathbf{E}_x[f]$ is bounded for all $f \in \mathcal{H}$, since

$$|T_p f| = |\mathbf{E}_x[f]| \leq \mathbf{E}_x[|f|] = \mathbf{E}_x[|\langle f, \phi(x) \rangle_{\mathcal{H}}|] \leq \mathbf{E}_x \left[\sqrt{k(x, x)} ||f||_{\mathcal{H}} \right]$$

Now, the Riesz representation theorem (theorem 3) gives the existance of $\mu_p \in \mathcal{H}$ such that $T_p f = \langle f, \mu_p \rangle$ □

Choosing $f = \phi(x) = k(\cdot, x)$ leads to $\mu_p(x) = \langle \mu_p, k(\cdot, x) \rangle_{\mathcal{H}} = \mathbf{E}_p \phi(x)$. I.e. the mean embedding of distribution p is the expected value of the canonical feature map under p.

Using the notation of the mean embedding, the MMD may be written in a simple form – as the distance between

Lemma 4. *[Borgwardt et al., 2006, Theorem 2.2] Given two probability distributions p, q and given that their mean embeddings μ_p, μ_q in a unit ball $\mathcal{F}$ in a RKHS $\mathcal{H}$ exist, then*

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = ||\mu_p - \mu_q||^2_{\mathcal{H}}$$

Proof. Directly compute

$$\begin{aligned}
\mathrm{MMD}^2[\mathcal{F}, p, q] &= \left[\sup_{||f||_{\mathcal{H}} \leq 1} (\mathbf{E}_x[f(x)] - \mathbf{E}_y[f(y)]) \right]^2 \\
&= \left[\sup_{||f||_{\mathcal{H}} \leq 1} \langle \mu_p - \mu_q, f \rangle_{\mathcal{H}} \right]^2 \\
&= \langle \mu_p - \mu_q, \mu_p - \mu_q \rangle_{\mathcal{H}} \\
&= ||\mu_p - \mu_q||^2_{\mathcal{H}}
\end{aligned}$$

where the first equality follows from the existance of the mean embedding and the second equality follows from the the fact that the supremum if reached for f having the same direction as $\mu_p - \mu_q$. □

2.4.2. MMD as Metric Allows to Distinguish any Distribution Pair

In order to be able to distinguish *any pair* of probability distributions, the MMD must take a unique value if and only if the distributions are equal. This corresponds to saying that the MMD is a metric[3].

The MMD is a metric when the underlying mean embeddings are injective, i.e. different distribution lead to two different embeddings. Note the connection to Lemma 1 which says that two distributions are equal if and only if the expected values of all functions are equal. When this important property is met, it allows to distinguish any pair of probability distributions. It is *not* guaranteed for every RKHS – only for so called *universal* RKHS. This requires the kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, defined on a compact metric space $\mathcal{X}$ to be continuous and $\mathcal{H}$ to be dense in $\mathcal{C}(\mathcal{X})$ with respect to the L_∞ norm (see [Steinwart and Christmann, 2008] for details and proofs that the RKHS induced be the Gaussian kernel, c.f. section 2.3.3, is universal). The next theorem states when the MMD is a metric.

Theorem 4. *See [Gretton et al., 2012a, Theorem 5] for references. Given a unit ball $\mathcal{F}$ in a universal RKHS $\mathcal{H}$ with corresponding continuous kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, defined on a compact metric space $\mathcal{X}$, then*

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = 0 \qquad \Leftrightarrow \qquad p = q$$

Proof. Direction $\Leftarrow$ is easy: $p = q$ implies that

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = ||\mu_p - \mu_q||^2_{\mathcal{H}} = 0$$

To proof direction $\Rightarrow$, assume that $\mathrm{MMD}^2[\mathcal{F}, p, q] = 0$, which implies that $\mu_p = \mu_q$. First, note that since $\mathcal{H}$ is universal, for any $\epsilon > 0$ and $f \in \mathcal{C}(\mathcal{X})$, there is a $g \in \mathcal{H}$ which is close to f, i.e.

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

Expanding gives

$$|\mathbf{E}_x f(x) - \mathbf{E}_x g(x)| \leq |\mathbf{E}_x f(x) - \mathbf{E}_x g(x)| - |\mathbf{E}_x g(x) - \mathbf{E}_y g(y)| + |\mathbf{E}_y g(y) - \mathbf{E}_y f(y)|$$

The middle term can be rewritten as

$$|\mathbf{E}_x g(x) - \mathbf{E}_y g(y)| = \langle g, \mu_p - \mu_q \rangle_{\mathcal{H}} = 0$$

The first and third term can be upper bounded using the fact that f and g are close to

[3] A metric on the set $\mathcal{X}$ is a distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for any $x, y, z \in \mathcal{X}$, it holds that

$$d(x, y) > 0 \qquad d(x, y) = 0 \Leftrightarrow x = y$$
$$d(x, y) = d(y, x) \qquad d(x, z) \leq d(x, y) + d(y, z)$$

each other.

$$|\mathbf{E}_x f(x) - \mathbf{E}_x g(x)| \leq \mathbf{E}_x |f(x) - g(x)| \leq \epsilon$$
$$|\mathbf{E}_y g(y) - \mathbf{E}_y f(y)| \leq \mathbf{E}_y |g(x) - f(y)| \leq \epsilon$$

The inequality becomes

$$|\mathbf{E}_x f(x) - \mathbf{E}_x g(x)| \leq 2\epsilon$$

This implies $p = q$ according to Lemma 1. □

2.4.3. A Kernel-Based Two-Sample Test

This section introduces statistics that will be used for two-sample testing in this work. The following definitions are fundamental in order to understand experimental results that are presented later on.

Finally, a statistical test is of interest where an estimator of the MMD is used as test statistic. Two such estimators are established now: one that can be computed in quadratic time, and one that can be computed in linear time. Only the latter will be extensively used throughout this work. However, since the quadratic statistic is important for kernel two-sample testing, and since the outlook of this work involves it, it will be described here anyway.

The MMD will be expressed in terms of expected values of kernel of kernel functions on sample data as in the following Lemma.

Lemma 5. *[Gretton et al., 2012a, Lemma 6] In a RKHS $\mathcal{H}$ with kernel k, given random variable x with distribution p and random variable y with distribution q, the squared population MMD is*

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = \boldsymbol{E}_{x,x'}\left[k(x, x')\right] - 2\boldsymbol{E}_{x,y}\left[k(x, y)\right] + \boldsymbol{E}_{y,y'}\left[k(y, y')\right]$$

where x', y' are independent *copies of x, y with the same distribution respectively.*

Proof. Using the expression for the MMD in Lemma 4, directly compute

$$\begin{aligned}
\mathrm{MMD}^2[\mathcal{F}, p, q] &= ||\mu_p - \mu_q||_{\mathcal{H}} \\
&= \langle \mu_p - \mu_q, \mu_p - \mu_q \rangle_{\mathcal{H}} \\
&= \langle \mu_p, \mu_p \rangle_{\mathcal{H}} - 2\langle \mu_p, \mu_q \rangle_{\mathcal{H}} + \langle \mu_q, \mu_q \rangle_{\mathcal{H}} \\
&= \mathbf{E}_{x,x'} \langle \phi(x), \phi(x) \rangle - 2\mathbf{E}_{x,y} \langle \phi(x), \phi(y) \rangle + \mathbf{E}_{y,y'} \langle \phi(y), \phi(y) \rangle \\
&= \mathbf{E}_{x,x'}\left[k(x, x')\right] - 2\mathbf{E}_{x,y}\left[k(x, y)\right] + \mathbf{E}_{y,y'}\left[k(y, y')\right]
\end{aligned}$$

□

The above expressions are population expressions, meaning that they hold for the infinite data case. In practice, estimates based on finite data have to be used. These of course, since they depend on sample data, have different values for each set of samples that is used. Computing an estimator therefore corresponds to sampling from either the

null (given $H_0 : p = q$ holds) or alternative distribution (given $H_A : p \neq q$ holds). Along with the estimators, expressions around these distributions are provided where needed.

Quadratic Time MMD statistic

A naive estimate is to incorporate as much information in data as possible. This is possible in quadratic time since all pairs of samples have to be considered. When expectations are taken over independent random variables, one basically needs to compute the mean of all pairs of samples, except for equal ones. This way, all available information of the sample data is taken account. Following this approach leads to a naive empirical estimate of the population MMD.

Lemma 6. *[Gretton et al., 2012a, Lemma 6] An unbiased estimator of the population expression for the squared MMD in Lemma 5 for two sets of samples X, Y of size m, n, drawn from p, q respectively, is given by*

$$\begin{aligned}\mathrm{MMD}_u^2[\mathcal{F}, X, Y] =& \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(y_i, y_j) \\ & - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j \neq i}^{n} k(x_i, y_j)\end{aligned}$$

The computational costs are in $\mathcal{O}((m+n)^2)$.

Due to the costs, this statistic does not work for large scale problems. On the other hand, it squeezes as much information out of the sample as possible. A biased statistic can obtained if the diagonals of the kernel matrices are also taken into account when averaging, i.e.

$$\begin{aligned}\mathrm{MMD}_b^2[\mathcal{F}, X, Y] =& \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} k(x_i, x_j) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(y_i, y_j) \\ & - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j \neq i}^{n} k(x_i, y_j) \end{aligned} \tag{2.2}$$

This estimator is sometimes used when actual tests are constructed. It is not very important in context of this work but rather given for completeness.

The null distribution of the quadratic time MMD has a complicated form: an infinite sum of χ^2 variables, which leads to a long tailed shape. Alternative distribution is Gaussian. Figure 2.2 shows an empirical estimate of both distributions. Formal details are omitted here since they are not needed. See [Gretton et al., 2012a, Theorem 12]. Since the outlook of this work will deal with the unbiased quadratic time statistic, expressions for population variance and quadratic time estimates will be provided there.

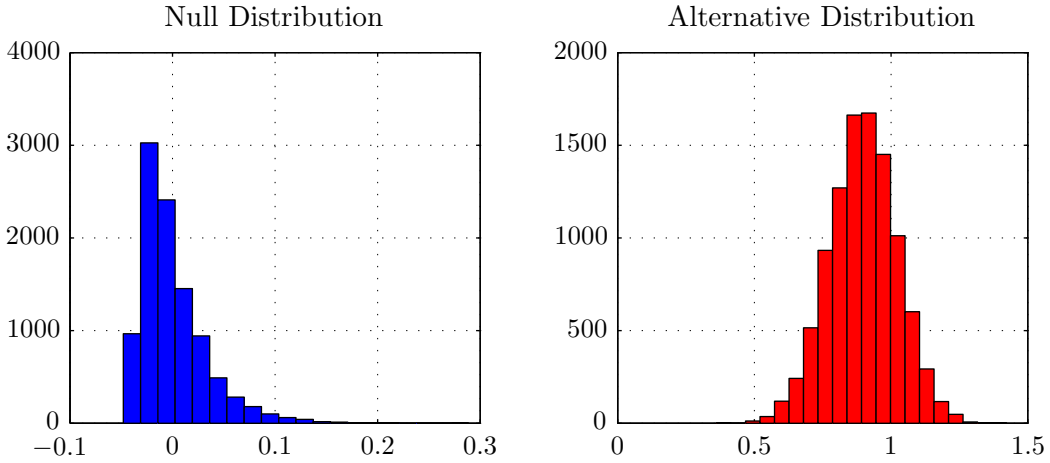


Figure 2.2.: Histograms of 10000 samples from null and alternative distribution of the quadratic time MMD. The null distribution has a long tailed form while alternative distribution is normal. Used data: two univariate normal distributions with different mean.

Linear Time MMD statistic

Sometimes, a fast test is desired, which at the same time does not loose too much accuracy. A statistic that can be computed in linear time and still incorporates every sample datum is to divide the data into two halves and use these as independent random variables. This statistic has many appealing properties and is the main methods used in this work.

Lemma 7. *[Gretton et al., 2012a, Lemma 14] An unbiased estimator of the population expression for the squared MMD in Lemma 5 for two sets of samples X, Y, assumed to have equal size m for notational ease, drawn from p, q respectively, is given by*

$$\begin{aligned}
\mathrm{MMD}_l^2[\mathcal{F}, X, Y] &= \frac{1}{m_2}\sum_{i=1}^{m_2} h(z_i, z_i') \\
&= \frac{1}{m_2}\sum_{i=1}^{m_2} \left(k(x_{2i-1}, x_{2i}) + k(y_{2i-1}, y_{2i}) - k(x_{2i-1}, y_{2i}) - k(x_{2i}, x_{2i-1})\right) \\
&= \frac{1}{m_2}\operatorname{tr}(K_{XX} + K_{YY} - K_{XY} - K_{YX}) \qquad (2.3)
\end{aligned}$$

where m_2 is $\lfloor \frac{m}{2} \rfloor$ and $h(z_i, z_i') := k(x_{2i-1}, x_{2i}) + k(y_{2i-1}, y_{2i}) - k(x_{2i-1}, y_{2i}) - k(x_{2i}, x_{2i-1})$. The indices of the kernel matrices stand for the first and second half of a dataset, i.e. $(K_{XX})_{ii} = k(x_{2i-1}, x_{2i})$. Note that only the diagonal of the matrices have to be evaluated. The computational costs are in $\mathcal{O}(m)$.

The linear time statistic has a useful property: its associated null *and* alternative-

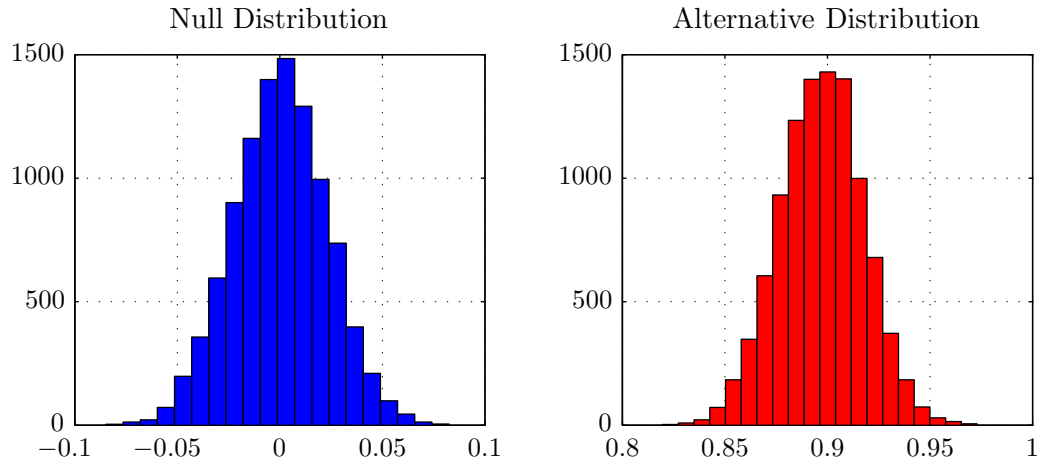


Figure 2.3.: Histograms of 10000 samples from null and alternative distribution of the linear time MMD. Both distributions are normal with equal variance. Used data: two univariate normal distributions with different mean.

distributions are normal with equal variance. This will exploited later to construct a kernel selection criterion. See figure 2.3 for empirical estimates. The null-distribution has zero mean since the statistic is unbiased, while the alternative distribution has a positive mean. Another appealing property that is induced by the equal variance is the fact that, in order to construct a test, only the variance has to be estimated – which will turn out to be solvable in linear time also. Since the statistic is just the sum of independent random variables, the central limit theorem [Serfling, 1980, Section 1.9] leads to the following Lemma.

Lemma 8. *[Gretton et al., 2012a, Corollary 16] Assume* $0 < \boldsymbol{E}(h^2) < \infty$. *Then* MMD_l^2 *converges in distributions to a Gaussian according to*

$$m^{\frac{1}{2}} \left(\mathrm{MMD}_l^2 - \mathrm{MMD}^2[\mathcal{F}, p, q]\right) \xrightarrow{D} \mathcal{N}(0, 2\sigma_l^2)$$

where

$$\sigma_l^2 = \left[\boldsymbol{E}_{z,z'} h^2(z, z') - [\boldsymbol{E}_{z,z'} h(z, z')]^2\right]$$

with the notational shortcut $\boldsymbol{E}_{z,z'} = \boldsymbol{E}_{z,z' \sim p \times q}$

This result is very useful for constructing two-sample tests using the linear time statistic. Since the null distribution is normal, it is very easy to approximate it: this is possible simply by computing its variance – it has zero mean. Consequently, computing a test threshold is straight-forward using the inverse Gaussian cumulative distribution function – a two-sample test may be performed easily once the variance is known. Details on how to construct a test will follow.

The variance expression provided above is a population expression and needs to be

estimated. This is possible in linear time by simply computing an unbiased empirical variance estimate $\hat{\sigma}_l^2$ of the observations in the $h(z_i, z_i')$ entries, i.e.

$$\begin{aligned}\hat{\sigma}_l^2 &:= \frac{1}{m_2 - 1} \sum_{i=1}^{m_2} \left(h(z_i, z_i') - \frac{1}{m_2} \sum_{j=1}^{m_2} h(z_j, z_j') \right)^2 \\ &= \frac{1}{m_2 - 1} \sum_{i=1}^{m_2} \left(h(z_i, z_i') - \mathrm{MMD}_l^2[\mathcal{F}, X, Y] \right)^2 \end{aligned} \tag{2.4}$$

This estimate can also be computed in linear time.

Usage Scenarios: Linear vs. Quadratic Time

Intuitively, MMD_u^2, MMD_l^2 and any other unbiased estimator of the population MMD are small when $p = q$, and large if $p \neq q$. Therefore, both are suited for two-sample testing. Since the linear time statistic does not consider all connections in the data (only diagonal entries of kernel matrices), it is less stable. On the other hand, it is able to process far larger amounts of data that the quadratic time statistic, since it does not have to store data in memory. Both statistics are therefore useful in practice – only use scenarios are different:

- The quadratic time statistic is useful in the finite data case, infinite time scenario – when the amount of data is limited and the test should be as accurate as possible, no matter how long it takes.
- The linear time statistic is useful in the infinite data case, finite time scenario – when the amount of data is nearly unlimited, but time is limited. Another important observation is that data needs not to be stored, so this statistic may easily be applied to streaming data.

In [Gretton et al., 2012a, Section 8.3], the observation was reported that the linear time statistic leads to a stronger performance than the quadratic time statistic when time is limited and data is unlimited.

2.5. Summary

This chapter introduced three fundamental concepts that are used in this work.

Two-Sample Testing Statistical hypothesis and in particular two-sample testing was described as a method to distinguish probability distributions on the base of drawn samples – using a test statistic that is compared against a certain threshold. If the probability that the statistic was generated by the null hypothesis $H_0 : p = q$ is low, the latter is rejected.

Kernels and RKHS Second, kernels and reproducing kernel Hilbert spaces were introduced. A chain of arguments in order to allow notation of these was described along with selected proofs. Kernels are inner products of feature maps, which map data from an input space into a possibly high-dimensional feature space. Their most important property is positive definiteness. RKHS are spaces of functions whose elements are feature maps of kernels, which is the same as a kernel with one argument fixed. RKHS and positive definite kernels are in a unique relationship.

MMD Notation and Empirical Estimates Third, the Maximum Mean Discrepancy was described as the distance of mean embeddings of probability distribution functions in a RKHS. It was shown that such mean embeddings exist and that they can be expressed as expected values of reproducing kernels. The MMD is a metric on these embeddings and therefore can be used for two-sample testing. Linear and quadratic time empirical estimates can be computed on the base of kernel functions on data. These have different distributions that will be important during this work.

3. Methods for Test Construction & Kernel Choice

Chapter Overview This chapter focusses on methods that are important throughout this work: constructing two-sample tests using MMD estimates and methods for kernel selection for these. As for kernel selection, existing state-of-the-art methods are described – followed by novel methods which are superior.

Section 3.1, describes how to construct two-sample tests on the base of MMD estimates as described in the previous chapter. In particular, this involves methods for approximating null distributions. A general method that works with any test statistic, bootstrapping, is described in section 3.1.1, a method for linear time MMD estimates is described in section 3.1.2, followed by a on-line version in section 3.1.3. Null distribution approximations for the quadratic time MMD are briefly mentioned in section 3.1.4.

Section 3.2, introduces and motivates the problem of kernel choice for two-sample testing. It then provides an overview of all investigated methods in this work. This starts with existing methods as the popular median based heuristic in section 3.2.1 – followed by the current state-of-the-art method of maximising MMD estimates in section 3.2.2. Connections to a related approach that interprets the MMD as a binary classifier with a linear loss function is described in sections 3.2.3 and 3.2.4. Finally, in section 3.3, downsides of existing methods are described. A new criterion for kernel selection for the linear time test is described in section 3.3.2. This includes an empirical estimate for use in practice, an argument why the method yields optimal kernel choice, and why it is superior to existing methods. Finally, another new method for single kernel selection based on cross-validation is described in section 3.3.3. The chapter closes with a summary of contained key points in section 3.4.

Literature & Contributions This chapter contains a number of methods that are taken from literature and a number of methods that have not yet been described.

Methods for approximating null distributions of the quadratic time MMD in section 3.1.4 come from [Gretton et al., 2012b]. The general bootstrapping method is described in [Gretton et al., 2012a] as does the linear time test construction in section 3.1.2. Explicit description of an on-line style test construction in section 3.1.3 is a minor original extension of the linear time test.

Existing kernel selection methods come from literature: the median heuristic, section 3.2.1, is suggested in [Gretton et al., 2012a], state-of-the-art kernel selection via maximising the MMD estimate (section 3.2.2) is described in [Sriperumbudur et al., 2009] as well as connections to binary classification in section 3.2.3. The linear loss based method in section 3.2.4 comes from [Sugiyama et al., 2011].

Algorithm 3.1 Bootstrapping the null-distribution of an MMD estimator.

Inputs are:

- X, Y, sets of samples from p, q of size m, n respectively

Output is:

- One sample from null-distribution

1: $Z \leftarrow \{X, Y\}$
2: $\hat{Z} = \{\hat{z}_1, ..., \hat{z}_{m+n}\} \leftarrow \text{randperm}(Z)$ (generate a random ordering)
3: $\hat{X} \leftarrow \{\hat{z}_1, ...\hat{z}_m\}$
4: $\hat{Y} \leftarrow \{\hat{z}_{m+1}, ...\hat{z}_{m+n}\}$
5: **return** Empirical estimate for MMD$[F, \hat{X}, \hat{Y}]$

All new methods for kernel selection in section 3.3 have not yet been described in literature: this includes problems of maximising MMD estimates alone as well as the new criterion for kernel selection in section 3.3.2. This method and also provided arguments on its optimality will also appear in a compressed form in [Gretton et al., 2012c], which was submitted while this thesis was written. Formal results such as theorem 5 will be established there. Section 3.3.2 references kernel limits of the new ratio for Gaussian kernel and data in appendix A.1. This derivation is solely found in this thesis. Same holds for the cross-validation based kernel selection method in section 3.3.3.

3.1. Test Construction via Null-Distribution Approximation

As mentioned in section 2.1.1 and depicted in figure 2.1, in order to construct a two-sample test, the null distribution has to be known or approximated. Given this information, a threshold can be chosen in such way that the probability that a test statistic comes from the null distribution, which corresponds to an upper bound on the type I error, is at some level α. Since different estimators of the MMD have different null distributions, it is hard to come up with an efficient general method. However, there exists a method that works for *any* two-sample test.

3.1.1. Sampling the Null-distribution: Bootstrapping

To approximate null distributions, there exists a naive and straight-forward technique, called *bootstrapping*, which can be in fact used to approximate any distribution: generate a large number of samples from it and compute quantiles and other quantities on base of these samples. Any MMD estimate is a sample from its corresponding null distribution when data was generated under the null hypothesis $H_0 : p = q$. Given finite data, this is easy to emulate by merging and permuting samples X and Y; it is formalised in algorithm 3.1

Bootstrapping is a useful technique to create ground-truth samples for a null-distribution.

However, it is rather costly because the statistic has to be re-computed for every sample. For example, when the quadratic time estimate of the MMD is used the costs are in $\mathcal{O}(\ell(m+n)^2)$ for ℓ samples. Precomputed kernel matrices give a massive performance gain. Nevertheless, theoretical costs do not change.

Once a bunch of samples $\{s_1, ..., s_\ell\}$ is sampled from a null distribution, these can be used to compute a threshold or a p-value. To this end, drawn samples have to be sorted to get an ordering

$$(s'_1, s'_2, ..., s'_\ell) \qquad \text{such that} \qquad s'_1 \leq s'_2 \leq ... \leq s'_\ell \qquad \text{and} \qquad \{s'_1, ..., s'_\ell\} = \{s_1, ..., s_\ell\}$$

Given this ordering, and a test level α, a threshold is simply the $(1-\alpha)$ quantile in the samples, i.e.

$$s'_t \qquad \text{where} \qquad t = \left\lfloor \frac{1-\alpha}{\ell} \right\rfloor$$

Test statistics can now be compared against the threshold in order to accept (test statistic below s_t) or reject (test statistic above s_t) the null hypothesis $H_0 : p = q$. Similarly, a p-value for a given test statistic x can be computed by finding the position of the x in the null samples, i.e.

$$p = \frac{y}{\ell} \qquad \text{where} \qquad s_{y-1} \leq x \leq s_{y+1}$$

Given the p-value for a certain statistic estimate, the null hypothesis $H_0 : p = q$ is rejected if this value is larger than a given test level α. Note that comparing a statistic against a threshold and comparing a p-value against a desired p-value is exactly the same thing. Though, p-values are a little more informative since they are not just a binary value but the probability that the statistic comes from the null distribution given that $H_0 : p = q$ is true.

3.1.2. Linear Time MMD: Gaussian Approximation

As already mentioned, when using the linear time MMD for performing two-sample-testing, the population null-distribution is normal, c.f. Lemma 8. The distribution has zero mean; a population expression for variance along with an empirical linear time estimate was given in section 2.4.3. Using this empirical variance estimate, it is possible to approximate the null distribution by simply plugging in zero mean and the variance estimate into the normal distribution. Once this is done, it is easy to compute thresholds and p-values. Note that this test is consistent. Another major advantage is that it is possible in linear time, and does not have to store data in memory.

Given a variance estimate $\hat{\sigma}_l^2$, the threshold for test level α is simply given by

$$\Phi^{-1}_{0,\hat{\sigma}_l^2}(1-\alpha)$$

where $\Phi^{-1}_{\mu,\sigma^2} : [0,1] \to \mathbb{R}$ is the inverse normal cumulative distribution[1] function for mean μ and variance σ^2, which returns the value x that lies at the $(1-\alpha)$ quantile of the specified normal distribution.

Similarly, a p-value for a statistic estimate x can be computed by evaluating the position of x in the normal distribution, i.e.

$$\Phi_{0,\hat{\sigma}_l^2}(x)$$

Since Φ is part of any computer algebra system, this is easily computed.

3.1.3. Large Scale Two-Sample Testing: On-line Mean and Variance

As already briefly mentioned in section 2.4.3, the linear time MMD statistic may be used in the *infinite* or *streaming data* case. This means that the data used for the two-sample-test does not fit in memory; and therefore requires an on-line way to compute statistic and to approximate null-distribution. *On-line* means that only a constant number of samples are kept in memory while constructing/performing the test.

Recall that the empirical linear time MMD statistic is simply a mean of diagonal terms of kernel matrices, see expression 2.3. Since none of these matrices can be stored in memory in order to compute the trace, the expression has to be evaluated as a *running average*: generally speaking, the mean of t numbers, $\bar{x}_t := \frac{1}{t}\sum_{i=1}^{t} x_i$, can be expressed using the sum of the first $t-1$ numbers, $(t-1)\bar{x}_{t-1} = \sum_{i=1}^{t-1} x_i$, and the last number x_t, i.e.

$$\bar{x}_t = \frac{1}{t}\sum_{i=1}^{t} x_i = \frac{1}{t}\left(\sum_{i=1}^{t-1} x_i + x_t\right) = \frac{(t-1)\bar{x}_{t-1} + x_t}{t} \tag{3.1}$$

Therefore, in order to compute a mean, only the sum of all terms so far has to be stored. The mean in expression 2.3 can be computed exactly in that fashion.

The same arguments hold for the empirical variance of all terms in expression 2.3, as given in expression 2.4. The Variance of t numbers

$$\operatorname{var} x_t = \frac{\sum_{i=1}^{t} x_i^2 - t\bar{x}}{t-1} \tag{3.2}$$

is composed of the sum of the first t numbers, $t\bar{x}_t = \sum_{i=1}^{t} x_i$, and the sum of the same numbers squared, $\sum_{i=1}^{t} x_i^2$. Therefore, in order to compute the variance, only these two numbers have to be stored. Once the variance is known, one can use the Gaussian approximation of the null-distribution, as described in section 3.1.2, to construct a two-sample test that has linear computational costs and constant space costs.

[1] For a probability distribution $p(x)$ and a real number y, the cumulative distribution function gives $p(x \leq y)$, i.e. the integral from zero to x of $p(x)$. The inverse cumulative distribution function of a number t returns an argument y such that $p(x \leq y) = t$.

On-line Bootstrapping In general, there is no point in performing a bootstrapping based two-sample test using the linear time MMD since the Gaussian approximation of the null distribution is a cheaper alternative. However, it may be for certain distributions p and q that the number of samples needed for the approximation to be correct is very large. In this case, bootstrapping may be a safer alternative to get a consistent test. In general, bootstrapping has to store all data in memory to permute it. However, since the use-case for the linear time statistic is that infinite data is available, instead of merging and permuting the original samples, simply a new set of samples can be drawn from p and q in each iteration. Algorithm 3.1 would have to be called with new samples X and Y for every null distribution sample. This way, it is possible to perform an on-line style approximation of the null distribution: only samples from it have to be stored whereas the data used to draw these samples is discarded every iteration.

In the experiments section, an empirical investigation of necessary sample size for Gaussian approximation is provided.

Parallel Computations

Note that in order to compute the linear time estimate or its variance from expressions 2.3 and 2.4, only the sum of (squared) terms of the form

$$h(z_i, z_i') = k(x_{2i-1}, x_{2i}) + k(y_{2i-1}, y_{2i}) - k(x_{2i-1}, y_{2i}) - k(x_{2i}, x_{2i-1})$$

have to be stored. These two number then can be used to compute running averages for MMD estimate and its variance using expressions 3.1 and 3.2. At every step, only four points are needed[2]. This allows a straight-forward way to perform the test on massively parallel computing machines: each job computes the sum of (squared) terms for a given amount of data along with the number of samples used. When this is done, they are simply added in order to get the final statistic/threshold combination.

This is a very useful property since the linear time MMD test is most attractive when problems are so hard that massive amounts of data are needed. This would normally lead to huge computation times. Nevertheless, since parallelisation is easy, it scales with available cluster computer capacity.

3.1.4. Quadratic Time MMD: Gamma & Spectrum Method

In the literature, there are two methods for constructing a two-sample test using the quadratic time MMD. Both are faster than the bootstrapping approach, as described in section 3.1.1, which can always be used to get consistent (but slow) results. The *gamma* method is an approximation that fits a gamma distribution to the null distribution of the MMD estimate. It is fast but –while being a heuristic only– it offers no guarantees. The *spectrum* method is a fast and consistent method. It has larger costs than *gamma* but is still much faster than bootstrapping while being consistent. Still, it may fail in

[2] In practice, this should be implemented using blocks of a few thousand points rather than single points in order to minimise computational overhead.

practice for specific distributions p and q. Both methods will be used for constructing two-sample tests when linear and quadratic MMD estimates are compared. However, they are not part of the main subject of analysis of this work and are therefore given for the sake of completeness only. This section may be skipped in a first reading.

Moment Matching of a Gamma Distribution

A very fast heuristic for approximating the null distribution is to fit the first two moments of a gamma distribution to the null distribution of the *biased* quadratic time MMD estimate as described in section 2.2. See [Gretton et al., 2012b]. This remains a heuristic and does not guarantee consistent results, i.e. fixed type I error. However, it is very fast ($\mathcal{O}(m^2)$) and gives good results in practice – although there exist distributions where it fails. Therefore, its quality of approximation should always be checked by observing the type I error. Moments are matched as in the following:

$$m\,\mathrm{MMD}_b[\mathcal{F}, X, Y] \sim \frac{x^{\alpha-1}\exp(-\frac{x}{\beta})}{\beta^\alpha \Gamma(\alpha)}$$

where

$$\alpha = \frac{(\mathbf{E}(\mathrm{MMD}_b[\mathcal{F}, X, Y]))^2}{\mathrm{var}(\mathrm{MMD}_b[\mathcal{F}, X, Y])} \quad \text{and} \quad \beta = \frac{m\,\mathrm{var}(\mathrm{MMD}_b[\mathcal{F}, X, Y])}{(\mathbf{E}(\mathrm{MMD}_b[\mathcal{F}, X, Y]))^2}$$

From the expressions, it is clear, that only the statistic itself and its variance (which can be estimated in quadratic time) are needed to approximate the null distribution. In this work, this method is only briefly mentioned since it will not be used widely. See [Gretton et al., 2012b] for (also empirical) details.

Once parameters of the gamma distribution are known, threshold and p-value may be easily computed using the cumulative distribution function of the gamma distribution[3] in the same fashion as it was done with the normal CDF in section 3.1.2.

A Fast, Consistent Test based on the Eigenspectrum of the Kernel

In [Gretton et al., 2012b], another fast method of approximating the null distribution is described. This test is based on the Eigenspectrum of the kernel matrix of the joint samples. It is a bit more expensive to compute since empirical eigenvalues have to be computed (depending on implementation around $\mathcal{O}(m^3)$); however, it is still faster than bootstrapping while offering the same theoretical guarantees, namely consistency.

The null distribution of the *biased* quadratic time MMD converges in distribution as

$$m\,\mathrm{MMD}_b^2[\mathcal{F}, X, Y] \to \sum_{l=1}^{\infty} \lambda_l z_l^2$$

[3]The probability density function of the Gamma distribution with shape parameter $k > 0$ and scale parameter $\theta > 0$ on random variable $x > 0$ is $p(x; k, \theta) = \theta^{-k}\frac{1}{\Gamma(k)}x^{k-1}\exp(-\frac{x}{\theta})$.

where $z_l \sim \mathcal{N}(0, 2)$ are i.i.d. normal samples and λ_l are Eigenvalues of expression 3 in [Gretton et al., 2012b]. These expressions can be estimated by $\hat{\lambda}_l = \frac{1}{m}\nu_l$, where ν_l are Eigenvalues of the centred kernel matrix of the joined samples X and Y. Using this approximation, the null distribution can easily be sampled using $\hat{\lambda}_l$ and normal samples. Sampling form this distribution is obviously much faster than via bootstrapping where the complete statistic has to be computed in every iteration. Also in practice, not all Eigenvalues are used which makes computation cheaper. For more details, including empirical evaluation, see the mentioned paper.

As for bootstrapping, once a set of samples is drawn from the null distribution, it is straight forward to compute threshold and p-value. See section 3.1.1.

3.1.5. Interim Summary

This section described how to construct two-sample tests on the base of different MMD estimates. Namely, a technique that works for any MMD estimate, called bootstrapping, was introduced in section 3.1.1. For the linear time MMD estimte, a linear time approximation of the null distributions was described in section 3.1.2 along with notes on how to compute it in an on-line fashion. This is the main method used in this work. Section 3.1.4 described two ways of constructing a two sample test for the quadratic time MMD estimate: one fast heuristic and one consistent version. Bootstrapping is also a valid choice for the quadratic time statistic, whereas it does not make that much sense to use it for the linear statistic.

3.2. On Kernel Choice and Existing Methods

As described in section 2.1, a statistical test should have a low type two error for a fixed type one error. The tests in this work have, generally speaking, parameters that influence the error level of the test. More precisely: the choice of the kernel (and its parameters). Even though in theory, some kernels (including the Gaussian that is used in this work) can in theory distinguish any two distributions, in practice their parameters have a large impact on the test's type II error. The Gaussian kernel's only parameter is its bandwidth. This parameter basically determines the length scale at which the kernel looks at data. In order so get a low type II error, this length scale has to be set to the size where differences in the two underlying distributions p and q appear. If being set too large or too small, the kernel is not able to detect these differences. A major part of kernel selection in this work will deal with finding optimal bandwidth parameters. Another large part deals with finding weights for finite non-negative combinations of arbritary kernels. The latte will be described in the next chapter. This chapter deals with selecting single parameters. Most methods are able to select parameters of *any* kernel type – they are not restricted to the Gaussian kernel; while some methods are only usable with the latter.

To select a parameter, one needs a *measure of quality*. Optimally, one would have access to the type II errors of a test with *any* (kernel-)parameters. If this was known,

simply the parameter (here: kernel) that leads to the lowest type II error was chosen. However, since distributions are unknown, this is impossible in general.

In the following, different methods for selecting a two-sample-test's kernel (or kernel parameter) are described. Existing methods will be explained along with possible shortcomings; then, a bunch of new methods that were developed during this work are introduced and set in contrast to existing approaches.

Disjoint Training and Testing Data

Before methods for kernel selection are described, it is important to note the following: asymptotic distributions of all MMD based tests, c.f. section 2.4.3 and 2.4.3, are only valid for *fixed* kernels. In particular, they only hold of the kernel is *not* a function of the data. Therefore, when a kernel is selected on the same samples that are used to perform a two-sample-test, *none* of the methods for approximating null distributions are valid – except for bootstrapping, as described in section 3.1.1, since it simply samples the null distribution. However, this is highly undesirable since bootstrapping is costly for quadratic time tests and pointless for linear time tests (the Gaussian approximation is much faster). As a consequence, parameter selection data and test data are kept *disjoint* in this work. Practically speaking, a kernel is selected on a *training set*; then, using this kernel, the two-sample-test is performed on different data forming the *test* set.

Overview

Since there will be introduced many methods, in order to not loose the overall picture, all methods will be briefly listed now. See also table 3.1. **Existing approaches** for kernel selection for MMD based two-sample tests are:

1. Setting the bandwidth of a Gaussian kernel to the median distance in samples. This approach is limited to Gaussian kernels and widely used in its context.
2. Choosing a kernel such that the MMD statistic itself is maximised. The hope is that a larger statistic reduces the chance for a type II error.
3. Minimising linear loss of a binary classifier, which corresponds to maximising the MMD itself. In order to avoid over-fitting, this is done via cross-validation.

The following methods are **newly described** in this work:

1. Maximising the linear time MMD statistic while minimising its variance. It will be argued that this is a better choice than just maximising the MMD.
2. Minimising the type II error directly using cross-validation. This approach is inspired by minimising test error in binary classification.

All methods are described in detail in the following. Every method will be motivated, possible advantages and disadvantages will be listed, and expectations for experiments will be formulated.

Existing Methods	New Methods
Maximise MMD	Maximise ratio: MMD over standard deviation
Minimise linear loss	Minimise type II error
Use Median data distance	

Table 3.1.: Existing methods for kernel selection in context of MMD-based two-tample tests and their newly described counterparts.

3.2.1. Gaussian Kernels: Median Data Distance

One of the first methods that was proposed for choosing the width of a Gaussian kernel is to use the median distance of underlying data, see for example [Schölkopf, 1997]. This heuristic method is motivated by the fact that the dominant scaling level of the data (which is exactly what the median distance is) might well reflect underlying structure and therefore is a good choice to focus on. Several empirical investigations have shown that the method works well in practice.

An advantage is that it is easy to compute; all pairwise distances of data have to be computed and sorted to find the median. Since the median is a stable statistic, this can be done on a small subset of data. This makes the method attractive for situations where kernel selection has not been investigated yet, since it may well give good results in certain contexts. However, it is easy to construct datasets where selecting the median completely fails in context of two-sample testing. In fact, all distribution pairs whose overall scaling is different, e.g. much larger, than the scale in which the actual difference appears will cause the method to fail. Another downside of the method is its restriction to selecting *single* Gaussian kernels. Other kernel classes, including combinations of kernels (more on this later) may not be approached.

For reasons of comparability of new methods, since the median distance is used in most literature on kernel based two-sample-testing, it will be evaluated in this work although it is expected that it is outperformed by its competitors.

3.2.2. A First General Approach: Maximising the MMD

A naive approach to achieve the goal of pushing the mean of the alternative distribution above a threshold of the null-distribution is to simply maximise the MMD statistic itself. In [Sriperumbudur et al., 2009], exactly this is done: the bandwidth σ of a Gaussian kernel k_σ is selected such that

$$k_\sigma^* = \arg\max_{k_\sigma} \left[\mathrm{MMD}^2[\mathcal{F}, p, q]\right]$$

The hope is that this pushes the mean of the alternative distribution further apart from any any high quantile of the null-distribution. The method is superior to the median based distance selection since it is applicable to *any* kernel type (recall the median method is only suitable for Gaussian kernels). This strategy is suited for any type of kernel. In addition, will is possible to use it for selecting weights of combined kernels.

However, there might be problems with the strategy itself, as will be described soon. However, before this is done, the following section examines the approach of maximising the MMD from the perspective of binary classification and expected risk. It establishes connections to a field that is well-known – leading to a second, alternative approach for maximising the MMD.

3.2.3. Connections to Binary Classification & Linear Loss

The so called *parzens windows classifier* [Schölkopf and Smola, 2001, Section 1.2] is a binary classification rule that computes the mean of the training data of two classes and then assigns each testing point to the mass centre that is closest to it. This can also be done in the feature space of a kernel and then is called *kernel classification rule*, i.e. given a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with feature map $\phi : \mathcal{X} \to \mathcal{H}$; given training data from $\mathcal{X} \times \{1, -1\}$: m_+ points labelled with 1 and m_- points labelled with -1, then for a new test point x, the label prediction y is

$$\begin{aligned} y = f(x) &= \text{sign}\left(\frac{1}{m_-}\sum_{y_j=-1} ||\phi(x) - \phi(x_j)||^2 - \frac{1}{m_+}\sum_{y_i=1} ||\phi(x) - \phi(x_i)||^2\right) \\ &= \text{sign}\left(\frac{1}{m_+}\sum_{y_i=1} k(x, x_i) - \frac{1}{m_-}\sum_{y_j=-1} k(x, x_j) + b\right) \end{aligned} \tag{3.3}$$

where b is a constant offset depending only on the training data; which becomes zero of both class means have equal distance to the origin. Hence, in this case, the class prediction is simply checking which of the terms in the above expression is larger.

Note that the above term basically is the empirical estimate of the expected value of a kernel function. This is closely related to the expression for the MMD, as for example in the proof of Lemma 4 in section 2.4.1. In fact, one can see the points as being generated by probability distributions p for points with $y = 1$, and q for $y = -1$. Define μ_+ and μ_- to be the true ratios of positive and negative labelled points in the joint distribution of p and q. The discriminant function $f : \mathcal{X} \to \mathbb{R}$ is based on a function from the feature space $\mathcal{F}$. Its linear loss loss is given by

$$L_1(f(x)) = -\frac{f(x)}{\mu_+} \qquad L_{-1}(f(x)) = \frac{f(x)}{\mu_-}$$

The minimum expected risk of the kernel classification rule is given by

$$\mathcal{R}_f = \inf_{f \in \mathcal{F}} \left(\mu_+ \int_{\mathcal{X}} L_1(f(x)) dp(x) + \mu_- \int_{\mathcal{X}} L_{-1}(f(y)) dq(y)\right) \tag{3.4}$$

The risk intuitively gets lower when the mass centres of the two classes are spatially further apart in the feature space. This fact intuitively translates to the MMD, which is the distance of the mean embeddings of two distribution in the feature space (see Lemma 4). It also supports the intuition that easily *distinguishable* distributions should

be easily *classifiable*. In [Sriperumbudur et al., 2009], these two intuitions are established as facts: a direct link between the MMD and binary classification is described. In fact, minimising the linear loss of the described kernel classification rule is exactly the same as maximising the MMD itself. Formally

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = -\mathcal{R}_f$$

where $\mathcal{F} = \{f \in \mathcal{H} \mid ||f|| \leq 1\}$ is the unit ball in the underlying RKHS. These arguments form another motivation for maximising the MMD itself.

3.2.4. Prevent Over-fitting: Cross-Validation

Due to the connection to minimising the linear loss, the above approach is closely related to *empirical risk minimisation* (see e.g. [Schölkopf and Smola, 2001, Chapter 5]) as it in general appears in binary classification and related fields such as (kernel) ridge regression [Shawe-Taylor and Cristianini, 2004, Chapter 2.2]. Approaches which minimise a loss on training data always take the risk of *overfitting* data. To prevent potential overfitting, usually *cross-validation* is performed.

Cross-validation is a general method to estimate a test-error for any algorithm that can be divided into a training and a testing phase. Roughly speaking, n-fold cross-validation partitions available data into n disjoint subsets. Afterwards, training is performed on $n-1$ of these subsets and validation is performed on the remaining one. This is done for all n combinations and the results are averaged. The binary classifier that corresponds to the MMD is described by the function $f : \mathcal{X} \to \mathbb{R}$ with

$$f(\cdot) = \frac{1}{|Z_{\text{train}}|} \sum_{x \in Z_{\text{train}}} k(x, \cdot)$$

where $Z_{\text{train}} = (X_{\text{train}}, Y_{\text{train}})$. In order to correctly classify a new point x, the result of $f(x)$ has to be as positive as possible for points $x \sim p$ and as negative as possible for points from distribution $x \sim q$. This is exactly minimising the expected risk under the linear loss, as in equation 3.4. Note the similarity to the kernel classification rule in expression 3.3 and to the expression for the MMD in the proof for Lemma 4 in section 2.4.1.

The training phase involves computing the function f via summing up all kernel values where one argument runs over training data from both distributions p and q and the other one is left open. This corresponds to computing the mean of the training data in the feature space. Implementation wise, this cannot be done since the results is a function of the feature space $\mathcal{F}$. Instead, this function is evaluated in the validation phase. The latter is to evaluate f on all validation data, i.e. computing

$$\frac{1}{|X_{\text{validation}}|} \sum_{x \in X_{\text{validation}}} f(x) - \frac{1}{|Y_{\text{validation}}|} \sum_{y \in Y_{\text{validation}}} f(y)$$

which is an estimate for the expected loss. Such a cross-validation based approach

is suggested in [Sugiyama et al., 2011, Sections 5,6]. They empirically show that this method leads to better results than the median-based heuristic.

Note that in order to evaluate the function f, it is necessary to evaluate the kernel on all pairs of training and validation folds. Therefore, it has quadratic time costs and has to store all data in memory; which disqualifies it from being used along with the linear time statistic.

However, even when being used for the quadratic time test, since cross-validation has large computational costs, it is interesting to compare its performance with kernel selection by simply maximising the MMD (which potentially overfits), which is the same approach but without a protection for over-fitting. If no over-fitting happens, i.e. the results are the same, then there is no point in performing cross-validation at all. Another advantage of cross-validation may be a more stable kernel choice since multiple results are averaged. A comparison between both methods has not been done so far and will therefore be included in the experiments section.

3.3. New Methods for Kernel Choice

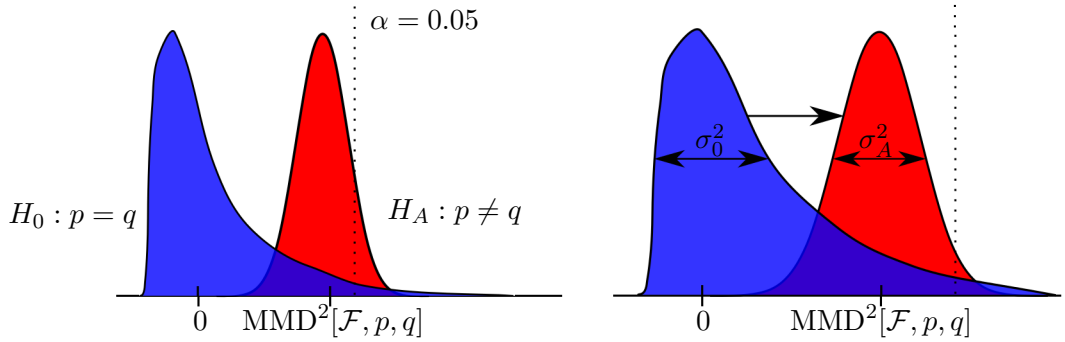
3.3.1. Problems with Maximising the MMD

Apart from theoretical considerations as given in the previous section, an intuitive hope when maximising the MMD itself is that this pushes the mean of the alternative distribution further apart from any any high quantile of the null-distribution. However, when using finite data, this remains a heuristic, since it does not guarantee that that variance of the alternative distribution does not grow. Pushing the mean upwards could at the same time increase the variance of the null- or alternative distribution. Consequently, the resulting test could even have worse performance than before since large parts of the alternative distribution tail would lie *under* a high quantile of the null-distribution – resulting in a higher type II error. See figure 3.1 for an illustration of the problem in context of the quadratic MMD statistic, as described in section 2.4.3. The same problem appears for the linear time MMD, where both null- and alternative distribution are normal.

Instead, it would be better to have a criterion that somehow controls the variance of the used statistic. Such a method will be described in the following.

3.3.2. A novel Criterion for the Linear MMD

This section proposes a new criteria for selecting a kernel in context of the linear time MMD statistic, as described in section 2.4.3. As mentioned in this section, both the null- and the alternative distributions of the statistic are normal. They only differ in their mean since they have the *same* variance. This is an appealing property that simplifies analysis.



For the quadratic time MMD, the null-distribution is an infinite sum of χ^2 variables, which leads to a long-tailed form. The resulting test for this constellation is not very good since large parts of the alternative distribution lie under an $\alpha = 0.05$ threshold of the null distribution. Given that $H_A : q \neq q$, maximising the MMD statistic without controlling the variance of null- and alternative distribution might lead to an even worse test; see right figure.

In this case, variances of both distributions grew more than can be compensated by a larger population MMD. Therefore, even though the MMD is larger here, even larger parts of the alternative distribution than in the left figure lie under an $\alpha = 0.05$ threshold – yielding a worse type II error.

Figure 3.1.: Illustration of problems that might occur when the MMD is maximised without controlling variance.

Basic Idea: Pull Tails of Null and Alternative Distribution Apart – Control Variance

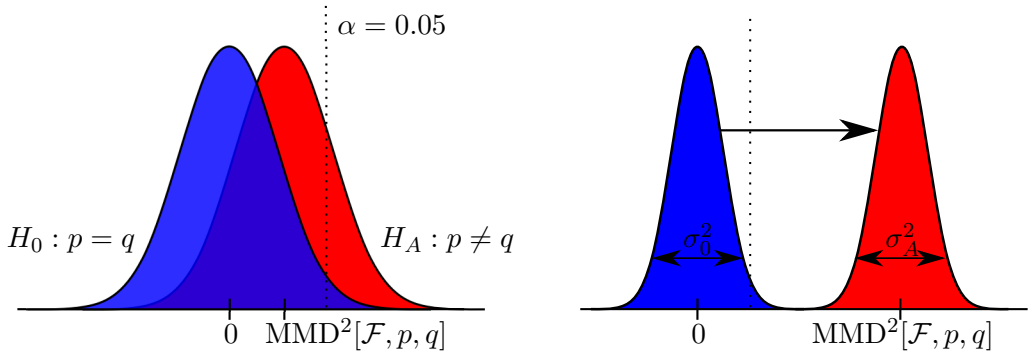
Given $H_A : p \neq q$ is true, in order to successfully reject $H_0 : p = q$, a sample from the alternative distribution has to lie above a high quantile (or threshold) of the null-distribution. To achieve a low type two error, this means that the mean and tail of the alternative distribution should lie above that threshold. The less probable it is for a sample from the alternative distribution to lie under the threshold, the lower the type two error becomes.

There are two way of achieving this goal: first, by maximising the MMD itself, as done in sections 3.2.2 and 3.2.3. Second, by minimising the statistic's variance. As described in section 3.3.1, increasing variance might break advantages of a larger MMD. Minimising it would avoids this problem.

Connection of Controlled Variance to Probability for Type II error

This is now described more formally. A type II error occurs when the used statistic, in this case the linear time MMD, falls below a threshold for a certain test level, or equally, if the p-value of the statistic is larger than the specified test level (c.f. section 2.1). In case of the linear time MMD, the distributions of the linear time MMD bother under $H_0 : p = q$ and $H_1 : p \neq q$ are normal with the *same* variance. Therefore the probability for the statistic to fall below a threshold can be expressed in terms of the cumulative distribution function Φ of the normal distribution.

Given a threshold t_α for a test level α, the empirical estimate $\mathrm{MMD}_l^2[\mathcal{F}, X, Y]$ of the



Bad Kernel: Given that $p \neq q$, the population MMD, and most of its distribution tail, lies *under* a reasonable threshold ($\alpha = 0.05$) of the null-distribution. It is very unlikely that a sample computed from data will lie above that threshold. Therefore, the test will not be able to reject $H_0 : p = q$ when it should – yielding a higher type two error.

Good Kernel: Minimizing the variance of MMD_l^2 shrinks tails of both null- and alternative distribution. Maximising MMD_l^2 at the same time pulls the distributions apart. Given that $p \neq q$, the population MMD and de-facto its complete distribution lies *above* the same threshold. Therefore it is now more likely that a sample from the population MMD lies above the threshold. This results in a lower type two error (in the above case almost zero).

Figure 3.2.: Illustration why the proposed criterion for selecting a kernel for the linear time MMD test works.

population $\mathrm{MMD}^2[\mathcal{F}, p, q]$ computed on m samples X, Y with variance $\mathrm{var}\left(\mathrm{MMD}_l^2[\mathcal{F}, X, Y]\right) = \sigma_l^2$, the probability for a type II error is

$$P(\mathrm{MMD}_l^2[\mathcal{F}, X, Y] < t_\alpha) = \Phi\left(\Phi^{-1}(1-\alpha) - \frac{\mathrm{MMD}^2[\mathcal{F}, p, q]\sqrt{m}}{\sigma_l\sqrt{2}}\right) \tag{3.5}$$

Since Φ is monotonic and the test level α as well as the sample size m are fixed, this probability decreases only with increasing ratio

$$\frac{\mathrm{MMD}^2[\mathcal{F}, p, q]}{\sigma_l} \tag{3.6}$$

This is a new criterion for selecting a kernel which indirectly minimises probability for type II errors. A depiction of the general idea can be found in figure 3.2.

A Linear Time Empirical Estimate for the Criterion

In practice, an empirical estimate using the linear time MMD statistic $\mathrm{MMD}_l^2[\mathcal{F}, X, Y]$ and its linear time variance estimate $\hat{\sigma}_l^2$ are used. For the infinite data limit, the resulting kernel equals to the one that is selected when the population values are used. In [Gretton et al., 2012c] the following result will be established as a fact (see paper for a proof).

Theorem 5. *Let $\mathcal{K} \subseteq \{k \mid k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}\}$ be a set of kernels on $\mathcal{X}$ that are bounded, i.e. $\sup_{k\in\mathcal{K};x,y\in\mathcal{X}} |k(x,y)| < K$ for some $K > 0$. Assume that for each kernel in $k \in \mathcal{K}$, the corresponding variance σ_l^2 and its estimate $\hat{\sigma}_l^2$ of the MMD are bounded away from zero. Then*

$$\left| \sup_{k\in\mathcal{K}} \frac{\mathrm{MMD}^2[\mathcal{F}, p, q]}{\sigma_l} - \sup_{k\in\mathcal{K}} \frac{\mathrm{MMD}_l^2[\mathcal{F}, X, Y]}{\hat{\sigma}_l} \right| = O_P\left(\sqrt{m}\right)$$

In addition,

$$\sup_{k\in\mathcal{K}} \frac{\mathrm{MMD}_l^2[\mathcal{F}, X, Y]}{\sigma_l} \quad \xrightarrow{P} \quad \sup_{k\in\mathcal{K}} \frac{\mathrm{MMD}^2[\mathcal{F}, p, q]}{\hat{\sigma}_l}$$

Limiting Cases of the Ratio

For the new criterion to work in optimisation, it has to be bounded from above for limiting cases of the used kernel. If this is not the case, its impossible to select an optimal kernel. It is hard to show this for the introduced criterion in general, however, for fixed, known contexts, it is possible.

Appendix A.1 examines the limiting expressions in context of Gaussian kernels and one-dimensional data where the distributions p and q have a different mean. The reasoning translates into higher dimensions but is formally more challenging and therefore avoided here. Gaussian kernels only have a single parameter that has to be selected: their bandwidth. The latter might be any positive number. It is not intuitively clear what happens in the limiting cases of zero and infinity bandwidth. Intuitively, for zero bandwidth, the Gaussian kernel always results in zero. For infinity bandwidth, it results in one. Since both $\mathrm{MMD}^2[\mathcal{F}, p, q]$ and σ_l^2 are differences of kernel expressions, both nominator and denominator of the criterion become zero.

Appendix A.1 establishes the fact that the ratio limit for very large or very small kernel sizes also is zero. Although not directly used in this work, it is a theoretical justification of why the ratio behaves well in practice.

Advantages over Existing Approaches

The proposed criterion for selecting a kernel for the linear time MMD has several advantages over existing methods

- **Linear time, constant space:** The criterion can be estimated in linear time, as described in section 3.3.2. In addition, data does not have to be stored in memory. Therefore, it may in theory be computed on arbitrarily large datasets – the only limiting factor is (linear) time costs. This is computationally much cheaper than cross-validation based minimisation of linear loss (section 3.2.3), which has to compute the complete kernel matrix. It is also cheaper than the median heuristic (section 3.2.1) since the latter has to compute all pairwise distances – although in practice, a subset of these is sufficient.

- **Indirectly Minimise Type II Error:** Since maximising the criterion implicitly minimises type II error of the underlying test, it is the optimal choice: if the type two error has a defined minimum, it is achieved. This makes it superior to simply maximising the MMD, c.f. section 3.2.2, where variance of the MMD estimate might cause problems, as depicted in figure 3.1. How this advantage manifests itself in practice will be evaluated in the experiments section.

- **Not restricted to single kernels:** The median heuristic and minimisation of linear loss using cross-validation are both criteria that may only select *single* kernels. However, as already mentioned, maximising the MMD is also possible on combinations of kernels (details follow). This makes the method far more flexible. The newly proposed method inherits this desirable property. Note that the cross-validation based approach may theoretically also work on kernel combinations – but these combinations would have to be tried *one by one*, which is computationally infeasible since every combination has quadratic time costs. In contrast, the new criterion, as maximising the MMD itself, can be optimised directly, as will be seen later.

3.3.3. A novel method using Cross-Validation

Another new method that is proposed in this work is inspired by cross-validation in context of regression and classification. In such a context, a method's true test error, i.e. performance on yet unseen data, is often approximated by applying the method to pairs of disjoint pairs of training and validation data and to average performance (for example accuracy in classification or sum of squared errors in regression) over all validation sets. This gives a nearly unbiased estimator of the true test error. In order to reduce variance, this approach is often repeated multiple times on a single dataset; where in each trial, the splits of training and validation data are different.

One very similar approach that has not yet been described for kernel selection in two sample testing works very similar: instead of the true test error, the true type II error is estimated using cross-validation. Data is split into n disjoint partitions. Then a two-sample test is performed on $n-1$ partitions while the remaining validation set is not used at all. For each two-sample test, the percentage of rejected $H_0 : p = q$ is computed. This estimator of the true type II error is averaged over all n combinations of the $n-1$ partitions. To further reduce variance, the partitioning and testing may be repeated with different partitions. See algorithm 3.2 for a formal description.

Using this approach allows to estimate type II error for a single evaluated kernel. Therefore, the method may be used for selecting kernels. As for the cross-validation based minimisation of the linear loss, c.f. 3.2.4, only *single* kernels can be evaluated. For combinations, one could try all weight combinations, but this is computational infeasible. Theoretical computational costs of the method are the same as for performing a complete two-sample test. The method performs one two-sample test per fold. Therefore, in practice, it is n times slower than a single two-sample test. Note the different to the cross-validation based minimisation of linear loss, as described in section 3.2.4: Whereas

Algorithm 3.2 Cross-validation of type II error for kernel two-sample testing.

Inputs are:

- $X, Y \subseteq \mathcal{X}$, sets of samples generated under $H_A : p \neq q$
- $\mathcal{T} : \mathcal{X}^{\hat{m}} \times \mathcal{X}^{\hat{n}} \to \{0, 1\}$, arbitary two-sample test as in definition 1: computes statistic and threshold; returns one when no type II error is made
- N, number of cross-validation folds, i.e. number of partitions
- M, number of runs cross-validation runs to perform

Output is:

- Type II error estimate for two-sample test $\mathcal{T}$.

1: **for** $i = 1 \to M$ **do**
2: Randomly partition X and Y into N disjoint sets $\mathcal{X}_1, ..., X_N$ and $\mathcal{Y}_1, ..., Y_N$.
3: **for** $j = 1 \to N$ **do**
4: $\hat{X} \leftarrow X \backslash \mathcal{X}_j$ and $\hat{Y} \leftarrow Y \backslash \mathcal{Y}_j$
5: $E_j \leftarrow 1 - \mathcal{T}(\hat{X}, \hat{Y})$
6: **end for**
7: $F_i \leftarrow \frac{1}{N} \sum_{j=1}^{N} E_j$
8: **end for**
9: **return** $\frac{1}{M} \sum_{i=1}^{M} F_i$

the latter always has quadratic costs, minimising type II error with cross-validation has linear time costs when the linear time statistic is used. This makes the method much more usable in practice.

An advantage that this method shares with the other cross-validation based approach is numerical stability – multiple results are averaged. More important, this method is the only one which *directly* minimised type II error of a test. Its closest competitor, maximising the linear time MMD statistic while minimising its variance, only implicitly minimises type II error. Therefore, this method might perform better in practice simply due to less numerical problems.

In the experiments section, this method will be compared against all competitors.

3.4. Summary

This chapter consists of three important parts.

Two-Sample Test Construction The first part introduced methods how to construct two-sample tests in practice. This is done via approximating the test statistic's null distribution. The linear time MMD is normal distributed with zero mean and therefore can be approximated by a linear time variance estimate. For the quadratic time MMD,

there exist two methods: the fast *gamma* heuristic and a consistent but slower way based on computing Eigenvalues of kernel matrices. Bootstrapping is a general but slow method that can be used with any statistic.

Existing Methods for Kernel Selection The second part of the chapter described existing methods for kernel selection for MMD-based two-sample tests. Existing methods involve using the median distance of numerical data, a heuristic based on maximising MMD estimates, and a cross-validation based method that uses the fact that the MMD can be seen as a binary classifier with a linear loss.

New Methods for Kernel Selection The last part described downsides of existing methods for kernel selection and introduced a new method for the linear time MMD. This method is based on maximising a ratio of MMD estimate and its standard deviation and yields *optimal* kernel selection in theory. Arguments for optimality exploit the fact that null and alternative distribution of the linear time MMD is normal with equal variance. At last, another new method for selecting single kernels via cross-validation of type II errors was described.

4. Combined Kernels for Two-Sample Testing

Chapter Overview This chapter describes a novel approach of how how non-negative linear combinations of a finite number of fixed kernels can be used for kernel based two-sample-testing.

After a brief description why combined kernels can be useful, section 4.1 explains why combined kernels are in fact valid kernels. In section 4.2, combined kernels are plugged into the MMD, which is written in terms of a combination of used baseline kernels. In order to construct tests, linear time estimates are needed and described in section 4.3. In addition, an expression for variance of the linear time estimate in terms of kernel weights is given in section 4.3.1. Finally the new criterion for kernel selection, as described in section 3.3.2 of chapter 3, is applied to the combined kernel case in section 4.3.2. A method for selecting kernel weights w.r.t. the criterion, based on convex optimisation, is described in section 4.3.3. A generalisation of the state-of-the-art method for kernel selection (maximising the MMD itself), concerning the use of combined kernels based on L_2 norm regularisation is provided in section 4.3.4. Finally, section 4.4 provides a summary of most important contents of this chapter.

Literature & Contributions This chapter contains mostly new methods that are related to others that have previously been published, but have not yet been applied within the context covered by this report.

In general, combined kernels have been studied under the term *multiple kernel learning (MKL)* in context of classification and regression, [Rakotomamonjy et al., 2008], [Sonnenburg et al., 2006], or [Argyriou et al., 2006]. While this field is well studied, statistical two-sample testing has not yet been connected with MKL-style approaches.

Arguments why combined kernels are valid kernels in section 4.1 can be found in books on kernels, such as [Schölkopf and Smola, 2001] or [Shawe-Taylor and Cristianini, 2004]. Expressions for the MMD in terms of kernel weights of combined kernels (section 4.2), as well as linear time estimates have not yet been published. The same holds for using the new criterion for kernel selection within this MKL-style framework in section 4.3.2. The optimisation procedure in section 4.3.3 and the generalisation of maximising the MMD itself in section 4.3.4 are also newly described methods. All these were included in the submission of [Gretton et al., 2012c].

Details on how to solve convex programs as they appear in this chapter can be found in [Boyd and Vandenberghe, 2004].

4.1. On Combinations of Kernels

One of the strengths of using kernels in general is that information provided by multiple kernels might be combined into a single one. Combining kernels therefore allows the incorporation of information from multiple domains into any method that is based on kernels. As an example, consider data where relevant differences for two-sample testing are contained within multiple scaling levels. Using a single kernel would not be able to capture this fact, but a combination of two kernels where one focusses on one length scale, would. Moreover, even different *types* of kernels (on the same domain) might be combined. For example if one wants to do a two-sample-test on sequence based biological data, one might combine kernels that only work on the sequence with kernels that work on bio-chemical properties.

This section will briefly describe why non-negative (or conical) linear combinations of kernels are valid kernels – a fact that is needed in order to allow the methods described in this chapter. Given a number of kernels $k_i : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, (1 \leq i \leq d)$, and coefficients $\beta_i \geq 0, (1 \leq i \leq d)$, then

$$\sum_{i=1}^{d} \beta_i k_i(x, y)$$

is a valid kernel. This can be seen easily by noting that kernels may be represented as positive definite matrices; the sum of such two matrices is positive definite by definition. Furthermore, scaling a positive definite matrix with a positive number does not change the sign of its Eigenvalues and therefore not change its positive definiteness. Formally, given two kernel kernels $k, l : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, data $\mathbf{x}_1, ..., \mathbf{x}_m \in \mathcal{X}$, the corresponding kernel matrices $(K)_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $(L)_{ij} = k(\mathbf{x}_i, \mathbf{x}_j), (1 \leq i, j \leq m)$ are positive definite, c.f. Lemma 2, i.e. for any $\mathbf{a} \in \mathbb{R}^m, \mathbf{a} \neq \mathbf{0}$

$$\mathbf{a}^T K \mathbf{a} \geq 0 \qquad \mathbf{a}^T L \mathbf{a} \geq 0$$

In order to combine k and l, let $\beta_1, \beta_2 \geq 0$ be non-negative scalars. The corresponding matrix of the construct $(\beta_1 k + \beta_2 l) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with $(\beta_1 k + \beta_2 l)(\mathbf{x}, \mathbf{y}) = \beta_1 k(\mathbf{x}, \mathbf{y}) + \beta_2 l(\mathbf{x}, \mathbf{y})$ is given by $H := \beta_1 K + \beta_2 L$. For any $\mathbf{a} \in \mathbb{R}^m, \mathbf{a} \neq \mathbf{0}$, it holds

$$\begin{aligned}
& \mathbf{a}^T H \mathbf{a} \\
= \; & \mathbf{a}^T (\beta_1 K + \beta_2 L) \mathbf{a} \\
= \; & \mathbf{a}^T \beta_1 K \mathbf{a} + \mathbf{a}^T \beta_2 L \mathbf{a} \\
= \; & \mathbf{a}^T \beta_1^{\frac{1}{2}} K \beta_1^{\frac{1}{2}} \mathbf{a} + \mathbf{a}^T \beta_2^{\frac{1}{2}} L \beta_2^{\frac{1}{2}} \mathbf{a} \\
= \; & \mathbf{b}_1^T K \mathbf{b}_1 + \mathbf{b}_2^T L \mathbf{b}_2 \\
\geq \; & 0
\end{aligned}$$

where the square roots of β_1, β_2 exist since $\beta_1, \beta_2 \geq 0$. Defining $\mathbf{b}_1 := \beta_1^{\frac{1}{2}}\mathbf{a}$ and $\mathbf{b}_2 := \beta_2^{\frac{1}{2}}\mathbf{a}$ leads to the result. The last inequality holds since K and L are positive definite. By induction, it holds that any non-negative linear combination of kernel matrices is a positive definite matrix and therefore the corresponding function is a valid kernel according to the opposite direction of Lemma 2.

4.2. Combined Kernels for the MMD

In order to effectively use the power of combined kernels within the context of the MMD, it has to be written in terms of base kernels of the linear combination. To this end, the MMD population expression is written in terms of kernel weights. This means that all MMD based expressions from section 2.4.3 have to be written in terms of these weights.

Let $\mathcal{F}_k$ be a RKHS with corresponding reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$; let p and q be two Borel probability measures; and let $\mu_k(p), \mu_k(q)$ be mean embeddings of p and q, as described in section 2.4.1. Define the (population) MMD for a single kernel k, corresponding to Lemma 5 in section 2.4.3, as

$$\begin{aligned}\eta_k(p,q) &:= \mathrm{MMD}^2[\mathcal{F}_k, p, q] \\ &= ||\mu_k(p) - \mu_k(q)||^2_{\mathcal{F}_k} \\ &= \mathbf{E}_{x,x'}k(x,x') + \mathbf{E}_{y,y'}k(y,y') - 2\mathbf{E}_{x,y}k(x,y)\end{aligned}$$

where $x, x' \sim p$ and $y, y' \sim q$ are iid. A shorter expression is obtained by introducing

$$h_k(x,x',y,y') = k(x,x') + k(y,y') - k(x,y') - k(x',y)$$

and then writing

$$\eta_k(p,q) = \mathbf{E}_{x,x',y,y'}h_k(x,x',y,y') =: \mathbf{E}_v h_k(\mathbf{v})$$

where $\mathbf{v} := (x, x', y, y')$ is a vector of random variables and $h_k(\mathbf{v}) := h_k(x, x', y, y')$

Now, kernels for this expression are combined from a family of baseline kernels: let $\{k_u\}_{u=1}^d$ be a set of positive definite functions of the form $k_u : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and let $D > 0$. The kernel family from which a kernel will be chosen is the set of all bounded non-negative combinations, i.e.

$$\mathcal{K} = \left\{ k = \sum_{i=1}^d \beta_i k_i \quad \text{where} \quad \sum_{i=1}^d \beta_i \leq D \text{ and } \beta_i \geq 0 \text{ for } 1 \leq i \leq d \right\} \tag{4.1}$$

where each $k \in \mathcal{K}$ is connected with a unique RKHS. Plugging such a kernel into the expression for the squared MMD gives

$$\eta_k(p,q) = ||\mu_k(p) - \mu_k(q)||^2_{\mathcal{F}_k} = \sum_{i=1}^d \beta_i \eta_i(p,q)$$

where $\eta_i(p,q)$ is the population MMD for a *single baseline kernel* k_i. Using

$$\begin{aligned}\eta_i &:= \eta_i(p,q) = \mathbf{E}h_i \\ \mathbf{E}h_i &:= \mathbf{E}_{\mathbf{v}}h_i(\mathbf{v}) \\ \mathbf{h} &:= (h_1,...,h_d)^T \in \mathbb{R}^{d\times 1} \\ \boldsymbol{\beta} &:= (\beta_1,...,\beta_d)^T \in \mathbb{R}^{d\times 1} \\ \boldsymbol{\eta} &:= (\eta_1,...,\eta_d)^T \in \mathbb{R}^{d\times 1}\end{aligned}$$

(where the first two expressions correspond to a single baseline kernel; the last two expressions are vectors where each component corresponds to a single baseline kernel), a shorter notation is

$$\eta_k(p,q) = \mathbf{E}(\boldsymbol{\beta}^T\mathbf{h}) = \boldsymbol{\beta}^T\boldsymbol{\eta} \tag{4.2}$$

This expression has an intuitive interpretation: the population MMD using a non-negative combination of baseline kernels is simply the same combination of population MMDs for the single kernels.

4.3. Linear Time Estimates and Weight Optimisation

In the following, a linear time estimate of the population MMD for combined kernels in expression 4.2 is established. Since the expression corresponds to a combination of MMDs for single kernels, linear time estimates for single MMDs, as described in section 2.4.3, can be utilised.

Assume m is even for notational ease and redefine $X = \{x_1,...,x_m\}, Y = \{y_1,...,y_m\}$ are i.i.d. samples from p and q respectively (these were random variables before). A linear time estimate for expression 4.2 can be obtained by using the same strategy as for the linear time estimate in expression 2.3 in section 2.4.3 – simply take an average of independent kernel terms, i.e.

$$\hat{\eta}_k := \frac{2}{m}\sum_{i=1}^{m/2} h_k(\mathbf{v}_i) = \boldsymbol{\beta}^T\hat{\boldsymbol{\eta}}$$

where

$$\begin{aligned}\mathbf{v}_i &= (x_{2i-1}, x_{2i}, y_{2i-1}, y_{2i}) \\ h_k(\mathbf{v}_i) &= k(x_{2i-1}, x_{2i}) + k(y_{2i-1}, y_{2i}) - k(x_{2i-1}, y_{2i}) - k(x_{2i}, y_{2i-1})\end{aligned}$$

and where $\hat{\boldsymbol{\eta}}$ is a vector where every component is the empirical (linear time) estimate of $\mathbf{E}_{\mathbf{v}}(h_k(\mathbf{v}))$ using a single baseline kernel. Note that this expression does not depend on random variables but on actual sample data. In fact, this is exactly the same expression as 2.3 in a new notation. This is done since the estimate needs to be written in terms of

kernel weights in order to select these later. The intuitive interpretation, again, is that the MMD for kernel combinations is simply the very same combination of linear time estimates of the underlying baseline kernels.

As in Lemma 8, this statistic is expected to be normally distributed with zero mean under the null-hypothesis $H_0 : p = q$, positive mean under the alternative hypothesis $H_A : p \neq q$ and equal variance for both hypotheses. This will now be investigated.

4.3.1. Null-distribution and Variance Estimates

Similar to the linear time MMD with a single kernel, the null-distribution for the combined kernel estimate is also Gaussian. This is, as before, due to the fact that $\hat{\eta}_k$ is a simple average of (independent) random variables – and therefore given by the central limit theorem (see for example [Serfling, 1980, Section 1.9]). As in Lemma 8, $\hat{\eta}_k$ converges in distribution to a Gaussian as

$$m^{\frac{1}{2}} \left(\hat{\eta}_k - \eta_k(p, q)\right) \xrightarrow{D} \mathcal{N}(0, 2\sigma_k^2)$$

where

$$\sigma_k^2 = \mathbf{E}_{\mathbf{v}} h_k^2(\mathbf{v}) - [\mathbf{E}_{\mathbf{v}} h(\mathbf{v})]^2$$

This is just another notation of already established results. However, in order to construct a test, the variance of this distribution needs to be known and estimated in terms of the kernel weights $\boldsymbol{\beta}$. To do so, simply apply the definition of variance and expand. The population expression ($\mathbf{h}$ is a random vector) then is

$$\begin{aligned}
\sigma_k^2 = \operatorname{var}(\boldsymbol{\beta}^T \mathbf{h}) &= \mathbf{E}[(\boldsymbol{\beta}^T \mathbf{h} - \boldsymbol{\beta}^T \mathbf{E}(\mathbf{h}))(\boldsymbol{\beta}^T \mathbf{h} - \boldsymbol{\beta}^T \mathbf{E}(\mathbf{h}))^T] \\
&= \mathbf{E}[(\boldsymbol{\beta}^T \mathbf{h} - \boldsymbol{\beta}^T \boldsymbol{\eta})(\boldsymbol{\beta}^T \mathbf{h} - \boldsymbol{\beta}^T \boldsymbol{\eta})^T] \\
&= \mathbf{E}[(\boldsymbol{\beta}^T (\mathbf{h}\mathbf{h}^T - 2\mathbf{h}\boldsymbol{\eta}^T + \boldsymbol{\eta}\boldsymbol{\eta}^T)\boldsymbol{\beta}^T] \\
&= \boldsymbol{\beta}^T (\mathbf{E}[\mathbf{h}\mathbf{h}^T] - 2\mathbf{E}[\mathbf{h}]\boldsymbol{\eta}^T + \boldsymbol{\eta}\boldsymbol{\eta}^T)\boldsymbol{\beta}^T \\
&= \boldsymbol{\beta}^T (\mathbf{E}[\mathbf{h}\mathbf{h}^T] - \boldsymbol{\eta}\boldsymbol{\eta}^T)\boldsymbol{\beta}^T \\
&= \boldsymbol{\beta}^T \operatorname{cov}(\mathbf{h})\boldsymbol{\beta}^T
\end{aligned} \tag{4.3}$$

To get an empirical estimate $\hat{\sigma}_k$, replace the covariance $\operatorname{cov}(\mathbf{h})$ by its empirical covariance matrix of the entries of the vector $\hat{\boldsymbol{\eta}}$. This is possible to compute in linear time.

Having described both a linear time statistic and its variance in terms of kernel weights β_i of a non-negative linear combination of kernels, it is now possible to select optimal weights for a two-sample test. This approach that generalises the described criterion of maximising linear time MMD statistic over its standard deviation, c.f. section 3.3.2 to a multiple kernel case. The strategy inherits all desirable properties from the described criterion – namely that the selected kernel weights are optimal. The next section describes how to optimise for $\boldsymbol{\beta}$ in practice.

4.3.2. Optimal Kernel Choice

Using the new criterion for optimal kernel selection as described in section 3.3.2, the goal is to select optimal weights $\boldsymbol{\beta}^*$ to get an optimal kernel $k^* = \sum_{i=1}^{d} \beta_i^* k_i$ from the family $\mathcal{K}$ in expression 4.1 in such way that it maximises the objective

$$k^* = \sup_{k \in \mathcal{K}} \frac{\eta_k(p, q)}{\sigma_k} \tag{4.4}$$

where $\eta_k(p, q)$ is the population MMD in terms of kernel weights $\boldsymbol{\beta}$ from expression 4.2; and σ_k is the square root of the variance of the linear time estimate for $\eta_k(p, q)$ from expression 4.3. Maximising this ratio minimises type II error of the resulting two-sample test due to the same reasons that applied for using the criterion with a single kernel, c.f. section 3.3.2.

In terms of kernel weights, the objective for maximisation is

$$\alpha(\boldsymbol{\beta}; \eta_k(p, q), \mathbf{h}) := \frac{\eta_k(p, q)}{\sigma_k} = \frac{\boldsymbol{\beta}^T \boldsymbol{\eta}}{\sqrt{\boldsymbol{\beta}^T \operatorname{cov}(\mathbf{h}) \boldsymbol{\beta}}} \tag{4.5}$$

In practice, both nominator and denominator of this expression are unknown and have to be estimated. This is done using the established linear time estimates from above. The objective becomes

$$\alpha(\boldsymbol{\beta}; \hat{\eta}_k, Q) = \frac{\hat{\eta}_k}{\hat{\sigma}_k} = \frac{\boldsymbol{\beta}^T \hat{\boldsymbol{\eta}}}{\sqrt{\boldsymbol{\beta}^T Q \boldsymbol{\beta}}} \tag{4.6}$$

where Q is the empirical estimate of $\operatorname{cov}(\mathbf{h})$ using entries of the vector $\hat{\boldsymbol{\eta}}$, which can be computed in linear time.

In the infinite sample limit, the kernel selected by maximising the empirical estimate in expression 4.6 converges to the one that would be selected if optimisation would have taken place on the popluation objective in expression 4.5. This is guaranteed due to theorem 5.

4.3.3. Convex Optimisation for Kernel Weights

To maximise expression 4.6, first note, that its solution is invariant to a scaled $\boldsymbol{\beta}$, since it is a homogeneous function[1] of order zero in $\boldsymbol{\beta}$. Therefore, the bound restriction $||\boldsymbol{\beta}||_1 \leq D$ that is inherited from the form of the kernel family in expression 4.1 can be dropped – leaving only the non-negativity constraint on $\boldsymbol{\beta}$. The problem in expression

[1] A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be homogeneous of degree k if for all $t > 0$, it holds $f(tx) = t^k f(x)$ for all $x \in \mathbb{R}^n$.

4.6 therefore becomes to select $\boldsymbol{\beta}$ such that

$$\boldsymbol{\beta}^* = \arg\max_{\boldsymbol{\beta} \succeq 0} \alpha(\boldsymbol{\beta}; \hat{\boldsymbol{\eta}}, \hat{Q}) = \arg\max_{\boldsymbol{\beta} \succeq 0} \frac{\boldsymbol{\beta}^T \hat{\boldsymbol{\eta}}}{\sqrt{\boldsymbol{\beta}^T \hat{Q} \boldsymbol{\beta}}} \tag{4.7}$$

In order to solve the problem in expression 4.7, the sign of $\hat{\boldsymbol{\eta}}$ will play a role. Since it is an unbiased estimator of the MMD, which under the null-distribution has zero mean, it may be negative. This case is rarely interesting in practice since it always is below any variance threshold (which is always positive). Therefore, any two-sample test would fail.

Two cases are considered now. The first is: at least one element of $\hat{\boldsymbol{\eta}}$ is positive. Then $\alpha(\boldsymbol{\beta}^*; \hat{\boldsymbol{\eta}}, Q)$ is for sure also positive since the component of $\boldsymbol{\beta}$ that belongs to the positive element of $\hat{\boldsymbol{\eta}}$ will take as much weight as is needed to make the function value of $\alpha(\boldsymbol{\beta}; \hat{\boldsymbol{\eta}}, \hat{Q})$ positive since the latter is maximised. Consequently, it is possible to square the objective. In addition, again since $\alpha^2(\boldsymbol{\beta}; \hat{\boldsymbol{\eta}}, \hat{Q})$ is scale-invariant in $\boldsymbol{\beta}$, the nominator may be fixed to $\boldsymbol{\beta}^T \hat{\boldsymbol{\eta}} = 1$. Using both facts allows to rewrite the problem

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^* &= \arg\max_{\boldsymbol{\beta} \succeq 0} \alpha(\boldsymbol{\beta}; \hat{\boldsymbol{\eta}}, Q) \\
&= \arg\max_{\boldsymbol{\beta} \succeq 0} \alpha^2(\boldsymbol{\beta}; \hat{\boldsymbol{\eta}}, Q) \\
&= \arg\max_{\boldsymbol{\beta} \succeq 0} \frac{\boldsymbol{\beta}^T \hat{\boldsymbol{\eta}}}{\boldsymbol{\beta}^T Q \boldsymbol{\beta}} \\
&= \arg\min_{\boldsymbol{\beta} \succeq 0, \boldsymbol{\beta}^T \hat{\boldsymbol{\eta}} = 1} \boldsymbol{\beta}^T Q \boldsymbol{\beta}
\end{aligned}$$

which is equivalent to the convex program

$$\min\{\boldsymbol{\beta}^T Q \boldsymbol{\beta} : \boldsymbol{\beta}^T \hat{\boldsymbol{\eta}} = 1, \boldsymbol{\beta} \succeq 0\} \tag{4.8}$$

See [Boyd and Vandenberghe, 2004] for details on convex optimisation. Expression 4.8 can directly be plugged into any solver for quadratic programs. The computational costs to solve it is quadratic in $\mathcal{O}(d^2)$, i.e. quadratic in the number of kernels.

4.3.4. Maximising the MMD via Weight Norm Regularisation

Related to the previous section, it is also possible to selects weights of combined kernels in such way that the MMD statistic itself is maximised. This corresponds to the existing method for kernel selection as described in section 3.2.2. Even though this method in theory is not optimal as the above strategy, it is a potentially useful generalisation of the strategy of maximising the MMD. It turns out that different norm regularisations correspond to selecting single or combined kernels. In addition, these methods work with the quadratic time statistic. Therefore, and in order to have a competitor for optimal kernel weight choice as described in the previous section, the approach is briefly described here.

Using L_2 norm regulaisation, the combined MMD can simply be maximised by solving the convex program

$$\min\{||\boldsymbol{\beta}||_2^2 : \boldsymbol{\beta}^T\hat{\boldsymbol{\eta}} = 1, \boldsymbol{\beta} \succeq 0\} \tag{4.9}$$

Note that this program is equal to the one in expression 4.8 with $Q = I$. The objective function is the L_2 norm of kernel weights. Since it is minimised, every kernel that gets weight is penalised in a quadratic way. Therefore, and because of the constraint $\boldsymbol{\beta}^T\hat{\boldsymbol{\eta}} = 1$, the kernels which contribute most MMD are selected: they allow to full-fill the constraints "fastest", i.e. with least mass on their weights and therefore with least objective function costs. Because of the quadratic costs, weights are smoothly distributed around the kernel which yields maximum MMD. As for expression 4.8, costs for solving this are quadratic in the number of kernels, i.e. in $\mathcal{O}(d^2)$.

A final observation is that maximising the MMD itself as described in section 3.2.2 simply corresponds to solving

$$\min\{||\boldsymbol{\beta}||_1^2 : \boldsymbol{\beta}^T\hat{\boldsymbol{\eta}} = 1, \boldsymbol{\beta} \succeq 0\}$$

which uses L_1 norm regularisation on the weights. Since the program minimises the absolute sum of weights, the one single kernel that yields maximum MMD is selected – it needs minimum mass to full-fill the constraints while not being punished for being too large (costs are linear).

4.4. Summary

This chapter generalised the notation of the MMD in terms of weights of non-negative combinations of a set of single baseline kernels. The MMD in that case simply becomes the same combination of MMDs that correspond to the single kernels. Same holds for linear empirical estimates. The variance of this estimate can also be expressed in terms of kernel weights, and a linear empirical estimate can be expressed using a covariance matrix computed from sample data. These expressions then can be used to generalise the described criterion for optimal kernel selection for the linear time MMD statistic to the multiple kernel case via solving a convex program. The method then allows to select optimal kernel weights for non-negative kernel combinations. A similar approach, based on L_2 regularisation, was described in order to select kernel weights in such way that the MMD statistic itself is maximised.

5. Experimental Results

Chapter Overview This chapter describes all experiments that were carried out during this work. This includes a description of existing and new benchmark datasets for two-sample testing, experimental results of all existing and new methods for kernel selection, and empirical investigations of earlier posed questions.

Section 5.1 begins by describing and illustrating existing and new benchmark dataset for two-sample testing. An analysis of difficulties and structure of underlying problems and expectations for experimental results are given, preceded by a short overview in section 5.1.1. An important requirement for linear time tests in this work is examined section 5.2: The linear time variance estimate that is used to construct tests is empirically compared to its ground truth. Section 5.3 continues by evaluating all previously introduced kernel selection methods on all introduced datasets. General experimental design is described in section 5.3.1. In particular, existing state-of-the-art methods are compared against newly proposed competitors. Results on each dataset are analysed and interpreted. Afterwards, in section 5.4, linear time tests are compared against quadratic time tests in two particular contexts: large-scale two-sample testing and infinite data/finite time constraints. Finally, section 5.5 summarises most important experimental results and their implications.

All experiments were performed on the cluster computer of the Gatsby Computational Neuroscience unit[1].

Literature & Contributions While some experiments are built on existing ones that have previously been published, none of them has yet been described anywhere in detail.

Datasets *mean*, *var*, and *sine* in section 5.1 have been used in [Gretton et al., 2012a] and [Sriperumbudur et al., 2009]. All other datasets, namely *blobs*, *selection*, *music* and *am* are newly described in this work. Evaluation of the number of samples that are necessary in order to safely perform the described linear time two-sample test in section 5.2 has not yet been described in published works. Datasets *mean*, *var*, and *sine* have been evaluated with existing methods for kernel selection in [Gretton et al., 2012a] and [Sriperumbudur et al., 2009]. However, comparison to newly described kernel selection methods, to cross-validation based methods, and to methods on combined kernels are an original contribution of this work. The same holds for all results on datasets *blobs*, *selection*, *music* and *am*. In addition, experiments to support use of using linear instead quadratic time tests in section 5.4 are a original contribution. Some results of this chapter also went into [Gretton et al., 2012c], which was submitted while this work was written.

[1] `http://www.gatsby.ucl.ac.uk/`

5.1. Datasets: Descriptions, Difficulties and Expectations

This section introduces all datasets on which methods described in this work will be evaluated. Most described datasets consist of synthetic data. This choice was made since it is easy to modify difficulty for two-sample tests on such data: it allows a more in-depth analysis of advantages and disadvantages of methods since they may be tried on a wider set of difficulties of one particular problem. In real world datasets, such wide difficulty ranges of one problem are usually are not available.

Another advantage of using synthetic data is that its structure and therefore the structure of the underlying problem in two-sample testing is known. Therefore, methods may be compared in their ability of solving problems of a particular structure – which gives hints when to use them. Results are also easier to interpret: in real world data, it is often not clear where the actual problem is situated, i.e. which parts of the data make two distributions distinct. Consequently, it is harder to guess why methods fail or succeed. Using synthetic data, such differences are known. In addition, synthetic datasets may be designed in such way that results demonstrate usefulness of a new method in an isolated way. However, in order to include some real world data, a hybrid dataset which is a mixture of synthetic and real-world data is also described.

In the following, descriptions of all dataset are provided along with analysis of underlying difficulties and expected performance of different methods of two-sample testing.

5.1.1. Overview

Since six datasets are introduced, a brief summary of existing and newly described datasets is given before these are introduced in detail. In order to understand the basic ideas of each dataset, these brief descriptions should be sufficient on a first reading. In particular, visualisations of data should be considered while detailed descriptions are skipped. The reader may refer back to these later when necessary.

Existing Datasets *Mean* and *Var* Datasets consist of simple Gaussian random vectors which differ in mean or variance in one dimension. Difficulty arises from the numerical extent of this difference as well as from the number of "useless" dimensions added. These datasets are taken from [Gretton et al., 2012a]. They are described in detail in sections 5.1.2 and 5.1.3. See figure 5.1 for visualisations.

The *Sine* dataset consists of one-dimensional Gaussian noise where in one distribution a sinusoidal perturbation is added to the generating PDF. Difficulty grows with a higher frequency of this perturbation. Introduced in [Sriperumbudur et al., 2009], described in detail in section 5.1.4. A visualisation can be found in figure 5.2.

For the above datasets, the number of samples necessary to show a difference between existing and new methods for kernel selection is very large as will be reported later in this chapter. While working on this thesis, it therefore became clear that there is a need for more difficult problems, where different methods' performance deviate – to reveal statistical (dis-)advantages. In fact, all three yet described datasets are in-favour of existing methods since differences of distributions are situated at length-scales similar

to the overall scale of used data. Therefore, these methods do work well – they focus on the overall data scaling. In order overcome this situation, datasets are designed in such way that they represent harder problems for two-sample testing.

New Datasets The *Blob* dataset consists of a grid-like mixture of two-dimensional Gaussians. A difference between two distributions is established via slightly stretching the Gaussian's shape and then rotating them individually. These de-formed blobs are then placed on the same grid as their normal Gaussian counterparts. It is clear that distinguishing characteristics between these two distributions are situated at the scale where the Gaussian blobs are located and not in the scaling of the overall grid. See section 5.1.5 for details and in particular figure 5.3 for data plots.

In order to simulate the above difficulty on real data, the *am* dataset consists of a sine carrier wave whose amplitude is modulated by music: each distribution contains a different piece of music. Differences between distributions then happen at the envelope of the resulting waves and not at the overall present sine carrier wave. In order to compare to baseline results, the music only forms another dataset called *music*. Described in detail in sections 5.1.7 and 5.1.8. See figure 5.5 for an illustration of the amplitude modulation technique and figure 5.6 for sample plots.

The *selection* dataset invented in order to illustrate usefulness of combined kernels for two-sample testing. It contains random Gaussian vectors where only a subset of dimensions are relevant, i.e. in these particular dimensions, the means are randomly shifted. This does not always happen, but instead with equal probability occurs in just one dimension at once. Consequently, if all relevant dimensions are considered at once, it is easier for a test to distinguish the two underlying distributions. Weights of combined kernels should reflect this behaviour. A simple visualisation is provided in figure 5.4.

5.1.2. Difference in Mean

The *mean* dataset contains samples from random variables $x \sim p$, and $y \sim q$, which are Gaussian random vectors where each component is standard normal distributed – except for the first dimension of y, whose mean is shifted by some fixed value ϵ. Formally,

$$
\begin{aligned}
p(x_i) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x_i^2}{2}\right) && (1 \leq i \leq d) \\
p(y_1) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_1 - \epsilon)^2}{2}\right) \\
p(y_i) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y_i^2}{2}\right) && (2 \leq i \leq d)
\end{aligned}
$$

See figure 5.1 for a depiction of samples and distributions. As the data dimension grows, distinguishing both distributions becomes more difficult with the same holding for decreasing mean shift ϵ. This dataset is one of the benchmarks used in published

works and therefore is important to compare against.

5.1.3. Difference in Variance

The *var* dataset contains samples from random variables $x \sim p$, and $y \sim q$, which are Gaussian random vectors where each component is standard normal distributed except for the first dimension of y – it has a fixed a non-unit variance σ. Formally,

$$\begin{aligned} p(x_i) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x_i^2}{2}\right) && (1 \leq i \leq d) \\ p(y_1) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y_1^2}{2\sigma^2}\right) \\ p(y_i) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y_i^2}{2}\right) && (2 \leq i \leq d) \end{aligned}$$

See figure 5.1 for a depiction of samples and distributions. As for the *mean* dataset, difficulty increases in higher dimensions and smaller difference σ. This is another benchmark dataset used in published works.

5.1.4. Sinusoidal Perturbation

The *sine* dataset contains samples from one-dimensional random variables $x \sim p$, and $y \sim q$, which are both normally distributed – but y has a sinusoidal perturbation of frequency ω added to its probability distribution function, i.e.

$$\begin{aligned} p(x) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \\ q(y) &= \frac{1}{Z} \sin(\omega y) \exp\left(-\frac{y^2}{2}\right) \end{aligned}$$

where Z is a normalisation constant to ensure that q is a valid probability distribution. In practice, rejection sampling is used so there is no need to compute Z. See figure 5.2 for an illustration of samples and distributions.

This dataset is different from the *mean* and *var* sets: both p and q are univariate distributions. The problem is harder than the above higher dimensional problems since distinguishing characteristics are more subtle. Difficulty increases with increasing frequency ω – drawn samples look more and more Gaussian. A downside is that for very high frequencies, difficulty is not caused by structural but mainly by numerical problems. One finding is that a Gaussian kernel has to have a bandwidth similar to the sine frequency in order to detect differences. This dataset is used as a benchmark in published works, see [Sriperumbudur et al., 2009].

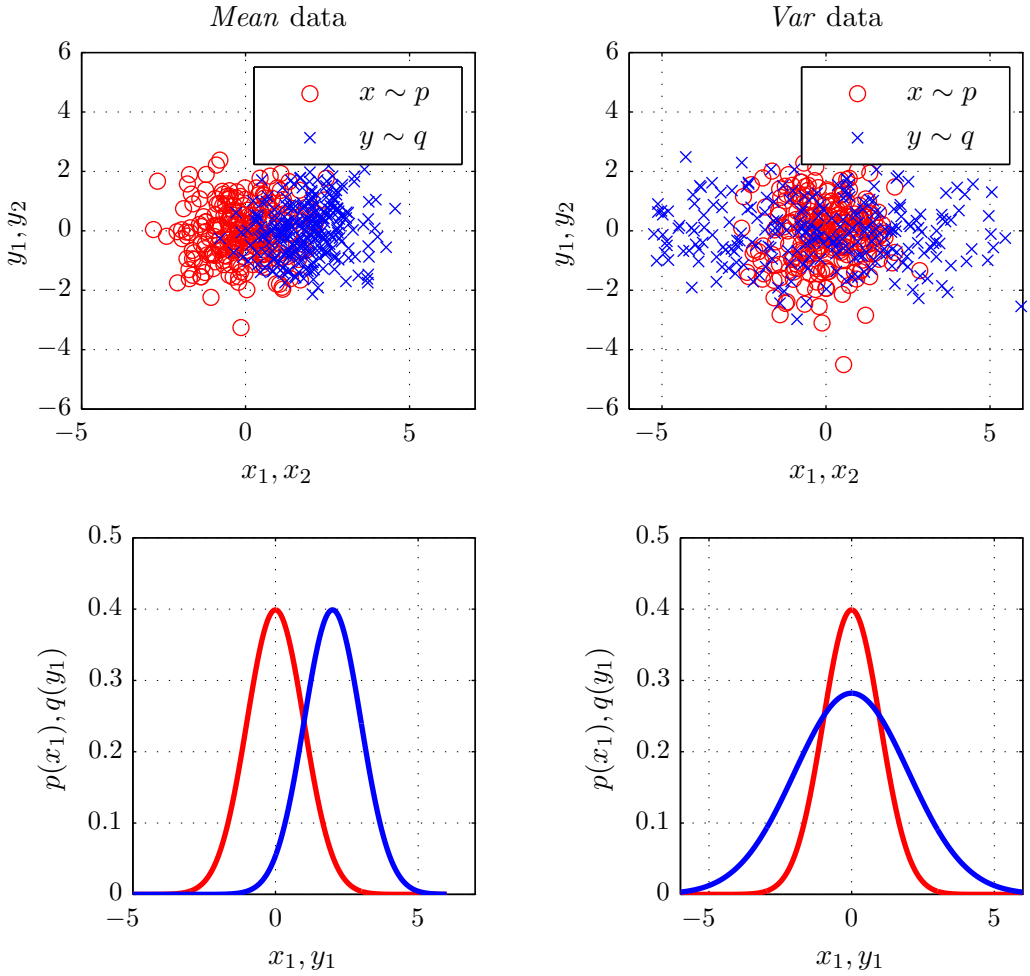


Figure 5.1.: Plots $m = 500$ samples from datasets *mean* and *var* in $d = 2$ dimensions (hard to plot for larger d). The probability distribution functions for the first dimension are drawn below. Difference in *mean* is is $\epsilon = 2$; difference in *var* is $\sigma^2 = 2$. Note that all potentially added dimensions have equal distributions for both datasets.

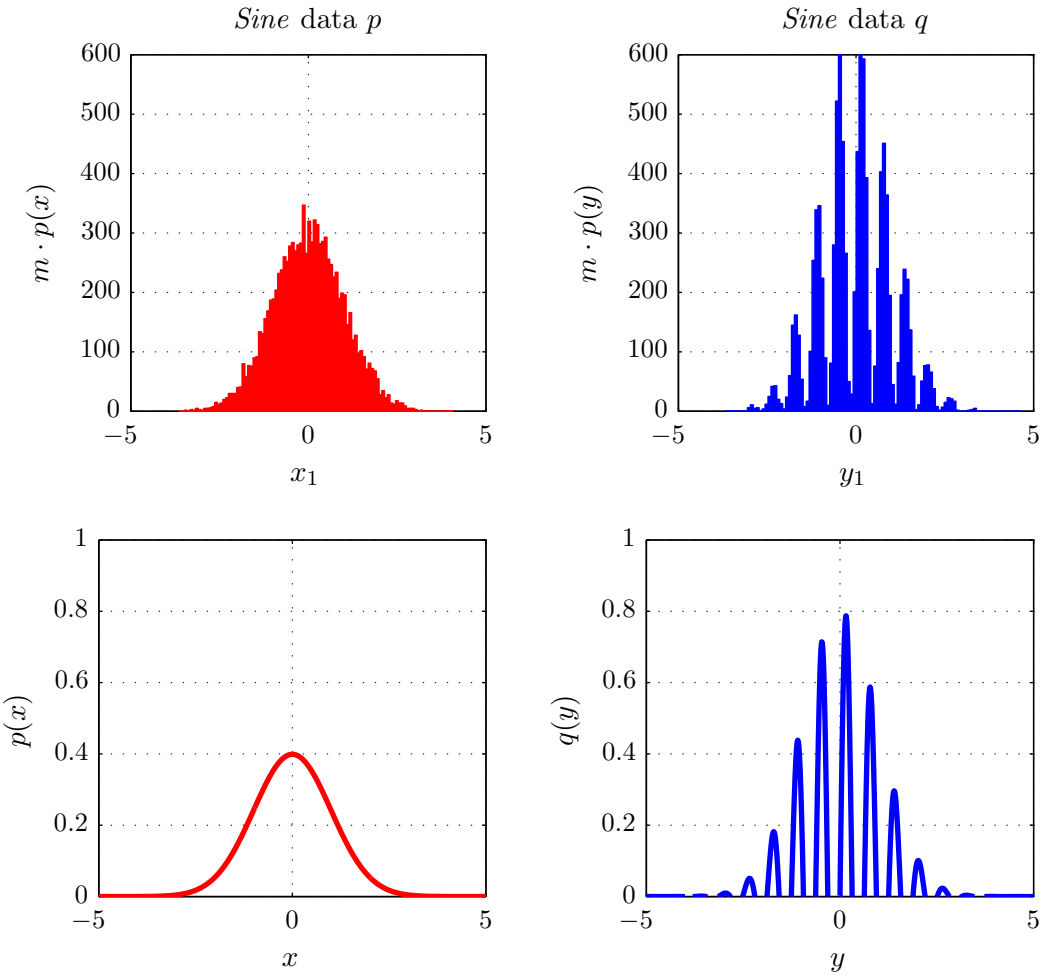


Figure 5.2.: Histograms of $m = 10000$ samples from datasets *sine* are shown on top. Data has fixed dimension $d = 1$. (Un-normalised) probability distribution functions are drawn below. Frequency of the sine is $\omega = 10$.

5.1.5. A harder Problem: Gaussian Blobs

The *blob* dataset contains samples from a grid-like mixture of t^2 equal distributed Gaussians in two dimensions. Every Gaussian in the mixture has equal probability. Random variable $x \sim p$, is a mixture of standard 2D-Gaussians whose means are distributed on a grid at distance Δ. Variable $y \sim q$ is a mixture of 2D-Gaussians on the same grid, but their shape is stretched by ϵ and rotated by angle α. Formally, this means that the first Eigenvalue of the covariance matrix is multiplied by a number ϵ, and multiplied by a 2D rotation matrix for α, i.e.

$$p(x) = \frac{1}{Z} \sum_{i=0}^{t^2-1} \mathcal{N}_x \left(\Delta \begin{pmatrix} \lfloor \frac{i}{t} \rfloor \\ \mathrm{mod}(i,t) \end{pmatrix}, I \right)$$

$$q(y) = \frac{1}{Z'} \sum_{i=0}^{t^2-1} \mathcal{N}_y \left(\Delta \begin{pmatrix} \lfloor \frac{i}{t} \rfloor \\ \mathrm{mod}(i,t) \end{pmatrix}, \Sigma \right)$$

$$\text{where } \Sigma = \left(\begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} \sqrt{\epsilon} & 0 \\ 0 & 1 \end{pmatrix} \right)^2$$

where Z and Z' are normalising constants. The covariance matrix of Gaussian in q decomposes into a rotation and stretch part. Note that the vector $(\mathrm{div}(i,t), \mathrm{mod}(i,t^2))^T$ sets coordinates in the grid for a running index i using the whole number division $\lfloor \frac{i}{t} \rfloor$ and the remaining rest of the same division $\mathrm{mod}(i,t) = i - \lfloor \frac{i}{t} \rfloor$. See figure 5.3 for a depiction of two problem datasets.

The dataset represents a complex problem for a two-sample-test to solve. Difficulty comes from two facts:

1. As with the *sine* dataset, difficulty may be increased while the dimension is fixed and small: $d = 2$. The *mean* and *var* datasets have to increase their dimension in order to effectively increase difficulty. However, this problem can be made *very* difficult by increasing the number of Gaussians t^2 and reducing the stretch of data from q. Even for humans, it is very hard to distinguish data as shown in the lower part of figure 5.3.

2. The difference of p and q appears at a length scale that is smaller than the overall scale of the data. While the *sine* dataset also has this property the ratio of relevant scaling to overall scaling shrinks with a growing number of blobs t and therefore is easily controllable without running into numerical problems as in the *sine* dataset. Due to this attribute of the *blob's* data, heuristic methods that orient themselves on overall data scaling, as for example the median distance based heuristic as described in 3.2.1, are likely to fail. In fact, any method for kernel selection has to be extremely sensible in order to take the right choice, so the expectation is that all described methods will differ a lot in their performance.

Another desirable property of this benchmark is its relation to real-world data: it simply is a mixture of Gaussian distributions. These models are well studied and have

proven to be useful in many contexts, [Barber, 2012, Section 20.3].

5.1.6. A Feature Selection Problem

The *selection* dataset was invented in order to support use of combined kernels for two-sample-testing, i.e. answering the questions which dimensions or features of data contain distinguishing characteristics of data and which ones can be easily dropped. Not only that answering this question may offer computational advantages since less data has to be taken into account, it also helps understanding an underlying problem – if important features are known, this might help to come up with models for data. While a description of feature selection for two-sample testing has not yet been published, this dataset is a standard way of illustrating feature selection for multiple kernels.

The *selection* dataset has its relevant parts in only a few of multiple dimensions. For a given dimension $d \geq 2$, data from p is standard normally distributed in every component. Data from q has its mean in certain dimensions shifted by ϵ – this occurs randomly and with equal probability in one of the first $\Delta \leq d$ components. Formally,

$$\begin{aligned} p(x_i) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x_i^2}{2}\right) && (1 \leq i \leq d) \\ q(y_i) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - \epsilon \cdot \delta(a \leq \pi))^2}{2}\right) && (1 \leq i \leq \Delta) \\ q(y_i) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y_i^2}{2}\right) && (\Delta + 1 \leq i \leq d) \end{aligned}$$

where a is drawn from the uniform distribution in $[0, 1]$ and δ is a function

$$\delta(x) := \begin{cases} 1 & x \text{ is true} \\ 0 & x \text{ is false} \end{cases}$$

See figure 5.4 for a depiction of samples and probability distribution functions. Feature selection can be performed by letting a test choose dimensions to focus on. Note that all dimensions are independent of each other here. There are two abilities of two-sample tests that can be tested on this dataset:

1. As mentioned, the dataset is suited to evaluate a test's ability in choosing appropriate dimensions, since only the first Δ dimensions are relevant – the others are completely indistinguishable. Any test that focusses on the latter will reach worse performance.

2. There is more than one dimension that is relevant to solving the problem – the first Δ. Because of that, any test that only chooses a single dimension will loose information and therefore reach a worse type II error. This can be interpreted in terms of selecting multiple combined kernels: if one univariate kernel is used for every dimension, selecting a single kernel will lead to bad results whereas combining

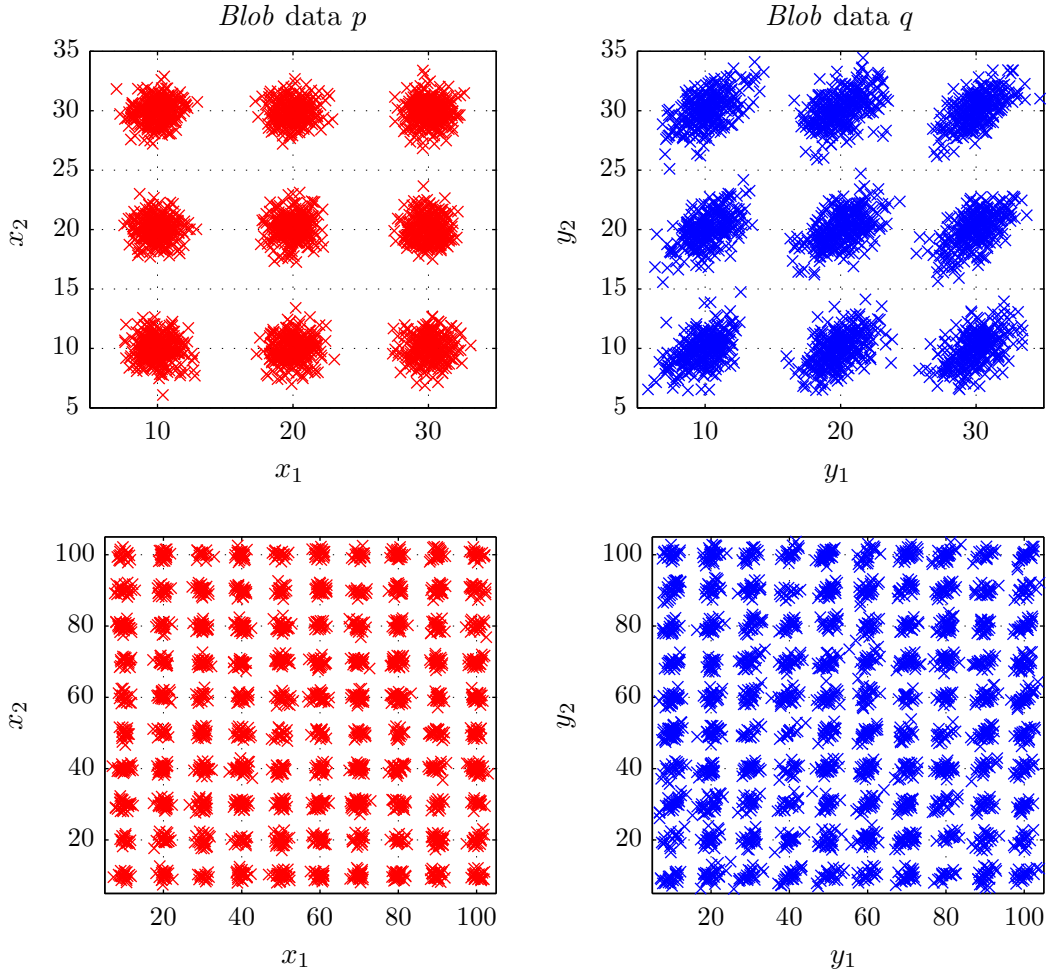


Figure 5.3.: Plots of $m = 5000$ samples from dataset *blob* in two difficulties (upper, $t = 3$, and lower, $t = 10$, plots). Data has fixed dimension $d = 2$. The angle was set to $\alpha = \frac{\pi}{4} \equiv 45$ degrees; first Eigenvalue of covariance matrix of q was set to $\epsilon = 3$. The stretch can be decreased along with the number of Gaussians being increased to create arbitrarily hard datasets, as can be seen in the lower plots.

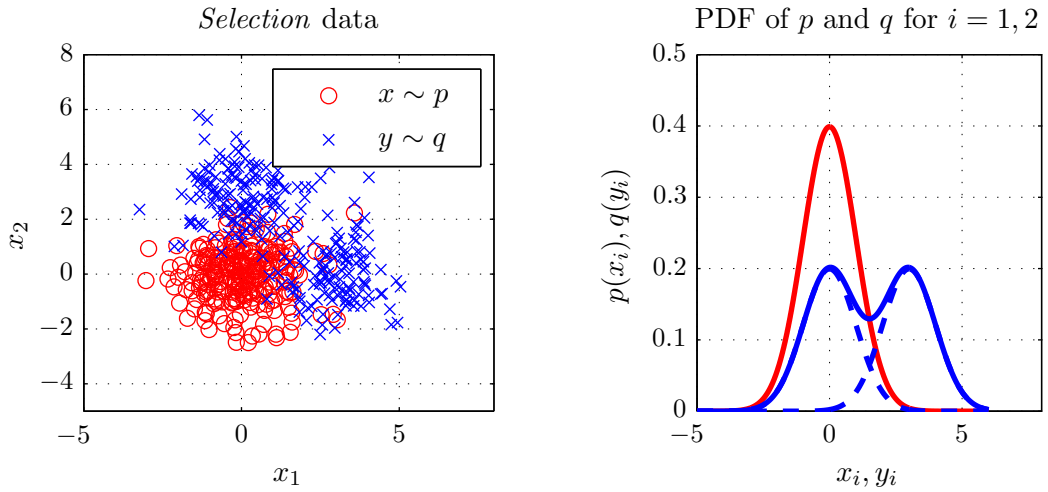


Figure 5.4.: Plot and PDF of $m = 500$ samples from dataset *selection* in $d = 2$ dimensions, where there are $\Delta = 2$ relevant dimensions. Note that the PDF are for both dimensions.

multiple kernels (here, those that focus on the first Δ dimensions) will not suffer from the described information loss. Chapter 4 deals with such combined kernels. This dataset allows the methods described there in to be tested in a known context.

5.1.7. Statistically Distinguishing Music

The *music* dataset presents the challenge of distinguishing digitalised music on the base of *unordered* small samples. It is solely based on real-world music. This approach is purely statistical in the sense that no time-induced structure is taken into account, as any time-series based model, e.g. hidden Markov models would do. The distinction of two pieces of music is simply performed on the base of present frequencies in sample data. This problem is considered since its results form a baseline that is needed in order to better understand the *am* dataset, where a carrier wave's amplitude is modulated in such way that is contains music signals – this is expanded upon later in the chapter.

The *music* dataset is based on two songs by the same artist. Pieces are selected in such way they subjectively sound similar. Music data of two songs is represented as large vectors where each entry corresponds to the music signal amplitude at a specific time (Sampling rate is 8kHz). This vector is cut into small windows of size $d = 10000$. In order to make the samples comparable to the following dataset, the songs are upsampled to a certain frequency before being split. This ensures that each sample in this dataset corresponds to the same part in the song as in the *am* dataset.

In contrast to all other data so far, difficulty of this problem is not varied by changing parameters. Therefore, results will be single numbers only. These may then be used to compare against the *am* dataset and are described when the experiment is evaluated.

5.1.8. Amplitude Modulation (AM) of Music

The *amplitude modulation (am)* dataset consists of amplitude modulated sine waves, where added signal is music. Amplitude modulation is a technique which might for example be used for transmitting information via carrier waves, as for example in radio transmission. The carrier waves amplitude is modulated in such way that its envelope reflects actual information. Carrier waves usually have a larger frequency than carried signal. See figure 5.5 for a depiction and figure 5.6 for sample plots.

There are various reasons why this kind of data is considered:

1. Distinguishing *am* data in a two-sample-test fashion is challenging since *signal*, i.e. meaningful parts, in the data has to be distinguished from the carrier wave itself. The scaling of contained information (namely scaling of the envelope) is different to the scaling of the carrier wave and therefore the overall shape of the signal. This is related to one of the difficulties present in the *blob* and *sine* datasets. However, the structure of both signal and non-signal in the *am* dataset is more complicated: non-signal is a sine wave and signal may be anything, which leads to the next point.

2. Amplitude modulation is widely used for radio broadcast – in order so transmit *speech or music*. Therefore, this dataset may be used to test methods on *real-world* data. Still, due to its hybrid nature, advantages of synthetic datasets are preserved: controllable difficulty and knowledge of underlying structure.

Formal Description

The *am* dataset uses sine waves as signal carrier. Samples consist of the same sinusoidal carrier wave of frequency ω that is modulated with two different but similar pieces of music (e.g. two songs from one CD) $M_p(i), M_q(i)$ (i is the time). Afterwards, Gaussian noise with standard deviation σ is added to increase difficulty. Multiple points are grouped to a consecutive window of length d, which is the dimension of resulting samples. Formally,

$$x_i = \begin{pmatrix} \sin(2\pi\omega(id))(\Delta + \epsilon M_p(id)) \\ \sin(2\pi\omega(id+1))(\Delta + \epsilon M_p(id+1)) \\ \vdots \\ \sin(2\pi\omega(id+d-1))(\Delta + \epsilon M_p(id+d-1) \end{pmatrix} + \Sigma$$

and

$$y_i = \begin{pmatrix} \sin(2\pi\omega(id))(\Delta + \epsilon M_q(id)) \\ \sin(2\pi\omega(id+1))(\Delta + \epsilon M_q p(id+1)) \\ \vdots \\ \sin(2\pi\omega(id+d-1))(\Delta + \epsilon M_q(id+d-1) \end{pmatrix} + \Sigma$$

Δ is an offset to prevent signal from dominating the carrier, ϵ is the scaling of the envelope in contrast to the carrier, and Σ is a random $d \times 1$ vector where each component is independently sampled from $\mathcal{N}(0, \sigma^2)$. Carrier wave frequency is set to a fixed value $\omega = 24$kHz during all experiments. Music is sampled at 8kHz.

Since music data is finite (about 30000-60000 samples in different songs), samples are taken by shuffling and then sampling without replacement in order to avoid collisions.

5.2. Convergence of Linear Time MMD Variance Estimate

As described in section 3.1.2, the linear time MMD statistic, as given in sections 2.4.3 (single kernel) and 4.3 (combined kernels), may be used along with a Gaussian approximation of the null distribution in order to construct a linear time two-sample test. To do so, the population variance of the MMD estimate has to be empirically estimated itself. This results in a consistent test, however, whether the resulting type I error is controlled properly in practice has yet to be confirmed. It is essential for the compatibility of two-sample tests that the type I error is controlled properly. In practice this can be checked by simply sampling the null distribution via bootstrapping (c.f. algorithm 3.1), and then computing its variance. This can then be compared against the linear time variance estimate. Another way to check type I errors is to compute a two sample test using the variance estimate, c.f. section 3.1.2, using data from the null hypotheses $H_0 : p = q$. The resulting type I error should be $1 - \alpha$.

Since it is very costly to estimate type I errors along with the type II errors, it is undesirable to compute them all the time. In order to empirically guarantee fixed type I errors, this section investigates the accuracy of the Gaussian approximation for the linear time MMD. The latter improves as sample size increases. Therefore, accuracy of the approximation will be investigated as a function of the sample size in order to deduce a number that is needed to get reliable results. In addition, it is instructive to see convergence of the estimate to its population value.

Figure 5.7 depicts described convergence. Underlying data is from *mean* and *blob*. The left plot is both variance estimates against the sample size. As can be seen, the variance drops for growing sample size m. However, this alone is not sufficient in order to be confident about an accurate type I error: the statistic may also decrease for larger sample size. It is the *relative* difference of variance and threshold that influences the test's type I error since thresholds are computed from the variance. In order to illustrate this, the right plot shows the difference of the variance estimates divided by the statistic itself – on a log-scale. It shows that from approximately 2^8 and 2^{13} samples from *mean* and *blob* respectively, the impact becomes negligible. This means that the approximation should only be used with at least $m = 1000$ samples. Since all experiments in this work will be done on the base of at least $m = 10000$ samples, there is no risk of getting a badly controlled type I error. The latter is usually checked in experiments anyway – while not being reported if it reveals no problems. Note that the variance of on the *blob* dataset is much larger. However, the error impact on the test also drops to almost zero from about 2^10 used samples.

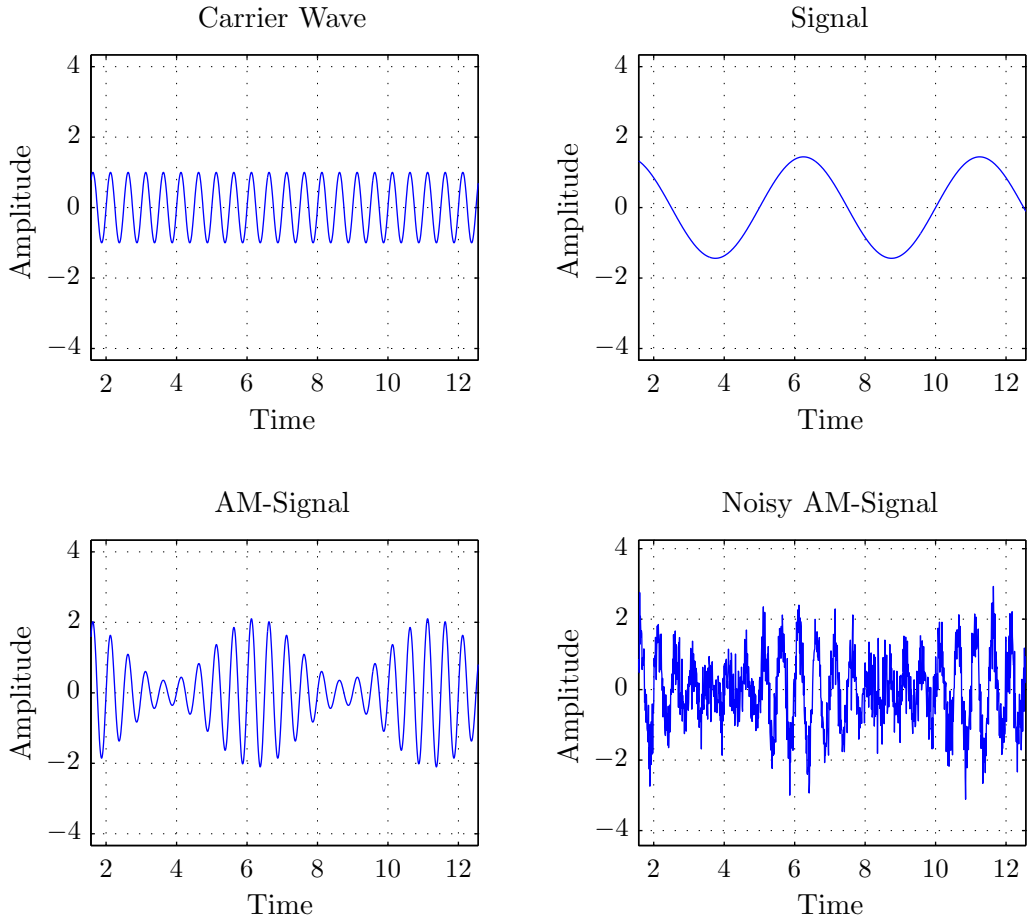


Figure 5.5.: Depiction of the modulation process used in the *am*-dataset. The amplitude of a carrier wave is modulated according to a signal (both sinusoidal here, signal has lower frequency). Note how the envelope of the am-modulated signal corresponds to the original signal. Added noise makes it harder to identify signal in the envelope. Carrier wave has frequency $\omega = 2$; signal has frequency 0.2 (which will be replaced by digitalised music); offset was set to $\Delta = 2$; added noise is standard normal with $\sigma = 1$. All data except carrier is normalised to have unit standard deviation. In order to get samples of dimension d, consecutive windows of d points are grouped. See figure 5.6 for sample plots.

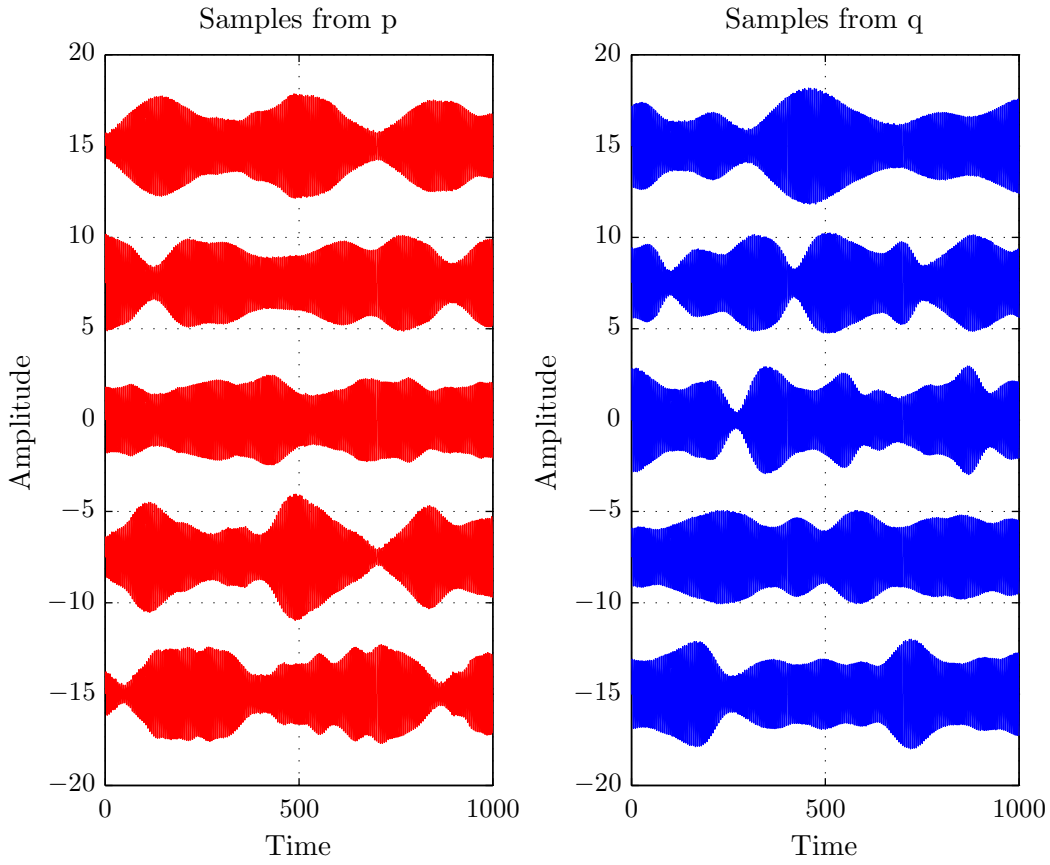


Figure 5.6.: Representative samples from *am*-dataset (Magnetic Fields). Offset $\Delta = 2$, envelope scale $\epsilon = 0.5$, added noise $\sigma^2 = 0.01$, carrier frequency $\omega = 24$kHz, music sampling rate 8kHz.

This initial experiment justifies usage of the linear time two-sample test that was described in section 3.1.2. It establishes the test in for practical usage.

5.3. Kernel Choice Strategies for Linear Time MMD

So far, experiments dealt with justification of used methods and providing examples of usefulness. This section describes experiments for selecting kernels. This is done for linear time two-sample tests based on the statistic described in section 2.4.3 and test construction described in section 3.1.2. The section provides main experimental results: comparison of existing and new methods for kernel selection in different contexts.

All methods for selecting kernels described in sections 3.2 (existing) and 3.3 (new) are tested on all datasets described in section 5.1. For purposes of overview, table 5.2 briefly summarises all evaluated strategies for kernel choice and also defines labels used in plots later on.

5.3.1. General Experimental Design

Since all experiments have a similar design, a general description is given once.

Estimating Type II Errors and Controlling Type I Errors Kernel selection methods are evaluated on different flavours of all datasets. In general, the goal is to produce plots where a measure of difficulty is drawn the x-axis while the method's type II error is drawn on the y-axis. This allows to compare performance of different methods as a function of difficulty. Table 5.1 contains the different measures of difficulty for all datasets.

In order to get stable type II error estimates, two-sample tests have to be performed a number of times. This allows to compute confidence intervals in which the true type II error lies with high probability. Throughout the experiments 95% wald-confidence intervals[2] are used.

To construct a two-sample test, an upper bound on the type I error has to be selected. The test level in all tests was is to $\alpha = 0.05$. This corresponds to an upper bound of 95%. Sample size is set to $m = 10000$ samples from each distribution. Since experiments in section 5.2 suggest that the linear time Gaussian approximation of the null distribution for the linear time test is accurate from about $m = 1000$ samples, this is a safe choice. Type I errors are not reported. However, they were checked for many cases while performing experiments in order to prevent possible problems with the used Gaussian approximation for threshold computation.

Implementation is done in such way that the program keeps increasing the number of trials as long as it runs – up to a limit of 5000 trials. The maximum number of trials ensures that confidence intervals are very tight. However, due to computational costs,

[2] Wald confidence intervals for a significance level of α can be constructed from a single mean value m and a number of trials n as $[m - \delta, m + \delta]$ with $\delta = \Phi^{-1}(1 - \frac{\alpha}{2})\sqrt{\frac{m(1-m)}{n}}$, where Φ^{-1} is the inverse normal cumulative distribution function.

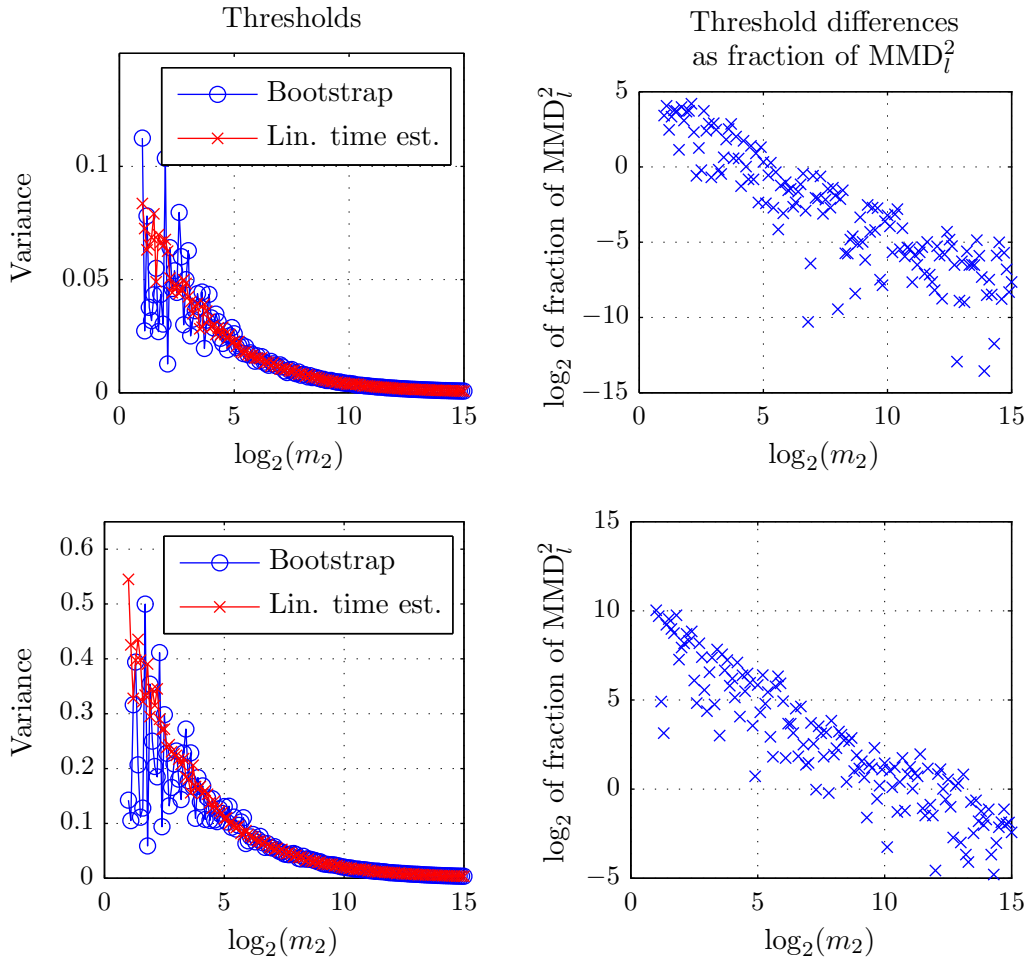


Figure 5.7.: Depiction of convergence of linear time variance estimate to ground truth (1000 bootstrapping iterations) estimate. Data of upper plots is from *mean* difference dataset, lower plots are based on *blobs* data. The right plots depicts the differences of the variances between bootstrapping and approximation as a fraction of the linear time MMD estimate on a log scale – this shows the impact on a two-sample-test in terms of how much the threshold is shifted. From a about 2^8 and 2^{13} samples from *mean* and *blobs* respectively, the impact is neglectable.

Dataset	Difficulty Induced via Changes in
Mean	Dimension
Var	Dimension
Sine	Frequency of sinusoidal perturbation
Blob	Number of rotated and scaled Gaussians
Selection	Dimension
AM	Noise added after amplitude modulation; relation of signal to carrier amplitude in contrast to added noise
Music	Subjective *similarity* of used songs (No parameters changed)

Table 5.1.: Difficulty of all used datasets can be varied using the above changes.

this number cannot always be reached in a reasonable time. Thus, experiments are sometimes stopped at a lower number of trials. Since every point of resulting error plots is handled by a separate job on a cluster computer, the number of runs for each point usually differ. Type II errors are still comparable due to usage of 95% Wald confidence intervals in the plots. Average number of trials ($\varnothing$) are provided above each error plot.

Type II errors are compared against each other easily: lowest errors are produced by best methods. Significant differences happen when confidence intervals do not overlap.

Kernel Weights and Their Visualisation All methods for kernel evaluation need an a-priori set of possible kernels to evaluate. While all methods for selecting a single kernel (c.f. sections 3.2 and 3.3) simply try out all kernels at once and then select the best one, methods for selecting non-negative combinations of kernels (c.f. chapter 4) return weights for a-priori specified kernels. Choosing single kernels can be seen as selecting weights for a kernel combination where only one weight may be set to one and all others are set to zero.

It is instructive to visualise weights in both cases – kernel weights for all trials are plotted as images where each column corresponds to one trial and each row corresponds to one vector of kernel weights. These are displayed in different shades of gray, minimum value is black, maximum value is white. Normalisation is done on the individual image in order to ensure comparability of all columns. Note that the y-axis is scaled logarithmically. Every line corresponds to $\log_2(\sigma)$, where σ is the used Gaussian kernel width.

Kernel Selection Methods To recall all described methods for kernel selection, see table 5.2. It also contains abbreviations used in plots.

5.3.2. Mean and Var Dataset

Results on these datasets are only described briefly. In order to show a difference of used methods, a massive number of samples and computational time has to be used. Still, differences in performance are very subtle. As mentioned before, these datasets do not

Existing Methods	
Median	Select the bandwidth of a Gaussian kernel according to the median distance in underlying data. See section 3.2.1
MaxMMD	Select a (here: single Gaussian) kernel in such a way that the linear time MMD statistic itself is maximised. Corresponds to L_1 norm regularisation on the weights. See section 3.2.2.
X-Val-Loss	Select a single (here Gaussian) kernel using cross-validation in such way that the expected risk of the parzen window style binary classifier with a linear loss is minimised. This corresponds to MaxMMD with a protection for overfitting. See section 3.2.4.
L_2	Select a non-negative combination of kernels in such way that the MMD statistic is maximised by using L_2 norm regularisation on the weights. See section 4.3.4.
New Methods	
MaxRat	Select a (here: single Gaussian) kernel in such a way that the ratio of the the linear time MMD statistic divided by its standard deviation, is maximised. This kernel choice is optimal. See section 3.3.2.
Opt	Select a non-negative combination of fixed baseline kernels, such that the above ratio is maximised. Optimal kernel choice for non-negative combinations. See section 4.3
X-Val-Type II	Select a single (here Gaussian) kernel using cross-validation in such way that its type II error estimate is minimised. See section 3.3.3.

Table 5.2.: All evaluated methods for kernel selection for linear time tests.

represent major difficulties in two-sample testing – distinguishing characteristics of the underlying distributions are obvious. Consequently, in order for new methods to show their statistical superiority of new kernel selection methods (optimal kernel choice), a very large number of samples is needed.

Experiments were performed as described in section 5.3.1; difference of the dataset is varied by changing dimension d of data.

Dataset parameters are: $m \in \{10000, 40000\}$ samples from each p and q; difference $\epsilon \in \{0.4, 0.5\}$; dimension $d \in \{2^1, ..., 2^5\}$.

Mean Dataset: No Significant Difference Figure 5.8 shows type II errors. There are two plots provided: since cross-validation, (in particular *X-Val-Loss*) is very expensive, it is left out in the upper plot ($\epsilon = 0.5$) to reach a larger number of trials in order to tighten confidence intervals. Still, only in $d = 2^2$, a very subtle difference between old and new methods is observed. The lower plot adds cross-validation based methods on a slightly harder difficulty ($\epsilon = 0.4$). There are no significant and consistent differences between methods – almost all confidence intervals overlap. A slight exception is *X-Val-Type II*, which seems to be most robust.

Var Dataset: Slight Advantage of new Methods As for *mean*, figure 5.8 does not include cross-validation based methods. With a very large sample size (which is non-feasible for *X-Val-Loss*) a slight significant difference between new and existing methods is revealed: optimal kernel selection performs better. With lower number of samples, all methods perform similar.

5.3.3. Sine dataset

As described in section 5.1.4, the main difficulty of this dataset is that differences between p and q are a very high frequency sinusoidal perturbation of the Gaussian density function – for which overall data scaling is not a good indicator. Therefore, the problem is numerically and structural more difficult than *mean* and *var* datasets. Experiments are performed as described in section 5.3.1; difference of the dataset is varied by changing the frequency ω of the sinusoidal perturbation.

Dataset parameters are: $m = 10000$ samples from each p and q; sinusoidal perturbation frequencies $\omega \in [7, 16] \subseteq \mathbb{N}$.

Median Fails: It Cannot Detect Signal Figure 5.10 depicts type II errors reached by the *Median* method. The first important observation is that the Median method for selecting a kernel completely fails. The type II error is around the worst possible value 95%. The reason is the large selected kernel width: $\log_2(\sigma) = -0.5$ (see figure 5.11)is way too large to detect differences induced by a sine wave with frequency $\omega \in [7, 16]$. The kernel size selected by the best method is around 2^{-5} – a massive difference. This supports the expectation that methods that orient kernel choice on the overall scaling of data fail when the the actual difference is hidden at another scale.

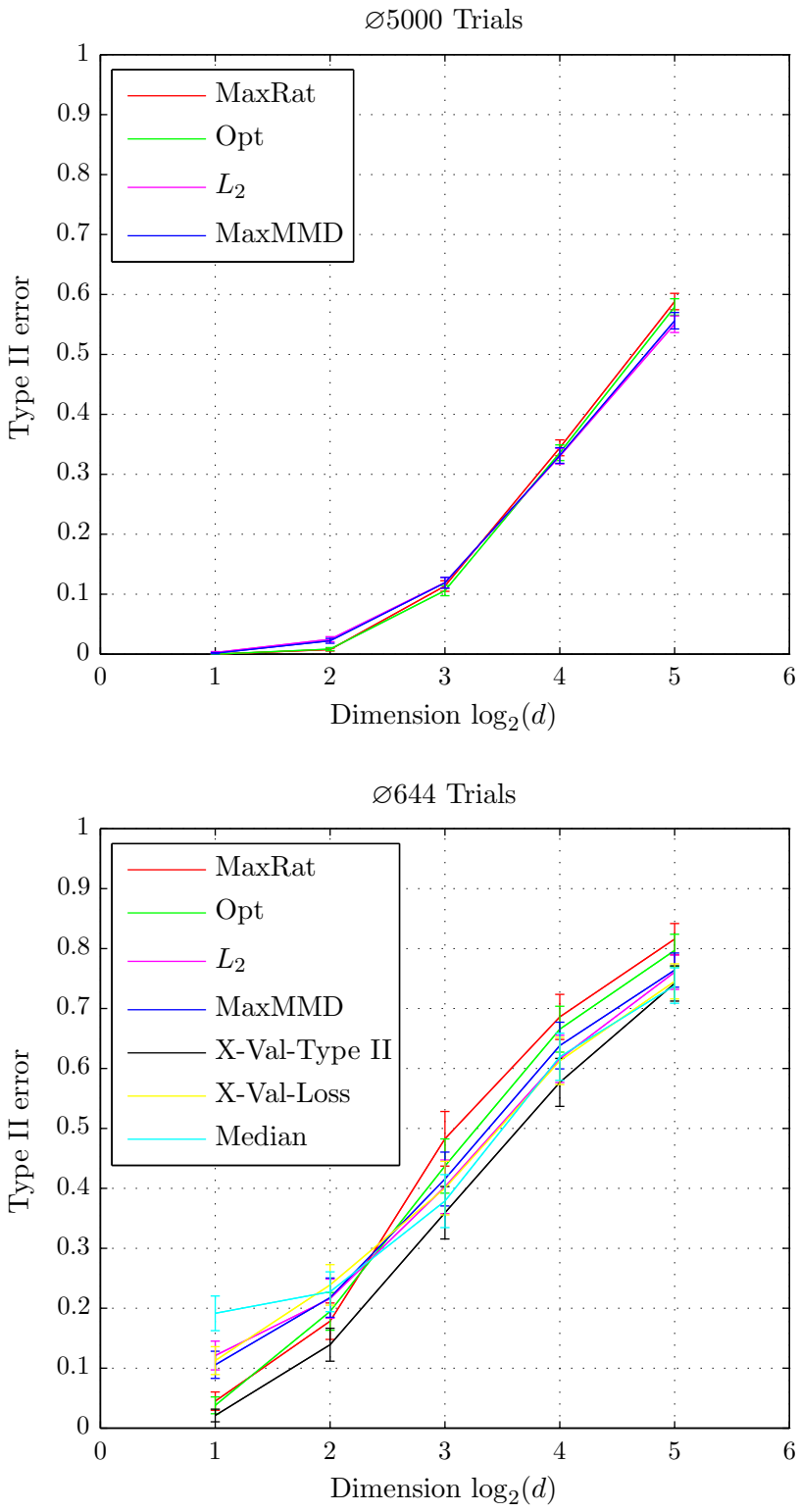


Figure 5.8.: Linear Time Kernel Selection for *mean*. $m = 10000$ samples from each p and q. Upper plot: difference $\epsilon = 0.5$, lower plot: difference $\epsilon = 0.4$.

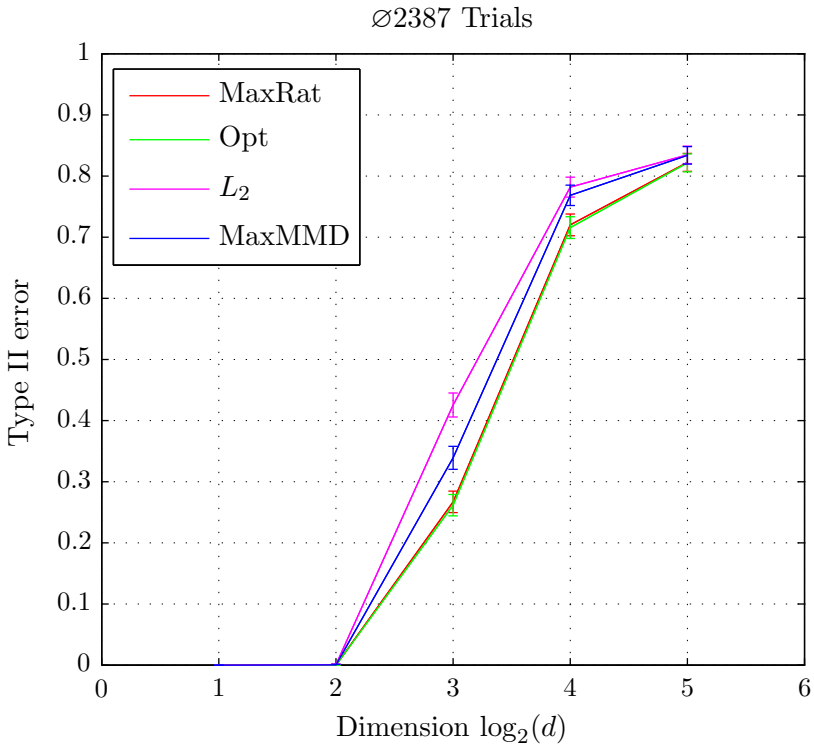


Figure 5.9.: Linear Time Kernel Selection for *var.* $m = 40000$ samples from each p and q; difference $\epsilon = 0.5$

L_2: Forced Combinations are not Sensible The L_2 method so far is the only method that is capable of selecting multiple kernels. However, it is unable to select a single kernel due to the L_2 norm constraints which favour smooth weight distributions around a maximum. For datasets where selecting multiple kernels does not lead to an advantage, this is a downside. The *sine* dataset is such a case: a single kernel is the best choice. In particular, this is the kernel whose detective range is closest to the shape of the sinusoidal signal. This is simply due to the fact that the sine perturbation has the same frequency everywhere. The L_2 method is unable to select this single kernel. Instead, it will select a smooth combination of kernel around the best one. This can be seen in figure 5.11. Figure 5.10 shows that the resulting type II errors loose against other methods.

Opt: Chooses a Single Best Kernel In contrast, the *Opt* method, which is also capable of selecting multiple kernels, does not have this problem. There is no constraint for selecting smooth weights as in L_2. Therefore, it is able to identify the single best kernel and to put all weight onto it. This can bee seen in 5.11. One problem that remains is numerical instability. The figure shows a quite large amount of noise in the kernel weights. Sometimes, there is weight put on very small kernels. This clearly lets the two-sample test fail. This probably is caused by the noisy nature of the *sine* dataset itself: with increasing frequency ω, detecting differences is hard due to numerical problems. Handling such noise is a point for further research. Despite these problems, most of the time, the correct (single) kernel size is selected by *Opt* – this also can be seen in the type II error which is not significantly different to the best methods.

X-Val-Loss is More Stable Than MaxMMD As described in section 3.2.4, *X-Val-Loss* is in its core the same as *MaxMMD* with a protection for overfitting and numerical stability. Figure 5.11 confirms the latter: *X-Val-Loss* looks less noisy. The protection for overfitting is not necessary here – both methods tend to select the same kernel size. Consequently, both methods lead to similar type II errors. This suggests that the computational efforts of minimising a linear loss risk by cross-validation is not necessary since it give the same results as maximising the MMD itself. The extend of gained numerical stability does usually also not change type II errors since *MaxMMD* is already very stable. An exception are very high frequencies ω of sinusoidal perturbation. In these very high frequency domains, numerical stability becomes more and more important: *X-Val-Loss* wins. However, this is done at large difference in computational costs: the method's costs are quadratic in the number of samples while *MaxMMD* in linear in this context. Therefore, in lower frequencies ω, *MaxMMD* outperforms *X-Val-Loss* in the way that almost the same results are achieved with less computational costs.

Remaining Methods: Similar Performance – X-Val-Type II is Most Stable All of the methods *Opt*, *MaxRat*, *MaxMMD*, *X-Val-Loss*, and *X-Val-Type II* perform similar. There are no massively significant differences. However, there is a small trend that *MaxMMD* is outperformed by the optimal choice of *MaxRat*. Especially with increasing difficulty, *MaxMMD* tends to get worse as can be seen in figure 5.10. For less difficult

problems, this difference is rather small and suggests that *MaxMMD* already selects a kernel size near to the optimum here. *Opt* is very close to *MaxRat* since it inherits the optimality of its choice. There is a clear trend that *X-Val-Type II* is the best method, here. However, the difference is not significant for many points. Computational costs of *X-Val-Type II* are slightly larger than of *MaxRat* which is the direct competitor.

Illustration of Adaptive Kernel Sizes Increasing difficulties in the *sine* dataset are induced by higher frequency sinusoidal perturbations on the Gaussian density. Clearly, in order to detect differences, the (Gaussian) kernel size has to shrink with increasing sinusoidal frequency ω. Figure 5.12 shows the evolution of the kernel size selected by the best method *X-Val-Type II*. Each histogram depicts the distribution of selected kernel sizes for one particular dataset difficulty ω. The mean weight is shown as a red line. Note how the optimal kernel size shrinks when ω increases. This depicts the need of adaptive methods: the *Median* method for example always selects the same kernel width while sensible methods adapt to data.

5.3.4. On the Need of More Difficult Datasets

So far, differences between existing and newly proposed methods for kernel selection have been minor or only happened at very large sample sizes. The reasons for this are either that datasets are too "easy" in the sense that differences of distributions are easily detected or that difficult problems also are mostly numerically challenging (*sine* dataset). During the work on this thesis, it became clear that tougher datasets are needed: those whose difficulty comes from structure of the data and not from numerical problems. Such methods are better suited to evaluate two-sample tests as methods are likely to perform more distinct. The following datasets are designed with these arguments in mind. They form a new set of benchmarks for adaptive two-sample testing and went into submission of [Gretton et al., 2012c].

5.3.5. Blobs dataset

As described in section 5.1.5, the main difficulty of this dataset is the fact that differences between distributions p and q are hidden in the overall scaling of the data. In contrast to the *sine* dataset, this difficulty does not come with increased numerical challenges. In case of the *blobs* dataset, it would be actually enough to look at a *single* Gaussian blob in order to distinguish p and q. Experiments are performed as described in section 5.3.1; numerical difficulty is varied by changing the scale ϵ of the rotated Gaussians in q. Overall difficulty is fixed.

Dataset parameters are: $m = 10000$ samples from each p and q; $\alpha = \frac{\pi}{4} \equiv 45$ degrees; rotated Gaussians' stretch $\epsilon \in [2, 15] \subseteq \mathbb{N}$; number of Gaussians $t^2 = 25$ at distance $\Delta = 5$.

Median Fails Again Similar to results on *sine* dataset, figure 5.13 shows that the *Median* method for selecting a kernel completely fails. Type II error is around the worst

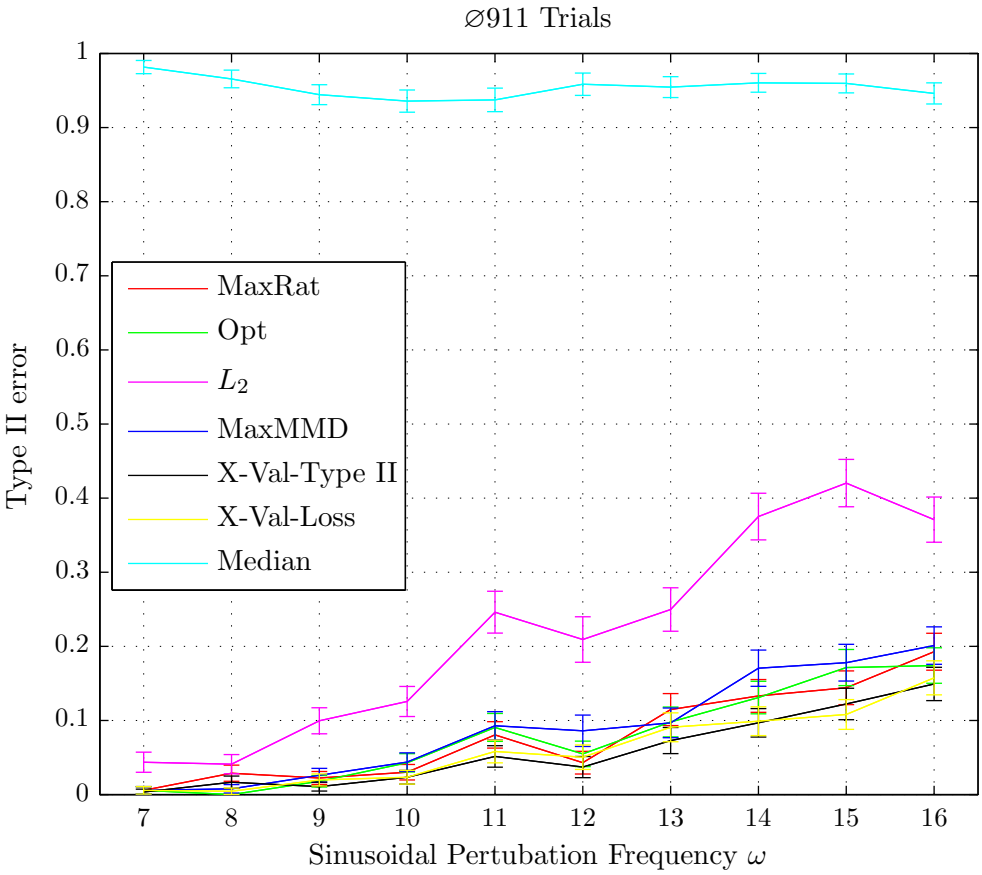


Figure 5.10.: Type II Errors for Linear Time Kernel Selection on *Sine* Dataset. Parameters: $m = 10000$ samples from each p and q; sinusoidal perturbation frequencies $\omega \in [7, 16] \subseteq \mathbb{N}$.

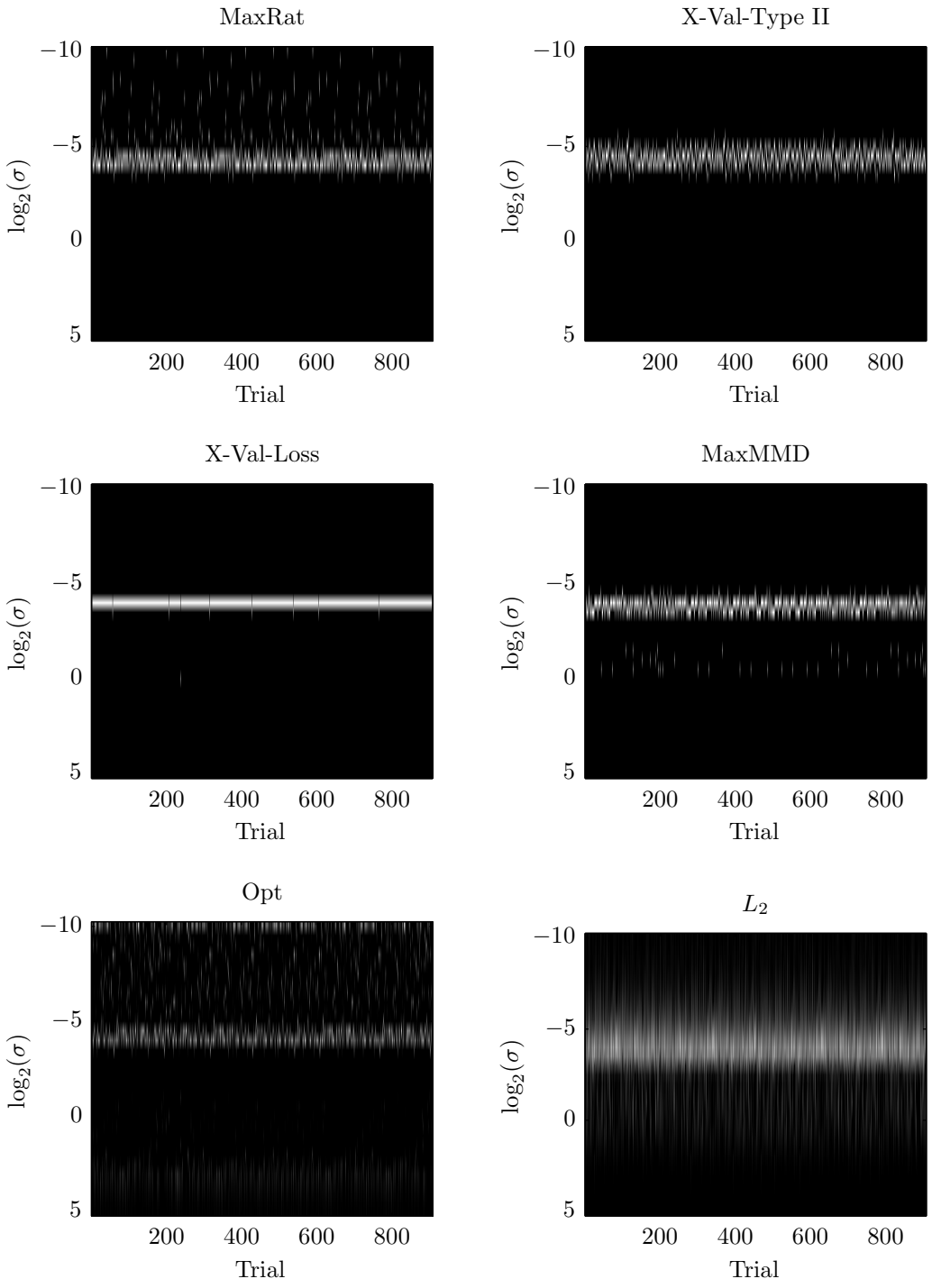


Figure 5.11.: Kernel Weights for Linear Time Kernel Selection on *sine* Dataset with sine frequency $\omega = 14$. The Median method always selected $\log_2(\sigma) = -0.5$.

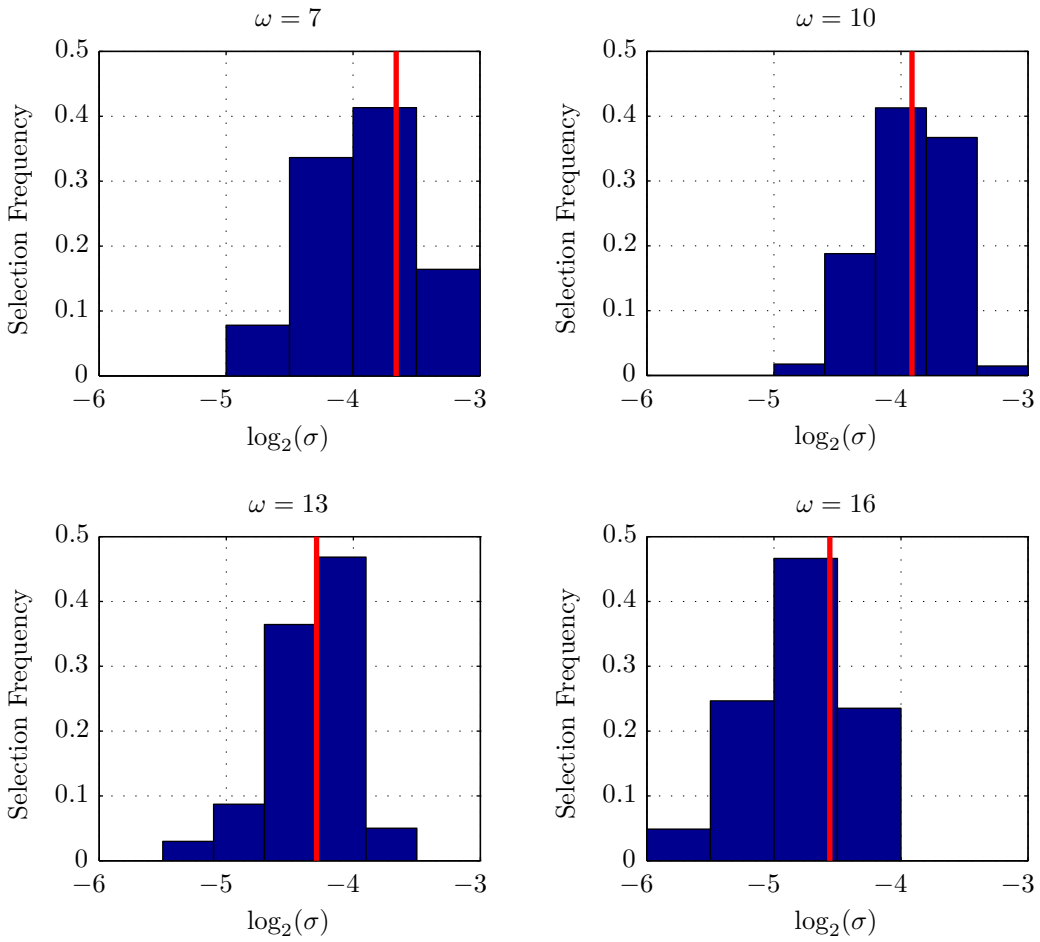


Figure 5.12.: Kernel Weight Evolution for for Linear Time Kernel Selection on *sine* Dataset. Each histogram depicts the distribution of selected kernel weights during all performed trials. Used kernel choice strategy is the best method *X-Val-Type II*. Red lines depict means of selected kernel weights. They clearly adapt to frequency ω of the sinusoidal perturbation.

possible value 95%. The reason is the selected kernel width: $\log_2(\sigma) = 4.5$. This kernel is too large; the difference to the best method around 2^5 – a massive difference for two dimensions. See figure 5.14. The conclusion is: the *Median* method should not be used on data where differences in distributions might be subtle.

Problems of MaxMMD and L_2 The next two competitors are both based on maximising the MMD: *MaxMMD* and L_2. The latter performs remarkably worse than *MaxMMD*. This seems odd at first but simply may be caused by the fact that L_2 tends to select smooth combinations of multiple kernels instead a single one. Figure 5.14 supports this. However, since structure of the data is similar in all dimensions, a single kernel gives a larger MMD than a smooth combination around this maximum. Still, even *MaxMMD* does not perform particularly well compared to other methods. Note that kernel weights selected by *MaxMMD* in figure 5.14 are sometimes in the same range as these selected by Median. This is most likely due to the problem described in section 3.3.1 which motivates the *MaxRat* and *Opt* method: even though the MMD is maximised, its variance grows even more and neutralises any benefits gained by a large MMD statistic.

Minimising Linear Loss Resolves Some Problems at Huge Costs The cross-validation based approach that minimises expected linear loss of a binary classifier performs (sometimes significantly) better than the *MaxMMD* approach. Since it basically is the same method plus a protection for overfitting, this result might indicate that this happens with *MaxMMD*. However, kernel weights in figure 5.14 reveal it is not the case. The only difference between *MaxMMD* and *X-Val-Loss* is that *X-Val-Loss* does not select a kernel that is too large, as *MaxMMD* does from time to time. Since these large outliers are obviously too large for the data, type II error gets worse. Cross-Validation averages multiple folds of training and validation data and then takes the kernel that the majority of cases supported. If a kernel that is too large is selected only in a few cases, this effect is being suppressed by averaging of folds. Therefore, *X-Val-Loss* does never select these outliers and gets a better type II error. This is an appealing property which makes the approach more stable. However, since its computational costs are huge, it is questionable whether it is useful when cheaper approaches perform better. See the following paragraph.

Flexibility and Accuracy of MaxRat and Opt As expected, the newly described methods for optimal kernel choice, *MaxRat* and *Opt*, perform significantly better than any existing method. Figure 5.14 also shows that their selected kernel is almost identical. In particular, the *Opt* method which selects weights of multiple baseline kernel puts the vast majority of weights onto the same kernel than *MaxRat*. There are a few exceptions where little weight (dark gray) is put onto smaller kernels. Stabilisation of such noisy kernel weights is a point for further investigation.

The fact that the majority of weights of *MaxRat* and *Opt* are equal is an illustration for the fact that a single kernel is sufficient. More important, the *Opt* method is able to capture this and therefore is a superset of the *MaxRat* method here: if a single kernel is

the optimal choice, it will be selected even though combinations of multiple kernels are also considered.

A Single Kernel Which Minimises Type II Error The other newly described approach, *X-Val-Type II*, selects a single kernel that minimises the type II error directly. In contrast to the direct competitors, *MaxRat* and *Opt*, it performs slightly better. Since *MaxRat* and *Opt* both minimise the probability for type II error indirectly by maximising a ratio of MMD statistic and its standard deviation, they suffer numerical problems. For example denominators in ratio may become very small. A direct approach is therefore likely to be superior in practice – especially in numerically challenging contexts (left part of plot). This suspicion is confirmed when one takes a look at the selected kernel weights in figure 5.14: the difference to *MaxRat* is rather the fact that it is more stable than a difference in overall selected kernel size. Note that *X-Val-Type II* performs almost identical to the other new methods when problems are numerically easy (large stretches).

As already mentioned, for the *blobs* dataset, a single kernel is sufficient. Therefore, and due to above reasons *X-Val-Type II* is the best of competing methods here – with a slight advantage over *Opt* and *MaxRat*. Computational costs are larger than for *MaxRat* and *Opt*, however, they are still way lower than for minimising linear loss.

5.3.6. (Feature) Selection Dataset

As described in section 5.1.6, purpose of the *selection* dataset is to illustrate how multiple combined kernels can be used for *feature selection* in two-sample testing. A second point is to motivate that in certain cases, combined kernels are a better choice than single ones. The experiment considers one fixed univariate kernel per dimension. Selecting weights then corresponds to selecting dimensions to consider in the underlying two-sample test. In all datasets so far, best kernels always have been single ones. In contrast and by construction, the *selection* dataset needs more than one dimension in order to capture all differences between the distributions p and q. Therefore, a method needs to select multiple kernels in order to reach good performance.

Experiments are performed as described in section 5.3.1; difference of the dataset is varied by changing the dimension $d \in [2, 29] \subseteq \mathbb{N}$ of which one of $\Delta = 2$ dimensions randomly has its mean shifted by $\epsilon = 0.5$. Note again that *one* fixed univariate kernel is used per dimension. This kernel has a fixed size of $\sigma = 2^3$; used are $m = 10000$ samples from each p and q.

The kernel size is fixed in each dimension for this dataset – based on a-priori knowledge of the data. This is done in order for clarity of the argument that is made: multiple kernels are sometimes better than single ones. The approach could easily be adapted such that a kernel size would be selected in every dimension. This would have to be followed by a rather large analysis of kernel sizes in each dimension. However, here, a single kernel per dimension is enough to make a point. In practice, this a-priori approach is not possible.

Cross-validation based approaches are not feasible for feature selection in practice. For single kernels, simply all baseline kernels can be tried one after another. However,

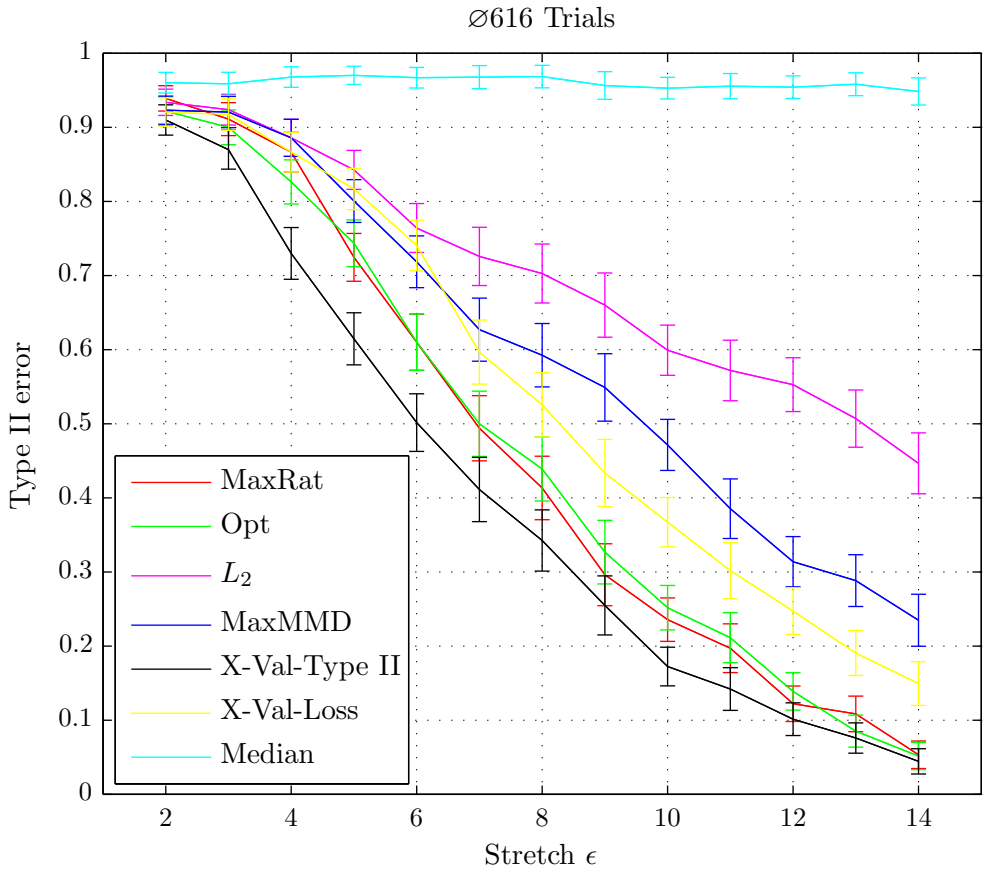


Figure 5.13.: Type II Errors for Linear Time Kernel Selection on *Blobs* Dataset. Parameters: $m = 10000$ samples from each p and q; $\alpha = \frac{\pi}{4} \equiv 45$ degrees; rotated Gaussians' stretch $\epsilon \in [2, 15] \subseteq \mathbb{N}$; number of Gaussians $t^2 = 25$ at distance $\Delta = 5$.

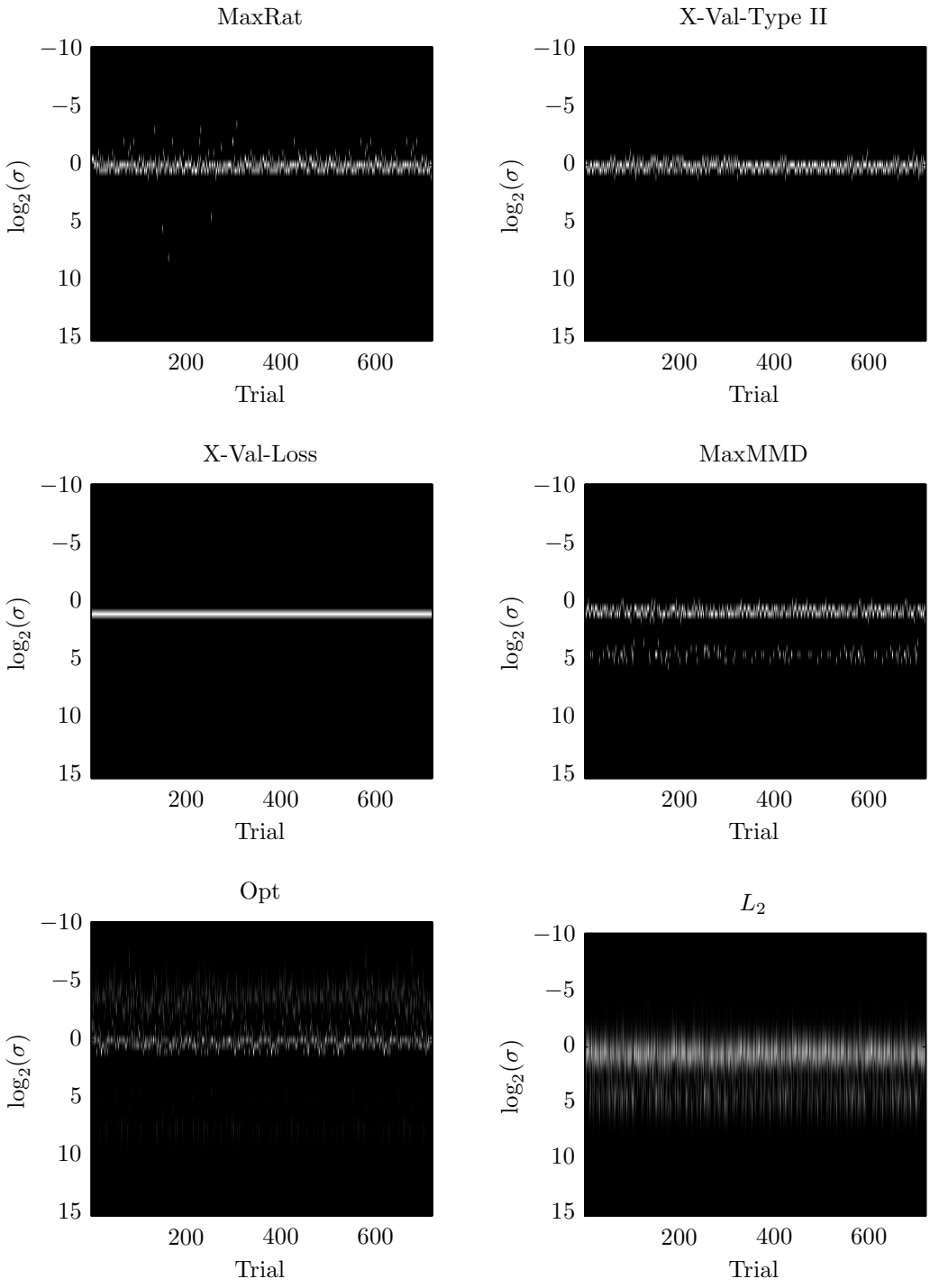


Figure 5.14.: Kernel Weights for Linear Time Kernel Selection on *Blobs* Dataset with stretch $\epsilon = 12$. The Median method always selected $\log_2(\sigma) = 4.5$.

for combined kernels there is an infinite number of non-negative combinations. Even if these are enumerated in a discrete grid, it is not possible to solve for best weights by trying out all combinations – computational costs would be too high.

MaxRat and MaxMMD: Single Kernels Fail As can be seen in figure 5.15, and as expected, selecting a single kernel/dimension leads to a worse type II error than a combination of kernels. Both *MaxRat* and *MaxMMD* perform worse than *Opt* and L_2. Figure 5.17 depicts selected kernel weights. These mostly vary between the first $\Delta = 2$ relevant dimensions, both dimensions are selected in about 50% of trials. Although only a single kernel is selected here, this reveals structure in the underlying data.

Opt and L_2: Multiple Kernels Succeed Figure 5.15 shows that combined kernels lead to a much better type II error. Figure 5.17 shows that the first two dimensions are selected in an equal combination most of the time. This equal combination of the first two dimensions is in fact the best choice for the underlying data. The ability to select weights of combined kernels in this case is a clear advantage.

Similarity of L_2 and Opt *Opt* and L_2 perform very similar – although *Opt* is theoretically statistically superior. An illustrative observation that connects both approaches for the *selection* dataset can be made in figure 5.16: The matrix Q used for optimisation in *Opt* is very close to being the identity (apart from some noise). This in fact reduces the convex program in expression 4.8 to the program in expression 4.9 which is solved for the L_2 method. The shape of Q is responsible for the almost identical selected weights that can be seen in figure 5.17.

The shape of the matrix also makes sense in context of the *selection* data: every dimension is independent so there are no inter-dimensional correlations. The only difference between Q and the identity matrix is noise. Therefore, *Opt* cannot perform better than L_2 on the *selection* dataset.

5.3.7. Music and AM Datasets

As described in section 5.1.7, purpose of the *music* dataset is have baseline results for songs that will be used as a in the *am* dataset. It is important to know, how hard they are to distinguish from one another when no additional difficulties (such as noise and amplitude modulation) are added. Difference in performance of all kernel selection methods on the pure music in contrast to its amplitude modulated counterpart is expected to give insight whether methods are able to detect an equal signal directly (*music*) and when hidden (*am*).

Due to the high dimension of data (window length $w = 10000$), computational costs to compute kernels are very large. Therefore, the cross-validation based methods *X-Val-Loss* and *X-Val-Type II* are not evaluated here. This is since the focus is different: showing an advantage of new ratio-based methods *MaxRat* and *Opt* over simply maximising the MMD in *MaxMMD* and L_2.

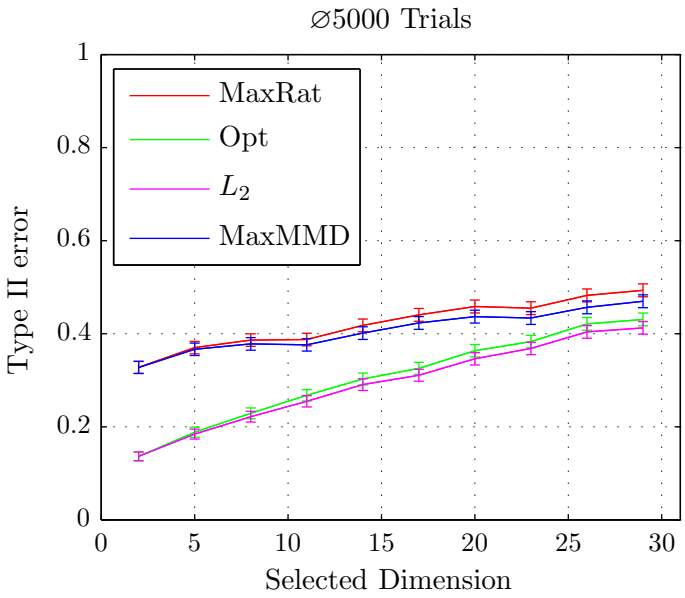


Figure 5.15.: Type II Errors for Linear Time Kernel Selection on *Selection* Dataset. Parameters: $m = 10000$ samples from each p and q; mean shift $\epsilon = 0.5$ randomly happening in one of $\Delta = 2$ of $d \in [2, 29] \subseteq \mathbb{N}$ dimensions; fixed kernel in every dimension with $\sigma = 2^3$.

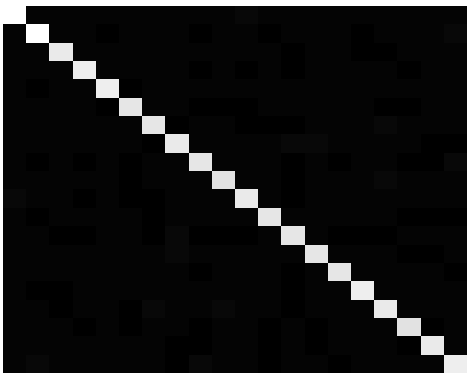


Figure 5.16.: Depiction of matrix Q for optimisation problem solved for *Opt* method. Same data as above, dimension is $d = 20$. The diagonal structure illustrates independence of dimensions and lets the convex program in expression 4.8 reduce to the L_2 approach in expression 4.9.

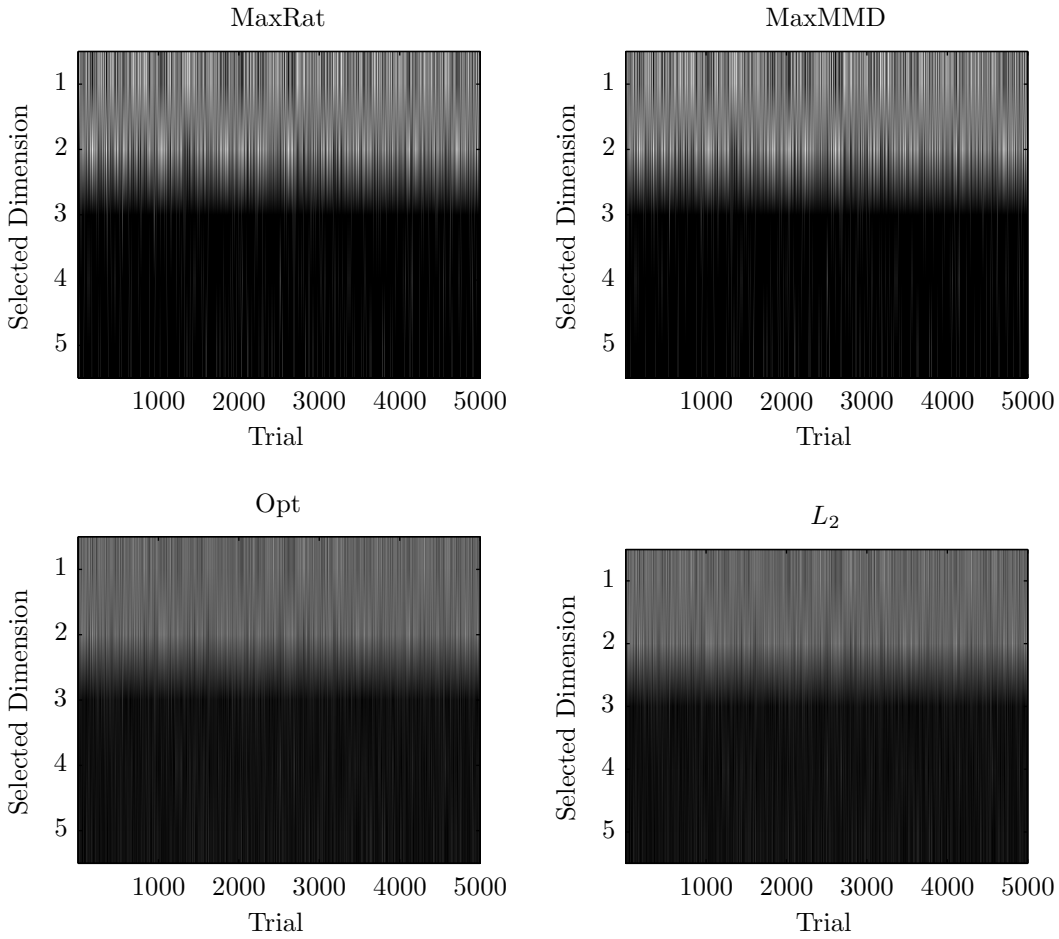


Figure 5.17.: Kernel Weights for Linear Time Kernel Selection on *selection* dataset for dimension $d = 5$. Note how only the first $\Delta = 2$ relevant dimensions are selected. Still, noise in the other selected dimensions is a problem.

	Type II Errors	
Method	***Music* (1362 trials)**	***AM* (3197 trials)**
MaxRat	$[0.013, 0.029]$	$[0.052, 0.069]$
Opt	$[0.017, 0.031]$	$[0.038, 0.052]$
MaxMMD	$[0.020, 0.038]$	$[0.022, 0.034]$
L_2	$[0.074, 0.105]$	$[0.082, 0.102]$
Med	$[0.504, 0.557]$	$[0.653, 0.686]$

Table 5.3.: Type II Errors for Linear Time Kernel Selection on *Music* Dataset. Shown is one representative pair of songs from the artist *Magnetic Fields*. Since parameters are not varied, there are only single type II error intervals. Zero noise AM cases are shown for comparison.

Results of one pair of songs will be presented for music data; multiple songs will be evaluated for *am* data. Since *music* data has no parameters to vary, results are simply type II errors of each method. Table 5.3 shows type II errors for two pairs of songs. Results are consistent with these seen so far. The performance order methods is: *MaxRat*, *Opt*, *MaxMMD*, L_2, *Med*.

The table also contains comparable (same window size, sampling rate of music) results for a two-sample test on *am* data with the same music, as will be described next. It confirms the intuition that *am* data is a more difficult problem. In addition, the *Med* method even performs worse there. More details and an interpretation will be given in the next section.

AM Dataset Results As described in section 5.1.8, purpose of the *am* dataset is to evaluate a test's ability to distinguish structured signal whose difference appears in the envelope of a carrier wave. Since signal is audio data, this is a real world data test.

One pair of songs is analysed in detail and compared to *music* dataset. For others, only error plots are reported. Experiments are performed as described in section 5.3.1; offset is set to $\Delta = 2$; difference of the dataset is varied by changing the envelope scaling of the am modulation $\epsilon \in \{0.1, 0.3, ..., 0.9\}$. Smaller ϵ make a harder problem. Results reported here have $\epsilon \in \{0.3, 0.7\}$. Note that setting ϵ too large will result in phase shifting and therefore is to be avoided since results are unpredictable.

Another difficulty is introduced by adding noise with variances $\sigma^2 \in \{0, 0.2, ..., 1.0\}$ to the am signal. When noise is added, music signal in the envelope becomes harder to detect. This is in particular happens with smaller values of the envelope scale σ. Since *music* dataset does not have any added noise, comparison is done on the $\sigma^2 = 0$ case. Comparison is done on one pair of songs.

Again, due to the high dimension of data (window length $d = 10000$) cross-validation based approaches are not evaluated.

Music is Easier Than AM – Median Fails As can bee seen in table 5.3, the zero noise case of the *am* version of the same audio as used in the *music* dataset, with a small

envelope scale of $\epsilon = 0.3$, is harder to distinguish for all evaluated methods. *Med* has most problems with the new setting. Even though the signal in the underlying data *has not changed* (same music) the method's type II error drops about 15% in the zero noise case. This can be explained by the small envelope scaling $\epsilon = 0.3$: since the median distance in data reflects the shape of the *overall* signal, any kernel which focusses on that looses most of the audio signal which is located at a smaller length scale. Other methods also reach slightly worse type II errors; however, these changes are not very distinct. With increasing noise, from $\sigma^2 = 0.4$, *Med* reaches the worst type II error possible as can bee seen in figure 5.18.

MaxRat Performs Better Than MaxMMD on Noisy AM-Signals Figure 5.18 shows type II errors for different levels of added noise in the *am* dataset. It reveals that the new method *MaxRat* is better in handling noisy am signals: its reached type II error is consistently below *MaxMMD*'s error. Especially for a noise level $\sigma^2 = 0.2$ that is around the same extend as the envelope scaling of the audio signal $\epsilon = 0.2$, this difference is highly significant: around 50%. For noise level $\sigma^2 = 0.4$, *MaxRat* still performs significantly better. With more noise, all methods but converge to a high level of type II error. This is no surprise: when noise dominates over audio signal, it literally becomes impossible to distinguish both songs.

Single Kernels Are Sufficient Figure 5.19 depicts kernel weights that are selected by all evaluated methods – for added noise $\sigma^2 = 0.2$. All weights are set around a similar value. Even *Med* selects a kernel that is close to other methods. However, note that the scale of the y-axis is logarithmically. L_2 also puts most weight around the same value. All these results suggest that for distinguishing the *am* signal, a single bandwidth is the best choice.

Differences figure 5.18 can be explained by different kernel choices. Both *Opt* and *MaxRat* focus on the same single kernel whereas the one selected by *MaxMMD* and L_2 is double sized during 50% of trials. *Med* always selects a single kernel that has a bandwidth four times larger than the best one.

Differences Among Different Music Figure 5.20 shows the same error plots as described above for different pairs of songs. They all contain the same trend in terms of how methods perform against each other: *Opt* and *MaxRat* are superior to *MaxMMD* and L_2; the *Med* method fails. However, some music pieces are better distinguishable than others in the chosen setting. There are many possible reasons for this: similar instruments, recording sessions, song dynamics – all these facts influence frequencies occuring throughout the songs. However, the point here was to illustrate the ability kernel selection strategies to detect signal which is hidden in other signal disturbed by noise. Top plots in figure 5.20 successfully show this. The lower ones indicate that the selected problem is too difficult to infer anything.

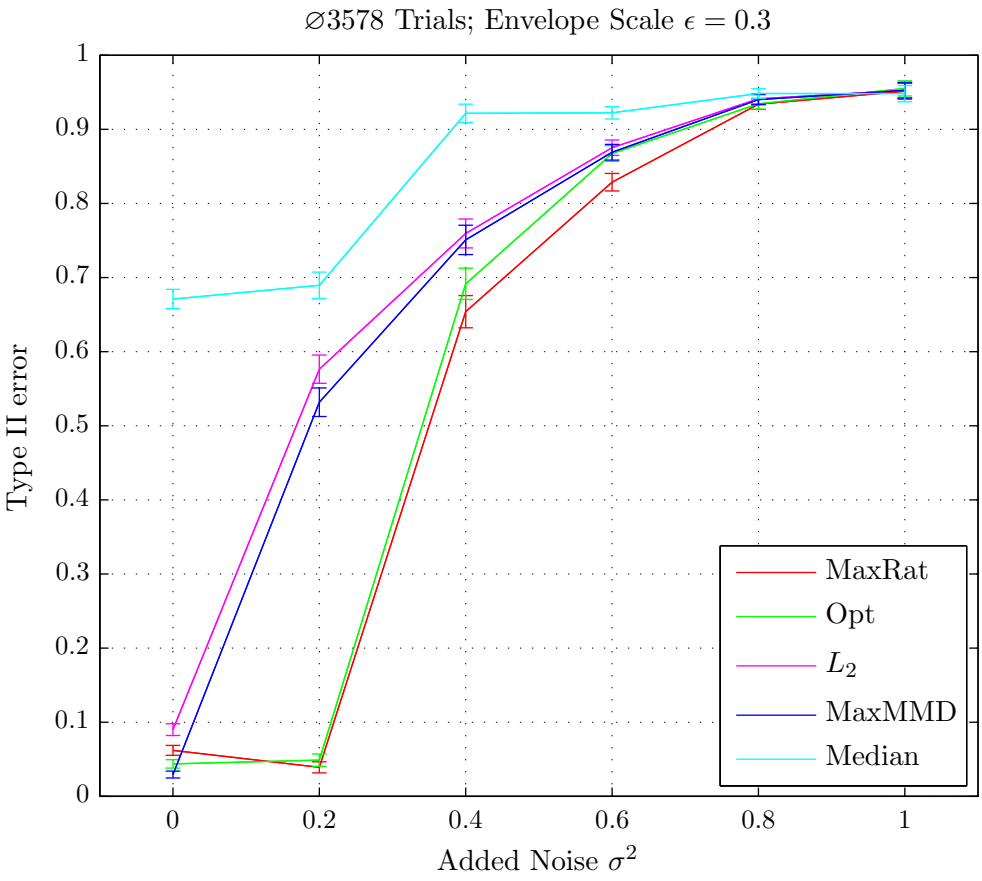


Figure 5.18.: Type II Errors for Linear Time Kernel Selection on AM Dataset (Magnetic fields). Parameters: $m = 10000$ samples from each p and q; window length $w = 10000$; envelopes scale $\epsilon = 0.3$; added noise with variances $\sigma^2 \in \{0, 0.2, ..., 1.0\}$.

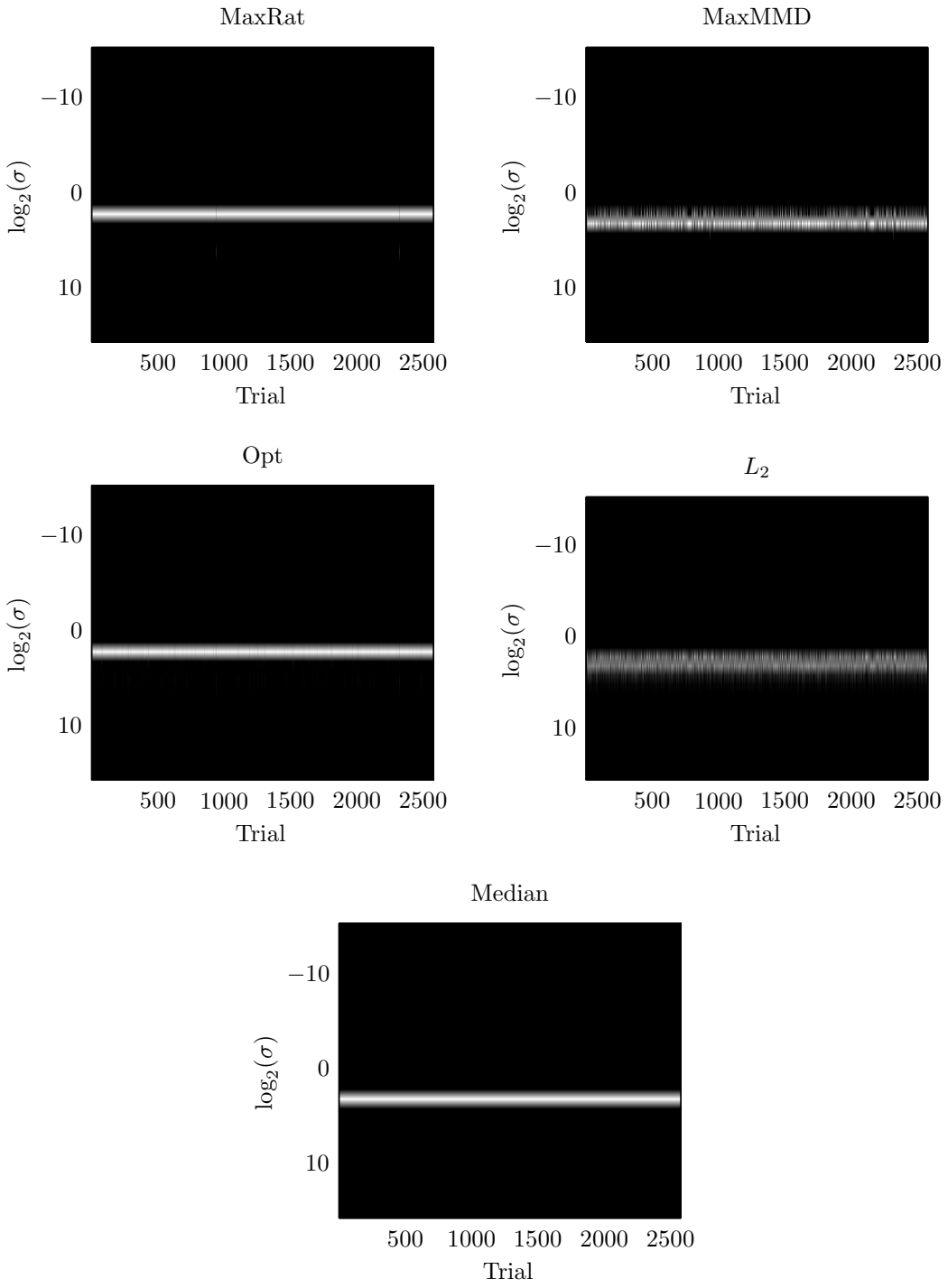


Figure 5.19.: Kernel Weights for Linear Time Kernel Selection on AM Dataset (Magnetic fields). Parameters: $m = 10000$ samples from each p and q; window length $w = 10000$; envelopes scale $\epsilon = 0.3$; added noise $\sigma^2 = 0.2$.

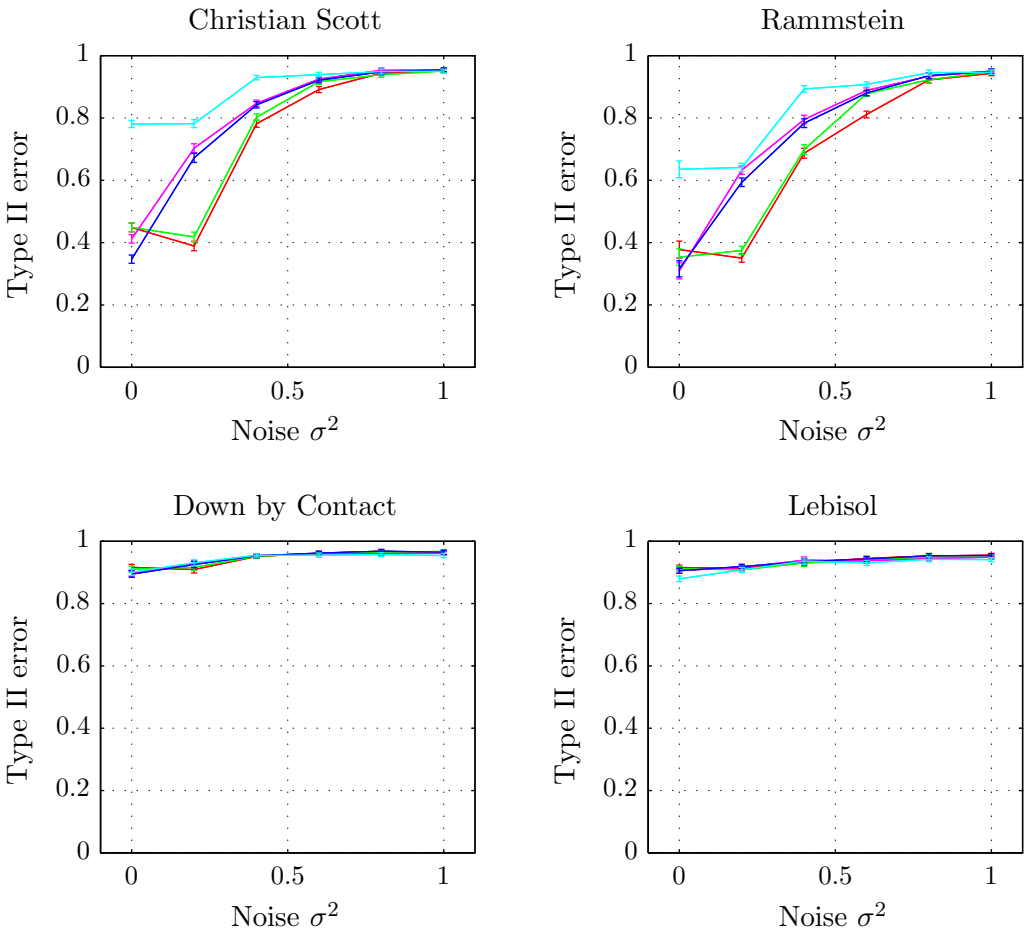


Figure 5.20.: Kernel Weights for Linear Time Kernel Selection on *AM* Dataset for additional pairs of songs. See figure 5.18 for legend. Results are similar to *Magnetic fields* in the above plots. Lower plots contain music that is too hard to distinguish in the chosen context.

5.4. Quadratic vs. Linear Time Tests

While the last section contained empirical results on new described kernel selection methods for the linear time MMD, this section motivates usage of such linear time tests. To this end, two experiments are described: one that illustrates a case when existing quadratic time methods fail but linear time methods do not; one experiment where using linear time tests leads to better results on the same problem.

5.4.1. Large Scale Two-Sample Tests – When the Quadratic Time Test Fails

One question that naturally occurs when using the linear time MMD statistic (section 2.4.3) is why to use a statistic that is by construction less stable than the quadratic time MMD statistic. It seems odd that accuracy is sacrificed to save computational costs. Usually, two-sample tests try to reach best results possible for given data – but the linear time test is not in line with this fashion. It was already shown that the linear time statistic has the appealing property that optimal kernel selection is possible due to the simplicity of null and alternative distribution, c.f. section 3.3.2. However, optimal kernel choice does not give any advantages when the underlying test is weak.

There actually *is* an advantage when using the linear time test: its streaming nature. It does not need to store all data in memory and more importantly, this also holds for constructing an efficient and accurate test – see experiments in section 5.2. In contrast to the quadratic time MMD test, it does not have to precompute kernel matrices of sample data in order to perform the test. For the quadratic MMD, consistent test constructions *have* to store kernel matrices in memory – c.f. section 3.1.4. One alternative, bootstrapping, (section 3.1.1) could be computed without precomputing kernel matrices since it simply computes the statistic a number of times. However, in practice this is computationally very ineffective since many hundreds of samples from the null distribution would have to be computed while each of them has quadratic costs. Consequently, the number of samples than can be processed with the quadratic time test is in practice limited to the maximum size of a (squared) kernel matrix that can be stored in memory.

The question is whether is there a two-sample-problem that is so hard that the amount of data needed to solve it (the kernel matrix) is larger than would fit into computer memory. Such a problem could be only solved by a test that uses the linear MMD. Such a problem is now described. This answers one of the questions that were posed in introduction section 1.1.6 to motivate usage of large-scale two-sample tests.

A Very Hard Two-Sample Problem

The *blob* dataset as described in section 5.1.5 may be used to create very tough two-sample problems. The dimension in figure 5.3 is only two – however, even humans have problems in distinguishing problems as depicted.

Used data has parameters: $t = 12$ Gaussian blobs at distance $\Delta = 15$, rotated by $\alpha = \frac{\pi}{4} \equiv 45$ degrees, with the first Eigenvalue of the covariance matrices of q set to

$\epsilon = 1.4$.

As mentioned before, for the quadratic time MMD, multiple kernel matrices have to be stored in memory in order to ensure reasonable run-times. Using 64 bit double precision floating point numbers, a 10000×10000 kernel matrix uses $10000^2 \cdot 64/8/1024^3 \approx 763$MiB. This is on the cutting edge of what is possible with modern office computers. (In practice, three matrices have to be stored. One for each combination of samples X and Y.)

Experimental Setup The quadratic time test is performed on $m = 5000$ and $m = 10000$ points. The dataset itself was designed in such way that the test fails, i.e. has large type two error. A two sample test is constructed using the fast $\mathcal{O}(m^2)$ *gamma* approximation of the null-distribution, as described in section 3.1.4. Type I errors are reported for the *gamma* method (which is heuristic, not consistent) to ensure that tests are comparable (which is only the case if they have equal type I errors).

In contrast, the linear time test is fed with the squared number of points (which corresponds to the same amount of data the quadratic test uses) and more. Thresholds are computed via the Gaussian approximation described in section 3.1.2. Since results in section 5.2 indicate that type I error is controlled properly these are not computed (sample size is in the millions, so the central limit theorem definitely works here). As described in section 3.1.3 the linear time MMD and approximation of its null-distribution in an on-line way.

Linear Time Beats Quadratic Time Table 5.4 shows errors of the described test. Note that the number of samples for the linear time test are averages per trial. This is due to data grouping in distributed computations on a cluster computer. As mentioned above, type I errors are only computed for the quadratic test since the *gamma* method is heuristic. The linear test is consistent and has guaranteed 95% type I error for large sample sizes due to earlier experimental results.

Clearly, the linear time MMD based test is able to outperform the quadratic time MMD based test. For the maximum number of samples for that a kernel matrix can be stored in memory, the quadratic time test reaches a bad type II errors of about 82% and 55% for $m = 5000$ and $m = 10000$ samples respectively. In contrast, the linear time test, fed with squared the number of points and more reaches a better type II error of 25%. For much more data it even reaches almost zero – the problem is solved when five hundred million samples from each p and q are used along with the linear time test.

This result shows that the linear time MMD is useful in large scale cases: for particularly hard problems which require huge amounts of data to be solved and when infinite data is available, it outperforms the quadratic time MMD test whose performance is in practice limited by available computer memory. It also shows that the linear time test is truly large scale: all computations for the on-line linear time MMD as in section 3.1.3 are trivially split into independent sub-jobs on a cluster computer. In the described experiment, the number of individual jobs that were performed in parallel were in the thousands. Such approach is the only way of dealing with five hundred million samples and more importantly, it scales up arbitrarily with the number of cluster computer

	m per trial	Type II error	Trials	∅ Type I error
Quadratic time	5000	$[0.7996, 0.8516]$	820	0.95475
	10000	$[0.5161, 0.6175]$	367	0.94732
Linear time	∅119580000	$[0.2250, 0.3049]$	468	– (0.95)
	∅185130000	$[0.1873, 0.2829]$	302	– (0.95)
	∅502430000	0.0270 ± 0.0302	111	– (0.95)

Table 5.4.: 95% wald error intervals for type two errors of linear time MMD test versus quadratic time MMD test on the described hard problem. The quadratic MMD test used as much data as still allows to store kernel matrices in memory ($m = 5000, 10000$); the linear time MMD test used the same number squared (and more) data. Clearly, additional data allows the linear time test outperform the quadratic one. Note that the number of samples used in linear time test are average per trials (due to distributed implementation).

nodes.

To knowledge of the author, there has not yet been performed an on-line two-sample test on five hundred million (and more) samples. This result also justifies why kernel selection strategies for the linear time test are interesting – there are cases when the linear time test is the only existing method that can be applied.

5.4.2. Fixed Computational Costs, Infinite Data

Another question posed in section 1.1.6 is: given a fixed computational time and infinite data, which test should be chosen? It turns out that described linear time tests in this case perform better than quadratic ones. This is another motivation to use the methods that were described during this work.

Experimental Setup: Limited Time, Infinite Data In the following, the linear time test (sections 2.4.3 and 3.1.2) is compared to the quadratic time test with the *gamma* heuristic (sections 2.4.3 and 3.1.4). This is the same setup as in the above large-scale experiment. Note that the quadratic test already has a downside: it is not consistent when the *gamma* method is used. In practice, this is often neglectable since this heuristic is often accurate. However, in order to get reliable results, bootstrapping (section 3.1.1) is also used to double-check both quadratic and linear test constructions.

Data is taken from the *blobs* dataset. Parameters are the same as for the kernel selection experiment in section 5.3.5: $\alpha = \frac{\pi}{4} \equiv 45$ degrees; rotated Gaussians' stretch $\epsilon \in [2, 15] \subseteq \mathbb{N}$; number of Gaussians $t^2 = 25$ at distance $\Delta = 5$. The *blobs* dataset is chosen here since it yields most distinctive results for kernel selection different methods.

In order to simulate fixed computational costs, the number of samples are chosen in such way that the number of terms that are averaged over in the MMD estimates are equal for both linear and quadratic case. This number is set to $10000, 40000$. This corresponds to $m = 20000, 80000$ samples from each p and q for the linear time statistic

and $m = 100,200$ for the quadratic time statistic. Since test construction is in the same computational cost class, the performed two test procedures have comparable computational costs.

Kernel Selection Since kernel selection strategies influence results significantly, these have to be chosen in such way that a comparison of linear and quadratic time tests is possible. For the quadratic time test, the state-of-the-art method is maximising the MMD statistic itself, [Sriperumbudur et al., 2009], section 3.2.2. A method similar to maximising the ratio of MMD and its standard deviation as for the linear time statistic (section 3.3.2) has not yet been described. Cross-validation based methods are be excluded in the comparison due to their large computational costs. Kernel combinations are not included since MKL-style kernel selection has not yet properly been described for quadratic time tests (This is a further field of work). The median based method is taken out since it usually fails, see section 5.3.5. Instead, a *fixed* kernel whose size has been chosen a-priori is also included. It serves as a baseline method to compare linear and quadratic time tests on a fixed kernel. Since it is chosen to be optimal (data is known), it defines best possible performance. For the linear time test, all methods that are evaluated for the quadratic test are included – plus the *MaxRat* method as described in section 3.3.2.

Linear Time Test Best Quadratic Time Test It turns out that the statistical advantage of being able to select the optimal kernel using *MaxRat* allows the linear test to outperform its quadratic counterpart with *MaxMMD*. In figure 5.21, the extend of this can be observed. For very hard problems, *MaxMMD* on the quadratic time statistic performs slightly better. However, as the problem gets easier, the linear time test using *MaxRat* strongly sets apart. It converges to zero type II errors very fast. In contrast the quadratic test using *MaxMMD* also reaches zero II error, but the problems have to be *much* easier for this to happen. Increasing the number of data does not change this as can bee seen in the plot for 40000 terms on the top right (no bootstrapping performed here since computational costs – there is no difference anyway as the 10000 terms case depicts). This is an important result: when unlimited data is available, the linear time tests in practice reaches a better type II error than the current state-of-the-art method.

A-priori Chosen Fixed Kernels Since data distributions are known, it is possible to select a fixed kernel that optimally fits given data. Using this kernel allows to compare the tests without the influence of an unstable kernel selection method that may cause differences. Results in figure 5.21 show that all methods are outperformed by the quadratic counterpart when using a fixed kernel. Note that the used test construction for the quadratic time test is the *gamma* heuristic described in section 3.1.4. To this end, results of a bootstrapping based version which avoid possible bad behaviour of type I error are shown below left in figure 5.21[3]. Further note that the kernel that performed

[3]In this plot, the quadratic test does not really compete due to the much larger computational costs. However it confirms the results of the *gamma* and Gaussian approximation tests.

best was chosen a-priori – this is not an adaptive method but rather the best that the quadratic time case could get. It is not possible on unknown data in practice. The adaptive method closest to this performance is the linear time test along with *MaxRat*.

There is a difference between the performance yielded by the a-priori chosen kernel and the *MaxRat* method. In theory, this should not happen since *MaxRat* selects the kernel that minimises type II error. However, as already mentioned, *MaxRat* can be unstable. This was observed in kernel weight figures of other experiments, see for example figures 5.11, 5.14. Stabilising the *MaxRat* method in order to reach performance close to optimal is a subject for further investigation.

5.5. Summary

The current chapter described all experimental results that were obtained while this thesis was written. These are grouped into the following parts.

Datasets At first, used datasets were described. These included three simply structured sets from published work and three new sets with a structure that induces difficulty in two-sample testing – differences of underlying distributions are hidden within a different length-scale than the overall data has. A purely synthetic (*blobs*) and a real-world dataset (*am*) were the central benchmarks during the chapter.

In addition, the *selection* dataset was described in order to illustrate usefulness of using multiple kernels in two-sample testing.

Investigation of Test Constructions for Linear Time MMD Since the linear time test relies on a linear time threshold approximation, it is important to investigate threshold accuracy as a function of sample size. The approximation was compared to its ground truth obtained by bootstrapping. Results show that variance of the approximation depends on the used dataset. In addition sample sizes used in later experiments ($m \geq 10000$) lead to reasonably accurate null distribution approximations and thresholds.

Comparison of Different Kernel Selection Methods The most important part of the chapter compared all described methods for kernel selection against each other. On datasets with simple structure, it was hard to show advantages of new methods: very large sample sizes were needed. However, for harder datasets as *blobs* and *am*, the newly descried method for selecting optimal kernels performed significantly and sometimes by magnitudes better than existing methods. In particular, the *median* based method completely failed on difficult datasets. *MaxRat* and *Opt* methods for optimal kernel choice yet in some cases have problems with noise – this remains an open problem. In contrast, *X-Val-Type II* which minimises type II error via cross-validation performed very robust, and in cases where single kernels are sufficient was the best evaluated method.

Results on the *selection* dataset also show that combined kernels can be a significant advantage in two-sample testing. The concept can also be used for feature selection.

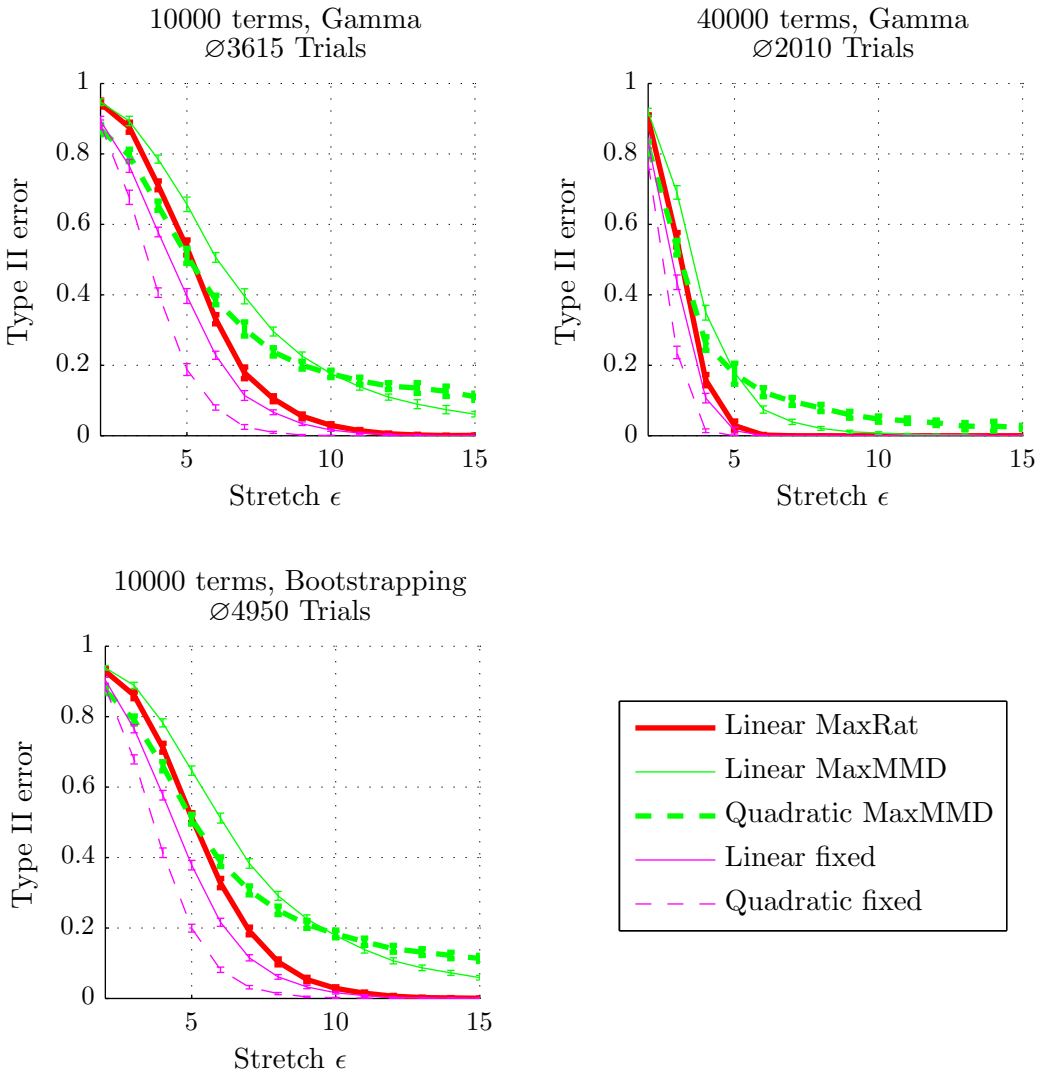


Figure 5.21.: Type II Errors for Linear Time Test vs. Quadratic Time Test. *Blobs* Dataset with parameters: $m = 10000$ samples from each p and q; $\alpha = \frac{\pi}{4} \equiv 45$ degrees; rotated Gaussians' stretch $\epsilon \in [2, 15] \subseteq \mathbb{N}$; number of Gaussians $t^2 = 25$ at distance $\Delta = 5$. The upper plots are computed using the *gamma* method for test construction. The lower plot uses bootstrapping to confirm above results – they do not differ significantly.

Linear Time vs. Quadratic Time Tests In order to argue for using the linear time two-sample test, two experiments that showed its usefulness were described. If problems are very difficult, i.e. the number of samples needed to solve them exceeds memory capacity (> 10000, kernel matrices do not fit into memory), quadratic time tests in practice cannot be used. In contrast, the linear time test is able to process arbitrary amounts of data. In the described experiment, it eventually reached almost zero type II error with about five hundred million processed samples.

Another motivation to use the linear time test is the new method for optimal kernel selection. In euqal computational time, i.e. on the same number of averaged terms in their statistics, the linear time with optimal kernel selection reaches lower type II error than the quadratic test with the state-of-the-art method for kernel selection. Therefore, whenever infinite data is available in practice, the linear time test is superior.

6. Outlook

Chapter Overview This chapter points out open questions for further directions of research that arose during this work and gives partial answers to them.

The chapter is mostly comprised of section 6.1, which is a conjecture for a kernel selection method for quadratic time tests with a criterion similar to the one used with the linear time MMD. The method is briefly described (technical derivations are provided in appendix A.2), and initial experimental results are reported. In addition, section 6.2 contains ideas for making introduced ratios for kernel selection more stable.

Literature & Contributions Section 6.1 suggests a conjecture for a new criterion for kernel selection for the quadratic time MMD. In order to formalise this idea, population expressions around MMD distributions from [Gretton et al., 2012a] are used. Empirical estimates are derived in appendix A.2 using a computational trick from [Song et al., 2007]. Neither this whole approach nor experiments in the section have yet been described elsewhere. While experiments suggest that it could work, a formal justification of the approach is yet missing. Section 6.2 contains brief descriptions of original ideas.

6.1. Kernel Selection for the Quadratic Time MMD

This section picks up the idea for optimal kernel choice that was described in section 3.3.2 and transfers it to context of the quadratic time MMD statistic.

6.1.1. Motivation and Review

In section 3.3, methods for optimal kernel selection for a two-sample test based on the linear time MMD were described. The linear time statistic was chosen since both its null and alternative distributions are normal and have equal variance (Lemma 8). This allows performing kernel selection by maximising a ratio of the linear time statistic to its standard deviation. The selected kernel is optimal in the sense that it has minimal type II error. Empirical estimates for optimisation in practice were provided. In chapter 4, the approach was generalised to selecting optimal weights of a non-negative linear combination of kernels via convex optimisation.

The quadratic time MMD statistic (section 2.4.3) is more stable than its linear counterpart since it considers all pairs of samples and sums over these dependent terms. This yields a much lower variance and therefore in a more stable estimate. The computational costs for this increase from linear to quadratic. To compute the statistic, all data has to be stored in memory whereas the linear time statistic can be computed in an on-line fashion (section 3.1.3). Since in practice, kernel matrices have to be precomputed in

order to ensure reasonable runtimes (see section 5.4.1), it has de-facto quadratic storage costs. All these attributes make the quadratic time statistic relevant to cases where the amount of available data is fixed (and fits into memory) and one wants to get the best possible result. Even with the median based kernel selection, the quadratic time MMD beats many competing methods [Gretton et al., 2012a]. It therefore is desirable to perform kernel selection in the same fashion as for the linear time statistic – choosing an (if possible, optimal) kernel. This section provides some first steps in this direction. Note that these are conjecture only and have not been investigated for theoretical guarantees as for the linear time statistic.

6.1.2. A Criterion Similar to MaxRat

The null distribution of the unbiased quadratic time MMD estimate $\mathrm{MMD}_u^2[\mathcal{F}, X, Y]$ has a complex form: an infinite sum of weighted χ^2 random variables, [Gretton et al., 2012a, Theorem 12]. In particular, it differs from the alternative distribution. Therefore, it is not possible to control both distributions' form by one variance term as done in the linear case where null and alternative distribution have equal variance. In order to pull the distributions apart (maximise MMD), and at the same time keep their tails narrow (minimise variance), both distributions' variances have to be controlled both at the same time.

An intuitive implementation of this is to use the sum of the variances of null and alternative distribution in order to construct a ratio to maximise. Similar to expression 3.6, maximise

$$\frac{\mathrm{MMD}^2[\mathcal{F}, p, q]}{\sqrt{\sigma_0^2 + \sigma_A^2}} \tag{6.1}$$

where σ_0 and σ_A are the standard deviations of null and alternative distributions respectively. Whether this is strategy is optimal in the sense that it minimises probability for type II errors has to be investigated. However, even as a heuristic it might be useful. Intuitively, as in section 3.3, null and alternative distribution are pulled apart while their variance is minimised. In practice, the empirical estimate

$$\frac{\mathrm{MMD}_u^2[\mathcal{F}, X, Y]}{\sqrt{\hat{\sigma}_0^2 + \hat{\sigma}_A^2}} \tag{6.2}$$

would be maximised. Quadratic time estimates for both σ_0^2 and σ_A^2 are derived in appendix A.2.

Initial experiments suggest that derived empirical estimates for σ_0^2 and σ_A^2 are accurate and work in practice. Using the ratio in expression 6.1 for kernel selection in practice seems to work; in performed initial experiments, selected kernels are reasonable in the sense that they are similar to those selected by linear time methods on the same data.

However, these are not reported here due to the lack of time and space to investigate methods and results properly. Next steps involve a comparison of variance estimates for σ_0^2 and σ_A^2 to their ground truth as performed for the linear time case in section 5.2; and a comparison of type II errors compared to other method for kernel selection.

6.1.3. A Generalisation for Combined Kernels

Similar to the approach taken in chapter 4, the above described criterion for kernel selection can be generalised to select weights $\boldsymbol{\beta}$ of finite linear non-negative combinations of kernels from the family in expression 4.1. The resulting population MMD is the same as described in expression 4.2. However, the empirical estimates are different. Redefine the estimate for a combined kernel k from the family in expression 4.1

$$\hat{\eta}_k := \boldsymbol{\beta}^T \hat{\boldsymbol{\eta}}$$

where each entry of $\hat{\boldsymbol{\eta}}$ is the unbiased quadratic time MMD estimator corresponding to one baseline kernel (Note that $\hat{\boldsymbol{\eta}}$ stood for linear time estimates before). This is in-line with the approach for the linear time MMD in section 4.2: the MMD estimate for combined kernels is simply the same combination of estimators for the single kernels' MMD estimates. Next step is to express the variance estimate of null and alternative distribution in terms of $\boldsymbol{\beta}$. This leads to an expression for the sum of null and alternative distribution variance that is similar to the linear time case described in section 4.3.2:

$$\sqrt{\boldsymbol{\beta}^T (Q_0 + 4Q_A)\boldsymbol{\beta}} \tag{6.3}$$

where Q_0 and Q_A are covariance-like matrices that correspond to the empirical covariance matrix Q in the linear time case. Expressions for Q_0 and Q_A in terms of variance estimates are derived in appendix A.2.3. Having access to these estimates, optimisation may be performed in exactly the same fashion as in the linear time case described in section 4.3.3. The resulting convex program is

$$\min\{\boldsymbol{\beta}^T (Q_0 + 4Q_A)\boldsymbol{\beta} : \boldsymbol{\beta}^T \hat{\boldsymbol{\eta}} = 1, \boldsymbol{\beta} \succeq 0\} \tag{6.4}$$

Solving this program with a quadratic solver corresponds to selecting kernel weights in the sense that the criterion in expression 6.2 is maximised.

Initial experiments show that estimated matrices Q_0 and Q_A are positive definite and that mass is put on a smooth "hill" in the middle of the matrix. Using their sum for solving the convex program in expression 6.4 leads to reasonable kernel weights – they are similar to those chosen by optimal linear time kernel selection. As in section 6.1.2, experiments and results are not reported here due to lack of time and space to analyse them properly. However, results are promising and empirically support the conjecture that the ratio in expression 6.1 might be used for selection of single and combined kernels.

Next steps should involve comparison of type II errors against existing methods and a try on the *selection* dataset in order to confirm that chosen kernel weights can be used

for feature selection.

6.2. Numerical Stability of Ratios

A problem that occurs in many experiments in chapter 5 is that kernel selection methods based on the ratio from section 3.3.2, namely *MaxRat* and *Opt*, have problems with numerical stability. Especially when underlying data is numerically difficult as for example the *sine* dataset (see figure 5.11), selected kernel weights contain a lot of noise. Every time a wrong kernel is selected, this produces a type II error – this is undesirable.

A suggestion for fixing this problem is the following. The *X-Val-Type II* method outperforms *MaxRat* almost always and is numerically very stable. This is due to averaging multiple folds in cross-validation. A similar approach could be devised for *MaxRat*: partition data into k disjoint subsets, compute the *MaxRat* criterion for all kernels on all combinations of $k-1$ subsets while discarding the remaining data and select a kernel, use a majority vote on the kernels or the median if these can be ordered. Multiple runs should be performed in order to average over different partitions. Initial experiments suggest this approach vastly increases stability of *MaxRat*.

A downside of this method is that it is only suitable for single kernels. Even worse, if multiple kernels are relevant such as in the *selection* dataset, this method might not select one of these but one in between – leading to worse results. A method that averages over kernel weights should therefore be able to "clean" selected kernel weights while not focussing on a single kernel only. Computing the mean of kernel weight vectors in every run is not suitable since it is not robust to outliers. A method for averaging vectors which encourages sparsity might be well suited. This is a point for further practical investigations.

7. Summary

This chapter collects all results and contributions of this work and gives a summarised overview. There are three main contributions: first, a detailed description of linear-time kernel two-sample tests, based on [Gretton et al., 2012a, Section 6], which can be used on streaming data. In particular, this on-line test was experimentally analysed for infinite data cases and it was shown that it outperforms quadratic tests in certain situations.

Second, a new criterion for (in theory) optimal kernel selection, which can be used along with the described on-line large-scale test. Combining this and the previous method in many cases is superior to state-of-the-art methods, which was shown via both theoretical arguments and empirical evaluations.

Third, both the linear time and quadratic time tests were generalised to work in a multiple-kernel-learning (MKL) style context: a method for selecting weights of finite non-negative linear combinations of kernels via convex optimisation was described. Weights are selected in such way that they optimise the above mentioned criteria that were used for single kernel based tests.

In addition, a number of experiments were made on datasets that were argued to be well suited for benchmarking two-sample tests.

7.1. Main Contributions

Large-Scale Two-Sample Testing One of the central contributions of this work is a detailed analysis of the linear-time two-sample test, as described in [Gretton et al., 2012a, Section 6]. The test is able to work with streaming data, i.e. it does not have to store all data in memory. This work provides a detailed description how it can be constructed in an on-line fashion (section 3.1.2). The accuracy of this approximation was empirically found to be very accurate for reasonable sample sizes as can be seen in the experiment in section 5.4.1.

The state-of-the-art method for kernel two-sample testing has quadratic costs, see section 3.1.4. In order to justify usage of the linear version, which is by definition less accurate on the same number of samples, various arguments were made.

1. It is able to solve yet unsolvable problems: if a problem is so hard that the amount of data (de-facto: size of kernel matrix) that would have to be stored in computer memory is larger than available capacities, it cannot be solved using a quadratic time test. Technically, there is a possibility of doing this via not storing kernel matrices and using bootstrapping, however, this would yield infeasible run times. Using the linear time test, *any* number of sample can be used for testing – arbitrarily hard problems can be solved. In section 5.4.1, an experiment was described

where exactly this happened: quadratic time tests using all available computer memory (ten thousand samples) failed; the linear time test could reach almost zero type II error when using around five hundred million samples.

2. Given a fixed limit of computational time and squared the number of samples compared to the quadratic test, the linear time test reaches lower type II error. Section 5.4.2 describes an experiment for this. The reason for superiority of the linear time test is the possibility of selecting the *optimal* kernel. In contrast, for quadratic time tests, the state-of-the-art method is to maximise the MMD itself in order to select kernels. This makes the linear time test the best available method for cases where infinite data but only finite computation time is available.

These two arguments support using the described linear time test in described situations.

A new Criterion for Kernel Selection for Linear-Time Two-Sample Tests The second main contribution of the work is the mentioned criterion for kernel selection in context of the linear time test, section 3.3.2. This result is also described in Gretton et al. [2012c], which was submitted while this work was written. It is constructed as a ratio of linear time MMD statistic over its standard deviation and is an extension of the method of maximising the MMD statistic only described in section 3.2.2 and in [Sriperumbudur et al., 2009]. The fact that null and alternative distribution of the linear test's statistic are both normal with equal variance can be used to show that this criterion leads to *optimal* kernel selection. For Gaussian data and a Gaussian kernel, a theoretical justification that the limits of described ratio for extreme kernel choices are zero is provided in appendix A.1. The criterion was tested on a number of datasets in section 5.3 and empirically found to have a lower type II error than state-of-the-art methods in many cases. This holds especially when distinguishing characteristics of underlying distributions are situated at a different scaling than of the overall data. See experiments in sections 5.3.3, 5.3.5, and 5.3.7. Moreover, in these cases, the popular method for selecting bandwidth of a Gaussian kernel via using the median data distance described in section 3.2.1 massively fails. Worst case performance of the new method still is comparable to competing methods. It was also illustrated that the criterion is truly adaptive in the sense that it puts weight on kernels that fit data best. See figure 5.12.

There remains one problem when using the criterion, which is the problem that it is not very robust and sometimes leads to very noisy kernel selections and loss of accuracy, see for example figure 5.11. This especially happens when there is much noise in very hard problems. In section 6.2, possible cures for this behaviour are mentioned: results could be averaged in a cross-validation based style. This is a point for further work.

This novel criterion for kernel selection is the reason why linear time tests are able to outperform quadratic tests.

Generalisation of Kernel Two-Sample Testing for Combined Kernels The third main contribution of this work is generalisation of MMD-based tests to using nonnegative

linear combinations of a finite number of kernels, as described in chapter 4. As the new criteria for kernel selection, this approach is also described in [Gretton et al., 2012c]. The MMD for combined kernels becomes the same combination of MMDs for each single kernel as was shown in section 4.2. The same happens to linear time MMD estimates (section 4.3). Construction of a linear time test uses an empirical covariance matrix estimate instead of a variance estimate.

Along with the newly described criterion for optimal kernel selection, a convex program in terms of kernel weights of the underlying combined kernel was derived whose solution corresponds to selecting the weights such way that the previously mentioned criterion is maximised (*Opt* method). See section 4.3.3. Resulting kernel weights are optimal in the sense of the new criterion for optimal kernel choice: they minimise type II error. This connects two of the main results of this work in a useful way. In addition, a similar approach for maximising the MMD, based on L_2 norm regularisation, was described for comparison against *MaxMMD* in the combined kernel case. The approach may be used for the quadratic time statistic. Along with the linear time test, it is outperformed by the optimal *Opt* method.

Usefulness of combined kernels in two-sample testing was demonstrated on a dataset where kernel selection corresponded to feature selection: usage of combined kernels resulted in lower type II errors, unnecessary features were not taken into account at all. Section 5.3.6. This approach opens two-sample testing in practice to approaches who have been successfully used in context of multiple kernel learning – multiple kernels of different domains may be combined to find out which domains are relevant for the problem. There has not yet any MKL-style two-sample test yet been published.

7.2. Other Results

Cross-Validation for Selecting Single Kernels In [Sugiyama et al., 2011], a cross-validation based method for selecting a kernel was described. It is based on minimising linear loss of a binary classifier that corresponds to the MMD and by that is related to maximising the MMD itself. See section 3.2.3 and [Sriperumbudur et al., 2009]. The approach was shown to outperform the median based approach described in section 3.2.1. Since the only fundamental difference between minimising linear loss and maximising MMD is a protection of overfitting, this work included both approaches in experiments in order to find out whether they yield different results. Comparison results show that no advantage is achieved by the protection of overfitting. Instead, the cross-validation based method is much more robust and therefore leads to less noisy kernel selections and with that to a lower type II error. This makes it slightly superior to maximising the MMD itself. However, it is not suitable for the linear time test due to its computational costs.

Inspired by the robustness of the above method, in section 3.3.3, an alternative cross-validation based method was suggested, which in practice turned out to be the best evaluated method for selecting *single* kernels. It is based on approximation of type II errors. A kernel is selected in such way that error estimates are minimised. Since

the method averages over a number of folds, it is very robust. Computational costs are number of used folds times costs for performing a complete test (compute statistic & threshold) – beating the cross-validation of linear loss in the linear time case. The approach may also be useful for quadratic time tests, but ultimately depends on an accurate method of computing a test threshold.

Comprehensive Benchmarks for Two-Sample Testing In order to effectively benchmark methods for two-sample testing, one needs synthetic datasets sampled from known distributions: only when distributions, are therefore structure of data, is known, performance of different methods can be interpreted in a meaningful way. If this structure is unknown, there can be made only guesses *why* methods perform differently. This work introduced a number of possible benchmarks for two-sample testing on numerical data. See section 5.1.

Related to that, and in addition, this work showed that difficulty in two-sample testing of numerical data mainly arises if signal that distinguishes distributions is hidden at a different length scale that the one of the overall data. It was shown that popular methods, such as the median distance heuristic, section 3.2.1, completely fail in such a context. However, in many cases in literature, the median method was successfully used. This shows that used datasets were easy in the sense that signal that distinguishes distributions dominates used samples. It also shows the need of more sophisticated benchmarks to evaluate methods for two-sample testing. This work closes this gap by having introduced such datasets. Especially the *blobs* and *am* datasets, sections 5.3.5, 5.3.7, are very well suited since distinguishing signal is hidden. Both datasets represent benchmarks on purely synthetic data and on a mixture of synthetic and real-world data. The *blobs* dataset is particularly well suited since it may be used to create arbitrarily difficult problems and since results on it are distinct for different methods and in-line with theoretical expectations of these.

A. Proofs Omitted in the Main Text

This appendix contains technical derivations that were skipped in the main text for reasons of readability.

A.1. Limits of Ratio Criterion: Gaussian Kernel & Data

This section examines limiting expressions of the ratio of linear time MMD and its standard deviation as described in section 3.3.2. A fixed context is chosen: Gaussian kernels and one-dimensional data where the distributions p and q have a different mean. See section 3.3.2 for a motivation on this.

A.1.1. Formal Problem Description

Let x, y be two univariate Gaussian random variables with unit variance and distance d, i.e.

$$p(x) = \mathcal{N}_x(0,1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right)$$
$$p(y) = \mathcal{N}_y(d,1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x-d)^2\right)$$

and let $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a Gaussian kernel with bandwidth α[1], as described in section 2.3.3, i.e.

$$k(x,y) = \exp\left(-\frac{(x-y)^2}{2\sigma}\right)$$

Using the notational short-cut $\mathbf{E}_{x,y}$ for $\mathbf{E}_{x\sim p, y\sim q}$, the problem is to come up with expressions for $\mathrm{MMD}_l^2[\mathcal{F}, p, q]$ and its variance σ_l^2 – in terms of kernel with α. In other words, an expression in terms of α for

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = \mathbf{E}_{x,x'}k(x,x') - 2\mathbf{E}_{x,y}k(x,y) + \mathbf{E}_{y,y'}k(y,y') \tag{A.1}$$

and

$$\sigma_l^2 = 2\left[\mathbf{E}_{z,z'}h^2(z,z') - [\mathbf{E}_{z,z'}h(z,z')]^2\right] \tag{A.2}$$

[1] Note that α here is *not* related to type I error of a two-sample test.

where x', y' are independent copies of x, y and $z = (x, y) \sim p \times q$ and

$$h((x, y), (x', y')) = k(x, x') + k(y, y') - k(x, y') - k(x', y)$$

is searched. In particular, the question of what happens to the population ratio

$$\frac{\mathrm{MMD}^2[\mathcal{F}, p, q]}{\sigma_l} \tag{A.3}$$

for the cases $\alpha \to 0$ and $\alpha \to \infty$ is investigated. In order to answer this, expectations of kernel functions and integrals of products of Gaussian distribution functions have to be carried out. The following two rules are used. They can for example be found in any book with probability theory, such as [Barber, 2012, Section 8.4].

Product of two Gaussian Densities The product of two Gaussian distribution functions is given as

$$\mathcal{N}_x(\mu_a, \sigma_a^2)\mathcal{N}_x(\mu_b, \sigma_b^2) = z\mathcal{N}_x(\mu_c, \sigma_c^2)$$

where

$$\begin{aligned}
\sigma_c^2 &= \frac{\sigma_a^2 \sigma_b^2}{\sigma_a^2 + \sigma_b^2} \\
\mu_c &= \frac{\sigma_a^2 \mu_b + \sigma_b^2 \mu_a}{\sigma_a^2 + \sigma_b^2} \\
z &= \mathcal{N}_{\mu_b}(\mu_a, \sigma_a^2 + \sigma_b^2) = \mathcal{N}_{\mu_a}(\mu_b, \sigma_a^2 + \sigma_b^2)
\end{aligned}$$

Integral of Products of Two Gaussian Densities The integral of the product of two Gaussian distribution functions is given as

$$\int_{-\infty}^{+\infty} \mathcal{N}_x(\mu_a, \sigma_a^2)\mathcal{N}_x(\mu_b, \sigma_b^2)dx = z \int_{-\infty}^{+\infty} \mathcal{N}_x(\mu_c, \sigma_c^2)dx = z$$

A.1.2. Population MMD in Terms of Kernel Width α

In order to write expression A.1 in terms of α, expectation of a kernel where both arguments are different random variables is considered, i.e.

$$\begin{aligned}
\mathbf{E}_{x,y} k(x, y) &= \int_y \int_x k(x, y) p(x, y) dx dy && \text{(Definition of expected value)} \\
&= \int_y \int_x k(x, y) p(x) p(y) dx dy && (p(x) \text{ and } p(y) \text{ are independent}) \\
&= \int_y \left(\int_x k(x, y) p(x) dx \right) p(y) dy && \text{(again independent)}
\end{aligned}$$

The inner integral is

$$\begin{aligned}
\int_x k(x,y)p(x)dx &= \int_x \exp\left(-\frac{(x-y)^2}{2\alpha}\right)\mathcal{N}_x(0,1)dx && \text{(Definition of } k \text{ and } p\text{)} \\
&= \int_x \sqrt{2\pi\alpha}\mathcal{N}_x(y,\alpha)\mathcal{N}_x(0,1)dx && (k \text{ as a Gaussian PDF}) \\
&= \sqrt{2\pi\alpha}\mathcal{N}_y(0,\alpha+1) && \text{(Product Integral Rule)}
\end{aligned}$$

Substitution into above term gives

$$\begin{aligned}
\int_y \left(\int_x k(x,y)p(x)dx\right)p(y)dy &= \int_y \sqrt{2\pi\alpha}\mathcal{N}_y(0,\alpha+1)p(y)dy && \text{(Substitution)} \\
&= \sqrt{2\pi\alpha}\int_y \mathcal{N}_y(0,\alpha+1)\mathcal{N}_y(d,1)dy && \text{(Definition of } p(y)\text{)} \\
&= \sqrt{2\pi\alpha}\mathcal{N}_d(0,\alpha+2) && \text{(Product Integral Rule)} \\
&= \frac{\sqrt{2\pi\alpha}}{\sqrt{2\pi(\alpha+2)}}\exp\left(-\frac{d^2}{2(\alpha+2)}\right) \\
&= \sqrt{\frac{\alpha}{\alpha+2}}\exp\left(-\frac{d^2}{2(\alpha+2)}\right)
\end{aligned}$$

Setting $d = 0$ leads to $p(x) = p(y)$ and $x = y$ – the exponential factor at the end becomes one. This gives the expectation for identical random variables, i.e.

$$\mathbf{E}_{x,x'}k(x,x') = \mathbf{E}_{y,y'}k(y,y') = \sqrt{\frac{\alpha}{\alpha+2}}$$

Therefore, the population MMD in terms of kernel width α is

$$\begin{aligned}
\mathrm{MMD}^2[\mathcal{F},p,q] &= 2\sqrt{\frac{\alpha}{\alpha+2}} - 2\sqrt{\frac{\alpha}{\alpha+2}}\exp\left(-\frac{d^2}{2(\alpha+2)}\right) \\
&= 2\sqrt{\frac{\alpha}{\alpha+2}}\left(1-\exp\left(-\frac{d^2}{2(\alpha+2)}\right)\right)
\end{aligned}$$

Note that for $d = 0$, this term becomes zero. To get an idea what happens for large kernel sizes (large α and fixed d), everything will be written in terms of magnitude of α using the symbol $\cong$, i.e.

$$\mathbf{E}_{x,x'}k(x,x') = \mathbf{E}_{y,y'}k(y,y') = \sqrt{\frac{\alpha}{\alpha+2}} \cong 1 \tag{A.4}$$

Then,

$$\mathrm{MMD}^2[\mathcal{F}, p, q] = 2\sqrt{\frac{\alpha}{\alpha+2}}\left(1 - \exp\left(-\frac{d^2}{2(\alpha+2)}\right)\right) \cong 1 - \exp\left(-\frac{d^2}{\alpha}\right)$$

Therefore, as suspected

$$\lim_{\alpha\to 0} \mathrm{MMD}^2[\mathcal{F}, p, q] \cong 1$$
$$\lim_{\alpha\to \infty} \mathrm{MMD}^2[\mathcal{F}, p, q] \cong 0$$

A.1.3. Variance of Linear Time MMD in Terms of Kernel Width α

In order to write expression A.2 in terms of α, consider both inner terms separately. The first inner term is

$$\begin{aligned}
\mathbf{E}_{z,z'} h^2(z, z') = \mathbf{E}_{z,z'} \quad & (k(x, x') + k(y, y') - k(x, y') - k(x', y))^2 \\
= \mathbf{E}_{z,z'} \quad & ([k(x, x') + k(y, y')] - [k(x, y') + k(x', y)])^2 \\
= \mathbf{E}_{z,z'} \quad & k^2(x, x') + k^2(y, y') + 2k(x, x')k(y, y') \\
+ \quad & k^2(x, y') + k^2(x', y) + 2k(x, y')k(x', y) \\
- \quad & 2k(x, x')k(x, y') - 2k(x, x')k(x', y) \\
- \quad & 2k(y, y')k(x, y') - 2k(y, y')k(x', y)
\end{aligned} \tag{A.5}$$

Writing this down exactly becomes infeasible since more than two Gaussian distributions are multiplied. Therefore, everything is written in magnitude of α – using $\cong$. The squared kernels in expression A.5 are analysed first. A squared Gaussian kernel is simply another Gaussian kernel with half bandwidth, i.e.

$$k_\alpha^2(x, y)\left(\exp\left(-\frac{(x-y)^2}{2\alpha}\right)\right)^2 = \exp\left(-\frac{(x-y)^2}{\alpha}\right) = k_{\frac{\alpha}{2}}(x, y)$$

Using expression A.4, this leads to

$$\mathbf{E}_{z,z'} k^2(x, x') = \mathbf{E}_{x,x'} k^2(x, x') = \mathbf{E}_{y,y'} k^2(y, y') \cong 1$$

Similarly

$$\mathbf{E}_{z,z'} k^2(x, y') = \mathbf{E}_{z,z'} k^2(x', y) = \mathbf{E}_{x,y} k^2(x, y) \cong \exp\left(-\frac{d^2}{\frac{\alpha}{2}}\right) \cong \exp\left(-\frac{2d^2}{\alpha}\right)$$

Next, use the fact that expectations of independent products factorise, i.e.

$$\mathbf{E}_{z,z'} k(x, x')k(y, y') = \mathbf{E}_{x,x'} k(x, x')\mathbf{E}_{y,y'} k(y, y') \cong 1 \cdot 1 \cong 1$$

and

$$\mathbf{E}_{z,z'}k(x,y')k(x',y) \cong \exp\left(-\frac{2d^2}{\alpha}\right)$$

Continue with terms in expression A.5 that depend on three random variables. Using similar arguments as in the MMD case,

$$\begin{aligned}\mathbf{E}_{z,z'}k(x,x')k(x,y') &\cong \int_{x'}\int_{x}\int_{y} k(x,x')k(x,y)p(x,x',y)dydxdx' \\ &\cong \int_{x'}\int_{x} k(x,x')\int_{y} k(x,y)p(y)dy \cdot p(x,x')dxdx'\end{aligned}$$

First, consider the inner integral

$$\begin{aligned}\int_{y} k(x,y)p(y)dy &\cong \int_{y} \sqrt{\alpha}\mathcal{N}_y(x,\alpha)\mathcal{N}_y(d,1)dy \\ &\cong \sqrt{\alpha}\mathcal{N}_x(d,\alpha)\end{aligned}$$

Plug into above expression

$$\mathbf{E}_{z,z'}k(x,x')k(x,y') \cong \sqrt{\alpha}\int_{x'}\int_{x} k(x,x')\mathcal{N}_x(d,\alpha)p(x)dx \cdot p(x')dx'$$

Consider the inner integral of this expression

$$\begin{aligned}\int_{x} k(x,x')\mathcal{N}_x(d,\alpha)p(x)dx &\cong \sqrt{\alpha}\int_{x} \mathcal{N}_x(x',\alpha)\mathcal{N}_x(d,\alpha)\mathcal{N}_x(0,1)dx \\ &\cong \sqrt{\alpha}\mathcal{N}_{x'}(d,\alpha)\int_{x} \mathcal{N}_x(x'+d,\alpha)\mathcal{N}_x(0,1)dx \\ &\cong \sqrt{\alpha}\mathcal{N}_{x'}(d,\alpha)\mathcal{N}_{x'}(-d,\alpha) \\ &\cong \sqrt{\alpha}\frac{1}{\alpha}\exp\left(-\frac{4d^2}{\alpha}\right)\mathcal{N}'_x(0,\alpha) \\ &\cong \exp\left(-\frac{4d^2}{\alpha}\right)\mathcal{N}'_x(0,\alpha)\end{aligned}$$

Plug in in again

$$\begin{aligned}
\mathbf{E}_{z,z'}k(x,x')k(x,y') &\cong \sqrt{\alpha}\int_{x'}\int_{x} k(x,x')\mathcal{N}_x(d,\alpha)p(x)dx\cdot p(x')dx' \\
&\cong \sqrt{\alpha}\int_{x'} \exp\left(-\frac{4d^2}{\alpha}\right)\mathcal{N}_{x'}(0,\alpha)p(x')dx' \\
&\cong \sqrt{\alpha}\exp\left(-\frac{4d^2}{\alpha}\right)\int_{x'}\mathcal{N}_{x'}(0,\alpha)\mathcal{N}_{x'}(0,1)dx' \\
&\cong \exp\left(-\frac{4d^2}{\alpha}\right)
\end{aligned}$$

Since all the variance's first term's terms with the factor two have the same structure, they all have the same value in magnitude of α. Together, this gives complete expression A.5 which is the first inner term of expression A.2 in terms of α

$$\begin{aligned}
\mathbf{E}_{z,z'}h^2(z,z') &\cong 4 + 4\exp\left(-\frac{2d^2}{\alpha}\right) - 8\exp\left(-\frac{4d^2}{\alpha}\right) \\
&\cong 2 + 2\exp\left(-\frac{2d^2}{\alpha}\right) - 4\exp\left(-\frac{4d^2}{\alpha}\right)
\end{aligned}$$

Now, the second inner term of expression A.2 is considered. This is easy, since the expected value is over the single kernels, use magnitude results from above:

$$\begin{aligned}
[\mathbf{E}_{z,z'}h(z,z')]^2 &\cong [\mathbf{E}_{z,z'}k(x,x') + k(y,y') - k(x,y') - k(x',y)]^2 \\
&\cong \left(2 - 2\exp\left(-\frac{d^2}{\alpha}\right)\right)^2 \\
&\cong 4 - 8\exp\left(-\frac{d^2}{\alpha}\right) + 4\exp\left(-\frac{2d^2}{\alpha}\right) \\
&\cong 2 + 2\exp\left(-\frac{2d^2}{\alpha}\right) - 4\exp\left(-\frac{d^2}{\alpha}\right)
\end{aligned}$$

The complete expression A.2 in terms of α is then

$$\begin{aligned}
\sigma_l^2 &= 2\left[\mathbf{E}_{z,z'}h^2(z,z') - [\mathbf{E}_{z,z'}h(z,z')]^2\right] \\
&\quad 2 + 2\exp\left(-\frac{2d^2}{\alpha}\right) - 4\exp\left(-\frac{4d^2}{\alpha}\right) - \left(2 + 2\exp\left(-\frac{2d^2}{\alpha}\right) - 4\exp\left(-\frac{d^2}{\alpha}\right)\right) \\
&\cong 4\exp\left(-\frac{d^2}{\alpha}\right) - 4\exp\left(-\frac{4d^2}{\alpha}\right) \\
&\cong \exp\left(-\frac{d^2}{\alpha}\right) - \exp\left(-\frac{4d^2}{\alpha}\right)
\end{aligned}$$

Therefore, as suspected

$$\lim_{\alpha \to 0} \sigma_l^2 \cong 0 - 0 \cong 0$$
$$\lim_{\alpha \to \infty} \sigma_l^2 \cong 1 - 1 \cong 0$$

A.1.4. Ratio Limits in Terms of α

So far, it was shown that both expression A.1 and expression A.2 converge to zero for $\alpha \to 0$ and $\alpha \to \infty$. The original question was what happens to the ratio limit in expression A.3.

$$\begin{aligned}
\frac{\mathrm{MMD}^2[\mathcal{F}, p, q]}{\sigma_l} &\cong \frac{1 - \exp\left(-\frac{d^2}{\alpha}\right)}{\sqrt{\exp\left(-\frac{d^2}{\alpha}\right) - \exp\left(-\frac{4d^2}{\alpha}\right)}} \\
&\cong \frac{\exp\left(\frac{2d^2}{\alpha}\right)}{\exp\left(\frac{2d^2}{\alpha}\right)} \frac{1 - \exp\left(-\frac{d^2}{\alpha}\right)}{\sqrt{\exp\left(-\frac{d^2}{\alpha}\right) - \exp\left(-\frac{4d^2}{\alpha}\right)}} \\
&\cong \frac{\exp\left(\frac{2d^2}{\alpha}\right) - \exp\left(\frac{d^2}{\alpha}\right)}{\sqrt{\exp\left(\frac{3d^2}{\alpha}\right) - 1}} \\
&\approx \frac{\frac{2d^2}{\alpha} + 1 - \frac{d^2}{\alpha} - 1}{\sqrt{\frac{3d^2}{\alpha} + 1 - 1}} \\
&= \frac{\frac{d^2}{\alpha}}{\sqrt{\frac{3d^2}{\alpha}}} = \frac{d^2}{\alpha} \frac{\sqrt{\alpha}}{\sqrt{3d^2}} = \frac{d}{\sqrt{\alpha 3}}
\end{aligned}$$

where the Taylor approximation $\exp(x) \approx 1 + x$ for small x was used. This term drops to 0 for $\alpha \to \infty$. The case $\alpha \to 0$ works out similar.

This completes showing that the ratio of population MMD and variance its linear time estimate in expression A.3 is bounded for the context of Gaussian data and kernels.

A.2. Variance Estimates for Quadratic Time MMD

This section established quadratic time estimate for null and alternative distribution of the unbiased quadratic time MMD estimate (section 2.4.3) which is used to sketch an idea for kernel selection in outlook section 6.1.

A.2.1. Variance of MMD_u^2 Under Null Hypothesis

The population variance of MMD_u^2 under H_0 is given in [Gretton et al., 2012a, Appendix B.3]

$$\begin{aligned}\sigma_0^2 &:= \frac{2}{m(m-1)}\mathbf{E}_{z,z'}[h^2(z,z')] \\ &= \frac{2}{m(m-1)}\mathbf{E}_{z,z'}(k(x,x')+k(y,y')-k(x,y')-k(x',y))^2 \end{aligned} \tag{A.6}$$

An unbiased empirical estimate con be obtained by averaging over the two variables while taking care that all terms are independent, i.e.

$$\begin{aligned}\hat{\sigma}_0^2 &= \frac{2}{m^2(m-1)^2}\sum_{i=1}^{m}\sum_{j\neq i}^{m}(k(x_i,x_j)+k(y_i,y_j)-k(x_i,y_j)-k(x_j,y_j))^2 \\ &= \frac{2}{m^2(m-1)^2}\mathbf{1}^T(\tilde{K}_{XX}+\tilde{K}_{YY}-\tilde{K}_{XY}-\tilde{K}_{YX})^{\cdot 2}\mathbf{1}^T \\ &= \frac{2}{m^2(m-1)^2}\mathbf{1}^T\tilde{H}^{\cdot 2}\mathbf{1}^T \end{aligned} \tag{A.7}$$

where $\tilde{K}_{XY}$ is the kernel matrix between X and Y with the modification $\operatorname{diag}\tilde{K}_{XY}=\mathbf{0}$, $K^{\cdot 2}$ is the element wise square of matrix K, and the matrix H is given by

$$H := K_{XX}+K_{YY}-K_{XY}-K_{YX}$$

$\tilde{H}$ is the same matrix with $\operatorname{diag}\tilde{H}=\mathbf{0}$. This estimate can be computed in quadratic time.

A.2.2. Variance of MMD_u^2 Under Alternative Hypothesis

The distribution of MMD_u^2 under H_A is Gaussian with variance $4\sigma_A^2$, where

$$\begin{aligned}\sigma_A^2 &:= \operatorname{var}(\mathbf{E}_z'(h(z,z'))) \\ &= \mathbf{E}_z[(\mathbf{E}_{z'}h(z,z'))^2]-(\mathbf{E}_{z,z'}h(z,z'))^2 \end{aligned} \tag{A.8}$$

See [Gretton et al., 2012a, Corollary 16]. An empirical estimate is now established by considering both terms in the above expression separately.

First term A naive empirical estimate for

$$\begin{aligned}\mathbf{E}_z[(\mathbf{E}_{z'}h(z,z'))^2] &= \mathbf{E}_z[\mathbf{E}_{z',z''}h(z,z')h(z,z'')] \\ &= \mathbf{E}_{z,z',z''}h(z,z')h(z,z'')\end{aligned}$$

can be obtained by averaging independent terms, i.e.

$$\frac{1}{m(m-1)(m-2)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} \sum_{k \neq i, k \neq j}^{m} H_{ij} H_{ik}$$

where H is defined as above. This is estimate however has cubic costs. Following [Song et al., 2007, Proof of Theorem 2], a cheaper estimate is

$$\frac{1}{m(m-1)(m-2)} \left(\mathbf{1}^T \tilde{H}^2 \mathbf{1} - \operatorname{tr}(\tilde{H}^2) \right)$$

where $\tilde{H}$ is, as above, H with its diagonal entries set to zero. This estimate can be computed in quadratic time.

Second term A naive estimate for

$$\mathbf{E}_{z,z'} h(z, z'))^2 = \mathbf{E}_{z,z',z'',z'''} h(z, z') h(z'', z''')$$

can be again obtained by averaging independent terms, i.e.

$$\frac{1}{m(m-1)(m-2)(m-3)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} \sum_{k \neq i, k \neq j}^{m} \sum_{l \neq i, l \neq j, l \neq k}^{m} H_{ij} H_{kl}$$

Following again [Song et al., 2007, Proof of Theorem 2], a quadratic time version is

$$\frac{1}{m(m-1)(m-2)(m-3)} \left(\left(\mathbf{1}^T \tilde{H} \mathbf{1} \right)^2 - 4 \mathbf{1}^T \tilde{H}^2 \mathbf{1} + 2 \operatorname{tr}(\tilde{H}^2) \right)$$

Complete Empirical Estimate Putting the above paragraphs together gives an unbiased quadratic time empirical estimate for σ_A

$$\begin{aligned} \hat{\sigma}_A^2 = & \frac{1}{m(m-1)(m-2)} \left(\mathbf{1}^T \tilde{H}^2 \mathbf{1} - \operatorname{tr}(\tilde{H}^2) \right) \\ & + \frac{1}{m(m-1)(m-2)(m-3)} \left(\left(\mathbf{1}^T \tilde{H} \mathbf{1} \right)^2 - 4 \mathbf{1}^T \tilde{H}^2 \mathbf{1} + 2 \operatorname{tr}(\tilde{H}^2) \right) \end{aligned} \tag{A.9}$$

This completes the empirical estimate of the ratio in expression 6.2.

A.2.3. Variances in Terms of Kernel Weights

Using the population expressions for variances as given in expressions A.6 and A.8, an empirical estimate for the sum of variances of null and alternative distributions is derived following section 4.3.1: the estimate in terms of kernel weights $\boldsymbol{\beta}$ for a combined kernel

k from the conical kernel family in expression 4.1 takes the form

$$\begin{aligned}\sqrt{\hat{\sigma}_{k,0}^2 + \hat{\sigma}_{k,A}^2} &= \sqrt{\boldsymbol{\beta}^T Q_0 \boldsymbol{\beta} + 4\boldsymbol{\beta}^T Q_A \boldsymbol{\beta}} \\ &= \sqrt{\boldsymbol{\beta}^T (Q_0 + 4Q_A)\boldsymbol{\beta}}\end{aligned}$$

where the matrices Q_0 and Q_A correspond to the empirical estimate of the covariance $\mathrm{cov}(\mathbf{h})$ in the linear time case. These can be computed using the expressions from previous sections. The matrix corresponding to the null distribution can be computed using expression A.7 where two $H^{(i)}, H^{(j)}$ matrices are used – one for each kernel

$$(Q_0)_{ij} = \frac{2}{m^2(m-1)^2} \mathbf{1}^T \tilde{H}^{(i)} \cdot H^{(j)} \mathbf{1}^T$$

Similarly, the covariance-like matrix for the alternative distribution can be computed using expression A.9 where again, each matrix $H^{(i)}, H^{(j)}$ corresponds to one kernel

$$\begin{aligned}(Q_A)_{ij} = &\frac{\mathbf{1}^T \tilde{H}^{(i)} \tilde{H}^{(j)} \mathbf{1} - \mathrm{tr}(\tilde{H}^{(i)} \tilde{H}^{(j)})}{m(m-1)(m-2)} \\ &+ \frac{\left(\mathbf{1}^T \tilde{H}^{(i)} \mathbf{1}\right)\left(\mathbf{1}^T \tilde{H}^{(j)} \mathbf{1}\right) - 4\mathbf{1}^T \tilde{H}^{(i)} \tilde{H}^{(j)} \mathbf{1} + 2\,\mathrm{tr}(\tilde{H}^{(i)} \tilde{H}^{(j)})}{m(m-1)(m-2)(m-3)}\end{aligned}$$

Both matrices can be computed in quadratic time if all d matrices H are precomputed and stored since the latter are used multiple times. Storing each of these has quadratic space costs. So for each additional considered kernel, a quadratic kernel matrix has be stored in order to ensure a reasonable runtime.

B. Open-Source Implementation: SHOGUN

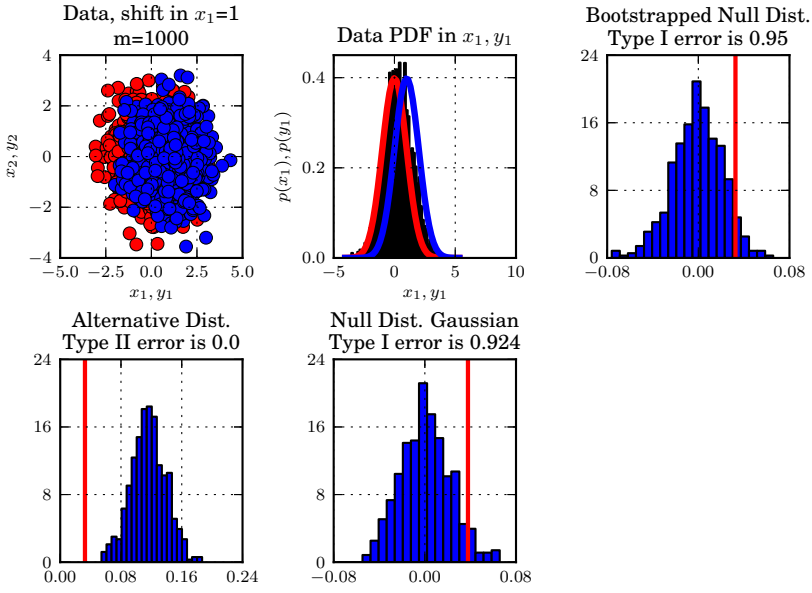
While working on this thesis, most described methods were implemented as a framework for statistical hypothesis testing into an open-source machine learning library called *SHOGUN*[1] [Sonnenburg et al., 2010]. This happened in context of the *Google Summer of Code 2012 (GSOC)*[2], which is a program that offers student stipends to write code for open source projects. The author's project proposal is available on-line[3].

SHOGUN implements many state-of-the-art machine learning algorithms such as many kernels, SVM and SVR, Gaussian processes and mixture models, clustering, LASSO and multitask-learning, dimensionality reduction, model selection and cross-validation, etc. While its core is written in C++ for efficiency, there exist modular bindings to many languages including Python, Matlab/Octave, R, Java, and others.

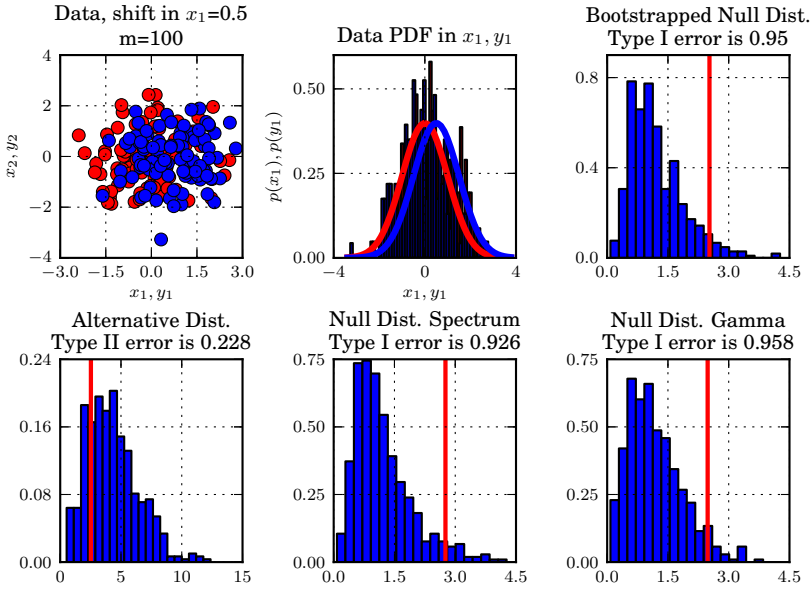
Along with kernel based two-sample tests based on the linear and quadratic time MMD, kernel based independence tests based on the *Hilbert-Schmidt Independece Criterion (HSIC)* were also implemented during this summer. Class documentation for MMD based two-sample tests can be found on-line[4], as can be examples[5]. Figure B.1 shows screenshots of graphical Python examples for linear and quadratic time MMD. As of early September 2012, all work is included in the official SHOGUN 2.0 release.

The release brings many concepts of this thesis to the public – including the selection of optimal kernel weights for the linear time MMD that is described in sections 3.3.2, 4.3 and also went into [Gretton et al., 2012c]. More importantly, it allows reproducing results of this thesis without the need of a large amount of programming. Well-designed, documented, and ready-to-use open-source software is an important addition for machine learning research, [Sonnenburg et al., 2007]. The author of this thesis is confident that including described methods in SHOGUN is useful for the machine learning community. Due to deadline constraints, not all single methods are included yet. However, these will be added in the near future as the author is member an active member[6] of the core development team of SHOGUN.

[1] `http://www.shogun-toolbox.org`
[2] `http://www.google-melange.com/gsoc/homepage/google/gsoc2012`
[3] `http://www.google-melange.com/gsoc/proposal/review/google/gsoc2012/heiko/11002`
[4] `http://www.shogun-toolbox.org/doc/en/current/classshogun_1_1CLinearTimeMMD.html`
`http://www.shogun-toolbox.org/doc/en/current/classshogun_1_1CQuadraticTimeMMD.html`
[5] `http://www.shogun-toolbox.org/doc/en/current/python_modular_examples.html`
[6] `https://github.com/karlnapf`



Linear Time MMD example



Quadratic Time MMD example

Figure B.1.: Screenshots of graphical Python examples for linear and quadratic time MMD along with *mean* dataset in SHOGUN.

Bibliography

Argyriou, A., Hauser, R., Micchelli, C., and Pontil, M. (2006). A DC-programming algorithm for kernel selection. In *Proceedings of the 23rd international conference on Machine learning*, pages 41–48. ACM.

Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Berger, G. and Casella, R. (2002). *Statistical Inference*. Duxbury Press.

Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J. (2006). Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics (Oxford, England)*, 22(14):e49–57.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Caruana, R. (1997). Multitask Learning. *Machine Learning*, 28(1):41–75.

Dudley, R. (2002). *Real Analysis and Probability*. Cambridge University Press.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012a). A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13:671–721.

Gretton, A., Fukumizu, K., Harchaoui, Z., and Sriperumbudur, B. K. (2012b). A fast, consistent kernel two-sample test. *Advances in Neural Information Processing Systems*, pages 673–681.

Gretton, A., Sriperumbudur, B., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., and Fukumizu, K. (2012c). Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems*.

Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2003). A practical guide to support vector classification. Technical Report 1.

Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. (2008). SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes in Machine Learning*. MIT Press.

Reed, M. and Simon, B. (1980). *Methods of Modern Mathematical Physics. Volume 1: Functional Analysis*, volume 1. Gulf Professional Publishing.

Schölkopf, B. (1997). *Support Vector Learning*. PhD thesis, TU Berlin.

Schölkopf, B. and Smola, A. J. (2001). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.

Serfling, R. (1980). *Approximation theorems of mathematical statistics*. Wiley, New-York.

Shawe-Taylor, J. and Cristianini, N. (2000). *An introduction to support Vector Machines and other kernel-based learning methods*. Cambridge University Press.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Song, L., Smola, A., Gretton, A., Bedo, J., and Borgwardt, K. (2007). Feature Selection via Dependence Maximization. *Journal of Machine Learning Research*, 1.

Sonnenburg, S., Braun, M. L., Ong, C. S., Bengio, S., Bottou, L., Holmes, G., Lecun, Y., Müller, K.-R., Pereira, F., and Rasmussen, C. E. (2007). The Need for Open Source Software in Machine Learning. *Journal of Machine Learning Research*, 8:2443–2466.

Sonnenburg, S., Rätsch, G., Henschel, S., Widmer, C., Behr, J., Zien, A., Bona, F., Binder, A., Gehl, C., and Franc, V. (2010). The SHOGUN machine learning toolbox. *The Journal of Machine Learning Research*, 99:1799–1802.

Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565.

Sriperumbudur, B., Fukumizu, K., Gretton, A., Lanckriet, G. R. G., and Schölkopf, B. (2009). Kernel choice and classifiability for RKHS embeddings of probability distributions. *Advances in Neural Information Processing Systems*.

Sriperumbudur, B., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. G. (2010). Hilbert space embeddings and metrics on probability measures. *Journal of-Machine Learning Research*, 11:1517–1561.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Verlag.

Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *The Journal of Machine Learning Research*, 8:985–1005.

Sugiyama, M., Suzuki, T., Itoh, Y., Kanamori, T., and Kimura, M. (2011). Least-squares two-sample test. *Neural networks : the official journal of the International Neural Network Society*, 24(7):735–51.