Using Score-based Methods for Unnormalisable Probability Density Estimation

Truncated Density Estimation and Parameter Derivative Estimation

By

Daniel Williams

Department of Engineering Mathematics
University of Bristol

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Engineering.

April 2023

Word count: 50,899

Abstract

Classical statistical modelling such as maximum likelihood estimation relies on knowledge of the normalising constant of a probability density model. Under certain cases, for example where data are observed on a generic truncated domain, the normalising constant is intractable. Whilst conventional methods usually approximate this term via numerical integration, methods such as score matching (Hyvärinen, 2005) and minimum Stein discrepancy estimators (Barp et al., 2019; Liu et al., 2016) bypass its evaluation entirely.

In chapter 3, we present a method that uses score matching to define a density estimator on a generic truncation domain. This method relies on a weighting function that is equal to zero at the boundary of the domain, for which we propose a distance function which arises naturally from maximising a Stein discrepancy. Numerical experiments show the potential of the proposed method across a range of experiments, including real world Chicago crime data, and correcting the over-compensation from outlier trimming.

In chapter 4, we extend the work from chapter 3 where data are both truncated and lie on the surface of a manifold. We present an application of this method to the sphere, and propose two distance functions as the corresponding weighting function. Experiments on the sphere show that this method achieves lower estimation error than prior Euclidean domain methods. We present a real-world application estimating the mean of storm locations truncated over the continental United States.

In chapter 5, we define an approximate Stein class for which the Stein identity holds approximately. This enables construction of a truncated kernelised Stein discrepancy, which only requires access to a set of boundary points. In experiments, we show that even with this relaxed set of assumptions, the method is comparable in performance to previous methods.

Finally, in chapter 6, we derive an estimator of the time-varying derivative of the parameter of an unnormalised exponential family density using time score matching (Choi et al., 2022). We present a changepoint detection method based on the asymptotic variance of this estimator which needs no prior assumptions on the type of changepoint expected. This highly flexible method provides comparable results to previous popular changepoint detection methods. We show its applicability to real-world applications, as well as the ability to detect when out-of-distribution images are introduced to a dataset using a neural network representation.

Dedication and Acknowledgements

For Romana
And Joe Abercrombie
But since Joe probably ain't that bothered
Mostly Romana

To my supervisor, Dr Song Liu, you have pushed me to being the best researcher I can be: you understood my abilities, flaws and limitations. This PhD would not have been as successful without you. This will have a lasting impact on my career and hence my life, and I cannot thank you enough.

To all my other colleagues at the University, thank you for being a good source of companionship and a constant source of additional help when I needed it. Most notably, Dom, for answering all my incessant messaging about changepoint detection. To Michael, Jake, Jack, Mingxuan, and everyone else in my cohort, thank you for your support and friendship.

To my friends everywhere else, whilst not directly associated to the research, times spent with you all kept me sane and focused, helping me to see the bigger picture in life. To Josh Worthington, you are my oldest friend and have always been there no matter what. To Johnny, Sam, Ali, Ed, Freya, Hannah, and Jess; friends old and new, thank you for the valuable time we've spent together.

To my wonderful fiancée and best friend, Romana. For reading through my work, keeping me calm, cooking all my dinners, making sure I stay grounded, and so many more things I cannot possibly list - thank you. You always make time for me, uprooted to Bristol with me so we can pursue PhDs together, and it goes without saying I wouldn't be where I am today without you. Thank you for your help, support, and neverending love.

Last, but the opposite of least, my family. To my brother, Matt, you have constantly raised my spirits over the years and kept me laughing. To my Mum, you alway ask about my work without ever understanding it, and have always been my inspiration. Finally, to my Dad. I would not be writing this if it were not for you. For sitting by my bedside late into the night, talking about the latest scientific article you read, encouraging me to pursue my dreams and not take any compromises.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED:  DATE: 11/04/2024

Table of Contents

List of Figures

> A journey will have pain and failure. It is not only the steps forward that we must accept. It is the stumbles. The trials. The knowledge that we will fail.
>
> But if we stop, if we accept the person we are when we fall, the journey ends. That failure becomes our destination.
>
> To love the journey is to accept no such end. I have found, through painful experience, that the most important step a person can take is always the next one.
>
> *Brandon Sanderson*

Chapter 1

Introduction

Statistical estimation procedures are well defined and tractable for proper probability distributions. For example, one can estimate the density of a dataset assuming it is Normally distributed in a variety of different ways, such as with Maximum Likelihood Estimation (MLE), method of moments estimation or Bayesian inference. However, in the age of modern statistics, researchers and practitioners are encountering more complex problems when a dataset lies on a non-typical domain, such as one that is truncated. When the domain is truncated, many typical assumptions break down and the classical estimation procedures are not suited. Additionally, the structure of large and elaborate datasets may not be adequately captured by classical probability distributions, and one may be interested in non-standard distributions that are more flexible. For example, one may use a neural network, which is a highly flexible model under which the distribution of the dataset can take almost any form.

The issue with estimation on a truncated domain, and with using these complex parameterisations for data distributions is that the data density function is *unnormalisable*. That is, the mathematical formulation of a density function requires it to integrate to 1; a condition that is included by definition of classical probability distributions. In these more complex parameterisations, this condition is impossible to verify without sacrificing flexibility of the model. Instead of losing flexibility, many methods seek to estimate the parameters of this density function *without enforcing this necessary condition*. The field containing these methods is referred to as *unnormalised* modelling.

Unnormalised modelling is useful across a range of domains, especially relevant to applications where the dataset at hand is too complex for classical statistical density estimation methods. Many modern generative machine learning tasks make use of unnormalised modelling for complicated tasks. For example, natural language processing, the study of the statistical properties of language, deals with highly autocorrelated and intricately linked sequential datasets, and as such

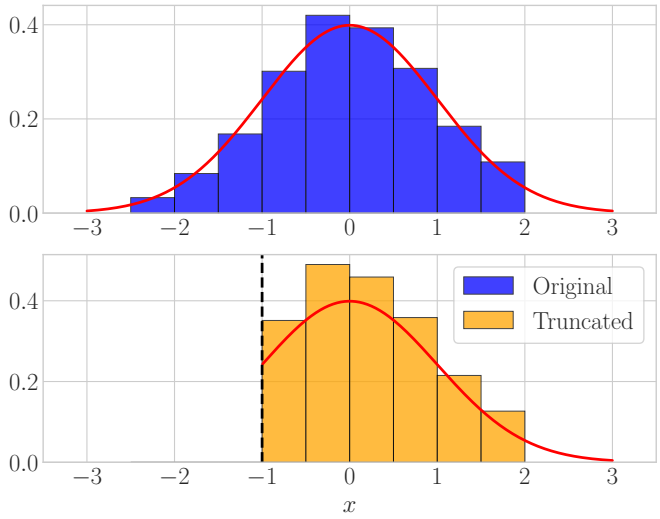


Figure 1.1. Example of a Normal distribution truncated at $x = -1$.

requires extremely flexible frameworks for estimating the densities of the embeddings of words or sentences, for example, Mikolov et al. (2013) or Devlin et al. (2019). Computer vision tasks often involve estimating the conditional density of images, which involves the spatially varying properties of pixel colours, and often relies on a convolutional neural network to capture these spatial variations (O'Shea and Nash, 2015). The most recent state of the art for image generation - diffusion models (Ho et al., 2020; Song and Ermon, 2019) - rely on score matching (Hyvärinen, 2005; Vincent, 2011), a popular framework for unnormalised density estimation. Other examples where unnormalised modelling has found application include hypothesis testing (Chwialkowski et al., 2016; Liu et al., 2016; Wu et al., 2022), Markov random fields for image analysis (Li, 2009), and independent component analysis (Teh et al., 2003)

In this thesis we focus primarily on another example of an unnormalisable model; where the data are observed on a *truncated* domain. Truncated density estimation seeks to estimate truncated distributions, which are probability distributions with a truncation boundary. See figure 1.1 for an example. These boundaries can be intrinsic to the dataset, or occur artificially by data collection methods. For example, image data consists of pixel values that are truncated to the region $[0, 1]$, and this data is truncated by definition. An example of artificially truncated data can be found in geographical applications, where the borders of a city, state or country stop data collection, even though there is no physical process that stops the potential observations of new data beyond these points.

Capturing a truncated probability density is an unnormalisable density estimation problem, as to ensure that the probability density function integrates to one, one needs to integrate with respect to the truncated domain, a task that is often intractable when the truncated domain is non-trivial. Whilst some methods exist for truncated density estimation that account for the normalisation of the density function for specific distributions (Daskalakis et al., 2018; Moreira

and de Uña Álvarez, 2012; Turnbull, 1976), none are applicable to any generic truncated domain nor a generic distribution.

The remainder of this section first gives an introduction and overview of generic density estimation. Then we detail the extension to unnormalised density estimation, and the inherent problems associated with this task. Finally, we give a summary of the remainder of the chapters in this thesis, and then detail our notation.

1.1 Problem Overview

Let the random variable $\boldsymbol{x}$ be distributed according to the *data density* function, $q(\boldsymbol{x})$, a probability density function (PDF) that defines the distribution of $\boldsymbol{x}$. We cannot access $q(\boldsymbol{x})$ and it is considered unknown. In practice, we may only access it through a collection of independent and identically distributed samples, $\{\boldsymbol{x}_i\}_{i=1}^n \sim q(\boldsymbol{x})$.

Estimating the data density has a large amount of interesting applications, and also has a long and important history within classical statistics. Knowing the structure of the density itself can have meaningful intuition on the dataset at hand. In the simplest example, calculating the sample mean of a dataset holds meaning in itself, and this calculation is equivalent to estimating the mean parameter, μ, of a Normal distribution with some variance σ^2, i.e. $\mathcal{N}(\mu, \sigma^2)$, using maximum likelihood estimation.

Density estimation can of course be extended to *conditional* density estimation, a framework that forms the basis for regression, classification, and other generic statistical and machine learning methods which learn the relationship between two or more dependent sets of variables. These topics are extremely broad and applicable to very many domains.

The methods to estimate $q(\boldsymbol{x})$ span the entire field of density estimation and are a subject of a wealth of literature. These methods can vary from completely non-parametric approaches, such as the widely popular kernel density estimation (Parzen, 1962; Rosenblatt, 1956), to specifically parameterising $q(\boldsymbol{x})$ under some known formulation, which will be the primary subject of this thesis. An example of density estimation can be seen in figure 1.2.

To parameterise $q(\boldsymbol{x})$ one can choose, due to some prior knowledge or otherwise, a function, $p(\boldsymbol{x}; \boldsymbol{\theta})$, which we presume can represent $q(\boldsymbol{x})$ under some value of the parameter vector $\boldsymbol{\theta}$. $\boldsymbol{\theta}$ defines the parameterisation of $p(\boldsymbol{x}; \boldsymbol{\theta})$; i.e. the PDF is fully specified through the values of $\boldsymbol{\theta}$. Learning how $p(\boldsymbol{x}; \boldsymbol{\theta})$ best approximates $q(\boldsymbol{x})$ is entirely accomplished through estimating $\boldsymbol{\theta}$ (for example by maximising the likelihood), and this forms the entire basis of parameteric density estimation. The general form of $p(\boldsymbol{x}; \boldsymbol{\theta})$ is given by

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{\bar{p}(\boldsymbol{x}; \boldsymbol{\theta})}{Z(\boldsymbol{\theta})}, \qquad Z(\boldsymbol{\theta}) = \int_V \bar{p}(\boldsymbol{x}; \boldsymbol{\theta}) d\boldsymbol{x}, \tag{1.1}$$

where V is the domain of $\boldsymbol{x}$, i.e. $\boldsymbol{x} \in V$, $\bar{p}(\boldsymbol{x}; \boldsymbol{\theta})$ is known analytically, and $Z(\boldsymbol{\theta})$ is the normalising constant which ensures that the probability density $p(\boldsymbol{x}; \boldsymbol{\theta})$ integrates to one. $p(\boldsymbol{x}; \boldsymbol{\theta})$ can also be referred to as a *statistical model*, or more simply a *model*.

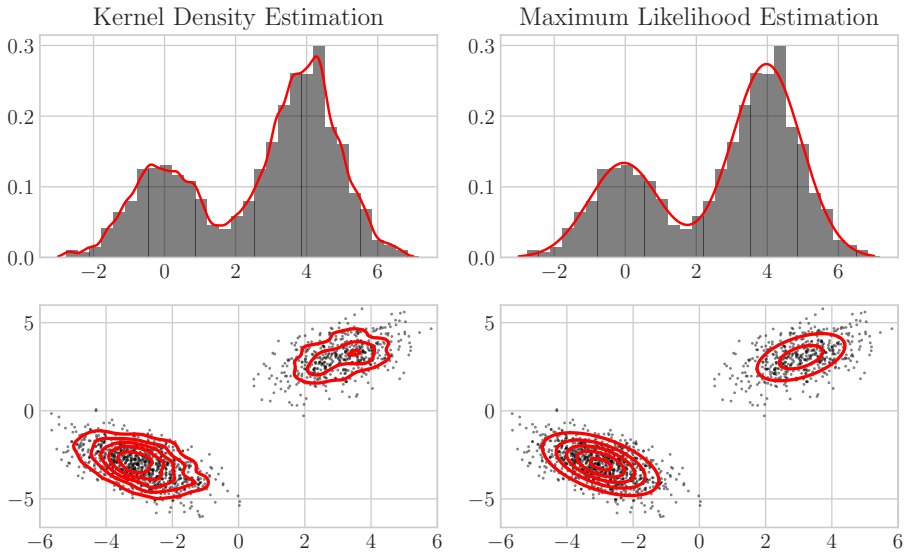


Figure 1.2. Example of density estimation for $d = 1$ (top) and $d = 2$ (bottom) for two methods: Kernel density estimation, which is non-parametric, and maximum likelihood estimation, which is parametric.

In the most common case, $V = \mathbb{R}^d$. However, V can be a truncated domain, e.g. $V \subset \mathbb{R}^d$, in which case things become a little more complicated. Truncated density functions share the same parametric form as their non-truncated counterparts up to the normalising constant $Z(\boldsymbol{\theta})$, since the integration is with respect to the truncated V instead of $\mathbb{R}^d$. In this case, it is likely that the integration is too difficult to analytically derive, and due to the complexity of the truncation domain, extremely hard to approximate also.

When $Z(\boldsymbol{\theta})$ is unknown or intractable, then we call $p(\boldsymbol{x}; \boldsymbol{\theta})$ an *unnormalised model*. The lack of ability to calculate the normalising constant is a problem for classical statistical methods such as maximum likelihood, which involves maximising the log likelihood function, given by

$$\sum_{i=1}^{n} \log p(\boldsymbol{x}_i; \boldsymbol{\theta}) = \sum_{i=1}^{n} \{\log \bar{p}(\boldsymbol{x}_i; \boldsymbol{\theta})\} - n \log Z(\boldsymbol{\theta})$$

which clearly involves the unknown normalising constant and therefore cannot be evaluated.

Instead of applying the traditional and unsuitable density estimation approaches such as maximum likelihood, we can use methods that bypass (or efficiently approximate or estimate) evaluation of the normalising constant. When unrestrained by the normalising constant, the class of models admitted by $p(\boldsymbol{x}; \boldsymbol{\theta})$ is significantly larger in scope. For example, $p(\boldsymbol{x}; \boldsymbol{\theta})$ can be a *truncated* density function. Before, the integration required by $Z(\boldsymbol{\theta})$ would be an integration over a highly complex truncated domain, which would be extremely difficult or intractable, whereas in an *unnormalised* setting there is no such limitation.

1.2 Thesis Overview

Chapter 2: Unnormalised Modelling Review This chapter is a literature review for current unnormalised density estimation methodologies, the most relevant of these being Score Matching and Stein Discrepancies. These methods serve as an important precursor to the original research presented in this thesis. Whilst most of the contributions of this thesis are independent of each other, there is a lot of overlap between methodologies between the works. This chapter is a review of the *shared* literature across all projects, but some more specific background works will be summarised in the project chapters themselves.

Chapter 3: Estimating Density Models with Truncation Boundaries using Score Matching This research studies parameter estimaton for truncated probability densities using an adaptation of score matching using a specific weighting function. The methodology is similar to prior implementations of score matching, but the novelty is its application to *any* truncated domain, provided a proper weighting function can be specified. We propose a weighting function that is a distance function to the boundary, which is easily applicable to many different boundary types. This choice of weight function also naturally arises from maximising the Stein discrepancy. We perform a series of experiments to show the promise of the method in estimating truncated densities. We present an application of the method on the Chicago crime data set in estimating the parameters of a truncated mixed multivariate Normal distribution, and also use it to correct outlier-trimming bias caused by aggressive outlier detection methods.

Chapter 4: Score Matching for Truncated Density Estimation on a Manifold This work studies a novel extension to truncated score matching in chapter 3, where the truncated density estimation is on any Riemannian manifold instead of the Euclidean domain. This extension is not straightforward as the integration by parts used to derive score matching no longer hold, but using a variation of Stokes' theorem enables its construction. We present applications for the von Mises-Fisher and Kent distributions on a two-dimensional sphere in $\mathbb{R}^3$, where instead of the Euclidean distance weighting function, we propose two variations: the Haversine distance on a sphere, and the projected Euclidean distance. We numerically verify the method via comparisons to truncated score matching and maximum likelihood estimation, and also present a real-world application on extreme storm observations in the U.S.

Chapter 5: Approximate Stein Classes for Truncated Density Estimation In this work we present an alternative to truncated density estimation than using score matching with a weighting function. We argue that it may not always be viable or possible to have access to a distance function from any point to the boundary of the truncated domain. Instead, we introduce approximate Stein classes, which is a flexible generalisation of the standard Stein class. From this we construct the Truncated Kernelised Stein Discrepancy (TKSD), which can be evaluated with only a sample of boundary points, and can be minimised to obtain parameters of a truncated density. This method implicitly uses a weighting function that is data-driven and flexible, rather than pre-defined like in prior works. Experiments show that TKSD is on-par with

prior methods even without the explicit functional form of the boundary. In some cases, TKSD is more accurate than previous methods, whilst maintaining a more relaxed set of conditions.

Chapter 6: Time Score Matching for Parameter Change Estimation and Changepoint Detection For the final project of this thesis, we adapt an extension of score matching for time-evolving densities, which estimates the parameter derivatives with respect to time. We call this method Time Score Matching for Derivative Estimation (TSM-DE). We first show that we can model the change rate of a time-evolving density using an *unnormalised* exponential family. Then, we derive the asymptotic distribution of our estimator under a no-change null hypothesis for changepoint detection, and unlike classical changepoint detection methods, TSM-DE uses a flexible model to capture the parameter changes, not needing *a priori* information of the type of changepoint expected. The performance of TSM-DE is comparable to previous works in the field, yet we argue it is a more general and versatile framework for modelling parameter change, and we show that it gives better performance than prior methods on more complex tasks. On real data we are able to detect increases in surface temperatures corresponding to climate change, as well as changes in stock correlations around the period of the global financial crisis. Finally, TSM-DE effectively identifies when out-of-distribution images are introduced to a neural network's output.

Chapter 7: Discussion Finally, we discuss the novelty of the research in the thesis, and the contributions that each of the projects have given to the field. We highlight limitations of the methods presented, as well as directions for future work in the area.

1.3 Notation

Vectors and Matrices A bold letter, e.g. $\boldsymbol{x}$, is used to denote a vector, often a d-dimensional vector, where d refers to the dimensionality, but not exclusively. A bold and upright letter, e.g. $\mathbf{x}$ is used to denote a vector, but is usually n-dimensional, where n refers to the sample size, but not exclusively.

Capital and bold letters, e.g. $\boldsymbol{X}$, are used to denote a matrix, often around the size of $d \times d$, but not limited to that. Usually the bold, upright and capital letter, e.g. $\mathbf{X}$, refers to a matrix that is around the size of $n \times n$, or $n \times d$.

Derivatives The symbol $\nabla_{\boldsymbol{x}}$ defines a derivative across all dimensions with respect to $\boldsymbol{x}$, i.e. for a differentiable function $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$,

$$\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(\boldsymbol{x}) \\ \vdots \\ \frac{\partial}{\partial x_d} f_d(\boldsymbol{x}) \end{bmatrix}.$$

The partial derivative, $\partial/\partial x$, is shortened to ∂_x for brevity. Likewise, the derivative with respect to the j-th dimension of $\boldsymbol{x}$ is denoted by ∂_{x_j}.

Sometimes, for convenience of notation, ∂_x will apply to the immediately succeeding term, i.e. in $\partial_x f(x)g(y)$, the partial derivative is taken with respect to the function f only. Where it is necessary to be clear, brackets will separate an equation such as $(\partial_x f(x))g(x)$ so that the partial derivative with respect to f only is apparent. In cases where the derivative applies to more than one function, the brackets will surround all functions, i.e. $\partial_x[f(x)g(x)]$.

Integration by parts For two differentiable functions, $u(x)$ and $v(x)$, and where $x \in D$ is a scalar, and D is some domain with lower and upper endpoints a and b respectively, the integration by parts formula is given by

$$\int_D u(x) \cdot \nabla v(x) dx = [u(x) \cdot v(x)]_b^a - \int_D v(x) \cdot \nabla u(x) dx.$$

where

$$[u(x) \cdot v(x)]_b^a = \lim_{x \to b} (u(x) \cdot v(x)) - \lim_{x \to a} (u(x) \cdot v(x)).$$

When $\boldsymbol{x} \in D$ and $\boldsymbol{x}$ is d-dimensional, for example $D = \mathbb{R}^d$, let us define $\boldsymbol{u} : D \to D$ as a function with a vector output, and $v : D \to \mathbb{R}$ as a function with a scalar output. We also can write $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x}_{-j}, x_j)$ and $v(\boldsymbol{x}) = v(\boldsymbol{x}_{-j}, x_j)$, where $\boldsymbol{x}_{-j}$ denotes $\boldsymbol{x}$ without the j-th dimension and is fixed. The lower and upper endpoints of for the j-th axis of D in this case are $a(\boldsymbol{x}_{-j})$ and $b(\boldsymbol{x}_{-j})$ respectively. Then integration by parts on the inner product $\langle \boldsymbol{u}(\boldsymbol{x}), \nabla v(\boldsymbol{x}) \rangle$ is given by

$$\int_D \langle \boldsymbol{u}(\boldsymbol{x}), \nabla_{\boldsymbol{x}} v(\boldsymbol{x}) \rangle \, d\boldsymbol{x} = [\langle \boldsymbol{u}(\boldsymbol{x}), v(\boldsymbol{x}) \rangle]|_{a(\boldsymbol{x}_{-j})}^{b(\boldsymbol{x}_{-j})} - \int_D \langle v(\boldsymbol{x}), \nabla_{\boldsymbol{x}} \boldsymbol{u}(\boldsymbol{x}) \rangle \, d\boldsymbol{x},$$

where

$$[\langle \boldsymbol{u}(\boldsymbol{x}), v(\boldsymbol{x}) \rangle]|_{a(\boldsymbol{x}_{-j})}^{b(\boldsymbol{x}_{-j})} = \sum_{j=1}^{d} \int \left[\lim_{x_j \to b(\boldsymbol{x}_{-j})} w(\boldsymbol{x}_{-j}, x_j) - \lim_{x_j \to a(\boldsymbol{x}_{-j})} w(\boldsymbol{x}_{-j}, x_j) \right] d\boldsymbol{x}_{-j},$$

and $w(\boldsymbol{x}_{-j}, x_j) = \langle \boldsymbol{u}(\boldsymbol{x}_{-j}, x_j), v(\boldsymbol{x}_{-j}, x_j) \rangle$.

Expectations The regular expectation with respect to a density function $q(\boldsymbol{x})$, with support V is defined as

$$\mathbb{E}_{\boldsymbol{x} \sim q(\boldsymbol{x})}[\cdot] = \mathbb{E}_q[\cdot] = \int_V q(\boldsymbol{x})(\cdot) d\boldsymbol{x}.$$

Score Function The score function is the first derivative of the log probability density function with respect to the input variable $\boldsymbol{x}$. For a given density function $q(\boldsymbol{x})$, it is defined by

$$\boldsymbol{\psi}_q(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}).$$

We also define the score function for the *model* density function, $p(\boldsymbol{x}; \boldsymbol{\theta})$, as

$$\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; \boldsymbol{\theta}).$$

Norms Given a vector $\boldsymbol{x} \in \mathbb{R}^d$, we denote the ℓ_1 norm, sometimes referred to as the *absolute value* norm, by

$$\|\boldsymbol{x}\|_1 = \sum_{j=1}^{d} |x_j|.$$

We denote the ℓ^2 norm, sometimes referred to as the *Euclidean* norm by

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{j=1}^{d} x_j^2}\,.$$

Sometimes we may refer to the ℓ^1 or the ℓ^2 *ball*, which is the set

$$B_q = \{\boldsymbol{x} \in \mathbb{R}^d \mid \|\boldsymbol{x}\|_q \leq r\},$$

where $q = 1$ corresponds to the ℓ^1 ball and $q = 2$ corresponds to the ℓ^2 ball, each of radius r. When $r = 1$, these are referred to as the *unit* ℓ^1 or ℓ^2 ball.

Chapter 2

Unnormalised Modelling Review

We review the literature relevant to the shared topics amongst all projects presented in this thesis, that is unnormalised modelling. A brief overview of the problem formulation for unnormalised modelling has been given in section 1.1. Here, we detail the current state-of-the-art methodologies for density estimation when the normalising constant is intractable or unavailable.

2.1 Score Matching

Let us briefly recall the aim of density estimation, which is to match two probability density functions $q(\boldsymbol{x})$ and $p(\boldsymbol{x};\boldsymbol{\theta})$ to be as close together as possible. Our aim is to find a $\boldsymbol{\theta}$ to satisfy

$$q(\boldsymbol{x}) = p(\boldsymbol{x};\boldsymbol{\theta})$$

where $\boldsymbol{x} \in \mathbb{R}^d$, and we denote d as the dimensionality of $\boldsymbol{x}$. In the unnormalised setting, methods to do the above fall short of being able to compute the normalising constant part of $p(\boldsymbol{x};\boldsymbol{\theta})$. Consider instead matching the *score functions*, that is to instead match the first derivatives of the logarithms of $q(\boldsymbol{x})$ and $p(\boldsymbol{x};\boldsymbol{\theta})$. It can be shown that

$$\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x};\boldsymbol{\theta}) \iff q(\boldsymbol{x}) = \frac{p(\boldsymbol{x};\boldsymbol{\theta})}{C}$$

for some constant C that is independent of $\boldsymbol{x}$. Therefore, when the score functions are equal, then the densities themselves are also equal up to a constant. Further, consider the score functions for $p(\boldsymbol{x};\boldsymbol{\theta})$,

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x};\boldsymbol{\theta}) = \nabla_{\boldsymbol{x}} \log \bar{p}(\boldsymbol{x};\boldsymbol{\theta}) - \nabla_{\boldsymbol{x}} \log Z(\boldsymbol{\theta}) = \nabla_{\boldsymbol{x}} \log \bar{p}(\boldsymbol{x};\boldsymbol{\theta}).$$

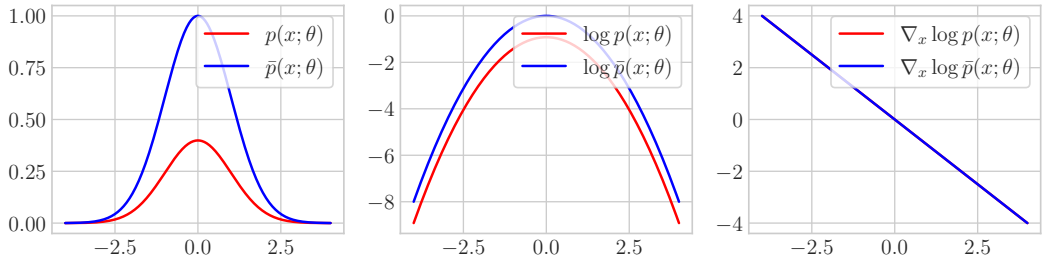


Figure 2.1. Example where $p(\boldsymbol{x};\boldsymbol{\theta})$ is a Gaussian PDF, showing, from left to right, the unnormalised and normalised PDFs, log PDFs and score functions respectively. Note that in the final panel, the normalised score function is invisible because it perfectly aligns, and is the same as, its unnormalised counterpart.

Since $Z(\boldsymbol{\theta})$ is independent of $\boldsymbol{x}$, its derivative is zero, and the score function for $p(\boldsymbol{x};\boldsymbol{\theta})$ does not require computing the normalising constant. Matching the score functions matches the probability density functions, and does not require knowledge of $Z(\boldsymbol{\theta})$ at all, and so is a natural choice of objective to consider for unnormalised density estimation. See figure 2.1 for a visual example of how the score function does not depend on the normalising constant.

This forms the basis of *score matching* (Hyvärinen, 2005), which minimises the expected squared difference between score functions, given by

$$J_{\mathrm{SM}}(\boldsymbol{\theta}) := \mathbb{E}_q\left[\|\boldsymbol{\psi}_q(\boldsymbol{x}) - \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})\|^2\right], \tag{2.1}$$

where $\boldsymbol{\psi}_q(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})$ and $\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x};\boldsymbol{\theta})$ are the data score function and model score function respectively. Similarly, we define $\psi_{q,j}(\boldsymbol{x})$ and $\psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{x})$ as the j-th component of $\boldsymbol{\psi}_q(\boldsymbol{x})$ and $\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})$ respectively.

Intuitively, minimising equation (2.1) with respect to $\boldsymbol{\theta}$ will match the score functions and hence the density functions themselves, but the problem is that it involves $\boldsymbol{\psi}_q(\boldsymbol{x})$, which in turn involves the unknown density function $q(\boldsymbol{x})$ and thus $J_{\mathrm{SM}}(\boldsymbol{\theta})$ is intractable as it currently stands. When $q(\boldsymbol{x})$ has an unbounded support (i.e. $\mathbb{R}^d$), Hyvärinen (2005) showed the following:

$$\begin{aligned} J_{\mathrm{SM}}(\boldsymbol{\theta}) &= \mathbb{E}_q\left[\|\boldsymbol{\psi}_q(\boldsymbol{x}) - \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})\|^2\right] = \int_{\mathbb{R}^d} q(\boldsymbol{x})\|\boldsymbol{\psi}_q(\boldsymbol{x}) - \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})\|^2 d\boldsymbol{x} \\ &= \int_{\mathbb{R}^d} q(\boldsymbol{x})\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})^\top \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) d\boldsymbol{x} - 2\int_{\mathbb{R}^d} q(\boldsymbol{x})\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})^\top \boldsymbol{\psi}_q(\boldsymbol{x}) d\boldsymbol{x} + C_q \\ &= \int_{\mathbb{R}^d} q(\boldsymbol{x})\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})^\top \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) d\boldsymbol{x} - 2\int_{\mathbb{R}^d} \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})^\top (\nabla_{\boldsymbol{x}} q(\boldsymbol{x})) d\boldsymbol{x} + C_q \end{aligned}$$

which follows from $q(\boldsymbol{x})\boldsymbol{\psi}_q(\boldsymbol{x}) = q(\boldsymbol{x})\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}) = q(\boldsymbol{x})(\nabla_{\boldsymbol{x}} q(\boldsymbol{x})/q(\boldsymbol{x})) = \nabla_{\boldsymbol{x}} q(\boldsymbol{x})$, and $C_q = \int_{\mathbb{R}^d} q(\boldsymbol{x})\boldsymbol{\psi}_q(\boldsymbol{x})^\top \boldsymbol{\psi}_q(\boldsymbol{x}) d\boldsymbol{x}$. Continuing by applying integration by parts to the second

term,

$$\begin{aligned} J_{\mathrm{SM}}(\boldsymbol{\theta}) &= \int_{\mathbb{R}^d} q(\boldsymbol{x}) \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})^\top \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) d\boldsymbol{x} - 2 \left[q(\boldsymbol{x}) \mathbf{1}^\top \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) \right] \Big|_{-\infty}^{+\infty} \\ &\qquad + 2 \int_{\mathbb{R}^d} q(\boldsymbol{x}) \mathrm{tr} \left(\nabla_{\boldsymbol{x}} \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) \right) d\boldsymbol{x} + C_q \\ &= \int_{\mathbb{R}^d} q(\boldsymbol{x}) \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})^\top \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) d\boldsymbol{x} + 2 \int_{\mathbb{R}^d} q(\boldsymbol{x}) \mathrm{tr} \left(\nabla_{\boldsymbol{x}} \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) \right) d\boldsymbol{x} + C_q \\ &= \mathbb{E}_q \left[\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})^\top \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) + 2 \mathrm{tr} \left(\nabla_{\boldsymbol{x}} \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) \right) \right] + C_q \end{aligned} \tag{2.2}$$

where the second equality has come from the boundary condition, given by

$$\lim_{\|\boldsymbol{x}\| \to \infty} q(\boldsymbol{x}) = 0. \tag{2.3}$$

This boundary condition is important, but it is an easy assumption to satisfy for an unbounded domain. For example, if $q(\boldsymbol{x})$ is a Gaussian density function, then this holds by definition as the tails of a Gaussian distribution tend to zero at $\pm\infty$. Since C_q is a constant with respect to $\boldsymbol{\theta}$, we can safely ignore it when interest lies in minimising $J_{\mathrm{SM}}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Equation (2.2) no longer contains any terms involving the unknown data density $q(\boldsymbol{x})$, and can be written as a single expectation over $q(\boldsymbol{x})$, and it is therefore a tractable objective.

In the practical setting, where we have observed a set of samples, $\{\boldsymbol{x}_i\}_{i=1}^n$, a sample version of equation (2.3) can be constructed by taking a Monte Carlo approximation to the expectation:

$$\hat{J}_{\mathrm{SM}}(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^n \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_i)^\top \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_i) + 2 \mathrm{tr} \left(\nabla_{\boldsymbol{x}} \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}_i) \right).$$

One can obtain estimates $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ by minimising $\hat{J}_{\mathrm{SM}}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, i.e.

$$\hat{\boldsymbol{\theta}} := \arg\min_{\boldsymbol{\theta}} \hat{J}_{\mathrm{SM}}(\boldsymbol{\theta}).$$

Score matching on an unbounded domain without any extensions is commonly referred to as *classical* score matching. When the support of $q(\boldsymbol{x})$ is non-negative, i.e. $\mathbb{R}_+^d$, Hyvärinen (2007) presented a slightly augmented objective, denoted as *non-negative* score matching, given by

$$J_{\mathrm{NNSM}}(\boldsymbol{\theta}) := \mathbb{E}_q \left[\left\| \boldsymbol{x} \odot \left(\boldsymbol{\psi}_q(\boldsymbol{x}) - \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) \right) \right\|^2 \right], \tag{2.4}$$

where $\odot$ denotes element-wise multiplication. Via a similar rearrangement and application of integration by parts, $J_{\mathrm{NNSM}}(\boldsymbol{\theta})$ can also be written tractably in a way that doesn't involve the unknown density $q(\boldsymbol{x})$, and therefore a sample version can be constructed for use in practice. Whilst the boundary condition in equation (2.3) is no longer satisfied, a similar condition is given by

$$\lim_{\substack{x_j \to 0^+ \\ x_j \to \infty^-}} x_j \, q(\boldsymbol{x}) = 0, \forall j = 1, \ldots, d.$$

The notation $x_j \to 0^+$ denotes the case where x_j approaches zero from above, and $x_j \to \infty^-$ denotes where x_j approaches infinity from below. Since $\boldsymbol{x}$ is non-negative, then as $x_j \to 0^+$, the condition is satisfied by x_j itself, and as $x_j \to \infty^-$, the condition is satisfied by $q(\boldsymbol{x})$, as before.

Yu et al. (2018, 2019) consider an extension of non-negative score matching for graphical models by replacing the weighting term $\boldsymbol{x}$ with a more general function $\boldsymbol{h}(\boldsymbol{x})^{1/2}$ in equation (2.4). This has been further extended by Yu et al. (2021) to consider score matching for a more general class of supports for $q(\boldsymbol{x})$ which is not necessarily limited to only $\mathbb{R}_+^d$. This method is denoted as *generalised* score matching, whose objective is given by

$$J_{\mathrm{GSM}}(\boldsymbol{\theta}) := \mathbb{E}_q \left[\| \boldsymbol{h}(\boldsymbol{x})^{1/2} \odot (\boldsymbol{\psi}_q(\boldsymbol{x}) - \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})) \|^2 \right]. \tag{2.5}$$

It is straightforward to see that when $h_j(\boldsymbol{x}) = x_j^2$ for all $j = 1, \ldots, d$, this recovers non-negative score matching from Hyvärinen (2007). Yu et al. (2021) show that $J_{\mathrm{GSM}}(\boldsymbol{\theta})$ can also be manipulated via integration by parts to form a tractable objective, but requires a different boundary condition of

$$\lim_{\substack{x_j \to a_j(\boldsymbol{x}_{-j})^+ \\ x_j \to b_j(\boldsymbol{x}_{-j})^-}} h_j(\boldsymbol{x}) q(\boldsymbol{x}) = 0, \forall j = 1, \ldots, d,$$

where $\boldsymbol{a}(\boldsymbol{x}_{-j})$ and $\boldsymbol{b}(\boldsymbol{x}_{-j})$ are the lower and upper endpoints of the general domain, respectively.

Many score matching extensions have been developed since its inception, including an application of score matching on a manifold (Mardia et al., 2016), for which we give a full explanation in section 4.2.3. Whilst score matching is a powerful tool for unnormalised density estimation, the trace term in equation (2.2), $\mathrm{tr}\left(\nabla_{\boldsymbol{x}} \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})\right)$, becomes extremely computationally expensive when the dimension d is high due to the necessity to calculate the hessian of $p(\boldsymbol{x}; \boldsymbol{\theta})$. A lot of research has been conducted on variations of score matching to alleviate this computational burden. Vincent (2011) present *denoising* score matching, which estimates the score of a perturbed data distribution according to some noise distribution, which circumvents calculation of the trace term. Meng et al. (2020) propose *composite* score matching, which instead optimises a decomposed score matching objective based on d separate conditional data and model densities for each dimension, which are conditional on the other dimensions. Pang et al. (2020) use finite-differencing to calculate the gradient of the score matching loss function in a computationally efficient way that does not require explicit computation graphs of the trace term. Finally, Song et al. (2020) detail *sliced* score matching, which instead of calculating the trace term, approximates it with random projections, equivalent to applying Hutchinson's trick for hessian approximations (Hutchinson, 1989).

Score matching has seen many applications in recent years due to its flexible approach to unnormalised model estimation, particularly in the context of EBMs (Song and Kingma, 2021), as well as latent variable EBMs (Bao et al., 2020). Most notably, score matching has been a key method in generative modelling (Song and Ermon, 2019) as well as the widely popular diffusion model (Song et al., 2021).

2.2 Stein Discrepancies

Originally developed for goodness-of-fit tests, *Stein discrepancies* are a family of methods which also use the score functions, $\boldsymbol{\psi}_q(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})$ and $\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; \boldsymbol{\theta})$, which can be used for density estimation. Originating from Stein's method (Chen et al., 2011; Stein, 1972), a *Stein class* of functions enables the construction of a family of discrepancy measures (Barp et al., 2019).

Definition 2.1 (Stein identity)**.** Let $q(\boldsymbol{x})$ be any smooth probability density supported on $\mathbb{R}^d$ and let $\mathcal{S}_q : \mathcal{F}^d \to \mathbb{R}$ be a map. $\mathcal{F}^d$ is a Stein class of functions, if for any $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$, $\boldsymbol{f} \in \mathcal{F}^d$,

$$\mathbb{E}_q \left[\mathcal{S}_q \boldsymbol{f}(\boldsymbol{x}) \right] = 0, \tag{2.6}$$

where $\mathcal{S}_q$ is called a Stein operator (Gorham and Mackey, 2015).

We refer to equation (2.6) as the Stein identity, which underpins a lot of existing work in unnormalised modelling. Whilst definition 2.1 holds for any Stein operator $\mathcal{S}_q$, the Langevin Stein operator (Gorham and Mackey, 2015), $\mathcal{T}_q$, is commonly used as it makes use of the score function and hence has applicability for unnormalised densities. This operator on $q(\boldsymbol{x})$ is given by

$$\mathcal{S}_q \boldsymbol{f}(\boldsymbol{x}) = \mathcal{T}_q \boldsymbol{f}(\boldsymbol{x}) := \sum_{j=1}^{d} \psi_{q,j}(\boldsymbol{x}) f_j(\boldsymbol{x}) + \partial_{x_j} f_j(\boldsymbol{x}),$$

where $\psi_{q,j}(\boldsymbol{x}) := \partial_{z_j} \log q(\boldsymbol{x})$. When the Langevin Stein operator is applied on $p(\boldsymbol{x}; \boldsymbol{\theta})$, i.e. $\mathcal{T}_{p_{\boldsymbol{\theta}}} \boldsymbol{f}(\boldsymbol{x})$, it is independent of the normalising constant, $Z(\boldsymbol{\theta})$, and $p(\boldsymbol{x}; \boldsymbol{\theta})$ is involved in the Stein operator only via its 'proxy' of the score function. It is straightforward to see that when the density has an unbounded support (i.e. $\mathbb{R}^d$), the Stein identity (equation (2.6)) holds under $\mathcal{T}_{p_{\boldsymbol{\theta}}}$:

$$\begin{aligned}
\mathbb{E}_q \left[\mathcal{T}_q \boldsymbol{f}(\boldsymbol{x}) \right] &= \int_{\mathbb{R}^d} q(\boldsymbol{x}) \sum_{j=1}^{d} \left\{ \psi_{q,j}(\boldsymbol{x}) f_j(\boldsymbol{x}) + \partial_{x_j} f_j(\boldsymbol{x}) \right\} d\boldsymbol{x} \\
&= \sum_{j=1}^{d} \int_{\mathbb{R}^d} \partial_{x_j} q(\boldsymbol{x}) f_j(\boldsymbol{x}) + q(\boldsymbol{x}) \partial_{x_j} f_j(\boldsymbol{x}) d\boldsymbol{x} \\
&= \sum_{j=1}^{d} \left\{ \left[q(\boldsymbol{x}) f_j(\boldsymbol{x}) \right]_{-\infty}^{\infty} - \int_{\mathbb{R}^d} q(\boldsymbol{x}) \partial_{x_j} f_j(\boldsymbol{x}) d\boldsymbol{x} + \int_{\mathbb{R}^d} q(\boldsymbol{x}) \partial_{x_j} f_j(\boldsymbol{x}) d\boldsymbol{x} \right\} \\
&= 0,
\end{aligned}$$

where the penultimate equality holds due to integration by parts and the final equality holds due to the boundary condition:

$$\lim_{\|\boldsymbol{x}\| \to \infty} q(\boldsymbol{x}) = 0. \tag{2.7}$$

The boundary condition is the exact same condition as for classical score matching, given in equation (2.3). Interestingly, the Stein identity given in definition 2.1 can be used in place of

the integration by parts in deriving the tractable score matching objective in equation (2.2), highlighting the importance of the Stein identity in unnormalised modelling.

The Stein identity is zero when the expectation is with respect to the same density that the Stein operator is applied to, e.g. both operations are with respect to $q(\boldsymbol{x})$. When these densities are not the same, one can measure the magnitude of $\mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}}\boldsymbol{f}(\boldsymbol{x})]$, which in essence measures the extent to which Steins identity is 'violated' between $q(\boldsymbol{x})$ and $p(\boldsymbol{x};\boldsymbol{\theta})$, and therefore the difference between the two densities. This motivates the definition of a *Stein discrepancy* : given by the supremum of the differences between expected Stein operators for the two densities $q(\boldsymbol{x})$ and $p(\boldsymbol{x};\boldsymbol{\theta})$ (Gorham and Mackey, 2015),

$$\begin{aligned} J_{\mathrm{SD}}(\boldsymbol{\theta}) &:= \sup_{\boldsymbol{f}\in\mathcal{F}^d} \left(\mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}}\boldsymbol{f}(\boldsymbol{x})] - \mathbb{E}_{p_{\boldsymbol{\theta}}}[\mathcal{T}_{p_{\boldsymbol{\theta}}}\boldsymbol{f}(\boldsymbol{x})]\right) \\ &= \sup_{\boldsymbol{f}\in\mathcal{F}^d} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}}\boldsymbol{f}(\boldsymbol{x})], \end{aligned} \tag{2.8}$$

where the second line holds due to equation (2.6), as $\mathcal{F}^d$ is a Stein class by definition. $\boldsymbol{f}$ is referred to as the *discriminatory function*, as it discriminates between the two densities, $q(\boldsymbol{x})$ and $p(\boldsymbol{x};\boldsymbol{\theta})$.

The maximisation over $\boldsymbol{f} \in \mathcal{F}^d$ finds the maximum 'violation' of the Stein identity, making the discrepancy as large as possible. The supremum across $\mathcal{F}^d$ is a challenging problem for optimisation (Gorham and Mackey, 2015), however, when we restrict the function class over which the supremum is taken to a Reproducing Kernel Hilbert Space (RKHS), we can derive the Kernelised Stein discrepancy (KSD), for which we follow a similar definition to Chwialkowski et al. (2016) and Liu et al. (2016). For a general overview of the RKHS, we refer to Chapter 4, Steinwart and Christmann (2008).

First, let $\mathcal{G}$ be an RKHS equipped with positive definite kernel k, and let $\mathcal{G}^d$ denote the product RKHS with d elements, where $\boldsymbol{g} = (g_1, \ldots, g_d) \in \mathcal{G}^d$, and is defined with inner product $\langle \boldsymbol{g}, \boldsymbol{g'} \rangle_{\mathcal{G}^d} = \sum_{j=1}^d \langle g_j, g'_j \rangle_{\mathcal{G}}$ and norm $\|\boldsymbol{g}\|_{\mathcal{G}^d} = \sqrt{\sum_{j=1}^d \langle g_j, g_j \rangle_{\mathcal{G}}}$. By the reproducing property, any evaluation of $\boldsymbol{g} \in \mathcal{G}^d$ can be written as

$$\boldsymbol{g}(\boldsymbol{x}) = \langle \boldsymbol{g}, k(\boldsymbol{x}, \cdot)\rangle_{\mathcal{G}^d} = \left[\langle g_1, k(\boldsymbol{x}, \cdot)\rangle_{\mathcal{G}},\ \ldots,\ \langle g_d, k(\boldsymbol{x}, \cdot)\rangle_{\mathcal{G}}\right]^\top . \tag{2.9}$$

Taking the supremum over $\mathcal{G}^d$, and including the restriction of $\boldsymbol{g}$ to the RKHS unit ball, i.e. $\|\boldsymbol{g}\|_{\mathcal{G}^d} \leq 1$, gives rise to the KSD

$$J_{\mathrm{KSD}}(\boldsymbol{\theta}) := \sup_{\substack{\boldsymbol{g}\in\mathcal{G}^d \\ \|\boldsymbol{g}\|_{\mathcal{G}^d}\leq 1}} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}}\boldsymbol{g}(\boldsymbol{x})] = \|\mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}}k(\boldsymbol{x}, \cdot)]\|_{\mathcal{G}^d}. \tag{2.10}$$

Unlike the classical Stein discrepancy, the KSD has a closed form expression which no longer involves the supremum. Moreover, the squared KSD can be expanded to a double expectation

$$J_{\mathrm{KSD}}(\boldsymbol{\theta})^2 = \|\mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}}k(\boldsymbol{x}, \cdot)]\|^2_{\mathcal{G}^d} = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{y}\sim q(\boldsymbol{x})}\left[\sum_{j=1}^d u_j(\boldsymbol{x}, \boldsymbol{y})\right] \tag{2.11}$$

where

$$u_j(\boldsymbol{x}, \boldsymbol{y}) = \psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{x})\psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{y})k(\boldsymbol{x}, \boldsymbol{y}) + \psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{x})\partial_{y_l}k(\boldsymbol{x}, \boldsymbol{y}) + \psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{y})\partial_{x_l}k(\boldsymbol{x}, \boldsymbol{y}) + \partial_{x_l}\partial_{y_l}k(\boldsymbol{x}, \boldsymbol{y}). \tag{2.12}$$

This objective contains no unknown terms, and is fully tractable. A sample version of equation (2.11) can be constructed by taking a Monte Carlo approximation of the expectation:

$$\hat{J}_{\mathrm{KSD}}(\boldsymbol{\theta})^2 := \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{d} u_l(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

$\hat{J}_{\mathrm{KSD}}(\boldsymbol{\theta})^2$, being a tractable formulation which measures discrepancies between densities, is a natural objective function to minimise to estimate $\boldsymbol{\theta}$ for density estimation. Further, Chwialkowski et al. (2016) showed that $J_{\mathrm{KSD}}(\boldsymbol{\theta})^2 = 0$ if and only if $p(\boldsymbol{x}; \boldsymbol{\theta}) = q(\boldsymbol{x})$, making $J_{\mathrm{KSD}}(\boldsymbol{\theta})^2$ a good discrepancy measure between distributions.

Stein discrepancies have seen a lot of development, including extensions to non-Euclidean domains (Shi et al., 2021; Xu and Matsuda, 2021), discrete operators (Yang et al., 2018), stochastic operators (Gorham et al., 2020) and diffusion-based operators (Gorham and Mackey, 2015; Gorham et al., 2020). Motivated by performing goodness-of-fit testing on truncated domains, Xu (2022) propose a modified Langevin Stein operator for any general bounded domain V with boundary ∂V, given by

$$\mathcal{T}_{p_{\boldsymbol{\theta}},h}\boldsymbol{g}(\boldsymbol{x}) = \sum_{j=1}^{d} \psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{x})g_j(\boldsymbol{x})h(\boldsymbol{x}) + \partial_{x_j}\left(g_j(\boldsymbol{x})h(\boldsymbol{x})\right), \tag{2.13}$$

for $\boldsymbol{g} \in \mathcal{G}^d$, where $h(\boldsymbol{x})$ is a weighting function. Using the Stein operator in equation (2.8), and taking the supremum over $\boldsymbol{g} \in \mathcal{G}^d$ gives an alternative definition to KSD for bounded domains, called bounded-domain kernelised Stein discrepancy (bd-KSD):

$$J_{\mathrm{bd-KSD}}(\boldsymbol{\theta}) = \|\mathbb{E}_q\left[\mathcal{T}_{p_{\boldsymbol{\theta}},h}k(\boldsymbol{x}, \cdot)\right]\|^2_{\mathcal{G}^d}.$$

Instead of the condition given in equation (2.7), this Stein divergence relies on the boundary condition

$$\lim_{\boldsymbol{x}\to\boldsymbol{x}'} q(\boldsymbol{x})h(\boldsymbol{x}) = 0$$

for which $\boldsymbol{x} \to \boldsymbol{x}'$ denotes $\boldsymbol{x}$ approaching its corresponding projection to the boundary denoted by $\boldsymbol{x}'$. This condition can be satisfied by choosing h such that $h(\boldsymbol{x}') = 0$ where $\boldsymbol{x}' \in \partial V$. The form of h is not explicitly defined, but the authors recommend a distance function.

2.3 Alternative Methods

So far, we have presented two primary methods for unnormalised density estimation: score matching and Stein discrepancies. Both methods underpin the majority of unnormalised modelling and could perhaps be considered as the most popular methods for this field. But there

exist a wealth more methods and literatures that introduce a diverse range of methodologies for unnormalised modelling, which we will summarise in this section to complete the picture of the field.

Instead of seeking a method which bypasses the evaluation of the normalising constant, Gutmann and Hyvärinen (2010) estimate $Z(\boldsymbol{\theta})$ as an additional parameter to the density parameters, i.e. consider $\boldsymbol{\theta} = [\boldsymbol{\theta}', \alpha]^\top$, where $\alpha = \log Z(\boldsymbol{\theta})$, and $\log p(\boldsymbol{x}; \boldsymbol{\theta}) = \log p(\boldsymbol{x}; \boldsymbol{\theta}') - \alpha$. Then, one can apply traditional parameter estimation methods on the new parameter vector $\boldsymbol{\theta}$. However, applying methods such as maximum likelihood on this augmented probability density function are infeasible, as maximisation on the likelihood would have unbounded growth due to a lack of constraint on α.

Noise-contrastive estimation (NCE) (Gutmann and Hyvärinen, 2010) instead seeks to discriminate between samples from the data distribution $q(\boldsymbol{x})$ (labelled as $y = 1$), and samples from a known noise distribution (labelled as $y = 0$), whose density is denoted as $p_n(\boldsymbol{x})$, which is for example a uniform or Normal distribution. Instead of learning the model density $p(\boldsymbol{x}; \boldsymbol{\theta})$ directly, the probabilities of the classes given the data, i.e.

$$\mathbb{P}(y = 1|\boldsymbol{x}; \boldsymbol{\theta}) = h(\boldsymbol{x}; \boldsymbol{\theta}), \quad \mathbb{P}(y = 0|\boldsymbol{x}; \boldsymbol{\theta}) = 1 - h(\boldsymbol{x}; \boldsymbol{\theta}),$$

can be learned via a classification approach such as logistic regression, where h is the classification function. Knowledge of the prior probabilities (the proportions of samples, $\mathbb{P}(y = 1)$ and $\mathbb{P}(y = 0)$), and the application of Bayes' theorem, enables the estimation of the class conditional probabilities via

$$\mathbb{P}(y = 1|\boldsymbol{x}; \boldsymbol{\theta}) = \frac{\mathbb{P}(\boldsymbol{x}|y = 1; \boldsymbol{\theta})\mathbb{P}(y = 1)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}; \boldsymbol{\theta})}{p(\boldsymbol{x}; \boldsymbol{\theta}) + (n_n/n_d)\, p_n(\boldsymbol{x})},$$

where the key learning target is $\mathbb{P}(\boldsymbol{x}|y = 1; \boldsymbol{\theta}) = p(\boldsymbol{x}; \boldsymbol{\theta})$, n_n is the number of samples from $p_n(\boldsymbol{x})$ and n_d is the number of samples from $q(\boldsymbol{x})$. Regular density estimation via learning the class conditional probabilities is well known, see for example Section 14.2.4 of Hastie et al. (2009), but Gutmann and Hyvärinen (2010) proved that unnormalised models can be estimated with the same principles via the introduction of estimating $Z(\boldsymbol{\theta})$ as an extra parameter, α.

NCE as a method has seen many applications, especially in the field of machine learning. For example, it has been applied to estimating natural image statistics (Gutmann and Hyvärinen, 2012), natural language processing (Mnih and Kavukcuoglu, 2013; Mnih and Teh, 2012), representation learning (Hjelm et al., 2019; Kong et al., 2020), and more. Whilst it is a powerful tool for unnormalised modelling, it relies heavily on choosing a good noise distributon $p_n(\boldsymbol{x})$ which is close to the data distribution (Gutmann and Hirayama, 2011; Song and Kingma, 2021), yet is easy to sample from and analytically tractable (Gorham et al., 2020). A 'bad' choice of a noise distribution leads to a flat loss landscape (Liu et al., 2022), and the learned model will have slow convergence rate and may fail to capture the data density $q(\boldsymbol{x})$ (Jiang et al., 2023).

Score matching and Stein discrepancies bypass the calculation of the normalising constant, noise-contrastive estimation estimates it as a parameter, but there is another family of methods

which *approximate* the normalising constant. In the simplest case, $Z(\boldsymbol{\theta})$ is an integration which is analytically unavailable, but approximation methods such as quadrature or Markov Chain Monte Carlo (MCMC) (Metropolis et al., 1953) could calculate the integral, albeit slowly.

Instead of approximating $Z(\boldsymbol{\theta})$ directly, consider naively maximising the likelihood via gradient updates of the form

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x}; \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log \bar{p}(\boldsymbol{x}; \boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}).$$

The first term, $\nabla_{\boldsymbol{\theta}} \log \bar{p}(\boldsymbol{x}; \boldsymbol{\theta})$, is known, or is straightforward to calculate, whereas the second term is still unknown. However, one can write

$$\nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}; \boldsymbol{\theta})}[\nabla_{\boldsymbol{\theta}} \log \bar{p}(\boldsymbol{x}; \boldsymbol{\theta})],$$

which is an expectation of a known quantity (Song and Kingma, 2021). If we can sample from $p(\boldsymbol{x}; \boldsymbol{\theta})$ then this term can be approximated via a Monte Carlo expectation over a finite number of samples. Obtaining samples from $p(\boldsymbol{x}; \boldsymbol{\theta})$ is not always straightforward. One can perform rejection sampling to sample from $p(\boldsymbol{x}; \boldsymbol{\theta})$, called rejection sampling MLE, but this is laborious and computationally expensive if $p(\boldsymbol{x}; \boldsymbol{\theta})$ is particularly complex.

An alternative approach is to apply MCMC methods to obtain samples from $p(\boldsymbol{x}; \boldsymbol{\theta})$. This is still computationally expensive however, since one would need to run an MCMC chain until convergence to obtain a sample $\boldsymbol{x} \sim p(\boldsymbol{x}; \boldsymbol{\theta})$. Hinton (2002) showed that by initialising the MCMC chain on the datapoint $\boldsymbol{x}$, one typically needs fewer MCMC steps until convergence. In fact, Hinton recommended that sometimes only one step would suffice. This method is known as *contrastive divergence*, which has been used in various applications since its inception (Chen and Murray, 2003; He et al., 2004; Teh et al., 2003), and has also been extended to more efficient variants (Gao et al., 2018; Tieleman, 2008).

Chapter 3

Estimating Density Models with Truncation Boundaries using Score Matching

Note on contribution This chapter is based on the paper entitled Estimating Density Models with Truncation Boundaries using Score Matching (Liu et al., 2022), which appeared in JMLR, authored by Song Liu, Takafumi Kanamori and Daniel J. Williams. My primary contribution was in the development of the algorithm and the experiments.

3.1 Introduction

In many applications of density estimation, we are unable to view a full picture of our dataset. We are instead given access to a smaller subsample of data that is likely artificially truncated by a boundary. As an example, a police department can only monitor crimes within their city's boundary, despite the fact that crimes do not automatically stop at the edge of the city's official borders. Other examples of truncation boundaries include limited number of medical tests, which can result in under-reported disease counts, or a country's borders preventing discoveries of habitat locations.

Truncation boundaries are often highly complex, for example, the boundary of the city of Chicago is a complex polygon (see Figure 3.1) which cannot be easily approximated by a bounding box or circle. Further, truncation introduces difficulties in statistical parameter estimation, which assumes access to the full dataset. If one applies these naive methods to a truncated dataset, the estimates will be bias. Whilst there exists methods for specific tasks, such as doubly-truncated data (Chen and Hanson, 2014; Moreira and de Uña Álvarez, 2012), truncated multivariate Normal data (Daskalakis et al., 2018) or truncated regression (Daskalakis et al., 2019), there exists no prior work which can perform density estimation on any generic truncated domain.

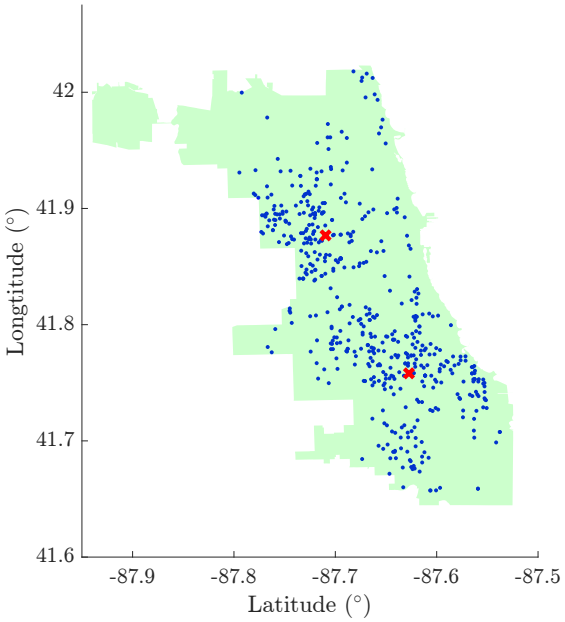


Figure 3.1. City of Chicago, where blue dots are locations of homicides in 2008, and the red crosses are estimates from a naive MLE implementation to estimate the two means of a mixed Gaussian distribution.

In this chapter, we propose a novel estimator for truncated density models using an adaptation of score matching Hyvärinen (2005); a popular framework for unnormalised density estimation. This estimator is a natural extension of previous works in the field, namely Hyvärinen (2007); Yu et al. (2018, 2019). We propose a weighting function that is the shortest distance from a data point to the truncation boundary. We show that this choice naturally arises from minimising a Stein Discrepancy.

3.2 Methodology

3.2.1 Generalised Score Matching on Truncated Domains

Let us begin by re-stating the generalised score matching objective for the non-negative domain from Yu et al. (2019), given by

$$J_{\mathrm{GSM}}(\boldsymbol{\theta}) := \mathbb{E}_q \left[\| \boldsymbol{h}(\boldsymbol{x})^{1/2} \odot (\boldsymbol{\psi}_q(\boldsymbol{x}) - \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})) \|^2 \right]. \tag{3.1}$$

Generalised score matching was originally proposed for the non-negative domain $\mathbb{R}^d_+$. For any generic truncated domain V (i.e. not necessarily $\mathbb{R}^d_+$), then the results derived by Yu et al. (2019) do not hold. We intend to extend this method to for any generic truncated domain V. Firstly, the following Lemma shows the validity of equation (3.1) for the domain V:

Lemma 3.1. *Suppose that V is a connected open subspace in $\mathbb{R}^d$, such that $V \subset \mathbb{R}^d$, $h_j(\boldsymbol{x}) > 0,\ \forall j = 1, \ldots, d$, and $q(\boldsymbol{x}) > 0,\ \forall \boldsymbol{x} \in V$. Then $J_{\text{GSM}}(\boldsymbol{\theta}) = 0$ if and only if $q(\boldsymbol{x}) = p(\boldsymbol{x}; \boldsymbol{\theta})$.*

The proof can be found in Lemma 2, Liu et al. (2022). The next theorem shows that the generalised score matching objective has a unique minimiser which is the true parameter.

Theorem 3.2. *Assume the same conditions as Lemma 3.1. Let $\mathcal{P} = \{p(\boldsymbol{x}; \boldsymbol{\theta}) | \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ be a family of parametric model density functions, from which $q(\boldsymbol{x}) = p(\boldsymbol{x}; \boldsymbol{\theta}^\star)$ for some true parameter $\boldsymbol{\theta}^\star$. Assume also that $p(\boldsymbol{x}; \boldsymbol{\theta})$ is identifiable, such that $p(\boldsymbol{x}; \boldsymbol{\theta}) \neq p(\boldsymbol{x}; \boldsymbol{\theta}')$ for $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$. Then $\arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} J_{\text{GSM}}(\boldsymbol{\theta})$ is unique and is $\boldsymbol{\theta}^\star$.*

The proof can be found in Theorem 1, Liu et al. (2022). This theorem states that minimising $J_{\text{GSM}}(\boldsymbol{\theta})$ will give us the true parameter $\boldsymbol{\theta}^\star$, provided the parametric family $\mathcal{P}$ contains the true data density.

3.2.2 Truncated Score Matching Objective

Hyvärinen (2007) and Yu et al. (2019) showed that the generalised score matching objective, $J_{\text{GSM}}(\boldsymbol{\theta})$, on the non-negative domain $\mathbb{R}^d_+$, could be written as

$$J_{\text{GSM}}(\boldsymbol{\theta}) = \sum_{j=1}^{d} \mathbb{E}_q \left[h_j(\boldsymbol{x}) \psi_{p_{\boldsymbol{\theta}}, j}(\boldsymbol{x})^2 + 2\partial_{x_j} (h_j(\boldsymbol{x}) \psi_{p_{\boldsymbol{\theta}}, j}(\boldsymbol{x})) \right] + C_q,$$

which follows under the mild boundary condition of

$$\lim_{\substack{x_j \to 0^+ \\ x_j \to \infty^-}} h_j(\boldsymbol{x}) q(\boldsymbol{x}) = 0,\ \forall j = 1, \ldots, d, \tag{3.2}$$

for which design of $\boldsymbol{h}(\boldsymbol{x})$ can satisfy this criteria. For example, Hyvärinen (2007) implicitly used $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{x}$, which naturally satisfies the condition. Let us now let the support of $q(\boldsymbol{x})$ be V instead of $\mathbb{R}^d_+$, then we can consider the following objective,

$$J_{\text{TSM}}(\boldsymbol{\theta}) := \mathbb{E}_q \left[\| \boldsymbol{g}(\boldsymbol{x})^{1/2} \odot (\boldsymbol{\psi}_q(\boldsymbol{x}) - \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})) \|^2 \right], \tag{3.3}$$

which is the same as $J_{\text{GSM}}(\boldsymbol{\theta})$ up to the support of $q(\boldsymbol{x})$.

Clearly, the weighting function $\boldsymbol{g}(\boldsymbol{x})$ inside of $J_{\text{TSM}}(\boldsymbol{\theta})$ will need to be different to its counterpart, $\boldsymbol{h}(\boldsymbol{x})$, inside of $J_{\text{GSM}}(\boldsymbol{\theta})$. More specifically, when considering a truncated domain V, one would need a different boundary condition to equation (3.2), as it would need to be specified across all points in the boundary set of V. Let ∂V denote the boundary of V, then one would instead require the following boundary condition,

$$\lim_{\boldsymbol{x} \to \boldsymbol{x}'} g_j(\boldsymbol{x}) q(\boldsymbol{x}) = 0,\ \forall j = 1, \ldots, d, \tag{3.4}$$

where $\boldsymbol{x}' \in V$ is denoted as a boundary point, and $\boldsymbol{x} \to \boldsymbol{x}'$ takes any point sequence converging to $\boldsymbol{x}'$ into account. This can be used in part to derive the following theorem.

Theorem 3.3 (Truncated Score Matching)**.** *Let equation* (3.4) *hold for some* $\boldsymbol{g}(\boldsymbol{x})$ *that is weakly differentiable. Further assume that* V *is a Lipschitz domain. Then,*

$$J_{\mathrm{TSM}}(\boldsymbol{\theta}) := \sum_{j=1}^{d} \mathbb{E}_q \left[g_j(\boldsymbol{x}) \psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{x})^2 + 2 g_j(\boldsymbol{x}) \partial_{x_j} \psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{x}) + 2 \partial_{x_j} g_j(\boldsymbol{x}) \psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{x}) \right]. \tag{3.5}$$

This theorem provides an objective function under a flexible formulation which only requires a *weakly* differentiable weighting function $\boldsymbol{g}(\boldsymbol{x})$. The proof can be found in Liu et al. (2022), which relies on Extended Greens Theorem (Proposition 7.6.1, Atkinson and Han (2005)), which is an analogue of integration by parts for weakly differentiable functions on Lipschitz domains. The theorem also relies on a number of assumptions, and therefore a carefully considered $\boldsymbol{h}(\boldsymbol{x})$, which will be addressed in the next section.

3.2.3 Choosing a Weighting Function

The weighting function $\boldsymbol{g}(\boldsymbol{x})$ must be chosen so that (i) it is weakly differentiable and (ii) equation (3.4) is satisfied. Further, it would be preferable to have intuitive reasoning for $\boldsymbol{g}(\boldsymbol{x})$ in the context of truncated density estimation. In this section, we consider a criterion for selecting $\boldsymbol{g}(\boldsymbol{x})$ based on maximising a Stein discrepancy. For simplicity, we assume that $q(\boldsymbol{x})$ and $\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})$ are both smooth in this section.

Since the task is density estimation, we can consider a density estimator that minimises a Stein discrepancy (as defined in equation (2.8)), i.e.

$$\sup_{\boldsymbol{f} \in \mathcal{F}^d} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}} \boldsymbol{f}(\boldsymbol{x})],$$

where $\mathcal{T}_{p_{\boldsymbol{\theta}}} \boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{d} \psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{x}) f_j(\boldsymbol{x}) + \partial_{x_j} f_j(\boldsymbol{x})$ is the Langevin-Stein operator (Gorham and Mackey, 2015). This minimum Stein discrepancy estimator is closely related to score matching (Barp et al., 2019).

Consider a function family given by

$$\mathcal{F}_0^d = \{ \boldsymbol{f} = \boldsymbol{h} \odot \boldsymbol{g}^{1/2} \mid \mathbb{E}_q[\|\boldsymbol{h}(\boldsymbol{x})\|^2] \leq 1 \},$$

where $\boldsymbol{h} : V \to \mathbb{R}^d$ is a smooth function.

Theorem 3.4. *Let* $\mathrm{Lip}_0^L(V)$ *be the set of all functions such that for* $g \in \mathrm{Lip}_0^L(V)$, $g : V \to \mathbb{R}^d$ *that are* L*-Lipschitz continuous with respect to* $\mathrm{dist}(\cdot, \cdot)$*, and for which* $g(\boldsymbol{x}') = 0 \ \forall \boldsymbol{x}' \in \partial V$*. Let* $\boldsymbol{f} \in \mathcal{F}_0^d$*, and* $g_j \in \mathrm{Lip}_0^L(V) \ \forall j = 1, \ldots, d$*. Then* $\mathcal{F}_0^d$ *is a Stein class of* $q(\boldsymbol{x})$ *defined on* V*, and*

$$\sup_{\boldsymbol{f} \in \mathcal{F}_0^d} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}} \boldsymbol{f}(\boldsymbol{x})] = \sqrt{L \cdot J_{\mathrm{TSM}_0}(\boldsymbol{\theta})},$$

(a) (b) (c) (d)

Figure 3.2. Some examples of $g_0(\boldsymbol{x})$ for (a) a circular boundary, (b) a square boundary, (c) a triangular boundary and (d) an irregular polygon. Here, $\text{dist}(\cdot, \cdot)$ is the Euclidean, or ℓ^2, distance function.

where $J_{\text{TSM}_0}(\boldsymbol{\theta})$ *denotes* $J_{\text{TSM}}(\boldsymbol{\theta})$ *with* $\boldsymbol{g} = [g_0, \ldots, g_0]^\top$, *and* g_0 *is given by*

$$g_0(\boldsymbol{x}) = \min_{\boldsymbol{x}' \in \partial V} \text{dist}(\boldsymbol{x}, \boldsymbol{x}'), \tag{3.6}$$

and $\text{dist}(\cdot, \cdot)$ *is a distance function.*

The proof of this theorem can be found in Theorem 3, Liu et al. (2022), as it partly based on the proof of Theorem 2, Barp et al. (2019). Some examples of $\text{dist}(\cdot, \cdot)$ include the ℓ^2 distance function, $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2$, or the ℓ^1 distance function, $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_1$. Some visual examples of the ℓ^2 distance function for various domains is shown in figure 3.2.

This theorem shows that when maximising the Stein discrepancy over a set of L-Lipschitz continuous functions, provided that the functions satisfy the boundary condition as given in equation (3.4), then the maximum is proportional to the truncated score matching objective with a specfic weighting function g_0. Therefore, $J_{\text{TSM}_0}(\boldsymbol{\theta})$ is a maximum Stein discrepancy.

We denote $J_{\text{TSM}_0}(\boldsymbol{\theta})$ as the *TruncSM* objective function, where *TruncSM* is the name of the method that uses the truncated score matching estimator, equation (3.3), with weighting function $\boldsymbol{g} = g_0$, the distance function.

Corollary 3.5. *Let* $\overline{\text{Lip}}_0^L(V) := \{g \in \text{Lip}_0^L(V) | g(\boldsymbol{x}) \leq 1\}$ *be the set of bounded* L*-Lipschitz continuous functions that satisfy the conditions from theorem 3.4. Let* $\boldsymbol{f} \in \mathcal{F}_0^d$, *and* $g_j \in \overline{\text{Lip}}_0^L(V)\ \forall j = 1, \ldots, d$. *Then*

$$\sup_{\boldsymbol{f} \in \mathcal{F}_0^d} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}} \boldsymbol{f}(\boldsymbol{x})] = \sqrt{J_{\text{TSM}_{\text{L}}}(\boldsymbol{\theta})},$$

where $J_{\text{TSM}_{\text{L}}}(\boldsymbol{\theta})$ *denotes* $J_{\text{TSM}}(\boldsymbol{\theta})$ *with* $\boldsymbol{g} = [\bar{g}_L, \ldots, \bar{g}_L]^\top$, *and* $\bar{g}_L$ *is given by*

$$\bar{g}_L(\boldsymbol{x}) = \min(1, L \cdot g_0(\boldsymbol{x})). \tag{3.7}$$

This thoerem shows that $J_{\text{TSM}_{\text{L}}}(\boldsymbol{\theta})$ is also a maximum Stein discrepancy, for which the proof can be found in Corollary 1, Liu et al. (2022). This *capped* weighting function is an alternative

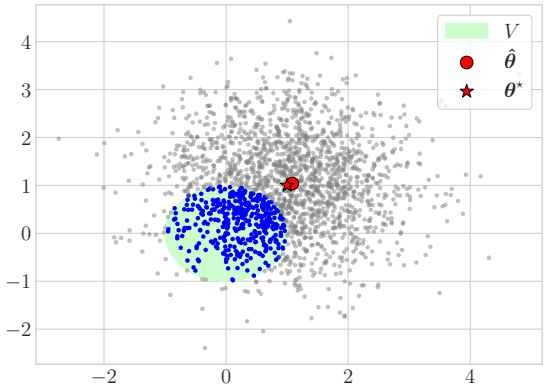


Figure 3.3. Estimating the mean of a Multivariate Normal distribution.

to g_0 based on a slight variation in the function family that is optimised over. A similar capped weighting function given by $g_j = \min(1, x_j)$ is discussed in Yu et al. (2018, 2019) for when $V = \mathbb{R}^d_+$, and this exact capped function is later discussed in the context of generalised score matching for an unbounded domain V by Yu et al. (2021).

Whilst theorem 3.4 and corollary 3.5 show that g_0 and g_L are natural candidates for weighting functions via $J_{\mathrm{TSM}_0}(\boldsymbol{\theta})$ and $J_{\mathrm{TSM}_\mathrm{L}}(\boldsymbol{\theta})$ being maximum Stein discrepancies, respectively, this does not guarantee efficiency in statistical inference tasks. In section 3.3, we investigate their performance empirically.

3.3 Synthetic Experiments

3.3.1 Illustrative Examples

Multivariate Normal Mean The first example we consider is estimation for a two-dimensional multivariate Normal distribution with fixed identity covariance matrix $\boldsymbol{I}_2$. In this case, the unnormalised density function is given by

$$\bar{p}(\boldsymbol{x}; \boldsymbol{\theta}) = \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\theta})^\top (\boldsymbol{x} - \boldsymbol{\theta})\right)$$

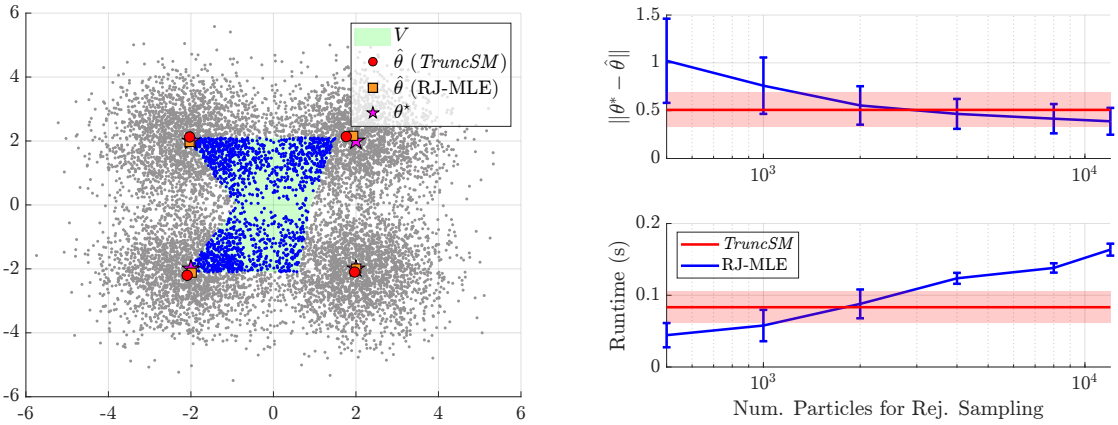
where $\boldsymbol{\theta}$ is the mean, and hence

$$\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) = -(\boldsymbol{x} - \boldsymbol{\theta}).$$

The experiment is set up as follows. First, a set of $n_0 = 2000$ samples is sampled via

$$\{\boldsymbol{x}_i\}_{i=1}^{n_0} \sim \mathcal{N}(\boldsymbol{\theta}^\star, \boldsymbol{I}_2),$$

where $\boldsymbol{\theta}^\star$ is the true mean to be estimated. We denote by n_0 the number of untruncated samples (observed and unobserved). These samples are then truncated to the unit ℓ^2 ball around the



(a) Example of estimating the mixed multivariate Normal data.

(b) The number of particles for RJ-MLE against the estimation error and runtime for the two methods.

Figure 3.4. Estimating the means of a mixed Multivariate Normal distribution using *TruncSM* and RJ-MLE.

origin, so only a subset of the samples are observed ($n = 393$) in a corner of the dataset. We employ *TruncSM* with the Euclidean distance function to estimate the mean, considering the covariance matrix known, for which the estimated mean, $\hat{\boldsymbol{\theta}}$ is close to the true mean, $\boldsymbol{\theta}^\star$. The experiment setup and result is shown in figure 3.3.

Mixed Multivariate Normal Means A more difficult example is given by estimating multiple means of a mixed multivariate Normal distribution with four nodes and identity covariance matrix, $\boldsymbol{I}_2$, for which the unnormalised density function is given by

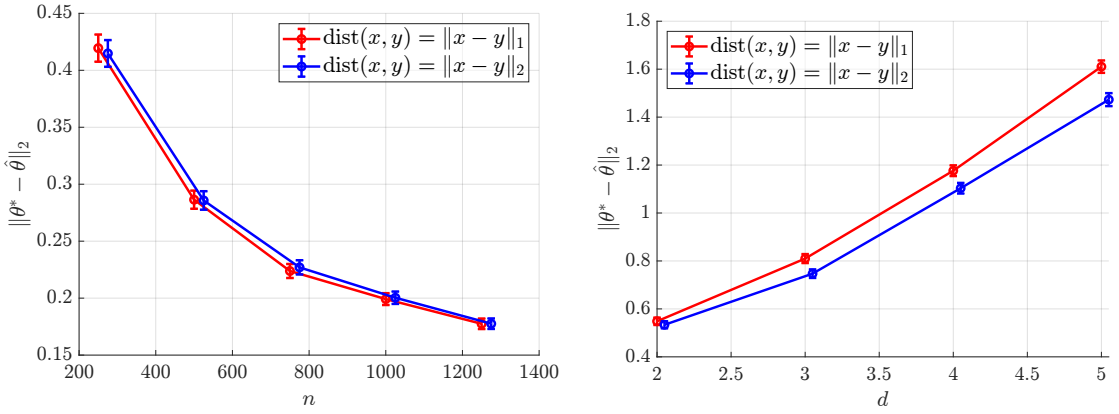
$$\bar{p}(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{k=1}^{4} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\theta}_k)^\top (\boldsymbol{x} - \boldsymbol{\theta}_k)\right)$$

for four distinct means at

$$\boldsymbol{\theta}_1^* = [2, 2], \;\; \boldsymbol{\theta}_2^* = [-2, 2], \;\; \boldsymbol{\theta}_3^* = [2 - 2], \;\; \boldsymbol{\theta}_4^* = [-2, -2].$$

The task in this case is to estimate all four means considering the covariance known. $n_0 = 10,000$ untruncated samples are simulated from the distribution, but only $n = 1417$ truncated samples are observed, where the truncation domain is a oddly shaped polygon positioned centrally in the dataset.

Both *TruncSM* and rejection sampling maximum likelihood estimation (RJ-MLE) are employed for estimation. RJ-MLE is a likelihood maximisation method which uses particles to approximate the normalising constant, $Z(\boldsymbol{\theta})$. The results from the experiment can be seen in figure 3.4. Figure 3.4(a) shows a visual example of the experiment, where RJ-MLE uses 500,000 particles, whereas figure 3.4(b) shows the estimation error and runtime of the methods, plotted against



(a) Sample size benchmark.

(b) Dimension benchmark.

Figure 3.5. Mean estimation error, with standard error bars, against sample size and dimension, independently, for *TruncSM*, where g_0 is either the ℓ^2 or ℓ^1 distance function.

the number of particles used in RJ-MLE. Both methods are given initial parameter estimates of $\boldsymbol{\theta}^\star + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$ is a small perturbation. *TruncSM* gives comparable results to RJ-MLE, and both methods estimate means that are close to the true values. Further, the runtime of *TruncSM* is considerably lower than RJ-MLE, which requires exponentially growing particles and hence computation time to achieve comparable results to *TruncSM*.

3.3.2 Choice of Distance Function: ℓ^1 vs ℓ^2

In section 3.2.3, we show that an optimal weighting function for truncated score matching is given by g_0, which is defined for some arbitrary distance function, $\text{dist}(\cdot, \cdot)$. In this section we explore two choices for dist, the ℓ^2 and ℓ^1 distance function, and compare their empirical performance across two experiments.

Sample Size Similarly to the first example in section 3.3.1, we simulate a set of untruncated samples from a d-dimensional multivariate Normal distribution via

$$\boldsymbol{x}_i \sim \mathcal{N}\left(\boldsymbol{\theta}^\star, \boldsymbol{I}_d\right),$$

where $\boldsymbol{\theta}^\star = \frac{1}{2} \cdot \mathbf{1}_d$ is the true mean to be estimated. In this case, we truncate samples to the upper half of the ℓ^1 ball, centred on the origin, i.e. we consider the domain given by

$$V = \{\boldsymbol{x} : \|\boldsymbol{x}\|_1 < 1, x_d > 0\}.$$

Any points for which the conditions in V are not satisfied are discarded, and we continue sampling $\boldsymbol{x}_i$ then potentially discarding until we have a set of truncated samples of size n. Then we estimate the mean with *TruncSM* using an ℓ^1 and ℓ^2 distance function for g_0, separately. Across 400 trials and for $d = 2$, the mean and standard error of the estimation error, $\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\|_2$, is reported for different values of n in figure 3.5(a).

There is a neglegible difference between the ℓ^1 and ℓ^2 distance function across all values of n. Additionally, from this experiment we can see that the mean estimation error decreases as expected with sample size n, which shows that the estimator is empirically consistent.

Dimension We also consider increasing the dimension of the data, d, and how it affects estimation error. The experiment setup is identical to the experiment above for sample size, except d varies and $n = 150$ is fixed. The results are shown in figure 3.5(b). Naturally the error increases with d as the problem becomes more difficult, but the increase is only linear, which is to be expected. The distance functions provide comparable estimation errors across all d, but for higher values of d, the ℓ^2 distance function has a marginally smaller error, implying it has a stronger performance for high dimensions.

Aside from a small difference in error at high dimension, from these experiments we can conclude that the choice of distance function is largely unimportant.

3.3.3 Capped Weighting Function

We investigate the performance of the capped weighting function,

$$\bar{g}_L(\boldsymbol{x}) = \min(1, L \cdot g_0(\boldsymbol{x})),$$

which stems from corollary 3.5. Firstly, we note that when L is small, then $\bar{g}_L$ will never be 'capped', where being capped refers to when $L \cdot g_0(\boldsymbol{x}) > 1$ for a given $\boldsymbol{x}$). In this case, $\bar{g}_L$ is equivalent to g_0 up to a multiplicative constant. We experiment the effect of capping by measuring the estimation error for three different values of L, $L = 0.1$, $L = 1$ and $L = 10$. Additionally, we vary the size of the domain V through a scaling factor b which controls the 'size' of the truncated region V, for which higher values of b result in a smaller proportion of data being truncated. We also record the number of data points which are capped. Similarly to earlier examples, n_0 samples are generated from a two-dimensional multivariate Normal distribution,

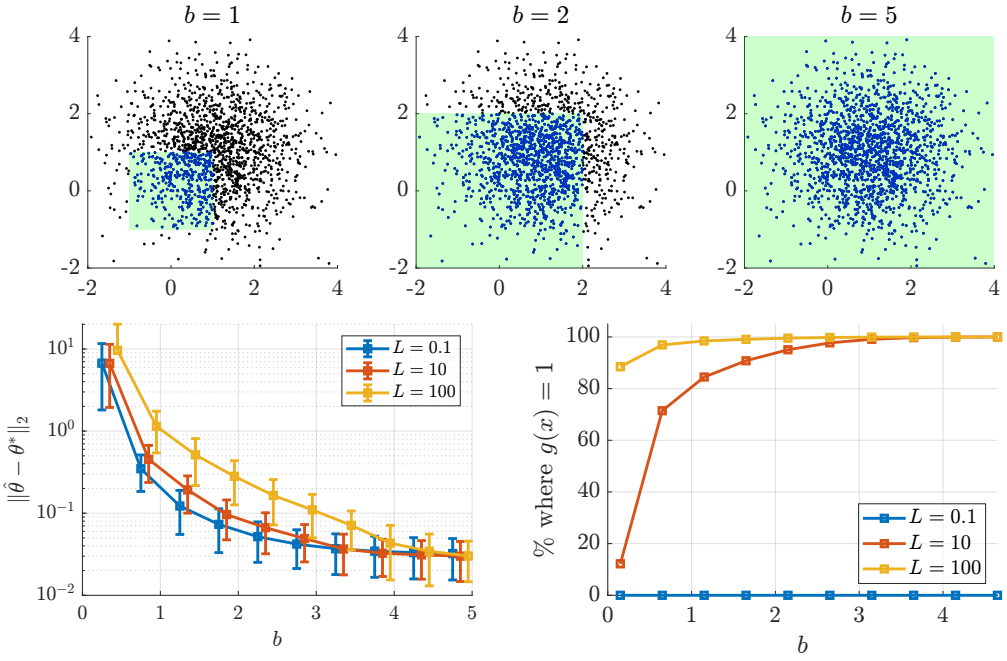
$$\{\boldsymbol{x}_i\}_{i=1}^{n_0} \sim \mathcal{N}\left(\boldsymbol{\theta}^\star, \boldsymbol{I}_2\right),$$

with true mean $\boldsymbol{\theta}^\star = \mathbf{1}_2$, and known covariance. The samples that are not within a given domain V are discarded and only those within the domain are considered as observed. We test for two distinct domains: a square boundary and a disjoint boundary, which are created by supplying set(s) of vertices Ω, detailing the locations of the polygonal truncation domain. The variable b controls the size of the domain by offsetting the supplied vertices in Ω. For the square, this is given by

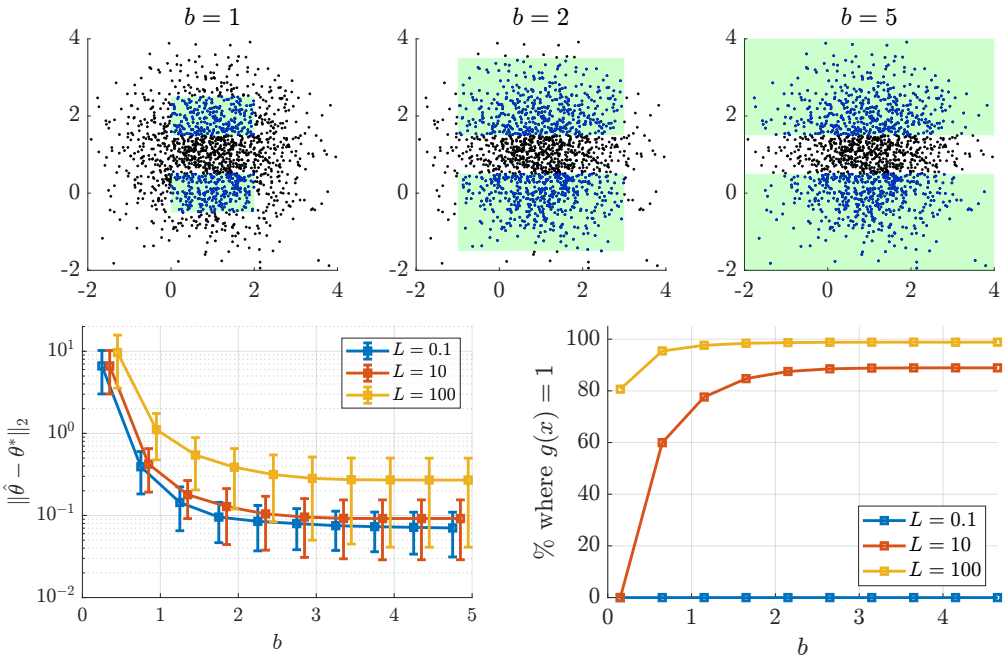
$$\Omega_{\text{sq}} := \{(-b, -b), (-b, b), (b, b), (b, -b)\}.$$

Increasing b in this scenario leads to the corners of the square boundary being shifted by an equal amount. For the disjoint boundary, two sets of vertices are given for two disjoint domains:

$$\begin{aligned}\Omega_{\text{dis1}} :=& \{(1-b, 0.5-b), (1-b, 0.5), (1+b, 0.5), (1+b, 0.5-b)\},\\ \Omega_{\text{dis2}} :=& \{(1-b, 1.5), (1-b, 1.5+b), (1+b, 1.5+b), (1+b, 1.5)\}.\end{aligned}$$



(a) Square domain.



(b) Disjoint domain.

Figure 3.6. Examples of truncated domains, estimation errors, and percentage of data points where $\bar{g}_L$ is 'capped', for two distinct domain types: (a) a square domain; and (b) a disjoint domain.

Increasing b in this case will enlarge the truncation domain while the 'disjoint' section in the middle remains unchanged. Examples of the truncated domain V under some values of b are given in the top rows of figures 3.6(a) and 3.6(b).

Figure 3.6 shows the results from this experiment. The results are in line with what is expected; when the size of the region V increases, and a more 'complete picture' of the dataset is given to the method, the estimation error decreases and *TruncSM* is more accurate. For almost all values of b, the square boundary provides a lower error than the disjoint boundary, due to the disjoint boundary always having a region where data are not observed.

The value of $L = 0.1$ seems to have the strongest empirical performance across all values of b, and in this case, $\bar{g}_L$ has no points that are capped. Therefore we can conclude that empirically, g_0 is the best performing weighting function. When the domain size increases to larger values, the square domain in figure 3.6(a) effectively becomes untruncated, and larger values of L reduces to classical score matching, as all evaluations of $\bar{g}_L$ are capped at 1. For the disjoint domain in figure 3.6(b), the domain is never fully untruncated at higher values of b, and smaller values of L remain the most performant throughout.

3.4 Applications

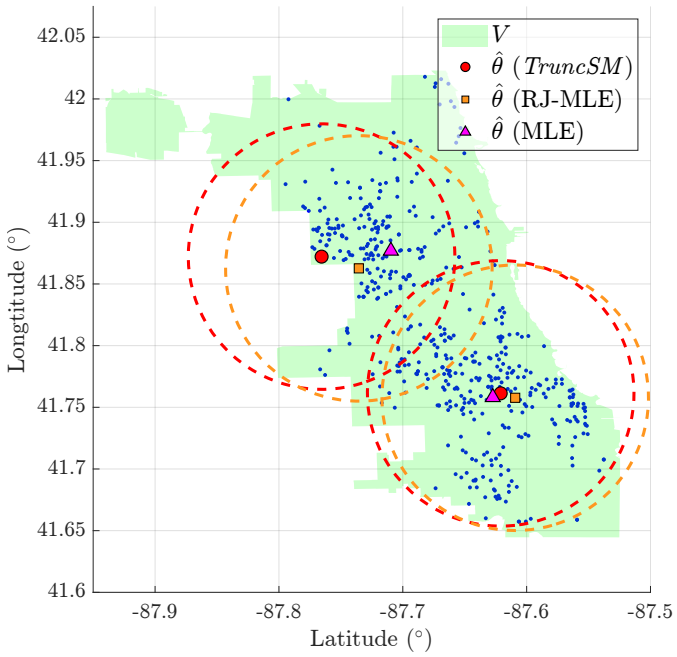
3.4.1 Chicago Crime Example

Crimes that happen within the city of Chicago are only recorded within the artificial confines of the city's boundary, even though they do not stop happening outside its borders. We can consider the region of Chicago as the truncated domain V, and can treat the homicide locations (longitude and latitude) reported by the police in 2008 as points from a mixed Multivariate Normal distribution. We assume a fixed standard deviation that roughly covers two 'widths' of the city, such that $\sigma = 0.06$. The dataset is obtained from Chicago Police Department (2019).

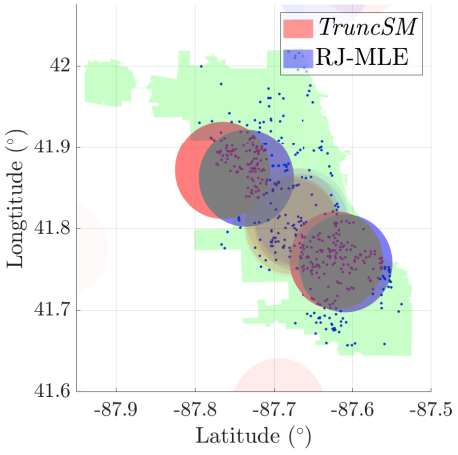
The task is to estimate two centres of crime locations assuming that the mixed distribution has two nodes, using the city boundary as the truncation boundary ∂V. We initialise parameter estimates at the sample mean of the dataset plus some perturbation, i.e. $\bar{\boldsymbol{x}} + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. This region is a complex polygon that cannot be written in an analytical form, and thus an approximation to the ℓ^2 distance function must be used for g_0. This approximation is based on minimising a sample version of equation (3.6), where instead of minimising over the full set ∂V, we minimise over a set of coordinates instead. The evaluations of g_0 are shown visually in figure 3.7(c).

Figure 3.7(a) shows single estimates given by *TruncSM*, RJ-MLE and regular (vanilla) MLE, as well as the 95% confidence region as a dashed line. All methods chose two means at the north and south of the city. Whilst MLE picked the northern mean close to the centre of the city, the two truncated methods, *TruncSM* and RJ-MLE, picked the northern mean as further west. The southern means were all consistently placed in the centre by all methods.

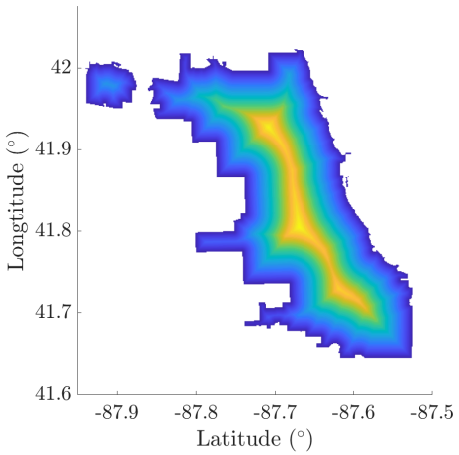
Figure 3.7(b) shows a collection of estimates for different initialisations across 500 trials for the



(a) Estimates of the mean and 95% confidence intervals from *TruncSM* and RJ-MLE, and vanilla MLE estimate of the mean.

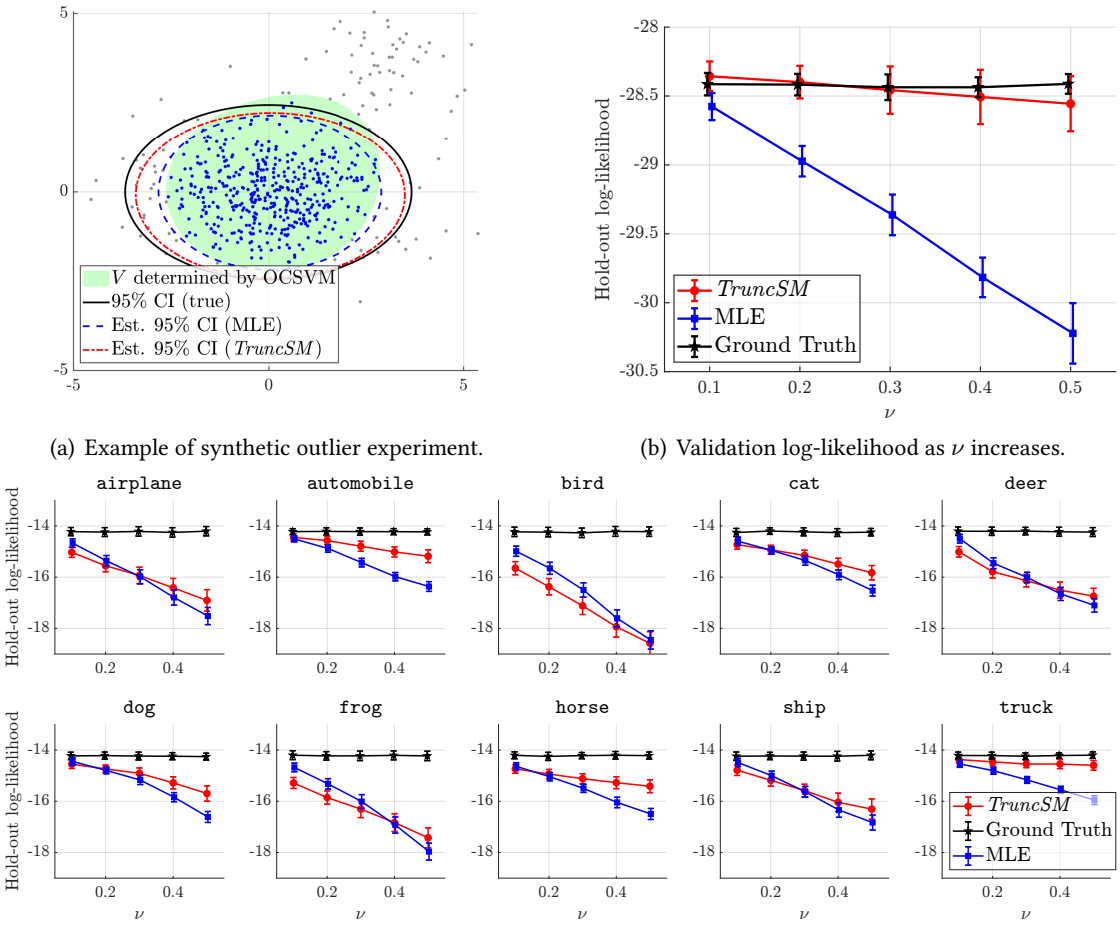


(b) Collection of mean estimates from *TruncSM* and RJ-MLE across 500 trials.



(c) Evaluations of $g_0(\boldsymbol{x})$ on the domain.

Figure 3.7. Estimating the mean location of reported homicides in Chicago in 2008 as a multivariate Normal distribution. Data are truncated at the city's borders.



(a) Example of synthetic outlier experiment.

(b) Validation log-likelihood as ν increases.

(c) Validation log-likelihood as ν increases across the 10 image classes in the CIFAR-10 dataset.

Figure 3.8. Outlier over-trimming compensation using *TruncSM*, in comparison with a naive MLE approach and the ground truth.

same methods. The purpose of this plot is to show how robust the estimator is to initialisations. For each trial, the mean is plotted with a lightly shaded ball with radius equal to the standard deviation. Therefore stronger colours represents where the estimates are placed more often. Both methods place their estimates consistently in the north and south of the city. Some means were placed outside of the city entirely, but these are rare events and can be safely discounted. Overall, the estimates seem consistent with figure 3.7(a) and we can say the estimator is reasonably robust to initialisation choice.

3.4.2 Outlier Trimming Over-compensation

Removing outliers from a corrupted dataset is necessary if one wants to perform density estimation on the uncorrupted dataset. Outlier removal methods come with risks - that any outliers you might exclude from your dataset have a chance of being 'inliers'; coming from the original, uncorrupted dataset, and not being an outlier at all. Further, the true number of outliers is often unknown, and it is sometimes necessary to over-compensate for outliers to ensure that you have removed the majority of them. We propose that *TruncSM* can be used to recover the density parameters of uncorrupted datasets, even when outlier detection methods are severe and remove a significant amount of inliers as well as outliers.

Methods such as One-Class SVM (OCSVM) (Schölkopf et al., 1999) construct a decision function

$$u(\boldsymbol{x}) := \sum_j \hat{\theta}_j \phi_i(\boldsymbol{x}),$$

where $\hat{\theta}_i$ are parameter estimates from the OCSVM model and $\phi_i(\boldsymbol{x})$ is a feature transform defined in advance. In this case, $\phi(\boldsymbol{x}) = k(\cdot, \boldsymbol{x})$, a Gaussian kernel. OCSVM also depends on the hyperparameter $\nu \in [0, 1]$ which governs the proportion of outliers in the dataset. For a given data point $\boldsymbol{x}_\star$, if $u(\boldsymbol{x}_\star) < 0$, then $\boldsymbol{x}_\star$ is labelled as an outlier. Likewise, if $u(\boldsymbol{x}_\star) > 0$, then $\boldsymbol{x}_\star$ is labelled as an inlier. OCSVM implicitly creates the domain V and boundary ∂V given by

$$V = \{\boldsymbol{x} \in \mathbb{R}^d | u(\boldsymbol{x}_\star) > 0\}, \qquad \partial V = \{\boldsymbol{x} \in \mathbb{R}^d | u(\boldsymbol{x}_\star) = 0\}.$$

V is a domain that is smaller in size when more aggressive (i.e. larger) values of ν are chosen, and more outliers (and hence a higher chance of more inliers) are trimmed from the dataset. Using *TruncSM*, we can treat V as a truncated domain with boundary ∂V and estimate the parameters of the uncorrupted density. We first experiment on a simulated example, and then show applicability to image data using the CIFAR-10 dataset.

Simulated Outlier Experiment Firstly, we sample sets of inliers and outliers as

$$\{\boldsymbol{x}_i^{\text{in}}\}_{i=1}^{n_{\text{in}}} \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1.5^2 & 0\\0 & 1\end{bmatrix}\right), \qquad \{\boldsymbol{x}_i^{\text{out}}\}_{i=1}^{n_{\text{out}}} \sim \mathcal{N}\left(\begin{bmatrix}3.5\\3\end{bmatrix}, \begin{bmatrix}0.8^2 & 0\\0 & 0.8^2\end{bmatrix}\right),$$

for $n_{\text{in}} = 500$ and $n_{\text{out}} = 50$. These samples are combined and we apply the OCSVM method to detect outliers. The outlier proportion parameter, ν, is set to $\nu = 0.2$, and so the number of outliers was overestimated.

The task is to estimate the mean and the diagonal of the covariance matrix for the inlier dataset, using only data points classified as inliers by OCSVM (i.e. points inside V), without any labels indicating inliers or outliers. Figure 3.8 shows this experiment, for which figure 3.8(a) shows the visualisation of this example, and the 95% confidence interval (CI) output by *TruncSM* which is significantly closer to the true CI than a naive MLE implementation.

In figure 3.8(b) we repeat the experiment for increasingly aggressive values of ν, which truncates more inliers as ν increases. This experiment is also performed for a much higher dimension,

$d = 20$. Across 36 trials for each value of ν, we record the log-likelihood values using estimates from *TruncSM* and MLE on a hold-out dataset that neither method accessed, and report the mean and standard deviation. In general, *TruncSM* is able to obtain a likelihood that is closer to the truth than the naive MLE implementation which does not account for any form of truncation, across all values of ν.

CIFAR-10 Image Dataset The CIFAR-10 image dataset (Krizhevsky, 2009) is a large collection of images of different objects, each labelled as one of ten classes. For each class, we artifically corrupt the dataset by including $0.1 \cdot n_0$ draws from a $\mathcal{N}(\mathbf{1}_d, \boldsymbol{I}_d)$ distribution alongside the images as outliers, where n_0 is the number of images in each class.

To reduce the computational complexity of the problem, we reduce the dimensionality of the dataset using PCA from a 32×32 image to a 10-dimensional vector. We use both *TruncSM* and MLE to estimate a multivariate Normal distribution on the corrupted data for each class in the reduced 10-dimensional space, and similarly to the earlier example, report the log-likelihood on a hold-out dataset that was previously unseen by all methods, across increasing values of ν. This is plotted in figure 3.8(c).

Across all classes, truncated score matching gives a hold-out log-likelihood that is comparable to MLE. For higher values of ν (i.e. for a heavier truncation setting), and for certain image classes, such as `automobile` and `truck`, truncated score matching gave a log-likelihood that is closer to the ground truth than MLE.

3.5 Discussion

We have presented a method based on score matching that bypasses the calculation of this normalising constant, called *TruncSM*, which is also applicable to any generic truncated domain. The method is derived from theoretical principles, and we have shown through experiments that it achieves results comparable to approximation-based methods. Experiments also show that *TruncSM* is empirically consistent. We also show its broad applicability to truncated density estimation for both multivariate Normal distributions and mixed multivariate Normal distributions. We present an application on real data for crime locations in the city of Chicago. One final example shows the use of *TruncSM* in accounting for over-compensation in outlier detection for synthetic data and an image dataset.

This work was developed to increase the flexibility of the score matching estimator to not only unbounded domains such as $\mathbb{R}^d$, or domains bounded in a specific way, such as $\mathbb{R}^d_+$, but to any generic domain V. This is specified for the Euclidean space only, but prior score matching works exist, such as Mardia et al. (2016), which are applicable to a Riemannian manifold. To extend *TruncSM* in this way, one would need to reformulate the objective function and likely find a new way of defining the distance function in g_0. This is explored in great detail in chapter 4.

TruncSM relies on a weighting function which is a distance function from a point to its closest corresponding point on the boundary of the truncation domain. We have presented different

choices for distance functions and experiments have shown little difference in performance between them. The weighting function, g_0, was chosen based on maximising a Stein discrepancy under a restrictive function class, but a more flexible function class may result in a better performing estimator. This serves as motivation for the work presented in chapter 5.

Chapter 4

Score Matching for Truncated Density Estimation on a Manifold

4.1 Introduction

Performing density estimation over the domain of a manifold such as the surface of the Earth, using methods that rely on a Euclidean data support, may yield inaccurate results due to the curvature of the planet's surface. This inaccuracy will be exacerbated when the region is significantly large, as the effect of the curvature becomes more pronounced. Earth is a manifold, and Euclidean-based methods do not account for this different shape.

Data collected in a geographic setting is likely to encounter a host of artificially occurring boundaries in the form of national or continental boundaries. For example, a country's borders may restrict data collection past an arbitrary stopping point, but the actual geographic structure of the Earth is unlikely to change beyond this point. Countries may only measure events that happen within their border, but not the phenomena which occur outside of their borders, for example in neighbouring countries or the ocean. Therefore it is likely that a dataset is truncated if the dataset originates from data collection confined within one country, making density estimation difficult. Regions defined by countries naturally span large areas of the Earth, making the curvature relevant to the task, exacerbating the difficulty for density estimation. Figure 4.1 shows an example of observed storm events in the U.S., which are only observed within the boundaries of the country due to the data collection process.

To our knowledge, there exists no such work that can perform truncated density estimation without a Euclidean approximation. In this project, we we seek to modify the existing truncated score matching estimator (*TruncSM*, from chapter 3) to account for density estimation on the manifold. This is not a straightforward task, as the integration by parts used to derive a tractable objective in prior score matching implementations is not applicable in the case of the manifold.

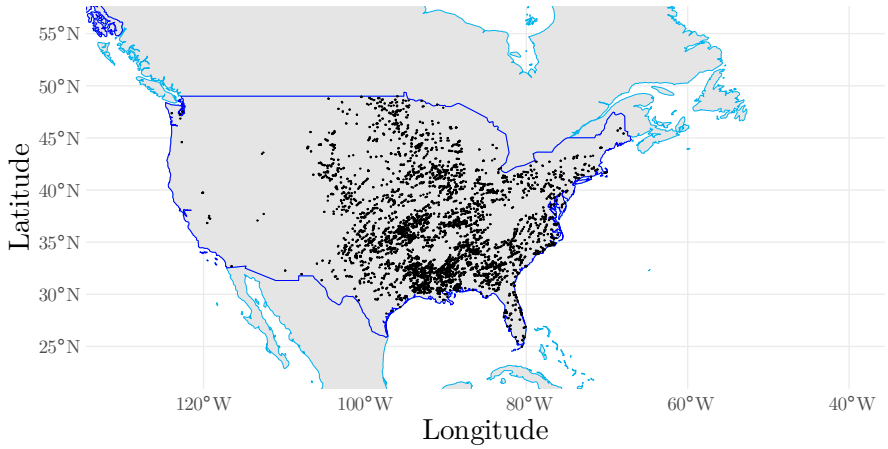


Figure 4.1. Observed locations of storms in the U.S. from 2017 to 2020.

Further to this, the distance function in the weighting function are Euclidean distance functions. Both of these aspects need to be cosidered to be able to achieve the goals of this project.

4.2 Background

The main literature related to this project is related to score matching, section 2.1. This project is an extension of the work described in chapter 3, there is a significant amount of overlap with section 3.2. We now summarise other, more specific, relevant literature to this work.

4.2.1 Manifold

Firstly, let us define some notation to distinguish between manifold space and Euclidean space. Let M be any *compact oriented Riemannian manifold*, and $\boldsymbol{z} \in M$ is a d_M-dimensional vector in *local* coordinates. M is *embedded* into the d-dimensional Euclidean space, $\mathbb{R}^d$, whose coordinates are denoted as $\boldsymbol{x} \in \mathbb{R}^d$.

Let $u(\boldsymbol{z})$ and $v(\boldsymbol{z})$ be two continuous twice differentiable real-valued functions on M, with $\tilde{u}(\boldsymbol{x})$ and $\tilde{v}(\boldsymbol{x})$ being extensions of $u(\boldsymbol{z})$ and $v(\boldsymbol{z})$ to a neighbourhood $N \subset \mathbb{R}^m$ of M. The manifold inner product between $u(\boldsymbol{z})$ and $v(\boldsymbol{z})$ can be written in terms of $\tilde{u}(\boldsymbol{x})$ and $\tilde{v}(\boldsymbol{x})$ as

$$\langle u(\boldsymbol{z}), v(\boldsymbol{z}) \rangle_M = \langle \mathbf{P}\tilde{u}(\boldsymbol{x}), \mathbf{P}\tilde{v}(\boldsymbol{x}) \rangle, \tag{4.1}$$

where $\mathbf{P}$ is the $m \times m$ orthogonal projection matrix onto the m-dimensional tangent hyperplane to M, for which $\mathbf{P} = \mathbf{P}^\top$. The gradient of $u(\boldsymbol{z})$ with respect to $\boldsymbol{z}$ on M is denoted by $\nabla_M u(\boldsymbol{z})$,

and we denote its Euclidean counterpart as $\nabla_E \tilde{u}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \tilde{u}(\boldsymbol{x})$, for which

$$\nabla_M u(\boldsymbol{z}) = \mathbf{P} \nabla_E \tilde{u}(\boldsymbol{x}). \tag{4.2}$$

The Laplace-Beltrami operator is analogous to trace of the hessian matrix, and can be written in terms of Euclidean coordinates as

$$\Delta_M u(\boldsymbol{z}) = \operatorname{tr} \left\{ \mathbf{P} \nabla_E^\top \left(\mathbf{P} \nabla_E \tilde{u}(\boldsymbol{x}) \right) \right\}. \tag{4.3}$$

Finally, Stokes' theorem enables the following integration by parts (Stewart et al., 2020),

Theorem 4.1 (Stokes' Theorem)**.** *Let f_1 and f_2 be continuous twice differentiable functions on a compact Riemannian manifold M, then*

$$\oint_{\partial M} f_1(\boldsymbol{z}) \frac{\partial \nabla_M f_2(\boldsymbol{z})}{\partial \mathbf{n}} \partial M = \int_M (\Delta_M f_2(\boldsymbol{z})) f_1(\boldsymbol{z}) d\boldsymbol{z} + \int_M \langle \nabla_M f_1(\boldsymbol{z}), \nabla_M f_2(\boldsymbol{z}) \rangle_M d\boldsymbol{z}. \tag{4.4}$$

where $\mathbf{n}$ is the unit normal vector.

See, e.g., Lee (2013) or Jost (2008) for a comprehensive overview of Riemannian manifolds.

It is possible that probability density functions on M are defined in terms of either Euclidean coordinates ($p(\boldsymbol{x}; \boldsymbol{\theta})$) or local coordinates ($p(\boldsymbol{z}; \boldsymbol{\theta})$). We assume that the true data density is given by $q(\boldsymbol{z})$, defined in terms of local coordinates.

4.2.2 Spherical Domain

Motivated by geographical data on the surface of the Earth, we primarily consider the unit $(d-1)$-dimensional hypersphere sphere embedded in $\mathbb{R}^d$, i.e. where $M = \mathbb{S}^{d-1}$, given by

$$\mathbb{S}^{d-1} := \{ \boldsymbol{x} \in \mathbb{R}^d \mid \|\boldsymbol{x}\|_2 = 1 \}.$$

On the 2D sphere, Euclidean coordinates have components $\boldsymbol{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$, whereas spherical coordinates have components $\boldsymbol{z} = (\vartheta, \varphi) \in S^2$, where $\vartheta \in [0, \pi)$ is the colatitude and $\varphi \in [0, 2\pi]$ is the longitude. Conversion between spherical and Euclidean coordinates is given by

$$\boldsymbol{x} = (r \cos \varphi, r \sin \varphi \cos \vartheta, r \sin \varphi \sin \vartheta),$$

and conversion from Euclidean coordinates to spherical coordinates is given by

$$\boldsymbol{z} = (\arccos(x_1/r), \arctan(x_3/x_2)), \tag{4.5}$$

where $r = (x_1^2 + x_2^2 + x_3^2)^{1/2}$. The projection matrix in this case is $\mathbf{P} = \mathbf{I}_3 - \boldsymbol{x}\boldsymbol{x}^\top$ (Mardia et al., 2016). Using this projection matrix, we can expand the Laplace-Beltrami operator equation (4.3)

on $u(\boldsymbol{z})$ as follows

$$\begin{aligned}\Delta_M u(\boldsymbol{z}) &= \operatorname{tr}\{\mathbf{P}\nabla_E^\top(\mathbf{P}\nabla_E\tilde{u}(\boldsymbol{x}))\} \\ &= \operatorname{tr}\{\mathbf{P}\nabla_E^\top[(\mathbf{I}_3 - \boldsymbol{x}\boldsymbol{x}^\top)\tilde{u}(\boldsymbol{x})]\} \\ &= \operatorname{tr}\left\{\mathbf{P}\left[\nabla_E^\top(\mathbf{I}_3 - \boldsymbol{x}\boldsymbol{x}^\top)\tilde{u}(\boldsymbol{x}) + (\mathbf{I}_3 - \boldsymbol{x}\boldsymbol{x}^\top)\nabla_E\tilde{u}(\boldsymbol{x})\right]\right\} \\ &= \operatorname{tr}\left\{\mathbf{P}\left[-2\tilde{u}(\boldsymbol{x})\boldsymbol{x}^\top + \mathbf{P}\nabla_E\tilde{u}(\boldsymbol{x})\right]\right\}. \end{aligned} \tag{4.6}$$

In implementation, we assume an independent and identically distributed dataset given by either

$$\{\boldsymbol{z}_i\}_{i=1}^n = \{(\vartheta_i, \varphi_i)\}_{i=1}^n,$$

drawn from $q(\boldsymbol{z})$, which can be mapped to

$$\{\boldsymbol{x}_i\}_{i=1}^n = \{(x_{i,1}, x_{i,2}, x_{i,3})\}_{i=1}^n$$

without loss of generality.

4.2.2.1 Spherical Distributions

We consider two primary probability distributions on $\mathbb{S}^{d-1}$ in $\mathbb{R}^d$: the Kent distribution and the von-Mises Fisher distribution. See Chapter 9, Mardia and Jupp (2009) for an overview of spherical distributions. Note that these spherical distributions are defined in terms of Euclidean coordinates, $\boldsymbol{x} \in \mathbb{R}^d$.

Kent Distribution The PDF for the Kent distribution is

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{Z(\kappa, \boldsymbol{\alpha})} \exp\{\kappa \boldsymbol{\mu}^\top \boldsymbol{x} + \sum_{j=1}^{d-1} \alpha_j (\boldsymbol{\gamma}_j^\top \boldsymbol{x})^2\},$$

for $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \kappa, \boldsymbol{\alpha})$, and where $Z(\kappa, \boldsymbol{\alpha})$ is the normalising constant that ensures $p(\boldsymbol{x}; \boldsymbol{\theta})$ integrates to one (Kent, 1982). Here, $\boldsymbol{\mu}$ is the mean direction, whilst $\boldsymbol{\gamma}_j$ are the axes which determine the orientation of the probability contours. The concentration parameter, κ, and the ovalness parameters, α_j, determine how elliptical the distribution is (Mardia and Jupp, 2009).

The Kent distribution is analogous to the Multivariate Normal (MVN) distribution on $\mathbb{R}^d$, and has similar properties, such as κ, α_j, and $\boldsymbol{\gamma}_j$ controlling the shape of the data distribution, similar to the role of the covariance matrix of the MVN, and the mean direction $\boldsymbol{\mu}$ being similar to the mean of the MVN. The Kent distribution is sometimes referred to as the Fisher-Bingham distribution.

von Mises-Fisher Distribution The von Mises-Fisher distribution is a special case of the Kent distribution where $\alpha = 0$. It has PDF

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{Z(\kappa)} \exp\{\kappa \boldsymbol{\mu}^\top \boldsymbol{x}\},$$

for $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{\theta} = (\boldsymbol{\mu}, \kappa)$. Here, $Z(\kappa)$ is the normalising constant, $\boldsymbol{\mu}$ is the mean direction and κ is the concentration parameter (Mardia and Jupp, 2009); all defined the same as in the Kent distribution.

The von Mises-Fisher distribution can also be considered as analogous to the MVN with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma} = \kappa^{-1}\mathbf{I}_d$.

4.2.3 Manifold Score Matching

Let us denote M_C as a *closed* oriented Riemannian manifold, such that M_C is compact and oriented *without* any edges or boundaries. Consider, for example, the sphere $\mathbb{S}^2$. Mardia et al. (2016) proposed a variant of score matching that is defined on M_C. Let $\boldsymbol{z} \in M_C$ and $\boldsymbol{z} \sim q(\boldsymbol{z})$. The *manifold* score matching objective is given by

$$J_{\text{MSM}}(\boldsymbol{\theta}) := \int_{M_C} q(\boldsymbol{z}) \left\| \boldsymbol{\psi}_q(\boldsymbol{z}) - \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \right\|^2 d\boldsymbol{x}, \tag{4.7}$$

where $\boldsymbol{\psi}_q(\boldsymbol{z}) := \nabla_M \log q(\boldsymbol{z})$ and $\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) := \nabla_M \log p(\boldsymbol{z}; \boldsymbol{\theta})$ are the score functions. Like preceding score matching methods, equation (4.7) is intractable in its current state due to the unknown quantity $\boldsymbol{\psi}_q(\boldsymbol{z})$. Mardia et al. (2016) made use of theorem 4.1 to rewrite the objective, noting that since M_C is closed, there is no need for boundary terms and the left hand side of equation (4.4) is zero. Hence, equation (4.7) can be written (up to a constant) as

$$J_{\text{MSM}}(\boldsymbol{\theta}) = \int_{M_C} q(\boldsymbol{z}) \left[\langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M + 2\Delta_M \log p(\boldsymbol{z}; \boldsymbol{\theta}) \right] d\boldsymbol{z}. \tag{4.8}$$

This can be written as an expectation, which does not involve evaluating the unknown data density $q(\boldsymbol{z})$. See Section 3, Mardia et al. (2016) for details. This step is an analogue to using integration-by-parts in classical score matching to derive a tractable objective function.

Expectations in the objective function can be replaced with their Monte Carlo approximations to yield a tractable equation for use in density estimation.

4.3 Truncated Manifold Score Matching

Let us now consider a compact oriented manifold *with boundary* as M_T, where the boundary is denoted as ∂M_T. For example, the upper hemisphere of a two-dimensional sphere, $\mathbb{S}^2$, is given by

$$\mathbb{H}^+ = \{\boldsymbol{z} \in \mathbb{S}^2 | z_2 > 0\}$$

and has a boundary, ∂M_T, given by the equator of $\mathbb{S}^2$. Instead of observing data that lie on the full M_C, we instead observe a subset on M_T, i.e. the support of $q(\boldsymbol{z})$ is the *truncated* domain M_T. In this case, the assumption by Mardia et al. (2016) that the boundary term in equation (4.4) is zero no longer holds.

4.3.1 Truncated Score Matching Objective on a Manifold

The aim of this section is to extend the approach of Mardia et al. (2016), using methods of truncated score matching described in chapter 3, to combine truncated and manifold score matching, which we entitle Truncated Manifold Score Matching (TMSM). Similar to previous score matching implementations, we propose to minimise the expected squared distance between the score functions of the statistical model and the data,

$$J_{\text{TMSM}}(\boldsymbol{\theta}) := \int_{M_T} q(\boldsymbol{z}) g(\boldsymbol{z}) \left\| \boldsymbol{\psi}_q(\boldsymbol{z}) - \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \right\|^2 d\boldsymbol{x}, \tag{4.9}$$

where the key difference from Mardia et al. (2016) is the integration over M_T instead of M_C. Similar to generalised score matching (Yu et al., 2018, 2019, 2021), $g(\boldsymbol{z})$ is a weighting function, for which we define the following boundary condition:

$$\lim_{\boldsymbol{z} \to \boldsymbol{z}'} g(\boldsymbol{z}) q(\boldsymbol{z}) = 0 \tag{4.10}$$

where $\boldsymbol{z} \in M_T$, $\boldsymbol{z}' \in \partial M_T$ and $\boldsymbol{z} \to \boldsymbol{z}'$ takes any point sequence converging to $\boldsymbol{z}'$ in account. This condition is key, but is easily verified by choice of g, which will be a focus in the coming sections. Along with theorem 4.1, this is used to derive the following theorem.

Theorem 4.2 (Truncated Manifold Score Matching Objective)**.** *Given a function g for which equation* (4.10) *holds, then $J_{\text{TMSM}}(\boldsymbol{\theta})$ can be expressed as*

$$\begin{aligned} J_{\text{TMSM}}(\boldsymbol{\theta}) = \int_{M_T} q(\boldsymbol{z}) g(\boldsymbol{z}) \langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M d\boldsymbol{z} + 2 \int_{M_T} q(\boldsymbol{z}) g(\boldsymbol{z}) \Delta_M \log p(\boldsymbol{z}; \boldsymbol{\theta}) d\boldsymbol{z} \\ + 2 \int_{M_T} q(\boldsymbol{z}) \langle \nabla_M g(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M d\boldsymbol{z}, \end{aligned} \tag{4.11}$$

where the integration is over M_T, $\langle \cdot, \cdot \rangle_M$ is the manifold inner product and Δ_M is the Laplace-Beltrami operator.

Proof. We can expand out (4.9) to give

$$J_{\text{TMSM}}(\boldsymbol{\theta}) = \int_{M_T} q(\boldsymbol{z}) g(\boldsymbol{z}) \Big[\langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M - 2 \langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \boldsymbol{\psi}_q(\boldsymbol{z}) \rangle_M \Big] d\boldsymbol{z} + C_q,$$

where $C_q = \int_{M_T} q(\boldsymbol{z}) g(\boldsymbol{z}) \langle \nabla_M \log q(\boldsymbol{z}), \nabla_M \log q(\boldsymbol{z}) \rangle_M d\boldsymbol{x}$. Note that we can write

$$\int_{M_T} q(\boldsymbol{z}) g(\boldsymbol{z}) \langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \boldsymbol{\psi}_q(\boldsymbol{z}) \rangle_M d\boldsymbol{z} = \int_{M_T} g(\boldsymbol{z}) \langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \nabla_M q(\boldsymbol{z}) \rangle_M d\boldsymbol{z},$$

due to $\nabla_M \log q(\boldsymbol{z}) = q(\boldsymbol{z})^{-1} \nabla_M q(\boldsymbol{z})$, and therefore

$$J_{\text{TMSM}}(\boldsymbol{\theta}) = \int_{M_T} q(\boldsymbol{z}) g(\boldsymbol{z}) \Big[\langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M - 2 \langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \nabla_M q(\boldsymbol{z}) \rangle_M \Big] d\boldsymbol{z} + C_q.$$

Now, noting that theorem 4.1 can be applied to M_T, and rearranged as

$$\int_{M_T} \langle \nabla_M f_1(\boldsymbol{z}), \nabla_M f_2(\boldsymbol{z}) \rangle_M d\boldsymbol{z} = \oint_{\partial M_T} f_1(\boldsymbol{z}) \frac{\partial \nabla_M f_2(\boldsymbol{z})}{\partial \mathbf{n}} \partial M_T - \int_{M_T} (\Delta_M f_2(\boldsymbol{z})) f_1(\boldsymbol{z}) d\boldsymbol{z},$$

and then letting $f_1 = g(\boldsymbol{z}) q(\boldsymbol{z})$ and $f_2 = \log p(\boldsymbol{x}; \boldsymbol{\theta})$, on our manifold M_T with boundary ∂M_T, we have

$$\begin{aligned} \int_{M_T} \langle \nabla_M [g(\boldsymbol{z}) q(\boldsymbol{z})], \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M d\boldsymbol{z} = \oint_{\partial M_T} g(\boldsymbol{z}) q(\boldsymbol{z}) \frac{\partial \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z})}{\partial \mathbf{n}} \partial M_T \\ - \int_{M_T} (\Delta_M \log p(\boldsymbol{z}; \boldsymbol{\theta})) g(\boldsymbol{z}) q(\boldsymbol{z}) d\boldsymbol{z}, \end{aligned}$$

where $\mathbf{n}$ is the unit normal vector. By equation (4.10), the boundary term is equal to zero, leaving

$$\int_{M_T} \langle \nabla_M [g(\boldsymbol{z}) q(\boldsymbol{z})], \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M d\boldsymbol{z} = - \int_{M_T} (\Delta_M \log p(\boldsymbol{z}; \boldsymbol{\theta})) g(\boldsymbol{z}) q(\boldsymbol{z}) d\boldsymbol{z}.$$

Expanding the left hand side via the chain rule and rearranging, gives

$$\begin{aligned} \int_{M_T} g(\boldsymbol{z}) \langle \nabla_M q(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M d\boldsymbol{z} = - \int_{M_T} (\Delta_M \log p(\boldsymbol{z}; \boldsymbol{\theta})) g(\boldsymbol{z}) q(\boldsymbol{z}) d\boldsymbol{z} \\ - \int_{M_T} q(\boldsymbol{z}) \langle \nabla_M g(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M d\boldsymbol{z} \end{aligned}$$

This can be substituted back into $J_{\text{TMSM}}(\boldsymbol{\theta})$ to obtain

$$\begin{aligned} J_{\text{TMSM}}(\boldsymbol{\theta}) = \int_{M_T} q(\boldsymbol{z}) g(\boldsymbol{z}) \langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M d\boldsymbol{x} + 2 \int_{M_T} q(\boldsymbol{z}) g(\boldsymbol{z}) \Delta_M \log p(\boldsymbol{z}; \boldsymbol{\theta}) d\boldsymbol{x} \\ + 2 \int_{M_T} q(\boldsymbol{z}) \langle \nabla_M g(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M d\boldsymbol{x}, \end{aligned}$$

which concludes the proof. ■

Provided that we have a set of samples, $\{\boldsymbol{z}_i\}_{i=1}^n \sim q(\boldsymbol{z})$, then using the law of large numbers, a sample version of equation (4.11) can be written as

$$\begin{aligned} \hat{J}_{\text{TMSM}}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n g(\boldsymbol{z}_i) \langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}_i), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}_i) \rangle_M + \frac{2}{n} \sum_{i=1}^n g(\boldsymbol{z}_i) \Delta_M \log p(\boldsymbol{z}_i; \boldsymbol{\theta}) \\ + \frac{2}{n} \sum_{i=1}^n \langle \nabla_M g(\boldsymbol{z}_i), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}_i) \rangle_M, \end{aligned} \tag{4.12}$$

provided that n is sufficiently large, and where $\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}_i) = \nabla_M \log p(\boldsymbol{z}_i; \boldsymbol{\theta})$. The TMSM estimator is given by

$$\hat{\boldsymbol{\theta}} := \arg\min_{\boldsymbol{\theta}} \hat{J}_{\text{TMSM}}(\boldsymbol{\theta}).$$

Although the weighting function, $g(\boldsymbol{z})$, has not yet been specified, we only require a g such that the condition for which $g(\boldsymbol{z}') = 0$ for all $\boldsymbol{z}' \in \partial M$ holds, provided it is at least once differentiable. In section 4.3.3, we propose two distance functions that naturally fulfill this criteria, and are suited for the sphere.

4.3.2 Score Matching on Spherical Distributions

The score matching approach detailed above is applicable to any manifold on which Stokes' theorem (theorem 4.1) holds. This includes the unit $(d-1)$-hypersphere in $\mathbb{R}^d$ from which the probability distributions detailed in section 4.2.2.1 are defined. These distributions are defined in terms of Euclidean coordinates, i.e. $p(\boldsymbol{x}; \boldsymbol{\theta})$. The TMSM objective requires specification of the score function, $\nabla_M \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z})$, its inner product with itself, $\langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M$, and the Laplace-Beltrami operator, $\Delta_M \log p(\boldsymbol{z}; \boldsymbol{\theta})$, all in terms of local coordinates. Making use of the relationship between these operations in Euclidean space and manifold space, via equations (4.1) to (4.3), we detail the required operations on the von-Mises Fisher and Kent distribution in this section.

Kent Distribution The score function and its derivative in Euclidean space can be calculated as

$$\begin{aligned} \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) &= \kappa\boldsymbol{\mu} + 2\sum_{j=1}^{d-1} \alpha_j \boldsymbol{\gamma}_j \odot (\boldsymbol{\gamma}_j^\top \boldsymbol{x}), \\ \nabla_E \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) &= 2\sum_{j=1}^{d-1} \alpha_j \boldsymbol{\gamma}_j \boldsymbol{\gamma}_j^\top, \end{aligned} \tag{4.13}$$

where $\odot$ represents elementwise product. Using the formulae for the inner product (equation (4.1)), gradient (equation (4.2)) and Laplace-Beltrami operator (equation (4.3)), for the Kent distribution we have

$$\begin{aligned} \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) &= \mathbf{P}\Big(\kappa\boldsymbol{\mu} + 2\sum_{j=1}^{d-1} \alpha_j \boldsymbol{\gamma}_j \odot (\boldsymbol{\gamma}_j^\top \boldsymbol{x})\Big), \\ \langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) \rangle_M &= (\mathbf{P}\kappa\boldsymbol{\mu})^\top (\mathbf{P}\kappa\boldsymbol{\mu}) + 4\left\| \mathbf{P}\left[\alpha_1 \cdot \boldsymbol{\gamma}_1(\boldsymbol{\gamma}_1^\top \boldsymbol{x}) + \alpha_2 \cdot \boldsymbol{\gamma}_2(\boldsymbol{\gamma}_2^\top \boldsymbol{x})\right] \right\|_2^2, \\ \Delta_M \log p(\boldsymbol{z}; \boldsymbol{\theta}) &= 2\text{tr}\{\mathbf{P}\big(\alpha_1 \cdot \boldsymbol{\gamma}_1\boldsymbol{\gamma}_1^\top + \alpha_2 \cdot \boldsymbol{\gamma}_2\boldsymbol{\gamma}_2^\top\big) - \mathbf{P}\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})\boldsymbol{x}^\top\}. \end{aligned}$$

von Mises-Fisher Distribution The score function and its derivative in Euclidean space are given by

$$\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) = \kappa\boldsymbol{\mu}, \tag{4.14}$$

$$\nabla_E \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}) = \mathbf{0}_{d\times d}, \tag{4.15}$$

where $\mathbf{0}_{d\times d}$ denotes the $d \times d$ matrix of zeros. Using the formulae for the inner product (equation (4.1)), gradient (equation (4.2)) and Laplace-Beltrami operator (equation (4.3)), for the von-Mises Fisher distribution we have

$$\begin{aligned}
\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{z}) &= \mathbf{P}(\kappa\boldsymbol{\mu}),\\
\langle \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x}), \boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})\rangle_M &= (\mathbf{P}\kappa\boldsymbol{\mu})^\top(\mathbf{P}\kappa\boldsymbol{\mu}),\\
\Delta_M \log p(\boldsymbol{x};\boldsymbol{\theta}) &= -2\mathrm{tr}\{\mathbf{P}\boldsymbol{\psi}_{p_{\boldsymbol{\theta}}}(\boldsymbol{x})\boldsymbol{x}^\top\}.
\end{aligned}$$

4.3.3 Choice of Weighting Function

The weighting function, $g(\boldsymbol{z})$, needs to satisfy the criteria that $g(\boldsymbol{z}') = 0\ \forall \boldsymbol{z}' \in \partial M_T$, i.e. the function takes zero at the boundary of the manifold. Choice of $g(\boldsymbol{z})$ is an important topic of study and will play a key role in the quality of the TMSM estimator.

We take inspiration from the methods proposed in chapter 3, in section 3.2.3, and aim to define a distance function. However, defining a distance function on the manifold is not as straightforward as it is for the Euclidean domain. Two propositions of distance functions for $\mathbb{S}^2$ are discussed in this section, which are then incorporated into a general weighting function.

Haversine distance For two points on $M_T \subset \mathbb{S}^2$, defined by $\boldsymbol{z} = (\varphi, \vartheta) \in M_T$ and $\boldsymbol{z}' = (\varphi', \vartheta') \in \partial M_T$, the *Haversine*, or *great circle* distance, measures the geodesic distance between these two points along the surface of a sphere, and is given by

$$\mathrm{dist}(\boldsymbol{z}, \boldsymbol{z}') := 2r \arcsin\left(\sqrt{u}\right),$$

where

$$u = \sin^2\left(\frac{\varphi' - \varphi}{2}\right) + \cos(\varphi)\cos\left(\varphi'\right)\sin^2\left(\frac{\vartheta' - \vartheta}{2}\right).$$

Projected Euclidean distance Points on the surface of the sphere, e.g. $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^3$ can be projected on a two dimensional plane in $\mathbb{R}^2$ via setting the j'-th coordinate to a fixed constant (in practice, we can set it to zero). The corresponding two-dimensional Euclidean distance will be measured on this plane. See Figure 4.2 for a visualisation of this method.

Let $\boldsymbol{x}_e = \mathrm{Proj}(\boldsymbol{z})$, where Proj is a function that defines the projection from the sphere to the 2D Euclidean plane, then for two projected points $\boldsymbol{x}_e, \boldsymbol{x}_e' \in \mathbb{R}^2$, the distance function is given by

$$\mathrm{dist}(\boldsymbol{z}, \boldsymbol{z}') := \|\mathrm{Proj}(\boldsymbol{z}) - \mathrm{Proj}(\boldsymbol{z}')\|_2 = \|\boldsymbol{x}_e - \boldsymbol{x}_e'\|_2.$$

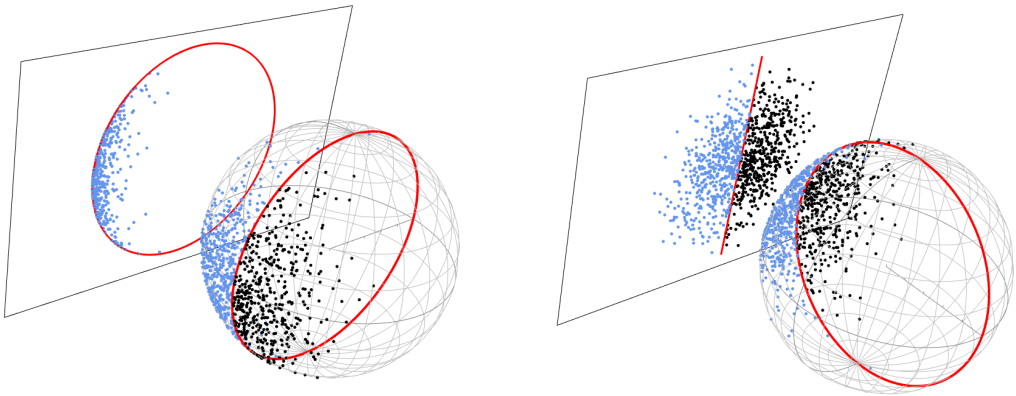


Figure 4.2. Two visualisations of the Euclidean projection, projecting across a different axis (fixing a different coordinate to zero), with simulated points from a von Mises-Fisher distribution. A hemispherical boundary is visualised in red, with some unobserved points in black and observed points in blue. The projected Euclidean distance is calculated on the projected 2D plane.

The gradient is easily defined as

$$\partial_{x_j} \text{dist}(\boldsymbol{z}, \boldsymbol{z}') = \frac{x_{e,j} - \tilde{x}_{e,j}}{\|\boldsymbol{x}_e - \tilde{\boldsymbol{x}}_e\|_2}, \quad \forall j \in \{1, \ldots, d\} \backslash \{j'\}$$

and $\partial_{x_{j'}} \text{dist}(\boldsymbol{z}, \boldsymbol{z}') = 0$, because the j'-th coordinate was fixed to a constant. Here, $x_{e,j}$ denotes the j-th element of $\boldsymbol{x}_e$, and $\tilde{x}_{e,j}$ denotes the j-th element of $\tilde{\boldsymbol{x}}_e$, which is the corresponding *projection point* of $\boldsymbol{x}_e$; the closest point on the boundary to $\boldsymbol{x}_e$.

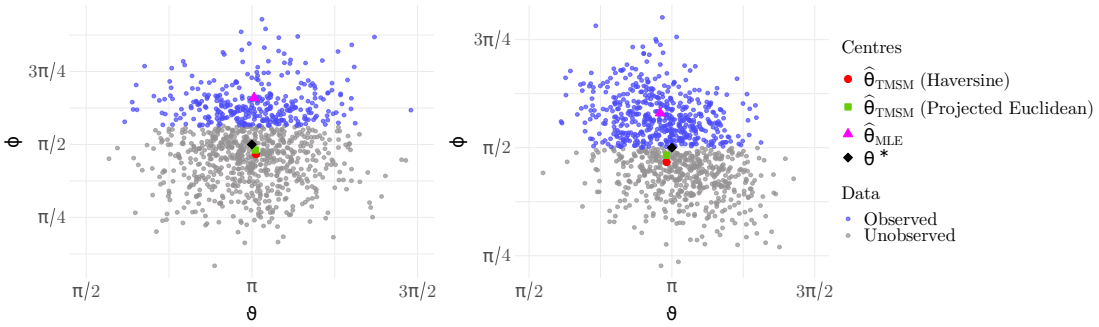
Weighting function Provided that a distance function is defined, we define the weighting function by the shortest distance from a point $\boldsymbol{z}$ to the boundary ∂M_T, given by

$$g(\boldsymbol{z}) = \min_{\boldsymbol{z}' \in \partial M_T} \text{dist}(\boldsymbol{z}, \boldsymbol{z}').$$

In the case of the projected Euclidean distance, the boundary itself would also need to be projected into the Euclidean space. For any distance function, this weighting function naturally satisifes the boundary condition required by TMSM detailed in equation (4.10).

4.4 Experiments

We focus on truncated domains on the sphere $\mathbb{S}^2$ in $\mathbb{R}^3$, as this manifold covers many important applications in geological or earth science applications where the datasets are collected from a truncated region on a the Earth.



(a) Von-mises Fisher distribution.

(b) Kent distribution.

Figure 4.3. Simulated experiments for TMSM.

4.4.1 Illustrative Example

We start with a synthetic data experiment, where data are simulated from a von Mises-Fisher distribution and Kent distribution using a rejection sampling algorithm, as recommended by Wood (1994). For the first example, we simulate n truncated samples from a von Mises-Fisher distribution with an unknown true mean direction $:= \boldsymbol{\mu}^\star = (\pi/2, \pi)$ and an unknown concentration parameter $\kappa^\star = 6$, where the truncation boundary is at a constant value of colatitude φ. The aim of the experiment is to produce an estimate, $\hat{\boldsymbol{\theta}}^\star := (\hat{\boldsymbol{\mu}}, \hat{\kappa})$ of $\boldsymbol{\theta}^\star := (\boldsymbol{\mu}^\star, \kappa^\star)$, the mean direction and the concentration parameter. Figure 4.3(a) shows the results of this example using the two different weighting functions proposed in section 4.3.3, compared to a naive MLE implementation which does not account for truncation. Both estimates of the mean direction lie roughly in the correct location, close to $\boldsymbol{\theta}^\star$.

The second example is a similar experiment, where samples are instead simulated from the Kent distribution with an unknown true mean direction $\boldsymbol{\mu}^\star = (\pi/2, \pi)$ and a known concentration parameter, $\kappa^\star = 10$, and ovalness parameter, $\alpha^\star = 3$. The aim of this experiment is to produce an estimate $\hat{\boldsymbol{\theta}} := \hat{\boldsymbol{\mu}}$ of $\hat{\boldsymbol{\theta}} := \hat{\boldsymbol{\mu}}$, the mean direction only. Figure 4.3(b) shows the results of this experiment. Similarly, the mean directions estimated appear close to the true value.

These plots serve as an example of the full capability of the method; even with half of the observations missing, the estimates of the proposed methods are closer to the true mean direction than MLE.

4.4.2 Estimation Accuracy

We repeat experiments, similar to those presented in section 4.4.1, across different setups. For each trial, we simulate and truncate samples until we have n truncated samples, simulated from either the von Mises-Fisher or Kent distribution. The truncation boundary in this case is the upper hemisphere, i.e. $M_T = \mathbb{H}^+$, where only datapoints for which $\varphi > \pi/2$ were observed. Across different values of sample size n, figure 4.4 shows the root mean squared error (RMSE),

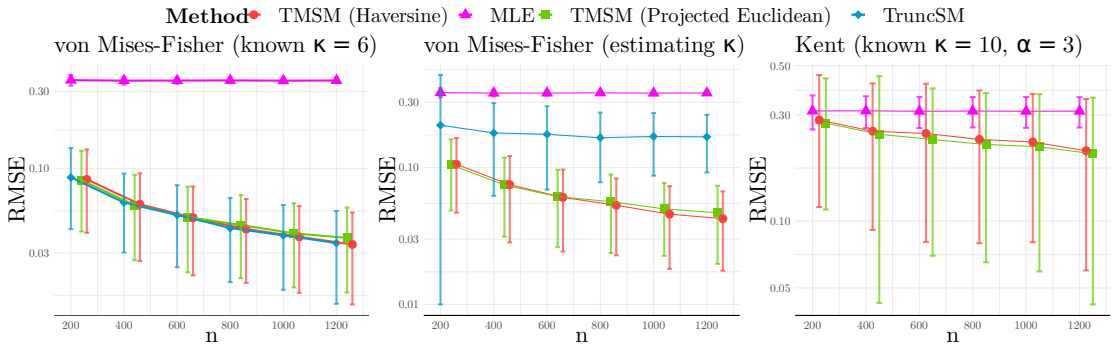


Figure 4.4. Mean estimation error and standard deviation against sample size n, on a log scale, from 512 trials for each method.

given by

$$\mathrm{RMSE}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star) = \frac{1}{d}\sqrt{\sum_{j=1}^{d}(\hat{\theta}_j - \theta_j^\star)^2}\,.$$

The three experiments are summarised as follows. First, we calculate the RMSE between the true mean direction, $\boldsymbol{\theta}^\star := \boldsymbol{\mu}^\star$, of the von Mises-Fisher distribution with a known concentration parameter κ, and the corresponding estimates $\hat{\boldsymbol{\theta}} := \hat{\boldsymbol{\mu}}$. The second experiment is the estimation of $\boldsymbol{\theta}^\star := [\boldsymbol{\mu}^\star, \kappa^\star]^\top$, and the RMSE is reported between $\kappa^\star$ and $\hat{\kappa}$. Finally, given a known concentration parameter κ and ovalness parameter α, the RMSE is calculated between the true mean direction, $\boldsymbol{\theta}^\star := \boldsymbol{\mu}^\star$ of the Kent distribution and the estimates $\hat{\boldsymbol{\theta}} := \hat{\boldsymbol{\mu}}$.

These estimates are compared across three methods: *TruncSM* (chapter 3), with a multivariate Normal assumption[1], our method Truncated Manifold Score Matching (TMSM), with either the Haversine or projected Euclidean distance, and MLE for a baseline.

As expected, in all cases the estimation error decreases as sample size increases, indicating empirical consistency. Comparing the two distance functions for $g(\boldsymbol{z})$, both give near identical results across all benchmarks. Our method achieves comparable performance to *TruncSM* in the most cases, and outperforms it when κ is also estimated. This is also expected as *TruncSM* incorrectly assumes a Euclidean domain.

4.4.3 Storm Events in the U.S.

Consider an application where the task is to estimate the probable locations of extreme storm events in the U.S., with dataset given by National Climatic Data Center (2022). Whilst storms are not restricted to land, observations of these events only happen on land and within the

[1]Whilst the TMSM methods can estimate κ directly, for *TruncSM* we let $\boldsymbol{\mu}_{\text{MVN}} = \boldsymbol{\mu}_{\boldsymbol{z}}$, i.e. the mean of the MVN distribution is the mean direction of the von Mises-Fisher distribution in polar coordinates, and $\boldsymbol{\Sigma}_{\text{MVN}} = \kappa^{-1}\mathbf{I}_d$, i.e. the covariance matrix of the MVN is isotropic. In this case, we treat κ^{-1} as the precision parameter and estimate it directly with *TruncSM*.

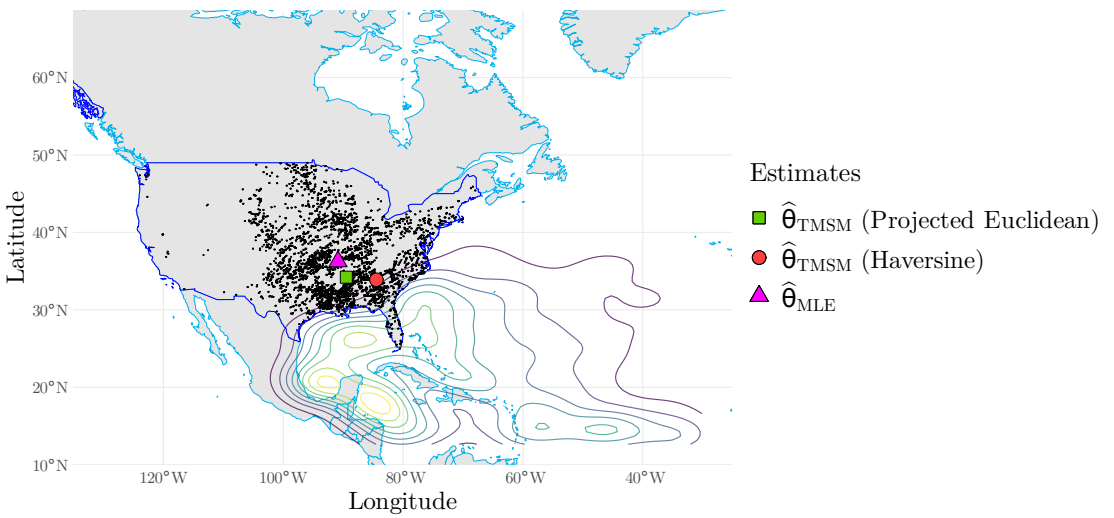


Figure 4.5. Observed storms as recorded by the U.S. from 2017 to 2020, with a kernel density estimate contour of storm origin locations over the Atlantic ocean.

country's borders. Furthermore, the large area spanned by the U.S. is significant enough that the curvature of its surface will affect any estimation, highlighting the need for considering density estimation on a truncated manifold.

We hypothesise that more storms are likely to occur on the East Coast of the U.S., and spread across the land from there. We assume there is a true storm 'centre' where most storms will be spread out from beyond the country's eastern border (Oliver, 2008).

Assuming the data can be modelled accurately by a von Mises-Fisher distribution, we produce an estimate of the mean direction, $\hat{\boldsymbol{\theta}} := \hat{\boldsymbol{\mu}}$. The estimates $\hat{\boldsymbol{\theta}}$ for both MLE and TMSM are given in Figure 4.5. Included in the figure is a kernel density estimate of storm points of origin, which are calculated via a post-storm analysis, after a storm is observed Landsea and Franklin (2013). Unsurprisingly, we see that $\hat{\boldsymbol{\theta}}$ given by MLE is at the center of the dataset, whereas the truncated manifold score matching estimate is further South East, closer to the storm origins.

4.5 Conclusion

This work can be considered an in-depth study in applying the methods presented in Liu et al. (2022), i.e. chapter 3, to a Riemannian manifold with boundary. The extension from manifold score matching (Mardia et al., 2016) to this method, truncated manifold score matching (TMSM), has produced a score matching method which is applicable to a compact and oriented Riemannian manifold with boundary. The objective function was made tractable by an application of Stokes'

theorem, and requires a weighting function to satisfy a necessary boundary condition. This weighting function is pivotal, and we proposed two choices for the specific application of the two-dimensional sphere, $\mathbb{S}^2$, embedded in $\mathbb{R}^3$. Through empirical experiments, we showed that these weighting functions provided comparable results to one another and effectively estimated parameters of two spherical distributions: the von Mises-fisher distribution and the Kent distribution. We also compared the method to a naive MLE implementation (without accounting for truncation) and *TruncSM* (which uses a Euclidean space assumption), for which TMSM had higher accuracy than both methods. A further application was presented on severe storms in the U.S., in which the score matching estimator gave a mean direction estimate lying on the East Coast, where the bulk of the storms originated.

A fully packaged implementation in R, with examples and a tutorial, can be found at the GitHub repository `https://github.com/dannyjameswilliams/truncsm`.

Chapter 5

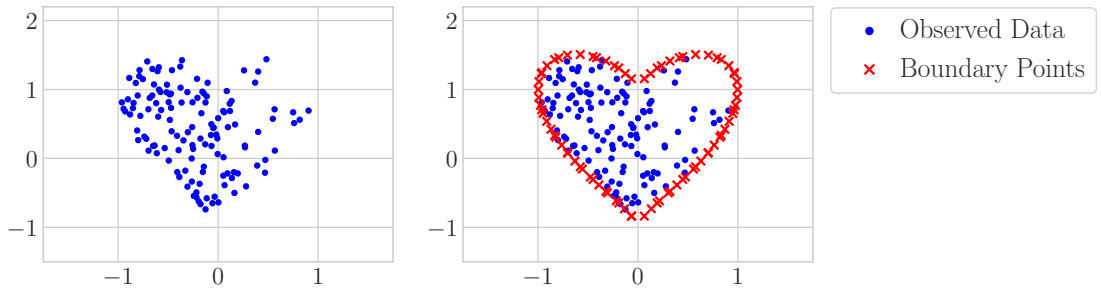
Approximate Stein Classes for Truncated Density Estimation

5.1 Introduction

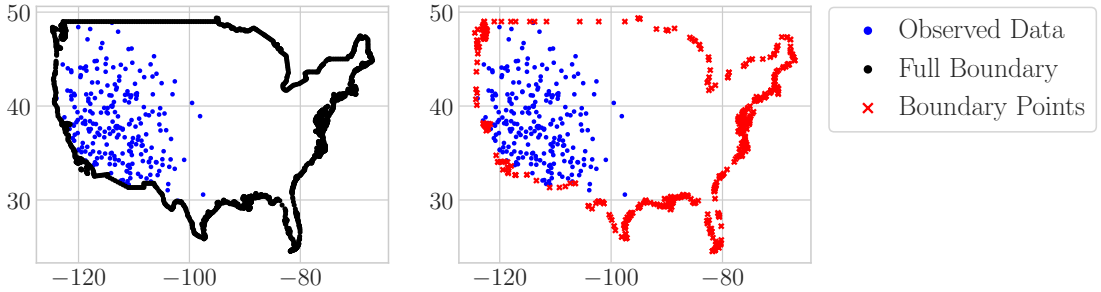
Suppose we wish to estimate a density model for paper submissions at an academic conference. By definition, we can only observe the accepted papers but not the rejected ones. The distinguishing factors between accepted papers and rejected papers form a boundary between acceptance and rejection, which is difficult or even impossible to define without knowing the true underlying criteria. However, we can easily identify *borderline* submissions from the lowest review scores in the observed dataset, which can be seen as *samples* from the boundary. This is an example of truncated density estimation where the truncation effect (the functional form of the boundary) is unknown.

The methods presented in the thesis so far, in chapters 3 and 4 (Liu et al., 2022; Williams and Liu, 2022), and other bounded-domain density estimation methods, e.g. Xu (2022); Yu et al. (2021), require a weighting function which measures the distance from a given data point to the boundary which is chosen in advance. The computation of this distance function can be challenging when the boundary is complex and high-dimensional. When the functional form of the boundary is unknown, and we only have access to a finite set of boundary points, this distance function is also unknown. With only a finite set of boundary points, no existing methods can utilise this information to perform truncated density estimation. Our aim is to derive a method that provides comparable results to prior works which *do* have access to a functional form of the boundary, but with only using this set of boundary points. This method should be a more flexible approach to truncated density estimation than existing methods.

Our work proceeds as follows. We first define *approximate Stein classes* and a corresponding



(a) An example where the shape or functional form of the boundary is not always apparant nor accessible, but can be inferred from the dataset.



(b) An example where we only have access to a set of boundary points that give us information towards the shape of the boundary. In this case, we access the truncation boundary of the U.S. through a set of readily available coordinates.

Figure 5.1. Some scenarios where the form of a truncation boundary is unknown or unavailable.

Stein discrepancy. In the truncated setting, we refer to such a discrepancy as a Truncated Kernelised Stein Discrepancy (TKSD), which is computationally tractable with only access to points on the boundary. By minimising the TKSD, we obtain a truncated density estimator when the boundary's functional form is unavailable. In experiments, we show that despite the approximate nature of the Stein class, density models can be accurately estimated from truncated observations. We also provide a theoretical justification of the estimator consistency.

5.2 Background

The task of this chapter is truncated density estimation and therefore unnormalisable density estimation, which is summarised in chapter 2. The main literature related to this project is related to Stein discrepancies, section 2.2, specifically the Stein identity, given in definition 2.1. In experiments, we compare with the methodology from chapter 3; or Liu et al. (2022). The remainder of this section first gives a brief reminder of Stein identities and Stein classes, and then a more detailed overview of bd-KSD (Xu, 2022), which was discussed briefly in section 2.2.

5.2.1 Stein Classes

Recall definition 2.1, which states that a function family $\mathcal{F}^d$ is a *Stein class* if, for any $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$, $\boldsymbol{f} \in \mathcal{F}^d$,

$$\mathbb{E}_q[\mathcal{S}_q \boldsymbol{f}(\boldsymbol{x})] = 0 \tag{5.1}$$

for a Stein operator $\mathcal{S}_q : \mathcal{F}^d \to \mathbb{R}$, defined for some probability density $q(\boldsymbol{x})$ (Gorham and Mackey, 2015). We refer to equation (5.1) as the *Stein identity*. From the Stein identity, Stein discrepancies are defined between two densities $q(\boldsymbol{x})$ and $p(\boldsymbol{x}; \boldsymbol{\theta})$ (Barp et al., 2019), given by

$$\sup_{\boldsymbol{f} \in \mathcal{F}^d} \mathbb{E}_q[\mathcal{S}_{p_{\boldsymbol{\theta}}} \boldsymbol{f}(\boldsymbol{x})].$$

When the Stein class is an RKHS, this maximisation has a closed form solution, which gives rise to the Kernelised Stein Discrepancy (KSD) (Chwialkowski et al., 2016; Liu et al., 2016), given in equation (2.11), and minimising the KSD .

5.2.2 Bounded-Domain KSD (bd-KSD)

Motivated by performing goodness-of-fit testing on truncated domains, Xu (2022) propose a modified Langevin Stein operator, given by

$$\mathcal{T}_{p_{\boldsymbol{\theta}},h} \boldsymbol{g}(\boldsymbol{x}) = \sum_{j=1}^{d} \psi_{p_{\boldsymbol{\theta}},j}(\boldsymbol{x}) g_j(\boldsymbol{x}) h(\boldsymbol{x}) + \partial_{x_l}(g_j(\boldsymbol{x}) h(\boldsymbol{x})) \tag{5.2}$$

for $\boldsymbol{g} \in \mathcal{G}^d$, an RKHS, where $h(\boldsymbol{x})$ is a weighting function which is constructed so that the boundary condition,

$$\lim_{\boldsymbol{x} \to \boldsymbol{x}'} q(\boldsymbol{x}) h(\boldsymbol{x}) = 0$$

is satisfied, for $\boldsymbol{x}' \in \partial V$. For example, any function for which $h(\boldsymbol{x}') = 0 \; \forall \boldsymbol{x}' \in \partial V$ satisfies this criteria by definition. This boundary condition is required so that the integration by parts in deriving Stein identity (equation (5.1)) holds.

The bounded-domain Kernelised Stein Discrepancy (bd-KSD) is defined by replacing $\mathcal{S}_{p_{\boldsymbol{\theta}}}$ in the kernelised Stein discrepancy with $\mathcal{T}_{p_{\boldsymbol{\theta}},h}$. Making this modification, then the Stein discrepancy between $q(\boldsymbol{x})$ and $p(\boldsymbol{x}; \boldsymbol{\theta})$ gives the bd-KSD objective, given by

$$J_{\mathrm{bd-KSD}}(\boldsymbol{\theta}) := \sup_{\boldsymbol{g} \in \mathcal{G}^d} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}},h} \boldsymbol{g}(\boldsymbol{x})] = \| \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}},h} k(\boldsymbol{x}, \cdot)] \|_{\mathcal{G}^d}, \tag{5.3}$$

which is a closed form objective function. One can use equation (5.3) for goodness-of-fit testing, to measure the difference between a model density $p(\boldsymbol{x}; \boldsymbol{\theta})$, and its counterpart data density $q(\boldsymbol{x})$. Alternatively, this setting can be modified to allow for truncated density estimation, since equation (5.3) can be minimised with respect to the parameter vector $\boldsymbol{\theta}$. The density functions admitted are not required to have a tractable normalising constant, as the densities only appear through the score function.

The authors recommend a weighting function, h, given by a distance function, similar to chapter 3. For example, if the truncated domain is the unit ball, one can set $h(\boldsymbol{x}) = 1 - \|\boldsymbol{x}\|^b$ for a chosen power b.

5.3 Approximate Stein Classes

Let the support of the data density, $q(\boldsymbol{x})$, be a truncated domain $V \subset \mathbb{R}^d$, whose boundary is denoted by ∂V. So far, previous works have assumed a functional form of ∂V, so that a weighting function can be precisely designed and Stein's identity holds exactly. In our setting, the functional form of the truncation boundary is unavailable, making the design of a Stein class infeasible; we cannot design a class of functions with appropriate boundary conditions that satisfy Stein's identity exactly, i.e. equation (5.1) will never hold. However, the truncation boundary is known to us approximately as a set of boundary points, and so we can design a class of functions within which Stein's identity holds 'approximately', as we will show below.

We begin by first defining an approximate Stein class, which is a more *relaxed* Stein class of functions for which the Stein identity holds in an approximate sense. From this we also define an approximate Stein identity.

Definition 5.1 (Approximate Stein Identity)**.** Let $q(\boldsymbol{x})$ be a smooth probability density function on $V \subseteq \mathbb{R}^d$. $\widetilde{\mathcal{F}}_m^d$ is an *approximate Stein class* of $q(\boldsymbol{x})$ if for all $\tilde{\boldsymbol{f}} : \mathbb{R}^d \to \mathbb{R}^d$, $\tilde{\boldsymbol{f}} \in \widetilde{\mathcal{F}}_m^d$,

$$\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\mathcal{S}_{q,m}\tilde{\boldsymbol{f}}(\boldsymbol{x})] = O_P(\varepsilon_m), \tag{5.4}$$

where ε_m is a monotonically decreasing function of m, and $\mathcal{S}_{q,m}$ is a Stein operator that depends on m in some way. $O_P(\varepsilon_m)$ denotes a sequence indexed by m which is bounded in probability by ε_m.

We denote equation (5.4) as the approximate Stein identity, and the approximate Stein class $\widetilde{\mathcal{F}}_m^d$ can be written more simply as

$$\widetilde{\mathcal{F}}_m^d = \{\tilde{\boldsymbol{f}} : V \to \mathbb{R} \mid \mathbb{E}_q[\mathcal{S}_{q,m}\tilde{\boldsymbol{f}}(\boldsymbol{x})] = O_P(\varepsilon_m)\}. \tag{5.5}$$

The classical Stein class of functions given by equation (5.1) requires Stein's identity to hold, whereas this approximate Stein class equation (5.5) defines a set of functions for which $\mathbb{E}_q[\mathcal{S}_{q,m}\tilde{\boldsymbol{f}}(\boldsymbol{x})]$ is bounded by a decreasing sequence indexed by m. Similarly to the classical Stein discrepancy, we propose the use of the Langevin Stein operator. We aim to show this is a flexible class which can be used across various applications, of which we provide two examples below.

Example: Latent Variable Models. Kanagawa et al. (2019) presented a variation of Stein discrepancies for testing the goodness-of-fit of models with unobserved latent variables $\boldsymbol{z}$, where the density of $\boldsymbol{x}$ is given by $q(\boldsymbol{x}) = \int_{\mathbb{R}^d} q(\boldsymbol{x}|\boldsymbol{z})q(\boldsymbol{z})d\boldsymbol{z}$, and thus the score function can be written as

$$\begin{aligned}
\boldsymbol{\psi}_q(\boldsymbol{x}) = \frac{\nabla_{\boldsymbol{x}} q(\boldsymbol{x})}{q(\boldsymbol{x})} &= \int_{\mathbb{R}^d} \frac{\nabla_{\boldsymbol{x}}(q(\boldsymbol{x}|\boldsymbol{z})q(\boldsymbol{z}))}{q(\boldsymbol{x})} \frac{q(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{x}|\boldsymbol{z})} d\boldsymbol{z} \\
&= \int_{\mathbb{R}^d} \frac{q(\boldsymbol{x}|\boldsymbol{z})q(\boldsymbol{z})}{q(\boldsymbol{x})} \left[\frac{\nabla_{\boldsymbol{x}} q(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{x}|\boldsymbol{z})}\right] d\boldsymbol{z} \\
&= \int_{\mathbb{R}^d} q(\boldsymbol{z}|\boldsymbol{x}) \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z} \\
&= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\psi_q(\boldsymbol{x}|\boldsymbol{z})],
\end{aligned} \tag{5.6}$$

where $\psi_q(\boldsymbol{x}|\boldsymbol{z}) = \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}|\boldsymbol{z})$. To evaluate the expectation over $q(\boldsymbol{z}|\boldsymbol{x})$, Kanagawa et al. (2019) recommend approximating each element with a Monte Carlo estimate

$$\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\psi_{q,j}(\boldsymbol{x}|\boldsymbol{z})] = \frac{1}{m}\sum_{i=1}^{m} \psi_{q,j}(\boldsymbol{x}|\boldsymbol{z}_i) + O_P(\varepsilon_m), \tag{5.7}$$

where $\psi_{q,j}(\boldsymbol{x}|\boldsymbol{z}) = \partial_{x_j} \log q(\boldsymbol{x}|\boldsymbol{z})$ and we assume we have access to m samples $\{\boldsymbol{z}_i\}_{i=1}^m \sim q(\boldsymbol{z}|\boldsymbol{x})$, and $O_P(\varepsilon_m)$ is the Monte Carlo approximation error which decreases as the number of samples on the boundary, m, increases.

We show that this existing modification of Stein discrepancies give rise to an Approximate Stein Class. Suppose $\boldsymbol{f} \in \mathcal{F}^d$, where $\mathcal{F}^d$ is a regular Stein class. Stein's identity with the score function from equation (5.6) can be written as

$$\begin{aligned}
\mathbb{E}_q[\mathcal{T}_q \boldsymbol{f}(\boldsymbol{x})] &= \mathbb{E}_q\left[\sum_{j=1}^{d} \psi_{q,j}(\boldsymbol{x}) f_j(\boldsymbol{x}) + \partial_{x_j} f_j(\boldsymbol{x})\right] \\
&= \mathbb{E}_q\left[\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\psi_{q,j}(\boldsymbol{x}|\boldsymbol{z})\right] f_j(\boldsymbol{x}) + \partial_{x_j} f_j(\boldsymbol{x})\right].
\end{aligned}$$

Substituting equation (5.7) into the above gives

$$\begin{aligned}
\mathbb{E}_q[\mathcal{T}_q \boldsymbol{f}(\boldsymbol{x})] &= \mathbb{E}_q\left[\sum_{j=1}^{d}\left\{\left(\frac{1}{m}\sum_{i=1}^{m}\psi_{q,j}(\boldsymbol{x}|\boldsymbol{z}_i) + O_P(\varepsilon_m)\right) f_j(\boldsymbol{x}) + \partial_{x_j} f_j(\boldsymbol{x})\right\}\right] \\
&= \mathbb{E}_q\left[\sum_{j=1}^{d}\left\{\frac{1}{m}\sum_{i=1}^{m}\psi_{q,j}(\boldsymbol{x}|\boldsymbol{z}_i) f_j(\boldsymbol{x}) + \partial_{x_j} f_j(\boldsymbol{x}) + O_P(\varepsilon_m) f_j(\boldsymbol{x})\right\}\right] \\
&\overset{\text{(a)}}{=} \mathbb{E}_q\left[\sum_{j=1}^{d}\left\{\frac{1}{m}\sum_{i=1}^{m}\psi_{q,j}(\boldsymbol{x}|\boldsymbol{z}_i) f_j(\boldsymbol{x}) + \partial_{x_j} f_j(\boldsymbol{x})\right\}\right] + O_P(\varepsilon_m) \\
&\overset{\text{(b)}}{=} O_P(\varepsilon_m),
\end{aligned}$$

where equality (a) follows from the fact that the error in the Monte Carlo approximation is accounted for by the $O_P(\varepsilon_m)$ term, and equality (b) follows by definition of $\mathcal{F}^d$ being a Stein class. Therefore, $\mathbb{E}_q[\mathcal{T}_q \boldsymbol{f}(\boldsymbol{x})] = O_P(\varepsilon_m)$, and this latent variable modification uses an Approximate Stein Class.

5.4 KSD for Truncated Density Estimation

Let $q(\boldsymbol{x})$ be a smooth probability density function with truncated support $V \subset \mathbb{R}^d$ with boundary ∂V. When $q(\boldsymbol{x})$ has a truncated support, $\mathcal{G}^d$ is not a Stein class. To show this, let

$\boldsymbol{g} \in \mathcal{G}^d$, then Stein's identity can be written as

$$\begin{aligned}
\mathbb{E}_q\left[\mathcal{T}_q \boldsymbol{g}(\boldsymbol{x})\right] &= \int_V q(\boldsymbol{x}) \Big(\sum_{l=1}^{d} \psi_{p_{\boldsymbol{\theta}},l}(\boldsymbol{x}) g_l(\boldsymbol{x}) + \partial_{x_l} g_l(\boldsymbol{x}) \Big) d\boldsymbol{x} \\
&= \sum_{l=1}^{d} \left\{ \int_V \partial_{x_l} q(\boldsymbol{x}) g_l(\boldsymbol{x}) d\boldsymbol{x} + \int_V q(\boldsymbol{x}) \partial_{x_l} g_l(\boldsymbol{x}) d\boldsymbol{x} \right\} \\
&= \sum_{l=1}^{d} \left\{ \oint_{\partial V} q(\boldsymbol{x}) g_l(\boldsymbol{x}) \hat{\mathrm{u}}_l(\boldsymbol{x}) ds - \int_V q(\boldsymbol{x}) \partial_{x_l} g_l(\boldsymbol{x}) d\boldsymbol{x} + \int_V q(\boldsymbol{x}) \partial_{x_l} g_l(\boldsymbol{x}) d\boldsymbol{x} \right\} \\
&= \oint_{\partial V} q(\boldsymbol{x}) \sum_{l=1}^{d} g_l(\boldsymbol{x}) \hat{\mathrm{u}}_l(\boldsymbol{x}) ds, \qquad (5.8)
\end{aligned}$$

where $\oint_{\partial V}$ is the surface integral over the boundary ∂V, $\hat{\mathrm{u}}_1(\boldsymbol{x}), \ldots, \hat{\mathrm{u}}_d(\boldsymbol{x})$ are the unit outward normal vectors on ∂V and ds is the surface element on ∂V.

To satisfy the Stein identity, equation (5.8) must equal zero. In the untruncated setting, equation (5.8) is zero due to the boundary condition on $q(\boldsymbol{x})$, but in the truncated setting the density is significantly nonzero on the boundary at ∂V. Prior methods such as Xu (2022) choose a pre-defined weighting function for which equation (5.8) is exactly zero at the boundary.

We instead aim to show that equation (5.8) is $O_P(\varepsilon_m)$ under an approximate Stein class, detailed in the remainder of this section.

5.4.1 Approximate Stein Class for Truncated Densities

Let us first consider a setting where the boundary ∂V is known analytically. Define a modified product RKHS as

$$\mathcal{G}_0^d = \{\boldsymbol{g} \in \mathcal{G}^d \mid \boldsymbol{g}(\boldsymbol{x}') = \boldsymbol{0}\ \forall \boldsymbol{x}' \in \partial V, \|\boldsymbol{g}\|_{\mathcal{G}^d}^2 \leq 1\}. \qquad (5.9)$$

Lemma 5.2. *Let $q(\boldsymbol{x})$ be a smooth density supported on V. For any $\boldsymbol{g} \in \mathcal{G}_0^d$, then*

$$\mathbb{E}_q[\mathcal{T}_q \boldsymbol{g}(\boldsymbol{x})] = 0. \qquad (5.10)$$

For proof, see section 5.8.1. Similar to the classical Stein identity, the proof follows from applying a simple integration by parts. In essence, provided the conditions in $\mathcal{G}_0^d$ hold for the function $\boldsymbol{g}$, then equation (5.8) is zero. Therefore lemma 5.2 shows that $\mathcal{G}_0^d$ is a proper Stein class of q. $\mathcal{G}_0^d$ defines a large class of functions, for which the proposed operator by Xu (2022) given in equation (5.2) is one such example of a function from this family.

Let us now consider the setting where ∂V is not known exactly. The information about the boundary is provided by $\widetilde{\partial V} = \{\boldsymbol{x}'_{i'}\}_{i'=1}^m$, a finite set of points randomly sampled from ∂V. Define

$$\mathcal{G}_{0,m}^d = \{\boldsymbol{g} \in \mathcal{G}^d \mid \boldsymbol{g}(\boldsymbol{x}') = \boldsymbol{0}\ \forall \boldsymbol{x}' \in \widetilde{\partial V}, \|\boldsymbol{g}\|_{\mathcal{G}^d}^2 \leq 1\}, \qquad (5.11)$$

which can be considered as an approximate version of $\mathcal{G}_0^d$ using the finite set $\widetilde{\partial V}$. The benefit of using $\mathcal{G}_{0,m}^d$ over $\mathcal{G}_0^d$ is that this class can be constructed using only 'partial' boundary information, i.e., $\widetilde{\partial V}$, without knowing the explicit expression of ∂V.

First, we show specific properties of the relationship between $\mathcal{G}_{0,m}^d$ and $\mathcal{G}_0^d$.

Lemma 5.3. *Let $\boldsymbol{g} \in \mathcal{G}_0^d$ and $\tilde{\boldsymbol{g}} \in \mathcal{G}_{0,m}^d$. Assume that $\widetilde{\partial V}$ is ε_m-dense in ∂V with some probability. Further assume that g_l and $\tilde{g}_l$ are C-Lipschitz continuous for all $l = 1, \ldots, d$. Then*

$$|g_l(\boldsymbol{x}') - \tilde{g}_l(\boldsymbol{x}')| = O_P(\varepsilon_m)$$

for any $\boldsymbol{x}' \in \partial V$.

For proof, see section 5.8.2. The proof follows from applying the Lipschitz continuous property on g_l and $\tilde{g}_l$ and then the triangle inequality. Lemma 5.3 establishes a connection between $\mathcal{G}_0^d$ and $\mathcal{G}_{0,m}^d$, relying on the assumption that $\widetilde{\partial V}$ is ε_m-dense in ∂V. We now show under some mild conditions this assumption holds with high probability.

Proposition 5.4. *Assume $\{\boldsymbol{x}_i'\}_{i=1}^m$ are samples drawn from the uniform distribution defined on ∂V. Let $\mathrm{L}(V)$ denote the $(d-1)$-surface area of a bounded domain $V \subset \mathbb{R}^d$ and $\mathrm{L}(V) < \infty$. Let $B_{\varepsilon_m}(\boldsymbol{x}')$ denote a ball of radius ε_m centred on $\boldsymbol{x}'$, and let $\xi(d) = \pi^{d/2}/\Gamma(\frac{d}{2}+1)$. For all ε_m such that*

$$\varepsilon_m \geq \left(\frac{\mathrm{L}(V)}{\xi(d)} \left[1 - 0.05^{1/m} \right] \right)^{1/d}. \tag{5.12}$$

we have

$$\mathbb{P}\left(\widetilde{\partial V} \text{ is } \varepsilon_m\text{-dense in } \partial V \right) = 0.95.$$

For proof, see section 5.8.3. Equation (5.12) shows the relationship between m and ε_m, which corresponds to how 'close' $\widetilde{\partial V}$ is to ∂V.

Remark 5.5. *Our numerical investigation (section 5.5.3) shows that the bound on ε_m, as defined in equation* (5.12)*, is a decreasing function of m, and is not sensitive to the value of $\mathrm{L}(V)$.*

Proposition 5.4 shows that lemma 5.3 holds with high probability, which in turn enables us to show that indeed $\mathcal{G}_{0,m}^d$ is an approximate Stein class.

Theorem 5.6. *Assume the conditions specified in lemma 5.3 hold. Let $q(\boldsymbol{x})$ be a smooth density supported on V. For any $\tilde{\boldsymbol{g}} \in \mathcal{G}_{0,m}^d$, then*

$$\mathbb{E}_q[\mathcal{T}_q \tilde{\boldsymbol{g}}(\boldsymbol{x})] = O_P(\varepsilon_m). \tag{5.13}$$

The proof can be found in section 5.8.4, and follows from integration by parts, then applying lemma 5.3. This result shows that $\mathcal{G}_{0,m}^d$ is an approximate Stein class, which paves the way for designing a new type of Stein divergence that measures differences between two distributions when their domain V is truncated.

5.4.2 Truncated Kernelised Stein Discrepancy (TKSD) and Density Estimation

Let $q(\boldsymbol{x})$ and $p(\boldsymbol{x};\boldsymbol{\theta})$ be two smooth densities supported on V. We can construct a Stein discrepancy measure, called the truncated Kernelised Stein Discrepancy (TKSD), given by

$$J_{\mathrm{TKSD}}(\boldsymbol{\theta}) := \sup_{\boldsymbol{g}\in\mathcal{G}^d_{0,m}} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}}\boldsymbol{g}(\boldsymbol{x})]. \tag{5.14}$$

Similarly to classical SD and KSD, TKSD can still be intuitively thought as the maximum violation of Stein's identity with respect to an *approximate* Stein class $\mathcal{G}^d_{0,m}$. It can be used to distinguish two distributions when their domain V is truncated, but the boundary is not known analytically.

$J_{\mathrm{TKSD}}(\boldsymbol{\theta})$ serves as the discrepancy measure between densities, for which, we propose to estimate a truncated density function by minimising this discrepancy. This is not straightforward as the supremum in equation (5.14) is a constrained optimisation problem with constraints given by those in the definition of $\mathcal{G}^d_{0,m}$. However, we can show that there is an analytic solution to this constrained optimisation problem by solving it as a Lagrangian dual problem, given in the following theorem.

Theorem 5.7. *$J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2$ can be written as*

$$\sum_{l=1}^{d} \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{y}\sim q(\boldsymbol{x})}\left[u_l(\boldsymbol{x},\boldsymbol{y}) - \mathbf{v}_l(\boldsymbol{x})^\top(\mathbf{K}')^{-1}\mathbf{v}_l(\boldsymbol{y})\right] \tag{5.15}$$

where $u_l(\boldsymbol{x},\boldsymbol{y})$ is given by

$$\begin{aligned} u_l(\boldsymbol{x},\boldsymbol{y}) = {} & \psi_{p_{\boldsymbol{\theta}},l}(\boldsymbol{x})\psi_{p_{\boldsymbol{\theta}},l}(\boldsymbol{y})k(\boldsymbol{x},\boldsymbol{y}) + \psi_{p_{\boldsymbol{\theta}},l}(\boldsymbol{x})\partial_{y_l}k(\boldsymbol{x},\boldsymbol{y}) \\ & + \psi_{p_{\boldsymbol{\theta}},l}(\boldsymbol{y})\partial_{x_l}k(\boldsymbol{x},\boldsymbol{y}) + \partial_{x_l}\partial_{y_l}k(\boldsymbol{x},\boldsymbol{y}), \end{aligned} \tag{5.16}$$

$\mathbf{v}_l(\boldsymbol{z}) = \psi_{p,l}(\boldsymbol{z})\boldsymbol{\phi}^\top_{\boldsymbol{z},\mathbf{x}'} + (\partial_{z_l}\boldsymbol{\phi}_{\boldsymbol{z},\mathbf{x}'})^\top$, $\boldsymbol{\phi}_{\boldsymbol{z},\mathbf{x}'} = [k(\boldsymbol{z},\boldsymbol{x}'_1),\ldots,k(\boldsymbol{z},\boldsymbol{x}'_m)]$ is a vector of kernel evaluations on boundary points, $\boldsymbol{\phi}_{\mathbf{x}'} = [k(\boldsymbol{x}'_1,\cdot),\ldots,k(\boldsymbol{x}'_m,\cdot)]^\top$ and $\mathbf{K}' = \boldsymbol{\phi}_{\mathbf{x}'}\boldsymbol{\phi}^\top_{\mathbf{x}'}$.

The proof can be found in section 5.8.5. This result gives a closed form solution to the supremum in $J_{\mathrm{TKSD}}(\boldsymbol{\theta})$ for which the constraints in $\mathcal{G}^d_{0,m}$ are enforced on a finite set of points in $\mathbb{R}^d$. Intuitively we can note that equation (5.15) can be decomposed into the 'KSD part', given by the $u_l(\boldsymbol{x},\boldsymbol{y})$ term, which is identical to equation (2.12), and the 'truncated part', given by $\mathbf{v}_l(\boldsymbol{x})^\top(\mathbf{K}')^{-1}\mathbf{v}_l(\boldsymbol{y})$, which comes from solving for the Lagrangian dual parameter.

Next, we show that the TKSD can be approximated with samples from $q(\boldsymbol{x})$.

Theorem 5.8. *Let $\{\boldsymbol{x}_i\}_{i=1}^n$ be a set of samples from $q(\boldsymbol{x})$, then*

$$\hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta})^2 = \frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n} h(\boldsymbol{x}_i,\boldsymbol{x}_j), \tag{5.17}$$

is an unbiased estimate of $J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2$*, where*

$$h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{l=1}^{d} u_l(\boldsymbol{x}_i, \boldsymbol{x}_j) - \mathbf{v}_l(\boldsymbol{x}_i)^\top (\mathbf{K}')^{-1} \mathbf{v}_l(\boldsymbol{x}_j), \tag{5.18}$$

assuming that $\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{y}\sim q(\boldsymbol{x})}[h(\boldsymbol{x}, \boldsymbol{y})^2] \leq \infty$.

The proof follows directly from the definition of a U-statistic (Serfling, 2009). Additionally, we could define a biased estimate of $J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2$ via a V-statistic

$$\hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta})^2 = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} h(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{5.19}$$

The U-statistic and V-statistic for TKSD allow us to evaluate $J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2$ via n samples from $q(\boldsymbol{x})$. In empirical experiments, the V-statistic seems to give better performance overall compared to the U-statistic.

Finally, we define our proposed estimator for unnormalised truncated density by minimising $\hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta})^2$ over the density parameter $\boldsymbol{\theta}$.

$$\hat{\boldsymbol{\theta}}_{n,m} := \arg\min_{\boldsymbol{\theta}} \hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta})^2. \tag{5.20}$$

In the next section, we study the theoretical properties of equation (5.20).

5.4.3 Consistency Analysis

In this section we aim to show that TKSD is a consistent estimator for large values of m. Since $\hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta})^2$ is a function of m, for the simplicity of the theorem, let us study equation (5.20) at the limit of m. Let $\hat{L}(\boldsymbol{\theta}) := \lim_{m\to\infty} \hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta})^2$ and define

$$\hat{\boldsymbol{\theta}}_n := \arg\min_{\boldsymbol{\theta}} \hat{L}(\boldsymbol{\theta}).$$

We now prove that $\hat{\boldsymbol{\theta}}_n$ converges to the true parameter under mild conditions:

Assumption 5.9 (Accurate Boundary Prediction)**.** *Let*

$$\boldsymbol{t} := \mathbb{E}_{\boldsymbol{y}}\left[(\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top [\nabla_{\boldsymbol{\theta}} \psi_{p,l}(\boldsymbol{y}))^\top]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}\right] \in \mathbb{R}^{m\times \dim(\boldsymbol{\theta})},$$
$$\boldsymbol{t}(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{y}}\left[(\boldsymbol{\varphi}_{\boldsymbol{y},\boldsymbol{x}}^\top [\nabla_{\boldsymbol{\theta}} \psi_{p,l}(\boldsymbol{y}))^\top]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}\right] \in \mathbb{R}^{\dim(\boldsymbol{\theta})}.$$

Assume the following holds:

$$\left[\oint_{\partial V} q(\boldsymbol{x}) \boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'} (\mathbf{K}')^{-1} \boldsymbol{t} \hat{\mathrm{u}}_l(\boldsymbol{x}) ds\right]_i = \left[\oint_{\partial V} q(\boldsymbol{x}) \boldsymbol{t}(\boldsymbol{x}) \hat{\mathrm{u}}_l(\boldsymbol{x}) ds\right]_i + O_P(\hat{\varepsilon}_m), \quad \forall i \in \{1, \ldots, \dim(\boldsymbol{\theta})\}, l \in \{1, \ldots, d\} \tag{5.21}$$

where $\hat{\varepsilon}_m$ is a positive decaying sequence with respect to m and $\lim_{m\to\infty} \hat{\varepsilon}_m = 0$.

One can see that $\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}(\mathbf{K}')^{-1}\boldsymbol{t}$ is the kernel least-square regression prediction of $\boldsymbol{t}(\boldsymbol{x}')$, for a $\boldsymbol{x}' \in \partial V$. Assumption 5.9 essentially states that the least squares 'trained' on our boundary samples, $\{\boldsymbol{x}'_{i'}\}_{i'=1}^m$, should be asymptotically accurate in terms of a testing error computed over a surface integral as m increases.

Assumption 5.10. *The smallest eigenvalue of the Hessian of $\hat{L}(\boldsymbol{\theta})$ is lower bounded, i.e.,*

$$\lambda_{\min}\left[\nabla^2_{\boldsymbol{\theta}}\hat{L}(\boldsymbol{\theta})\right] \geq \Lambda_{\min} > 0$$

with high probability.

In fact, we could make the same assumption on the population version of the objective function, and the convergence of the sample hessian matrix would guarantee that assumption 5.10 holds with high probability. To simplify our theoretical statement we stick to the simpler version.

Theorem 5.11. *Suppose there exists a unique $\boldsymbol{\theta}^\star$ such that $q = p_{\boldsymbol{\theta}^\star}$, assumption 5.9 and assumption 5.10 hold, then*

$$\|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^\star\| = O_P\left(\frac{1}{\sqrt{n}}\right).$$

For proof, see section 5.8.6. This result shows that although TKSD is an approximate version of KSD, the density estimator derived from TKSD is still a consistent estimator. We empirically verify this consistency in section 5.5.4.

5.5 Supporting Experiments

Before testing the method on various real-world style examples, we first show empirical results which confirm some of the key results outlined in section 5.4.

5.5.1 Experimental Details

The following information is relevant to all the experiments in this section.

Computational Considerations We first give some computational details on TKSD, the following information is related to the model selection process, the scalability of the method and a few implementation 'tricks' to improve efficiency of the method.

- TKSD requires the selection of hyperparameters: the number of boundary points, m, the choice of kernel function, k, and the corresponding kernel hyperparameters. For this work, we focus on the Gaussian kernel, $k(\boldsymbol{x},\boldsymbol{y}) = \exp\{-(2\sigma^2)^{-1}\|\boldsymbol{x}-\boldsymbol{y}\|^2\}$, and the bandwidth parameter σ is chosen heuristically as the median of pairwise distances on the data matrix.

- The inversion of $\mathbf{K}'$ is the largest computational expense when it comes to evaluating the U-statistic or V-statistic ((5.18) and (5.19) respectively). Since $\mathbf{K}'$ is an $m \times m$ matrix, there is a trade-off between accuracy (larger m) and computational expense (smaller m). In experiments, we let m scale linearly with d^2.

  For computationally simplicity we often invert $\mathbf{K}' + \epsilon \mathbf{I}_m$ instead, where $\epsilon > 0$ is small. Additionally, $\mathbf{K}' + \epsilon \mathbf{I}_m$ is symmetric and positive definite, so we can exploit its Cholesky decomposition to make the inversion faster and more stable.

  Overall, this inversion looks like

$$(\mathbf{K}')^{-1} \approx (\mathbf{K}' + \epsilon \mathbf{I}_m)^{-1} = \mathbf{L}^{-1}(\mathbf{L}^{-1})^\top,$$

  where $\mathbf{L}$ is the corresponding lower triangular matrix from the Cholesky decomposition of $\mathbf{K}' + \epsilon \mathbf{I}_m$. The inversion of $\mathbf{L}$, a lower triangular matrix, requires half as many operations as inverting $\mathbf{K} + \epsilon \mathbf{I}_m$ directly (Krishnamoorthy and Menon, 2013).

- We make use of methods presented in Jitkrittum et al. (2017) for fast computation when constructing all kernel matrices in Python.

Methods Across this section and section 5.6, we will be consistently comparing the following methods

- TKSD: our method as described in section 5.4, using a uniformly randomly sampled $\widetilde{\partial V}$.
- *TruncSM* (exact): the implementation described in chapter 3 (Liu et al., 2022), where the distance function is computed exactly using a known boundary.
- *TruncSM* (approximate): the implementation described in chapter 3 (Liu et al., 2022) with distance function *approximated* by

$$\min_{\boldsymbol{x}' \in \widetilde{\partial V}} \|\boldsymbol{x} - \boldsymbol{x}'\|_\alpha^\gamma, \tag{5.22}$$

  for each dataset point $\boldsymbol{x}$. γ and α can be chosen based on the application, for which we use $\gamma = 1$ and $\alpha = 2$. This method uses the same corresponding $\widetilde{\partial V}$ as given to TKSD. This may not provide an accurate approximation to the true distance function when m is small. This method is likely used when the boundary's functional form is unknown.
- bd-KSD (exact): a variation of the implementation by Xu (2022) for density estimation, where the distance function is computed exactly using a known boundary.
- bd-KSD (approximate) a variation of the implementation by Xu (2022) for density estimation, where the distance function is *approximated* via equation (5.22). The same criteria as above apply.

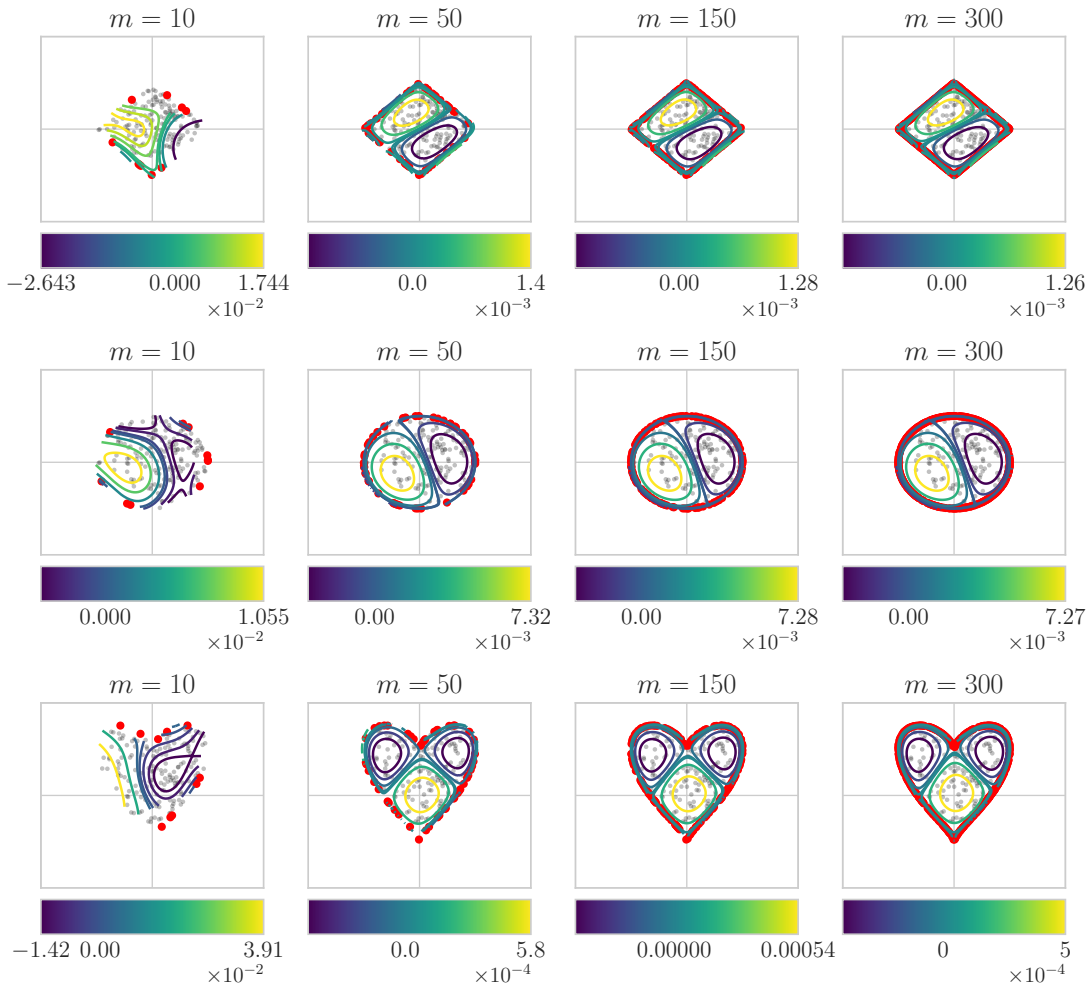


Figure 5.2. Contour lines of the estimated $\tilde{g}_1$ output by TKSD across different values of m and differently shaped truncation boundaries: the ℓ^1 ball (top), the ℓ^2 ball (middle) and a heart shape (bottom). Red points are all m points in $\widetilde{\partial V}$ and grey points are samples from the truncated dataset. Note that we plot only the first dimension of $\tilde{\boldsymbol{g}}$ (i.e. g_1), but we observe the same pattern with the second dimension.

5.5.2 Empirical Convergence of $\tilde{g}$ to g

In lemma 5.3 we proved that under some conditions, for any $\boldsymbol{g} \in \mathcal{G}_0^d$ and $\tilde{\boldsymbol{g}} \in \mathcal{G}_{0,m}^d$, both evaluated on $\boldsymbol{x}' \in \partial V$, the difference between these evaluations is bounded in probability by a decreasing sequence of m with high probability. In this section we aim to show that $\tilde{\boldsymbol{g}}$ converges to a specific function as m increases, which we hypothesise is the 'true' $\boldsymbol{g} \in \mathcal{G}_0^d$.

Figure 5.2 demonstrates empirical convergence of $\tilde{\boldsymbol{g}}$ as m increases for a given dataset. The experiment is setup as follows: we simulate points from a $\mathcal{N}(\mathbf{0}_2, \boldsymbol{I}_2)$ distribution (a unit Multi-

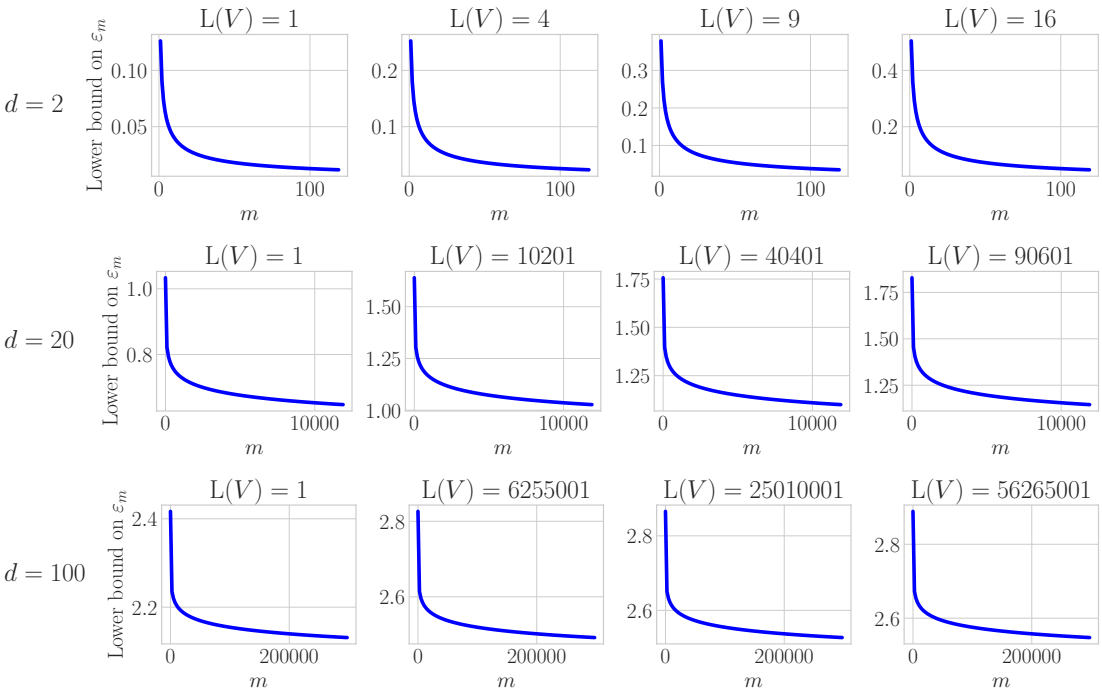


Figure 5.3. Lower bound on ε_m (given in equation (5.12)) against m, the number of finite boundary points, plotted for different values of fixed dimension d and boundary 'size' $L(V)$, which scales quadratically.

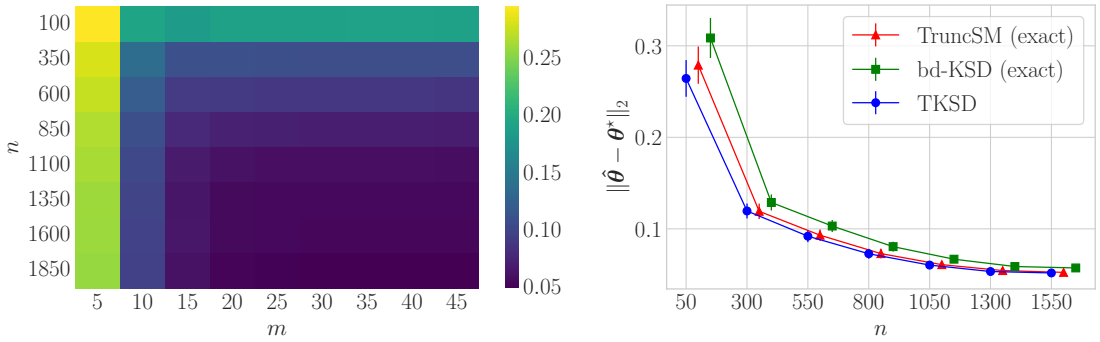
variate Normal distribution in 2D), then truncate data points to within the shape of the boundary based on location, and repeat until we acquire $n = 150$ truncated points. For each value of m, we minimise the TKSD objective to estimate $\boldsymbol{\theta} := \mathbf{0}_2$, the two-dimensional vector of zeros, and plug in the estimated $\hat{\boldsymbol{\theta}}$ to the formula for $\tilde{\boldsymbol{g}}$.

For lower values of m, there are not enough boundary points to enforce the constraint well on $\tilde{\boldsymbol{g}}$, and the approximate Stein identity has a high error, and the estimator is poor. The evaluations of $\tilde{\boldsymbol{g}}$ across the space do not match the $\tilde{\boldsymbol{g}}$ output for higher values of m. As m increases, $\tilde{\boldsymbol{g}}$ begins to converge to a particular shape, and the difference between function evaluations for $m = 150$ and $m = 300$ is small.

5.5.3 Evaluating increasing values of m in proposition 5.4

In figure 5.3 we show that ε_m, as defined in proposition 5.4, is bounded below by a decreasing function of m. In this example, we plot the value of the lower bound in equation (5.12) against m, the number of samples on the boundary set, and $\mathrm{L}(V)$, the $(d-1)$-surface area of the bounded domain V, which can be considered as a proxy to measure the complexity of the boundary.

We let $\mathrm{L}(V)$ scale quadratically with dimension d, and treat it as a constant. m scales linearly, and measure the effect on the lower bound of ε_m. As $\mathrm{L}(V)$ increases, the bound on ε_m increases



(a) Estimation error as n and m increases for TKSD only, for a fixed $d = 2$.

(b) Estimation error and standard error as n increases, for a fixed $m = 32$.

Figure 5.4. Mean estimation error as either m or n increase.

slightly. As m increases, the bound on ε_m decreases rapidly. Even at extremely high values of ε_m, relatively small values of m are required to shrink the bound. This result highlights how to choose m in experiments - as m increases, the 'denseness' of $\widetilde{\partial V}$ in ∂V also increases, as expected. Whilst larger values of m will always be better, there is a less pronounced effect when m becomes significantly larger than zero. As discussed in section 5.5.1, larger m comes with computational cost, and therefore this experiment shows that it may be beneficial to choose smaller values of m. This is explored further in the following section.

5.5.4 Empirical Consistency

We verify that (5.20) is consistent estimator via empirical experiments. Note that for consistency as proven in theorem 5.11, we require taking the limit as $m \to \infty$, after which the estimator is consistent for n. We show consistency as m and n increase empirically in figure 5.4, for a simple experiment setup, similar to the setup from section 5.6.2. We simulate a set of n samples given by $\{\boldsymbol{x}_i\}_{i=1}^n$, where each data point is sampled as follows. First,

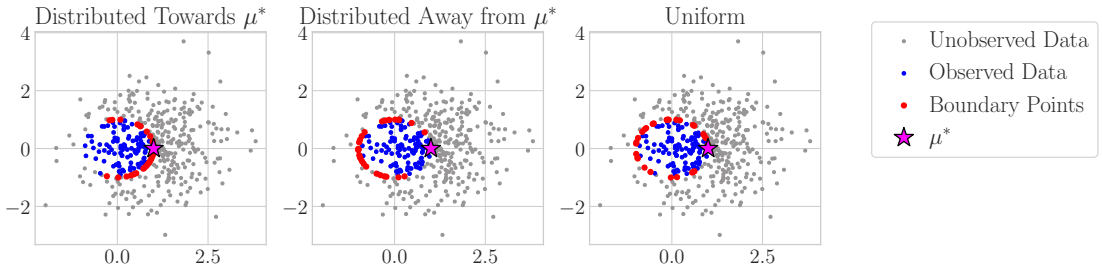
$$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}^\star, I_2),$$

where $\boldsymbol{\mu}^\star = [0.5, 0.5]^\top$, and I_2 is considered known. This sample is discarded if it is outside of the ℓ^2 ball, i.e. the domain

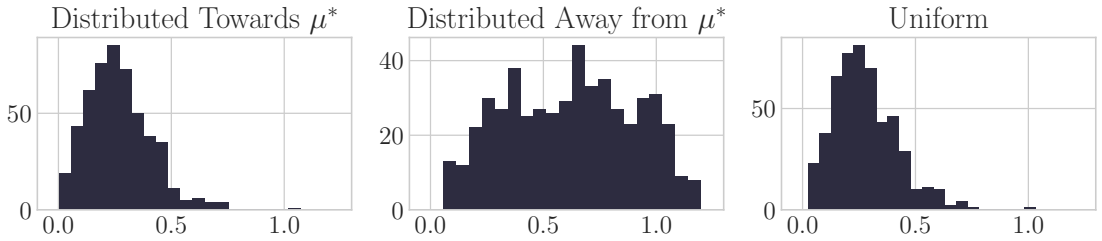
$$V = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_2 \leq 1\}.$$

We repeat the process of sampling and potentially discarding until we reach n truncated samples.

The aim of the experiment is to produce an estimate, $\hat{\boldsymbol{\theta}} := \hat{\boldsymbol{\mu}}$, of the mean, $\boldsymbol{\theta}^\star := \boldsymbol{\mu}^\star$. Averaged over 64 trials, figure 5.4 shows plots of the mean estimation error, given by $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|_2$, for the three different methods. If the estimation error decreases, this indicates the estimate converging to the true value. In figure 5.4(a), we show that as both n and m increase, the estimation error for TKSD decreases towards zero. In figure 5.4(b), for a fixed m, we show that the rate of



(a) Example experiment setup.



(b) Frequency of estimation errors ($\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|_2$).

Figure 5.5. Quantification of effect of distribution of boundary samples, for three types of skewed boundary samples: towards the centre, away from the centre and uniformly distributed.

convergence as n increases for TKSD matches that of bd-KSD and *TruncSM*. Similar to the experiment in section 5.5.3, figure 5.4(a) shows that as m becomes significantly larger than zero, there are diminishing returns on estimation accuracy.

5.5.5 Quantifying Effect of Boundary Point Distribution

We test whether the effect of boundary point sampling distribution has an effect on the robustness of TKSD. With the exception of proposition 5.4 (which assumes uniformity of boundary samples), we make no assumption on the distribution of samples from the boundary in deriving the theory for TKSD. However, we hypothesise that these samples need to be 'representative' in some way of the full boundary ∂V. To test this hypothesis, consider an experiment setup where $n_0 = 400$ samples are simulated via

$$\{\boldsymbol{x}\}_{i=1}^{n_0} \sim \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{I}_2),$$

where $\boldsymbol{\mu}^\star = [1, 1]^\top$. Any samples outside of the unit ℓ_2 ball around the origin are discarded. Those samples that remain are used for estimating the parameter $\hat{\theta}^\star := \boldsymbol{\mu}^\star$. We let $m = 30$ be fixed, and use TKSD to prduce an estimate $\hat{\boldsymbol{\theta}} := \hat{\boldsymbol{\mu}}$ under three scenarios:

1. **Distributed towards $\boldsymbol{\mu}^\star$** Boundary points are distributed towards $\boldsymbol{\mu}^\star$, i.e. samples from $\widetilde{\partial V}$ are closer to the centre of the dataset.

2. **Distributed away from $\boldsymbol{\mu}^\star$** Boundary points are distributed away from $\boldsymbol{\mu}^\star$, i.e. samples from $\widetilde{\partial V}$ are closer to the edge of the dataset.

3. **Uniform** Boundary points are sampled uniformly across ∂V.

Figure 5.5 shows the results of this experiment: figure 5.5(a) gives a visual representation of the three scenarios, and figure 5.5(b) shows the estimation errors, given by $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|_2$, across 512 trials. These are shown individually for all three scenarios described above. The aim for this experiment is to test whether one scenario provides a lower error on average overall.

Scenario 1 and 3 provide comparable error distributions which are skewed towards lower values of estimation error. Scenario 3 has a broader range of estimation errors, and on average has higher estimation errors. This implies that estimation performance is improved if either the boundary is 'covered' fully, or the boundary samples are representative of where the truncation effect is most significant.

5.6 Synthetic Experiments and Benchmarks Results

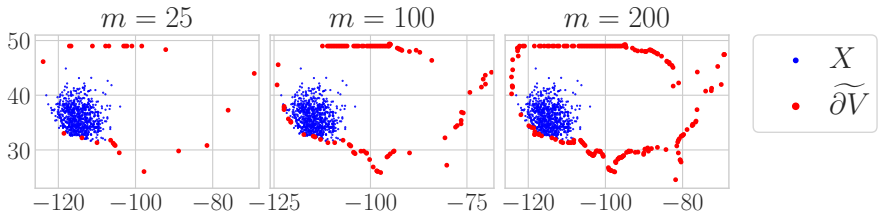
To show the validity of our proposed method, we experiment on benchmark settings against *TruncSM* and an adaptation of bd-KSD for truncated density estimation.

5.6.1 Density Truncated by the Boundary of the United States

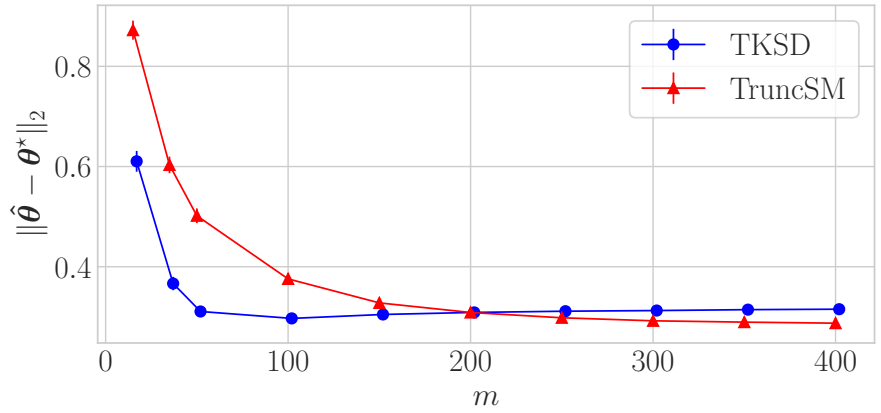
Let us consider the complicated boundary of the United States (U.S.). Let V be the interior of the U.S. and let $\widetilde{\partial V}$ be a set of coordinates (longitude and latitude) which define the country's borders. The boundary of the U.S. is a highly irregular shape, and as such, there is no explicit expression of the boundary in this case. *TruncSM* and bd-KSD must use an approximate distance function (given by equation (5.22)), whereas TKSD can readily use the set of coordinates that give rise to the boundary.

The experiment is set up as follows. Let $\boldsymbol{\theta}^\star := \boldsymbol{\mu}^\star = [-115, 35]$ and $\boldsymbol{\Sigma} = 10 \cdot I_d$. Samples are simulated from $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$ and those samples which are outside of the interior of the U.S. are discarded. We repeat sampling until we reach $n = 400$ points. Assuming $\boldsymbol{\Sigma}$ is known, we estimate $\hat{\boldsymbol{\theta}} := \hat{\boldsymbol{\mu}}$ using TKSD and compare it to the same estimation using *TruncSM*. We also vary m by uniformly sampling from the perimeter of the U.S., demonstrated in figure 5.6(a).

Figure 5.6(b) shows the mean and standard error of the estimation error between $\boldsymbol{\theta}^\star$ and $\hat{\boldsymbol{\theta}}$, measured over 256 trials for each value of m. *TruncSM* improves with higher values of m as the approximate distance function increases in accuracy, whilst TKSD performs significantly better with fewer boundary points.



(a) Example of increasing the number of boundary points m.



(b) Mean estimation error with standard error for TKSD and *TruncSM* as m increases, averaged across 256 trials.

Figure 5.6. Density estimation when the truncation boundary is the border of the U.S.

5.6.2 Estimation Error Across Dimension

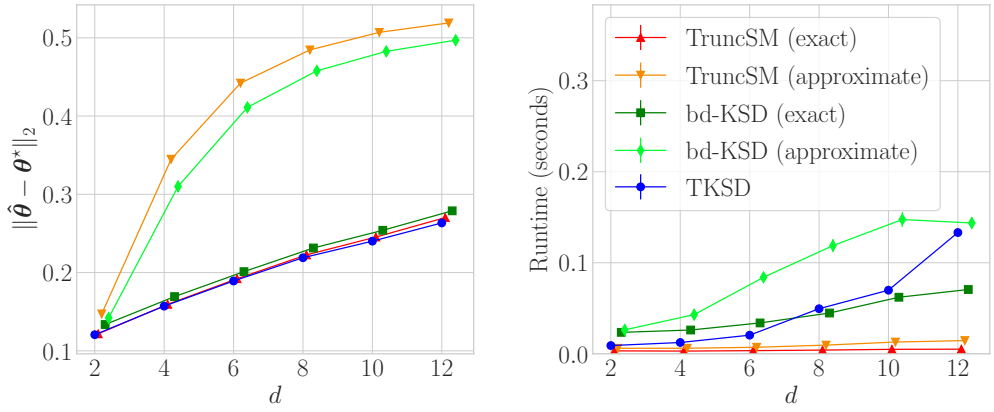
Consider an experiment setup where samples are restricted within the ℓ^c ball of radius r, i.e. the domain is given by

$$V = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_c \leq r\}.$$

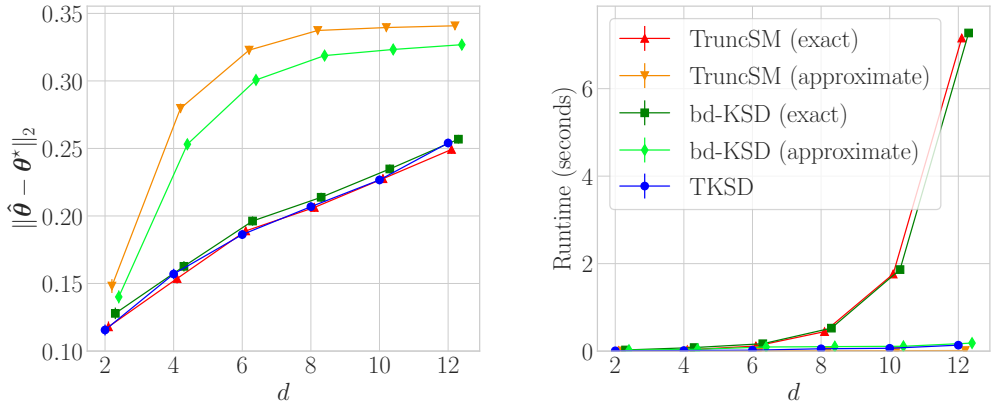
Samples are simulated from $\mathcal{N}(\boldsymbol{\mu}_d^\star, \boldsymbol{I}_d)$, where $\boldsymbol{\theta}^\star := \boldsymbol{\mu}^\star = \mathbf{1}_d \cdot 0.5$, and those samples outside of the ℓ_c ball are discarded. We repeat the process of sampling and then potentially discarding until we reach $n = 300$ samples.

The experiment is repeated for $c = 1$ and $c = 2$, for which we choose radii values of $r = d^{0.98}$ and $r = d^{0.53}$ for the ℓ^1 ball and ℓ^2 ball respectively. These values were chosen via trial and error such that the percentage of points simulated from the given Normal distribution remaining after truncation did not vary significantly across dimension. These percentages are plotted across different values of d in figure 5.8.

We estimate $\boldsymbol{\theta}^\star$ with $\hat{\boldsymbol{\theta}}$, and measure the ℓ^2 estimation error, $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|_2$, as d increases. Each boundary point $\boldsymbol{x}' \in \widetilde{\partial V}$ is simulated from $\mathcal{N}(\mathbf{0}_d, \boldsymbol{I}_d)$, normalized by $\|\boldsymbol{x}'\|_c$, and multiplied by r, resulting in a set of random points on the boundary.



(a) ℓ^2 ball.

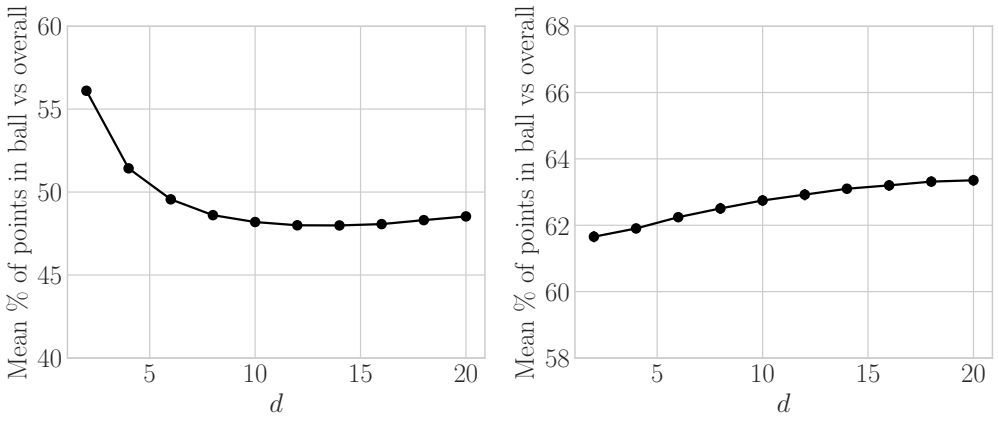


(b) ℓ^1 ball.

Figure 5.7. Mean estimation error averaged across 256 seeds, with standard error bars, as dimension d increases (left) and runtime for each method (right).

Figure 5.7 shows the error and computation time across a range of different methods. For the ℓ^2 ball, TKSD marginally outperforms all competitors across all dimensions, at only a slight computational expense. For the ℓ^1 ball, TKSD, bd-KSD and *TruncSM* have similar estimation errors across all dimensions. However, *TruncSM* (exact) and bd-KSD (exact) have an increasing computation time due to the costly evaluation of the distance function. In this implementation, we follow the advice of Liu et al. (2022), Section 7, where the distance function to the ℓ^1 ball is calculated via a closed-form expression, for which the computational complexity increases combinatorically with dimension.

TruncSM (approximate) and bd-KSD (approximate) have significantly higher estimation error than other methods across all benchmarks. TKSD is able to achieve the same level of accuracy as the exact methods, using the same finite set of boundary points, $\widetilde{\partial V}$.



(a) For the ℓ^2 ball of radius $d^{0.53}$.

(b) For the ℓ^1 ball of radius $d^{0.98}$.

Figure 5.8. Mean percentage of data points simulated from $\mathcal{N}(\boldsymbol{\mu}_d^\star, \boldsymbol{I}_d)$ which remain after truncation as dimension d increases.

5.6.3 Gaussian Mixture

As an additional experiment to show the capability of the method, we test on a more complex problem, estimating several means of a Gaussian Mixture distribution. The estimation task is as follows. Fix $d = 2$ and $m = 200$. Let all of the mixture modes be given as follows,

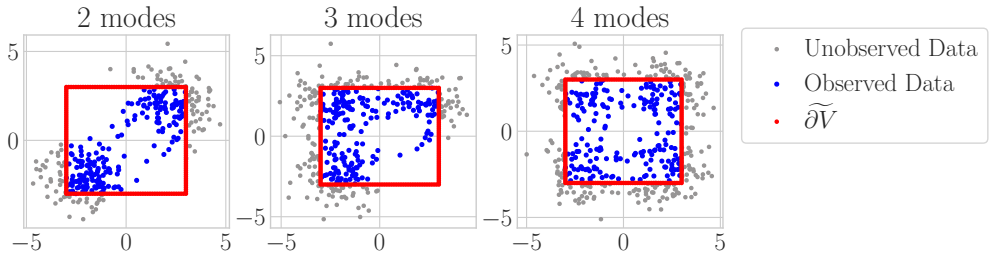
$$\boldsymbol{\mu}_1^* = [1.5, 1.5]^\top,\ \boldsymbol{\mu}_2^* = [-1.5, -1.5]^\top,\ \boldsymbol{\mu}_3^* = [-1.5, 1.5]^\top,\ \boldsymbol{\mu}_4^* = [1.5, -1.5]^\top.$$

For a given number of mixture modes, we independently simulate samples from $\mathcal{N}(\boldsymbol{\mu}_j^*, I_d)$, for $j = 1, \ldots, n_m$, where m is the number of mixture modes, and discard samples that are outside a box which has vertices at $[-3, -3]$, $[3, -3]$, $[-3, 3]$ and $[3, 3]$, until we reach a total of n samples after truncation. Figure 5.9(a) shows an example of this experiment setup for different amounts of n_m.

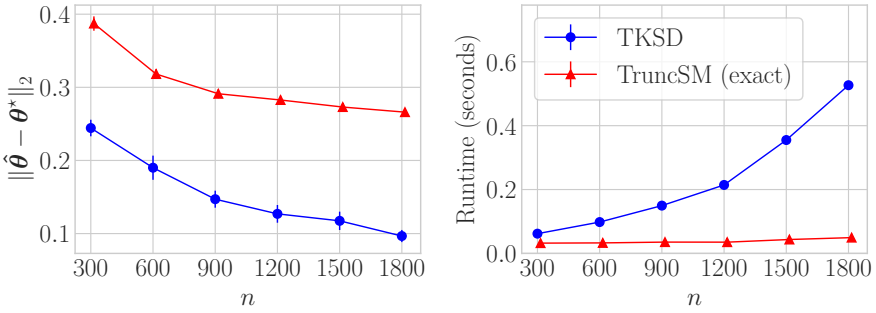
The task is to estimate $\boldsymbol{\theta}^\star := \boldsymbol{\mu}_i^\star$ for all i. To ensure a well specified experiment, we set the initial conditions for the numerical optimisation routine as a perturbation from the true value, i.e. $\boldsymbol{\theta}_i^{\text{ini}} = \boldsymbol{\theta}_i^\star + \boldsymbol{z}$, $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, 0.5 \cdot I_2)$. We estimate $\hat{\boldsymbol{\theta}}$ with TKSD and compare it to a corresponding estimate by *TruncSM* across a range of different values of n and number of mixture modes n_m, shown in figure 5.9. Overall, TKSD significantly outperforms *TruncSM* in this experiment across all variations at the slight cost of runtime. Whilst the runtime of TKSD seems to be quadratically increasing, it still remains under two seconds across all experiments.

5.6.4 Regression

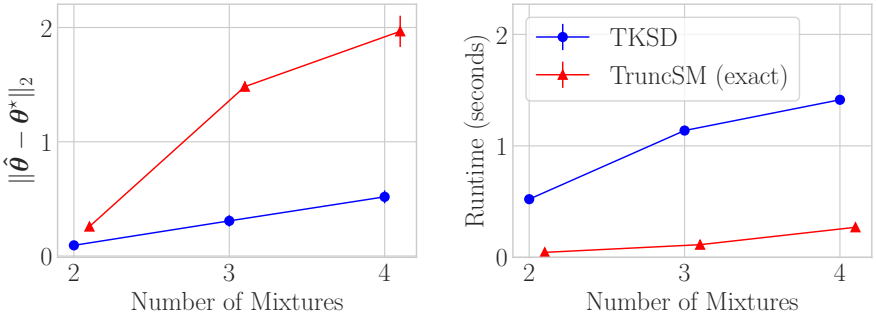
We provide a further example of using TKSD to estimate the parameters of a regression problem. We provide two examples in this section, for synthetic data and for real data.



(a) Example setup for the experiment as the number of mixture modes increases from 2 to 4.



(b) Estimation error as n increases for 2 mixture modes, averaged across 256 seeds.



(c) Estimation error as the number of mixture modes increases for a fixed $n = 300$, averaged across 256 seeds.

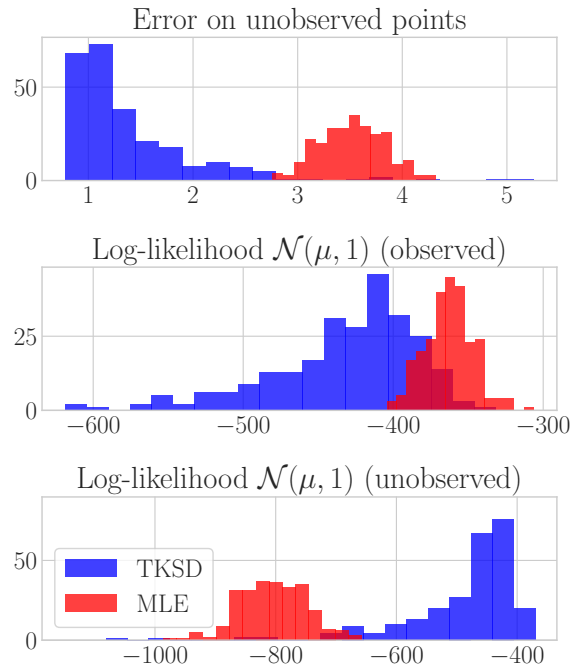
Figure 5.9. Estimation of Gaussian mixture modes using TKSD and *TruncSM*.

Synthetic Data First, we simulate data in the following way:

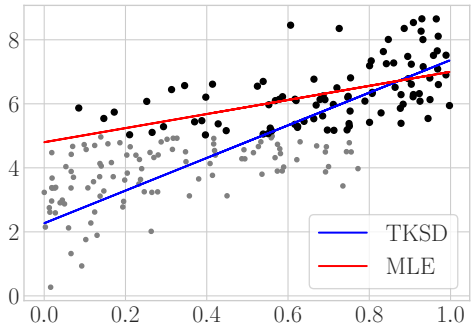
$$y_i \sim \mathcal{N}(\mu_i, 1), \quad \mu_i = \beta_0^\star + \beta_1^\star x_i, \quad x_i \sim \text{Uniform}(0, 1)$$

where we assume knowledge of the true values, $\beta_0^\star = 3$ and $\beta_1^\star = 4$. We truncate the dataset to where $y_i \geq 5 \ \forall i$, so only a portion of both y and x are observed. We then estimate the conditional density $p(y|x)$ with TKSD via estimation of $\boldsymbol{\theta}^\star := [\beta_0^\star, \beta_1^\star]^\top$.

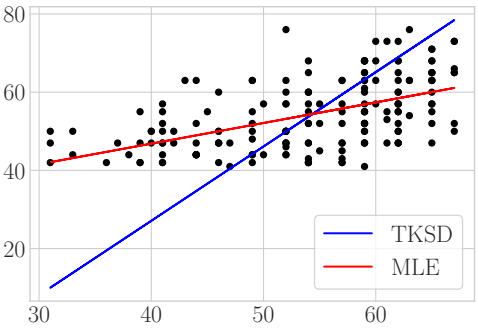
We obtain the log-likelihood of the Normal distribution under the estimate of $\hat{\boldsymbol{\theta}} := [\hat{\beta}_0, \hat{\beta}_1]^\top$ output by TKSD, split across the observed dataset (within the truncation boundary) and the



(a) Mean squared errors only on the observed dataset (top). Log-likelihoods on the observed dataset (middle) and the unobserved dataset (bottom), both for a $\mathcal{N}(\mu, 1)$ distribution and over 256 trials.



(b) An example of TKSD used to fit regression on the synthetic data on one trial.



(c) TKSD used to fit regression on student test scores data.

Figure 5.10. Estimation of regression parameters using TKSD.

unobserved dataset (outside the truncation boundary). We compare these log-likelihoods to ones obtained by a naive MLE approach which does not account for truncation. As an additional measure, we calculate the unobserved error, given by

$$\text{MSE}(\mathbf{y}, \hat{\mathbf{y}}) := \sum_{i=1}^{n_{\text{unobs}}} (\hat{y}_i - y_i)^2, \quad \hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i,$$

where n_{unobs} is the number of *discarded* data points due to truncation, and this error is calculated over these discarded data points only. This forms a measure which evaluates how well the density estimation method is predicting on unseen data points. Instead of traditional test error, it is the error on data points that have been truncated and hence no longer observed by either method.

A histogram of log-likelihoods and errors across 256 trials is given in figure 5.10(a), and an example of one such simulated regression is given in figure 5.10(b). The log-likelihoods for the observed points are higher for MLE, since the objective of this method is to directly maximise these likelihoods. However, on the unobserved dataset, the log-likelihoods obtained by TKSD

are significantly higher than MLE. The unobserved errors are significantly smaller for TKSD, showing the improvement of TKSD over the naive approach.

Real Data: Student Test Scores We also experiment on a real-world dataset given by UCLA: Statistical Consulting Group (2022) (Example 1). This dataset contains student test scores in a school for which the acceptance threshold is 40 out of 100, and therefore the response variable (the test scores) are truncated below by 40 and above by 100. Since no scores get close to 100, we only consider one-sided truncation at $y = 40$. The aim of the regression is to model the response variable, the test scores, based on a single covariate of each students' corresponding score on a different test. Figure 5.10(c) shows the plotted dataset and the regression line fit by TKSD and naive MLE. Whilst we have no true baseline value to compare to, the TKSD regression line seems to account for the truncation, whilst as expected, MLE does not.

5.7 Conclusion

We have proposed an alternative to the classical Stein class, called an *approximate Stein class*, bounded by a decreasing sequence instead of being strictly equal to zero. By maximising the KSD objective over this approximate Stein class, we have constructed a truncated density estimator based on KSD, called truncated KSD (TKSD), and shown that it is consistent. TKSD has advantages over the prior works by Xu (2022) and Liu et al. (2022), as it does not require a functional form of a boundary.

Some limitations of this method include the requirement of selecting hyperparameters, such as the kernel function k and its associated hyperparameters, and the number of samples from the boundary, m. Choice of m may depend on applications. Still, a larger value is preferred when the complexity of the truncation boundary is greater, or the dimension is larger. However, even with the heuristic approaches presented in this paper, TKSD provides competitive results.

In experimental results, we have shown that even though we assume no access to a functional form of the boundary, TKSD performs similarly to previous methods which require the boundary to be computed. In some scenarios, TKSD performs better than these methods, or takes less time to achieve the same result.

All results in this paper can be reproduced using the GitHub repository located at `https://github.com/dannyjameswilliams/tksd`.

5.8 Proofs

5.8.1 Proof of lemma 5.2

Proof. Let $\boldsymbol{g} \in \mathcal{G}_0^d$. Begin by writing the expectation as an integral,

$$\begin{aligned}\mathbb{E}_q[\mathcal{T}_q \boldsymbol{g}(\boldsymbol{x})] = \int_V q(\boldsymbol{x}) \mathcal{T}_q \boldsymbol{g}(\boldsymbol{x}) d\boldsymbol{x} &= \sum_{l=1}^d \int_V q(\boldsymbol{x}) \partial_{x_l} \log q(\boldsymbol{x}) g_l(\boldsymbol{x}) + \partial_{x_l} g_l(\boldsymbol{x}) d\boldsymbol{x} \\ &= \sum_{l=1}^d \int_V \partial_{x_l} q(\boldsymbol{x}) g_l(\boldsymbol{x}) + \partial_{x_l} g_l(\boldsymbol{x}) d\boldsymbol{x}.\end{aligned}$$

This can be expanded by integration by parts,

$$\begin{aligned}\sum_{l=1}^d \int_V \partial_{x_l} q(\boldsymbol{x}) g_l(\boldsymbol{x}) + \partial_{x_l} g_l(\boldsymbol{x}) d\boldsymbol{x} &= \sum_{l=1}^d \oint_{\partial V} q(\boldsymbol{x}) g_l(\boldsymbol{x}) \hat{\mathrm{u}}_l(\boldsymbol{x}) ds \\ &\qquad + \sum_{l=1}^d \int_V q(\boldsymbol{x}) (\partial_{x_l} g_l(\boldsymbol{x}) - \partial_{x_l} g_l(\boldsymbol{x})) d\boldsymbol{x} \\ &= \sum_{l=1}^d \oint_{\partial V} q(\boldsymbol{x}) g_l(\boldsymbol{x}) \hat{\mathrm{u}}_l(\boldsymbol{x}) ds = 0,\end{aligned}$$

where the final equality comes from the boundary conditions on g, such that all evaluations $g_l(\boldsymbol{x}') = 0 \; \forall \boldsymbol{x}' \in \partial V$. ■

5.8.2 Proof of lemma 5.3

Proof. Let $\boldsymbol{g} \in \mathcal{G}_0^d$ and $\tilde{\boldsymbol{g}} \in \mathcal{G}_{0,m}^d$. First note that $\boldsymbol{g}$ and $\tilde{\boldsymbol{g}}$ agree on $\widetilde{\partial V}$, i.e.

$$g_l(\tilde{\boldsymbol{x}}') = \tilde{g}_l(\tilde{\boldsymbol{x}}'), \quad \forall l = 1, \ldots, d \tag{5.23}$$

for all $\tilde{\boldsymbol{x}}' \in \widetilde{\partial V}$, since all $\tilde{\boldsymbol{x}}'$ are elements of ∂V also, as $\widetilde{\partial V} \subset \partial V$. First, we note that since we assume that g_l and $\tilde{g}_l$ are both Lipschitz continuous, then

$$|g_l(\boldsymbol{x}') - g_l(\tilde{\boldsymbol{x}}')| \leq C_1 \|\boldsymbol{x}' - \tilde{\boldsymbol{x}}'\| \leq C_1 \varepsilon_m \tag{5.24}$$

$$|\tilde{g}_l(\tilde{\boldsymbol{x}}') - \tilde{g}_l(\boldsymbol{x}')| \leq C_2 \|\boldsymbol{x}' - \tilde{\boldsymbol{x}}'\| \leq C_2 \varepsilon_m, \tag{5.25}$$

where $\|\boldsymbol{x}' - \tilde{\boldsymbol{x}}'\| \leq \varepsilon_m$. This follows under the assumption that $\widetilde{\partial V}$ is ε_m-dense in ∂V, therefore the distance $\|\boldsymbol{x}' - \tilde{\boldsymbol{x}}'\|$ is at most ε_m.

We seek to quantify

$$|g_l(\boldsymbol{x}') - \tilde{g}_l(\boldsymbol{x}')|,$$

which is how far apart the 'approximate' $\tilde{g}_l$ is from the 'true' g_l for any point $\boldsymbol{x}' \in \partial V$. Note that this includes points that are not in $\widetilde{\partial V}$. Write

$$\begin{aligned} g_l(\boldsymbol{x}') - \tilde{g}_l(\boldsymbol{x}') &= (g_l(\boldsymbol{x}') - \tilde{g}_l(\boldsymbol{x}')) + (g_l(\tilde{\boldsymbol{x}}') - g_l(\tilde{\boldsymbol{x}}')) + (\tilde{g}_l(\tilde{\boldsymbol{x}}') - \tilde{g}_l(\tilde{\boldsymbol{x}}')) \\ &\overset{\text{(a)}}{=} (g_l(\boldsymbol{x}') - \tilde{g}_l(\boldsymbol{x}')) - g_l(\tilde{\boldsymbol{x}}') + \tilde{g}_l(\tilde{\boldsymbol{x}}') \\ &= (g_l(\boldsymbol{x}') - g_l(\tilde{\boldsymbol{x}}')) + (\tilde{g}_l(\tilde{\boldsymbol{x}}') - \tilde{g}_l(\boldsymbol{x}')), \end{aligned}$$

where equality (a) is by equation (5.23). Using equation (5.24) and equation (5.25), we have the following inequality,

$$|g_l(\boldsymbol{x}') - \tilde{g}_l(\boldsymbol{x}')| \leq |g_l(\boldsymbol{x}') - g_l(\tilde{\boldsymbol{x}}')| + |\tilde{g}_l(\tilde{\boldsymbol{x}}') - \tilde{g}_l(\boldsymbol{x}')| \leq (C_1 + C_2)\varepsilon_m,$$

where the last step follows by the triangle inequality. Therefore, since $\widetilde{\partial V}$ is ε_m-dense in ∂V with some probability, we have

$$|g_l(\boldsymbol{x}') - \tilde{g}_l(\boldsymbol{x}')| = O_P(\varepsilon_m)$$

as desired. ■

5.8.3 Proof of proposition 5.4

Proof. First, let $\mathrm{L}(V)$ denote the $(d-1)$-surface area of a bounded domain $V \subset \mathbb{R}^d$ and $\mathrm{L}(V) < \infty$. For example, in 2D, $\mathrm{L}(V)$ corresponds to the line length of ∂V. We also define $B_{\varepsilon_m}(\boldsymbol{x}')$ as the ball of radius ε_m centred on $\boldsymbol{x}'$.

Before continuing to the proof, recall the definition of an ε-dense set: $\forall \boldsymbol{x}' \in \partial V, \exists \widetilde{\boldsymbol{x}}' \in \widetilde{\partial V}$ such that $d(\boldsymbol{x}', \widetilde{\boldsymbol{x}}') \leq \varepsilon$, where d is some measure of distance. The statement $d(\boldsymbol{x}', \widetilde{\boldsymbol{x}}') \leq \varepsilon_m$ is equivalent to $\widetilde{\boldsymbol{x}}' \in B_{\varepsilon_m}(\boldsymbol{x}')$. Now write that the probability that the definition of a ε_m-dense set holds for $\widetilde{\partial V}$ being dense in ∂V is equal to 0.95:

$$\begin{aligned} 0.95 = \mathbb{P}(\forall \boldsymbol{x}' \in \partial V, \exists \widetilde{\boldsymbol{x}}' \in \widetilde{\partial V}, \widetilde{\boldsymbol{x}}' \in B_{\varepsilon_m}(\boldsymbol{x}')) &= 1 - \mathbb{P}(\exists \boldsymbol{x}' \in \partial V, \forall \widetilde{\boldsymbol{x}}' \in \widetilde{\partial V}, \widetilde{\boldsymbol{x}}' \notin B_{\varepsilon_m}(\boldsymbol{x}')) \\ &= 1 - \mathbb{P}\Big(\bigcup_{\boldsymbol{x}' \in \partial V} \forall \widetilde{\boldsymbol{x}}' \in \widetilde{\partial V}, \widetilde{\boldsymbol{x}}' \notin B_{\varepsilon_m}(\boldsymbol{x}') \Big). \end{aligned}$$

Equivalently, by rearranging the above, we have

$$\mathbb{P}\Big(\bigcup_{\boldsymbol{x}' \in \partial V} \forall \widetilde{\boldsymbol{x}}' \in \widetilde{\partial V}, \widetilde{\boldsymbol{x}}' \notin B_{\varepsilon_m}(\boldsymbol{x}') \Big) = 0.05.$$

This probability can be bounded as follows

$$0.05 = \mathbb{P}\Big(\bigcup_{\boldsymbol{x}' \in \partial V} \forall \widetilde{\boldsymbol{x}}' \in \widetilde{\partial V}, \widetilde{\boldsymbol{x}}' \notin B_{\varepsilon_m}(\boldsymbol{x}') \Big) \geq \mathbb{P}\Big(\forall \widetilde{\boldsymbol{x}}' \in \widetilde{\partial V}, \widetilde{\boldsymbol{x}}' \notin B_{\varepsilon_m}(\boldsymbol{x}_0') \Big)$$

where $\boldsymbol{x}_0'$ is a single point in ∂V. This follows by basic laws of probability. Note that the probability on the right hand side can be written as

$$\begin{aligned} \mathbb{P}\left(\forall \widetilde{\boldsymbol{x}}' \in \widetilde{\partial V}, \widetilde{\boldsymbol{x}}' \notin B_{\varepsilon_m}(\boldsymbol{x}_0') \right) &= \mathbb{P}\left(\widetilde{\boldsymbol{x}}_0' \notin B_{\varepsilon_m}(\boldsymbol{x}_0') \right)^m \\ &= \left[1 - \mathbb{P}\left(\widetilde{\boldsymbol{x}}_0' \in B_{\varepsilon_m}(\boldsymbol{x}_0') \right) \right]^m \\ &= \left[1 - \frac{\mathrm{Area}(\partial V \cap B_{\varepsilon_m}(\boldsymbol{x}_0'))}{\mathrm{L}(V)} \right]^m \leq 0.05 \end{aligned} \tag{5.26}$$

where the first equality comes from all $\widetilde{\boldsymbol{x}}' \in \widetilde{\partial V}$ being independently distributed. $\mathrm{Area}(S)$ denotes the surface area of the boundary set S. For example, $\mathrm{Area}(\partial V) = \mathrm{L}(V)$. Therefore $\mathrm{Area}(\partial V \cap B_{\varepsilon_m}(\boldsymbol{x}'_0))$ represents the size of the region of ∂V that is inside $B_{\varepsilon_m}(\boldsymbol{x}'_0)$, which will be the area of the boundary hyperplane of ∂V which passes through $B_{\varepsilon_m}(\boldsymbol{x}'_0)$. We obtain the probability in the final equality by assuming that all $\widetilde{\boldsymbol{x}}'$ are uniformly sampled from ∂V, so the probability is the proportion of ∂V inside $B_{\varepsilon_m}(\boldsymbol{x}'_0)$ as a ratio of the full ∂V. This probability exists under the assumption that $\mathrm{L}(V) < \infty$.

Next, we can rearrange equation (5.26) to

$$\mathrm{Area}(\partial V \cap B_{\varepsilon_m}(\boldsymbol{x}'_0)) \geq \mathrm{L}(V) \left[1 - 0.05^{1/m}\right].$$

Consider an upper bound for $\mathrm{Area}(\partial V \cap B_{\varepsilon_m}(\boldsymbol{x}'_0))$ given by the volume of $B_{\varepsilon_m}(\boldsymbol{x}'_0)$, i.e.

$$\mathrm{Area}(\partial V \cap B_{\varepsilon_m}(\boldsymbol{x}'_0)) \leq \xi(d)\varepsilon_m^d, \qquad \xi(d) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)},$$

where Γ is the Gamma function. Combining these bounds gives

$$\xi(d)\varepsilon_m^d \geq \mathrm{L}(V) \left[1 - 0.05^{1/m}\right]$$

and therefore

$$\varepsilon_m \geq \left(\frac{\mathrm{L}(V)}{\xi(d)} \left[1 - 0.05^{1/m}\right]\right)^{1/d},$$

which completes the proof. ■

5.8.4 Proof of theorem 5.6

Proof. Let $\boldsymbol{g} \in \mathcal{G}_0^d$ and $\tilde{\boldsymbol{g}} \in \mathcal{G}_{0,m}^d$. Begin with writing the expectation in its integral form,

$$\begin{aligned}
\mathbb{E}_q[\mathcal{T}_q\tilde{\boldsymbol{g}}(\boldsymbol{x})] &= \int_V q(\boldsymbol{x}) \left[\sum_{l=1}^d \partial_{x_l} \log q(\boldsymbol{x})\tilde{g}_l(\boldsymbol{x}) + \sum_{l=1}^d \partial_{x_l}\tilde{g}_l(\boldsymbol{x})\right] d\boldsymbol{x} \\
&= \sum_{l=1}^d \int_V q(\boldsymbol{x})\partial_{x_l} \log q(\boldsymbol{x})\tilde{g}_l(\boldsymbol{x})d\boldsymbol{x} + \sum_{l=1}^d \int_V q(\boldsymbol{x})\partial_{x_l}\tilde{g}_l(\boldsymbol{x})d\boldsymbol{x} \\
&\overset{\text{(a)}}{=} \sum_{l=1}^d \int_V \partial_{x_l} q(\boldsymbol{x})\tilde{g}_l(\boldsymbol{x})d\boldsymbol{x} + \sum_{l=1}^d \int_V q(\boldsymbol{x})\partial_{x_l}\tilde{g}_l(\boldsymbol{x})d\boldsymbol{x} \\
&\overset{\text{(b)}}{=} \sum_{l=1}^d \oint_{\partial V} q(\boldsymbol{x})\tilde{g}_l(\boldsymbol{x})\hat{\mathrm{u}}_l(\boldsymbol{x})ds - \sum_{l=1}^d \int_V q(\boldsymbol{x})\partial_{x_l}\tilde{g}_l(\boldsymbol{x})d\boldsymbol{x} + \sum_{l=1}^d \int_V q(\boldsymbol{x})\partial_{x_l}\tilde{g}_l(\boldsymbol{x})d\boldsymbol{x} \\
&= \sum_{l=1}^d \oint_{\partial V} q(\boldsymbol{x})\tilde{g}_l(\boldsymbol{x})\hat{\mathrm{u}}_l(\boldsymbol{x})ds
\end{aligned}$$

where $\oint_{\partial V}$ is the surface integral over the boundary ∂V, (a) comes from the identity that $q(\boldsymbol{x})\partial_{x_l} \log q(\boldsymbol{x}) = \partial_{x_l} q(\boldsymbol{x})$, and (b) is from integration by parts. Substitute $\tilde{g}_l(\boldsymbol{x}) = \tilde{g}_l(\boldsymbol{x}) + g_l(\boldsymbol{x}) - g_l(\boldsymbol{x})$ to obtain

$$\begin{aligned}
&\sum_{l=1}^{d} \oint_{\partial V} q(\boldsymbol{x})(\tilde{g}_l(\boldsymbol{x}) + g_l(\boldsymbol{x}) - g_l(\boldsymbol{x}))\hat{\mathrm{u}}_l(\boldsymbol{x})ds \\
=&\sum_{l=1}^{d} \oint_{\partial V} q(\boldsymbol{x})(\tilde{g}_l(\boldsymbol{x}) - g_l(\boldsymbol{x}))\hat{\mathrm{u}}_l(\boldsymbol{x})ds + \sum_{l=1}^{d} \oint_{\partial V} q(\boldsymbol{x}) g_l(\boldsymbol{x})\hat{\mathrm{u}}_l(\boldsymbol{x})ds.
\end{aligned}$$

By lemma 5.3, $|\tilde{g}_l(\boldsymbol{x}') - g_l(\boldsymbol{x}')| = O_P(\varepsilon_m)$, leaving

$$\mathbb{E}_q[\mathcal{T}_q \tilde{\boldsymbol{g}}(\boldsymbol{x})] = O_P(\varepsilon_m) + \sum_{l=1}^{d} \oint_{\partial V} q(\boldsymbol{x}) g_l(\boldsymbol{x})\hat{\mathrm{u}}_l(\boldsymbol{x})ds$$

As all evaluations of $g_l(\boldsymbol{x}) = 0\ \forall \boldsymbol{x} \in \partial V$ by definition of $\mathcal{G}_0^d$, the second term equals zero, leaving only

$$\mathbb{E}_q[\mathcal{T}_q \tilde{\boldsymbol{g}}(\boldsymbol{x})] = O_P(\varepsilon_m)$$

as desired. ■

5.8.5 Proof of theorem 5.7

Proof. Recall $\mathcal{G}_{0,m}^d = \{\boldsymbol{g} \in \mathcal{G}^d \mid \boldsymbol{g}(\boldsymbol{x}') = \boldsymbol{0}\ \forall \boldsymbol{x}' \in \widetilde{\partial V}, \|\boldsymbol{g}\|_{\mathcal{G}^d}^2 \leq 1\}$, where $\mathcal{G}^d$ is the product RKHS with d elements, $\boldsymbol{g} = (g_1, \ldots, g_d)$, and $g_l \in \mathcal{G}, \forall l = 1, \ldots, d$.

Begin with the definition of TKSD as defined in equation (5.14),

$$\sup_{\boldsymbol{g} \in \mathcal{G}_{0,m}^d} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}} \boldsymbol{g}(\boldsymbol{x})].$$

To solve this supremum analytically, we can reframe it as a constrained maximisation problem,

$$\begin{aligned}
&\max_{\boldsymbol{g} \in \mathcal{G}^d} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}} \boldsymbol{g}(\boldsymbol{x})], \\
&\text{subject to } g_l(\boldsymbol{x}') = 0\ \forall \boldsymbol{x}' \in \widetilde{\partial V},\ \forall l = 1, \ldots, d \\
&\qquad \|\boldsymbol{g}\|_{\mathcal{G}^d} \leq 1,
\end{aligned} \tag{5.27}$$

where the constraints in the definition for $\mathcal{G}_{0,m}^d$ have been included as optimisation constraints, and the maximisation is now with respect to the RKHS function family $\mathcal{G}^d$ only. To solve this, we can formulate a Lagrangian dual function (Boyd and Vandenberghe, 2004).

$$\inf_{\boldsymbol{\nu}^{(1)}, \ldots, \boldsymbol{\nu}^{(d)}, \lambda} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{g}, \boldsymbol{\nu}^{(1)}, \ldots, \boldsymbol{\nu}^{(d)}, \lambda) \tag{5.28}$$

$$\mathcal{L} = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{g}, \boldsymbol{\nu}^{(1)}, \ldots, \boldsymbol{\nu}^{(d)}, \lambda) := \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}}} \boldsymbol{g}(\boldsymbol{x})] + \sum_{l=1}^{d} \sum_{i'=1}^{m} \nu_{i'}^{(l)} g_l(\boldsymbol{x}'_{i'}) + \lambda(\|\boldsymbol{g}\|_{\mathcal{G}^d}^2 - 1), \tag{5.29}$$

for $\boldsymbol{\nu}^{(l)} \in \mathbb{R}^m \; \forall l$, $\lambda \geq 0$ and $\mathcal{L}$ is our Lagrangian. The overall optimisation problem that needs solving is given by

$$\min_{\boldsymbol{\nu}^{(1)},\ldots,\boldsymbol{\nu}^{(d)},\lambda} \; \max_{\boldsymbol{g}\in\mathcal{G}^d} \; \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{g}, \boldsymbol{\nu}^{(1)}, \ldots, \boldsymbol{\nu}^{(d)}, \lambda). \tag{5.30}$$

By solving the dual problem, equation (5.30), we solve the primal problem, equation (5.27). We can rewrite equation (5.29) as

$$\mathcal{L} = \mathbb{E}_q \left[\sum_{l=1}^{d} \partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) g_l(\boldsymbol{x}) + \partial_{x_l} g_l(\boldsymbol{x}) \right] + \sum_{l=1}^{d} \sum_{i'=1}^{m} \nu_{i'}^{(l)} g_l(\boldsymbol{x}'_{i'}) + \lambda(\|\boldsymbol{g}\|_{\mathcal{G}^d}^2 - 1),$$

and we can expand evaluations of $g_l(\boldsymbol{x})$ via the reproducing property of $\mathcal{G}$, given by $g_l(\boldsymbol{x}) = \langle g_l, k(\boldsymbol{x}, \cdot)\rangle_{\mathcal{G}}$, to

$$\begin{aligned}
\mathcal{L} = \mathbb{E}_q \left[\sum_{l=1}^{d} \partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \left\langle g_l, k(\boldsymbol{x}, \cdot)\right\rangle_{\mathcal{G}} + \left\langle g_l, \partial_{x_l} k(\boldsymbol{x}, \cdot)\right\rangle_{\mathcal{G}} \right] \qquad (5.31) \\
+ \sum_{i'=1}^{m} \sum_{i=1}^{d} \nu_{i'}^{(l)} \left\langle g_l, k(\boldsymbol{x}'_{i'}, \cdot)\right\rangle_{\mathcal{G}} + \lambda \left(\sum_{l=1}^{d} \langle g_l, g_l\rangle_{\mathcal{G}} - 1 \right) \\
= \sum_{l=1}^{d} \mathbb{E}_q \left[\left\langle g_l, \partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) k(\boldsymbol{x}, \cdot) + \partial_{x_l} k(\boldsymbol{x}, \cdot) + \sum_{i'=1}^{m} \nu_{i'}^{(l)} k(\boldsymbol{x}'_{i'}, \cdot) \right\rangle_{\mathcal{G}} \right] + \lambda \left(\sum_{l=1}^{d} \langle g_l, g_l\rangle_{\mathcal{G}} - 1 \right) \\
= \sum_{l=1}^{d} \left\langle g_l, \mathbb{E}_q[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) k(\boldsymbol{x}, \cdot) + \partial_{x_l} k(\boldsymbol{x}, \cdot)] + \sum_{i'=1}^{m} \nu_{i'}^{(l)} k(\boldsymbol{x}'_{i'}, \cdot) \right\rangle_{\mathcal{G}} + \lambda \left(\sum_{l=1}^{d} \langle g_l, g_l\rangle_{\mathcal{G}} - 1 \right). \qquad (5.32)
\end{aligned}$$

The final equality holds provided that $\mathbb{E}_q[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) k(\boldsymbol{x}, \cdot) + \partial_{x_l} k(\boldsymbol{x}, \cdot) + \sum_{i'=1}^{m} \nu_{i'}^{(l)} k(\boldsymbol{x}'_{i'}, \cdot)] < \infty$, i.e. the term inside the expectation is Bochner integrable (Steinwart and Christmann, 2008). This same assumption was made in Chwialkowski et al. (2016), as we can consider $\sum_{i'=1}^{m} \nu_{i'}^{(l)} k(\boldsymbol{x}'_{i'}, \cdot)$ as a constant with respect to the expectation.

We solve for each parameter via differentiation to obtain a closed form solution. Across dimensions, each g_l in equation (5.32) appears only additively to another, and so we can consider the l-th element of the derivative, and solve the inner maximisation for each g_l to give a solution for $\boldsymbol{g}$. This differentiation gives

$$\frac{\partial \mathcal{L}}{\partial g_l} = \mathbb{E}_q \left[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) k(\boldsymbol{x}, \cdot) + \partial_{x_l} k(\boldsymbol{x}, \cdot)\right] + \sum_{i'=1}^{m} \nu_{i'}^{(l)} k(\boldsymbol{x}'_{i'}, \cdot) + 2\lambda g_l = 0.$$

Rearranging for g_l gives the solution

$$g_l^\star = -\frac{1}{2\lambda} \mathbb{E}_q \left[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) k(\boldsymbol{x}, \cdot) + \partial_{x_l} k(\boldsymbol{x}, \cdot)\right] - \frac{1}{2\lambda} \sum_{i'=1}^{m} \nu_{i'}^{(l)} k(\boldsymbol{x}'_{i'}, \cdot) = -\frac{z_l}{2\lambda},$$

where in this context the $\star$ notation denotes the closed form solution to an optimisation problem, additionally for convenience we have denoted

$$z_l = \mathbb{E}_q \left[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) k(\boldsymbol{x}, \cdot) + \partial_{x_l} k(\boldsymbol{x}, \cdot)\right] + \sum_{i'=1}^{m} \nu_{i'}^{(l)} k(\boldsymbol{x}'_{i'}, \cdot).$$

Substituting this back into equation (5.32) gives

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{g}^\star, \boldsymbol{\nu}^{(1)}, \ldots, \boldsymbol{\nu}^{(d)}, \lambda) = \sum_{l=1}^{d} \left\langle -\frac{z_l}{2\lambda}, z_l \right\rangle_{\mathcal{G}} + \lambda \left(\sum_{i=1}^{d} \left\langle -\frac{z_l}{2\lambda}, -\frac{z_l}{2\lambda} \right\rangle_{\mathcal{G}} - 1 \right) = \sum_{l=1}^{d} \frac{1}{4\lambda} \langle z_l, z_l \rangle_{\mathcal{G}} + \lambda, \quad (5.33)$$

where $\boldsymbol{g}^\star = (g_1^\star, \ldots, g_d^\star)$. Solve for λ by differentiating

$$\frac{\partial \mathcal{L}}{\partial \lambda} = -\frac{1}{4\lambda^2} \sum_{i=1}^{d} \langle z_l, z_l \rangle_{\mathcal{G}} + 1 = 0.$$

Rearranging for λ gives $\lambda = \pm\sqrt{\sum_{l=1}^{d} \langle z_l, z_l \rangle_{\mathcal{G}} / 4}$. Since $\lambda \geq 0$, we take the positive solution, giving $\lambda^\star = \sqrt{\sum_{l=1}^{d} \langle z_l, z_l \rangle_{\mathcal{G}}}/2$. Substituting $\lambda^\star$ into equation (5.33) gives

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{g}^\star, \boldsymbol{\nu}^{(1)}, \ldots, \boldsymbol{\nu}^{(d)}, \lambda^\star) = \sum_{l=1}^{d} \frac{1}{4\lambda^\star} \langle z_l, z_l \rangle_{\mathcal{G}} + \lambda^\star = \sqrt{\sum_{l=1}^{d} \langle z_l, z_l \rangle_{\mathcal{G}}}. \quad (5.34)$$

To solve for the final Lagrangian parameters, $\boldsymbol{\nu}^{(1)}, \ldots, \boldsymbol{\nu}^{(d)}$, let us first introduce some notation. Denote

$$\boldsymbol{\phi}_{\mathbf{x}'} = \begin{bmatrix} k(\boldsymbol{x}'_1, \cdot), \\ \vdots, \\ k(\boldsymbol{x}'_m, \cdot) \end{bmatrix}, \qquad \boldsymbol{\phi}_{\boldsymbol{z}, \mathbf{x}'} = \begin{bmatrix} k(\boldsymbol{z}, \boldsymbol{x}'_1) & \ldots & k(\boldsymbol{z}, \boldsymbol{x}'_m) \end{bmatrix}, \qquad \mathbf{K}' = \boldsymbol{\phi}_{\mathbf{x}'} \boldsymbol{\phi}_{\mathbf{x}'}^\top,$$

where $\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_m \in \widetilde{\partial V}$. Now consider the equivalence

$$\boldsymbol{\nu}^{(l)\star} := \arg\min_{\boldsymbol{\nu}^{(l)}} \sqrt{\sum_{l=1}^{d} \langle z_l, z_l \rangle_{\mathcal{G}}} = \arg\min_{\boldsymbol{\nu}^{(l)}} \sum_{l=1}^{d} \langle z_l, z_l \rangle_{\mathcal{G}} = \arg\min_{\boldsymbol{\nu}^{(l)}} \langle z_l, z_l \rangle_{\mathcal{G}},$$

as each $\boldsymbol{\nu}^{(l)}$ is independent. By rewriting z_l as $z_l = \mathbb{E}_q \left[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) k(\boldsymbol{x}, \cdot) + \partial_{x_l} k(\boldsymbol{x}, \cdot) \right] + {\boldsymbol{\nu}^{(l)}}^\top \boldsymbol{\phi}_{\mathbf{x}'}$, we solve this again by differentiating,

$$\begin{aligned} \frac{d}{d\boldsymbol{\nu}^{(l)}} \langle z_l, z_l \rangle_{\mathcal{G}} = 2 \left\langle \frac{dz_l}{d\boldsymbol{\nu}^{(l)}}, z_l \right\rangle_{\mathcal{G}} &= 2 \left\langle \boldsymbol{\phi}_{\mathbf{x}'}, \mathbb{E}_q \left[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) k(\boldsymbol{x}, \cdot) + \partial_{x_l} k(\boldsymbol{x}, \cdot) \right] + {\boldsymbol{\nu}^{(l)}}^\top \boldsymbol{\phi}_{\mathbf{x}'} \right\rangle_{\mathcal{G}} \\ &= 2\, \mathbb{E}_q \left[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \boldsymbol{\phi}_{\boldsymbol{x}, \mathbf{x}'} + \partial_{x_l} \boldsymbol{\phi}_{\boldsymbol{x}, \mathbf{x}'} \right] + 2 {\boldsymbol{\nu}^{(l)}}^\top \boldsymbol{\phi}_{\mathbf{x}'} \boldsymbol{\phi}_{\mathbf{x}'}^\top \\ &= 2\, \mathbb{E}_q \left[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \boldsymbol{\phi}_{\boldsymbol{x}, \mathbf{x}'} + \partial_{x_l} \boldsymbol{\phi}_{\boldsymbol{x}, \mathbf{x}'} \right] + 2 {\boldsymbol{\nu}^{(l)}}^\top \mathbf{K}'. \end{aligned}$$

Setting equal to zero and rearranging gives

$$\boldsymbol{\nu}^{(l)\star} = -(\mathbf{K}')^{-1} \mathbb{E}_q \left[\partial_{x_l} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \boldsymbol{\phi}_{\boldsymbol{x}, \mathbf{x}'}^\top + \left(\partial_{x_l} \boldsymbol{\phi}_{\boldsymbol{x}, \mathbf{x}'} \right)^\top \right]. \quad (5.35)$$

Before substituting back into equation (5.34), let us first square it and expand it as

$$\begin{aligned}\mathcal{L}^2 = \sum_{l=1}^{d} \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{y}\sim q(\boldsymbol{x})}[u_l(\boldsymbol{x},\boldsymbol{y})] + \left(\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\partial_{x_l}\log p_{\boldsymbol{\theta}}(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}^\top + (\partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'})^\top\right]\right)^\top \boldsymbol{\nu}^{(l)} \\ + {\boldsymbol{\nu}^{(l)}}^\top \mathbb{E}_{\boldsymbol{y}\sim q(\boldsymbol{x})}\left[\partial_{y_l}\log p_{\boldsymbol{\theta}}(\boldsymbol{y})\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top + (\partial_{y_l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'})^\top\right] + {\boldsymbol{\nu}^{(l)}}^\top \mathbf{K}'\boldsymbol{\nu}^{(l)}\end{aligned} \tag{5.36}$$

where $u_l(\boldsymbol{x},\boldsymbol{y}) = \psi_{p,l}(\boldsymbol{x})\psi_{p,l}(\boldsymbol{y})k(\boldsymbol{x},\boldsymbol{y}) + \psi_{p,l}(\boldsymbol{x})\partial_{y_l}k(\boldsymbol{x},\boldsymbol{y}) + \psi_{p,l}(\boldsymbol{y})\partial_{x_l}k(\boldsymbol{x},\boldsymbol{y}) + \partial_{x_l}\partial_{y_l}k(\boldsymbol{x},\boldsymbol{y})$. We can substitute $\boldsymbol{\nu}^{(l)\star}$ from equation (5.35) into $\mathcal{L}^2$, denoting $\mathcal{L}^{\star 2} := \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{g}^\star, \boldsymbol{\nu}^{(1)\star}, \ldots, \boldsymbol{\nu}^{(d)\star}, \lambda)$, giving

$$\begin{aligned}\mathcal{L}^{\star 2} &= \sum_{l=1}^{d} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim q(\boldsymbol{x})}[u_l(\boldsymbol{x},\boldsymbol{y})] \\ &+ \left(\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\psi_{\boldsymbol{x},l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}^\top + (\partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'})^\top\right]\right)^\top \left(-(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}\sim q(\boldsymbol{x})}\left[\psi_{\boldsymbol{y},l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top + (\partial_{y_l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'})^\top\right]\right) \\ &+ \left(-(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\psi_{\boldsymbol{x},l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}^\top + (\partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'})^\top\right]\right)^\top \mathbb{E}_{\boldsymbol{y}\sim q(\boldsymbol{x})}\left[\psi_{\boldsymbol{y},l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top + (\partial_{y_l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'})^\top\right] \\ &+ \left(-(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\psi_{\boldsymbol{x},l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}^\top + (\partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'})^\top\right]\right)^\top \mathbf{K}' \left(-(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}\sim q(\boldsymbol{x})}\left[\psi_{\boldsymbol{y},l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top + (\partial_{y_l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'})^\top\right]\right),\end{aligned} \tag{5.37}$$

where, for brevity, we have written $\psi_{\boldsymbol{x},l} = \partial_{x_l}\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $\psi_{\boldsymbol{y},l} = \partial_{y_l}\log p_{\boldsymbol{\theta}}(\boldsymbol{y})$. This can be further simplified to

$$\begin{aligned}\mathcal{L}^{\star 2} = \sum_{l=1}^{d} &\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim q(\boldsymbol{x})}[u_l(\boldsymbol{x},\boldsymbol{y})] \\ &- \left(\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\psi_{\boldsymbol{x},l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}^\top + (\partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'})^\top\right]\right)^\top (\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}\sim q(\boldsymbol{x})}\left[\psi_{\boldsymbol{y},l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top + (\partial_{y_l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'})^\top\right],\end{aligned} \tag{5.38}$$

and equivalently

$$\mathcal{L}^{\star 2} = \sum_{l=1}^{d} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim q(\boldsymbol{x})}\left[u_l(\boldsymbol{x},\boldsymbol{y}) - \left(\psi_{\boldsymbol{x},l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}^\top + (\partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'})^\top\right)^\top (\mathbf{K}')^{-1}\left(\psi_{\boldsymbol{y},l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top + (\partial_{y_l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'})^\top\right)\right], \tag{5.39}$$

giving the desired result. ■

5.8.6 Proof of theorem 5.11

First, we state a general result for proving the consistency of an empirical estimator of $\boldsymbol{\theta}$, which is second-order differentiable with respect to $\boldsymbol{\theta}$.

Lemma 5.12. *Define the unique solution of empirical objective* $\hat{\boldsymbol{\theta}} := \arg\min_{\boldsymbol{\theta}} \hat{L}(\boldsymbol{\theta})$ *and the population* $\boldsymbol{\theta}^\star := \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$*, where* $L(\boldsymbol{\theta}) := \mathbb{E}[\hat{L}(\boldsymbol{\theta})]$*. If* $\lambda_{\min}\left[\nabla^2_{\boldsymbol{\theta}}\hat{L}(\boldsymbol{\theta})\right] \geq \Lambda_{\min} > 0, \forall \boldsymbol{\theta}$ *and* $\|\nabla_{\boldsymbol{\theta}}\hat{L}(\boldsymbol{\theta}^\star)\| = O_P(\frac{1}{\sqrt{n}})$*, then* $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\| = O_P(\frac{1}{\sqrt{n}})$*. Here,* $\lambda_{\min}(M)$ *is the smallest eigenvalue of a matrix* M*.*

Proof. First, we define a function $h(\boldsymbol{\theta}) := \langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star, \hat{L}(\boldsymbol{\theta}) \rangle$. Using the mean value theorem, we obtain $h(\hat{\boldsymbol{\theta}}) - h(\boldsymbol{\theta}^\star) = \langle \nabla_{\boldsymbol{\theta}} h(\bar{\boldsymbol{\theta}}), \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star \rangle$, where $\bar{\boldsymbol{\theta}}$ is the midpoint between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}^\star$ in a co-ordinate wise fashion. Therefore, due to the fact that $\nabla_{\boldsymbol{\theta}} \hat{L}(\hat{\boldsymbol{\theta}}) = 0$,

$$0 - h(\boldsymbol{\theta}^\star) = \langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star, \nabla_{\boldsymbol{\theta}} \hat{L}(\boldsymbol{\theta}^\star) \rangle = \langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star, \nabla^2_{\boldsymbol{\theta}} \hat{L}(\bar{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star) \rangle.$$

This implies the following inequality

$$\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\| \|\nabla_{\boldsymbol{\theta}} \hat{L}(\boldsymbol{\theta}^\star)\| \geq \|(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star)^\top \boldsymbol{\nabla}^2_{\boldsymbol{\theta}} \hat{L}(\hat{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star)\| \geq \lambda_{\min} \|(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star)\|^2.$$

Assume $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\| \neq 0$, $\frac{1}{\lambda_{\min}} \|\nabla_{\boldsymbol{\theta}} \hat{L}(\boldsymbol{\theta}^\star)\| \geq \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|^2$. ■

For the remainder, we need only to show that $J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2$ satisfies lemma 5.12.

Proof. First, we need to show that

$$\left[\nabla_{\boldsymbol{\theta}} J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2 \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} \right]_i = O_P(\hat{\varepsilon}_m), \forall i. \tag{5.40}$$

For brevity let us define $\mathbb{E}_{\boldsymbol{x}} = \mathbb{E}_{\boldsymbol{x} \sim q(\boldsymbol{x})}$ and similarly $\mathbb{E}_{\boldsymbol{y}} = \mathbb{E}_{\boldsymbol{y} \sim q(\boldsymbol{x})}$. Recall

$$J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2 = \sum_{l=1}^{d} \mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{y}} \left[u_l(\boldsymbol{x}, \boldsymbol{y}) - \mathbf{v}_l(\boldsymbol{x})^\top (\mathbf{K}')^{-1} \mathbf{v}_l(\boldsymbol{y}) \right] \tag{5.41}$$

where

$$\begin{aligned} u_l(\boldsymbol{x}, \boldsymbol{y}) &= \psi_{p,l}(\boldsymbol{x}) \psi_{p,l}(\boldsymbol{y}) k(\boldsymbol{x}, \boldsymbol{y}) + \psi_{p,l}(\boldsymbol{x}) \partial_{y_l} k(\boldsymbol{x}, \boldsymbol{y}) + \psi_{p,l}(\boldsymbol{y}) \partial_{x_l} k(\boldsymbol{x}, \boldsymbol{y}) + \partial_{x_l} \partial_{y_l} \boldsymbol{y}), \\ \mathbf{v}_l(\boldsymbol{z}) &= \psi_{p,l}(\boldsymbol{z}) \boldsymbol{\phi}^\top_{\boldsymbol{z}, \mathbf{x}'} + (\partial_{z_l} \boldsymbol{\phi}_{\boldsymbol{z}, \mathbf{x}'})^\top, \end{aligned}$$

$\boldsymbol{\phi}_{\boldsymbol{z}, \mathbf{x}'} = [k(\boldsymbol{z}, \boldsymbol{x}'_1), \ldots, k(\boldsymbol{z}, \boldsymbol{x}'_m)]$, $\boldsymbol{\phi}_{\mathbf{x}'} = [k(\boldsymbol{x}'_1, \cdot), \ldots, k(\boldsymbol{x}'_m, \cdot)]^\top$ and $\mathbf{K}' = \boldsymbol{\phi}_{\mathbf{x}'} \boldsymbol{\phi}^\top_{\mathbf{x}'}$. Let us take the derivative of (5.41) with respect to $\boldsymbol{\theta}$. We first consider for the l-th dimension of the sum, and then note that this will apply to all d dimensions. Therefore consider

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2 \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{y}} \left[u_l(\boldsymbol{x}, \boldsymbol{y}) - \mathbf{v}_l(\boldsymbol{x})^\top (\mathbf{K}')^{-1} \mathbf{v}_l(\boldsymbol{y}) \right] \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} \\ &= \mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{y}} \left[\nabla_{\boldsymbol{\theta}} u_l(\boldsymbol{x}, \boldsymbol{y}) - \boldsymbol{v}_l(\boldsymbol{x})^\top (\mathbf{K}')^{-1} (\nabla_{\boldsymbol{\theta}} \mathbf{v}_l(\boldsymbol{y})) - (\nabla_{\boldsymbol{\theta}} \boldsymbol{v}_l(\boldsymbol{x}))^\top (\mathbf{K}')^{-1} \mathbf{v}_l(\boldsymbol{y}) \right] \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}. \end{aligned} \tag{5.42}$$

We begin by expanding the first term in (5.42):

$$\begin{aligned} \mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{y}} \left[\nabla_{\boldsymbol{\theta}} u_l(\boldsymbol{x}, \boldsymbol{y}) \right] \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} = \mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{\boldsymbol{y}} \big[& \psi_{p,l}(\boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{y}) (\nabla_{\boldsymbol{\theta}} \psi_{p,l}(\boldsymbol{y})) + (\nabla_{\boldsymbol{\theta}} \psi_{p,l}(\boldsymbol{x})) k(\boldsymbol{x}, \boldsymbol{y}) \psi_{p,l}(\boldsymbol{y}) \\ & + (\nabla_{\boldsymbol{\theta}} \psi_{p,l}(\boldsymbol{x})) \partial_{y_l} k(\boldsymbol{x}, \boldsymbol{y}) + (\nabla_{\boldsymbol{\theta}} \psi_{p,l}(\boldsymbol{y})) \partial_{x_l} k(\boldsymbol{x}, \boldsymbol{y}) \big] \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}. \end{aligned} \tag{5.43}$$

Now expand the first term of (5.43), giving

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{y}}\left[\psi_{p,l}(\boldsymbol{x})k(\boldsymbol{x},\boldsymbol{y})(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y}))\right]\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} &= \int_V q(\boldsymbol{x})\psi_{p,l}(\boldsymbol{x})\mathbb{E}_{\boldsymbol{y}}\left[k(\boldsymbol{x},\boldsymbol{y})(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y}))\right]d\boldsymbol{x}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} \\ &= \int_V \partial_{x_l}q(\boldsymbol{x})\mathbb{E}_{\boldsymbol{y}}\left[k(\boldsymbol{x},\boldsymbol{y})[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}\right]d\boldsymbol{x} \qquad (5.44)\end{aligned}$$

due to

$$q(\boldsymbol{z})\psi_{p,l}(\boldsymbol{z})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} = q(\boldsymbol{z})\partial_{y_l}\log p_{\boldsymbol{\theta}^\star}(\boldsymbol{z}) = q(\boldsymbol{z})\frac{\partial_{y_l}q(\boldsymbol{z})}{q(\boldsymbol{z})} = \partial_{y_l}q(\boldsymbol{z}).$$

Then, via integration by parts, (5.44) becomes

$$\oint_{\partial V} q(\boldsymbol{x})\mathbb{E}_{\boldsymbol{y}}\left[k(\boldsymbol{x},\boldsymbol{y})[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}\right]\hat{\mathrm{u}}_l(\boldsymbol{x})ds - \int_V q(\boldsymbol{x})\left[\partial_{x_l}k(\boldsymbol{x},\boldsymbol{y})[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}\right].$$

We can expand the second term of (5.43) in a similar way:

$$\begin{aligned}(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{x}))k(\boldsymbol{x},\boldsymbol{y})\psi_{p,l}(\boldsymbol{y}) = \oint_{\partial V} q(\boldsymbol{y})\mathbb{E}_{\boldsymbol{x}}\left[[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{x})]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}k(\boldsymbol{x},\boldsymbol{y})\right]\hat{\mathrm{u}}_l(\boldsymbol{y})ds \\ - \int_V q(\boldsymbol{y})\left[[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{x})]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}\partial_{y_l}k(\boldsymbol{x},\boldsymbol{y})\right]\end{aligned}$$

Both of these together, as they appear in (5.43), gives

$$\begin{aligned}B := \oint_{\partial V} q(\boldsymbol{x})\mathbb{E}_{\boldsymbol{y}}\left[k(\boldsymbol{x},\boldsymbol{y})[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}\right]\hat{\mathrm{u}}_l(\boldsymbol{x})ds \\ + \oint_{\partial V} q(\boldsymbol{y})\mathbb{E}_{\boldsymbol{x}}\left[[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{x})]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}k(\boldsymbol{x},\boldsymbol{y})\right]\hat{\mathrm{u}}_l(\boldsymbol{y})ds.\end{aligned} \qquad (5.45)$$

Now consider the second term in (5.42),

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{y}}\left[\boldsymbol{v}_l(\boldsymbol{x})^\top(\mathbf{K}')^{-1}(\nabla_{\boldsymbol{\theta}}\mathbf{v}_l(\boldsymbol{y}))\right]\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} \qquad (5.46)\\ &= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{y}}\left[(\psi_{p,l}(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'} + \partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'})(\mathbf{K}')^{-1}(\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y}))^\top)\right]\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} \\ &= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{y}}\Big[\psi_{p,l}(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}(\mathbf{K}')^{-1}(\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y}))^\top) \qquad (5.47)\\ &\qquad + (\partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'})(\mathbf{K}')^{-1}(\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y}))^\top))\Big]\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} \\ &= \mathbb{E}_{\boldsymbol{x}}\left[\psi_{p,l}(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}}\left[\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})^\top)\right]\right]\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} \\ &\qquad + \mathbb{E}_{\boldsymbol{x}}\left[(\partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'})(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}}\left[\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^\top(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y}))^\top\right)\right]\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star} \qquad (5.48)\end{aligned}$$

Consider only the first term of the above,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x}}\left[\psi_{p,l}(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}}\left[(\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^{\top}(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y}))^{\top})\right]\right]\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}& \\ = \int_V q(\boldsymbol{x})\psi_{p,l}(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}}\left[\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^{\top}(\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y}))^{\top}\right]d\boldsymbol{x}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}& \\ = \int_V \partial_{x_l}q(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}}\left[\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^{\top}[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})^{\top}]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}\right]d\boldsymbol{x}.&\end{aligned}$$

Integration by parts can be used to expand this to

$$\begin{aligned}\oint_{\partial V} q(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}}\left[\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^{\top}[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})^{\top}]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}\right]\hat{\mathrm{u}}_l(\boldsymbol{x})ds& \\ - \int_V q(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}}\left[\partial_{x_l}\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^{\top}[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})^{\top}]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}\right]\nu(\boldsymbol{x})d\boldsymbol{x}.&\end{aligned} \tag{5.49}$$

Assumption 5.9 indicates that we have the following equivalence (in a coordinate-wise fashion):

$$\begin{aligned}\oint_{\partial V} q(\boldsymbol{x})\boldsymbol{\phi}_{\boldsymbol{x},\mathbf{x}'}(\mathbf{K}')^{-1}\mathbb{E}_{\boldsymbol{y}}\left[\boldsymbol{\phi}_{\boldsymbol{y},\mathbf{x}'}^{\top}[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})^{\top}]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}\right]\hat{\mathrm{u}}_l(\boldsymbol{x})ds& \\ = \oint_{\partial V} q(\boldsymbol{x})\mathbb{E}_{\boldsymbol{y}}\left[k(\boldsymbol{x},\boldsymbol{y})[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})^{\top}]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}\right]\hat{\mathrm{u}}_l(\boldsymbol{x})ds + O_P(\hat{\varepsilon}_{m,1}).&\end{aligned} \tag{5.50}$$

We interpret this as follows. If we consider $\mathbf{x}' \in \widetilde{\partial V}$ as our training set, then our regression function is trained on the approximate set of boundary points. The surface integral denoted by $\oint_{\partial V}$ is evaluating on all $\boldsymbol{x} \in \partial V$, the true boundary. Therefore the prediction based on $\boldsymbol{x} \in \partial V$ is going to be well approximated by the regression function that is trained on $\mathbf{x}'$, and the approximation error, $\hat{\varepsilon}_{m,1}$, will decrease as m, the number of training points, increases.

We can apply the same operations (from (5.48) onwards) to the third term in (5.42), which gives

$$\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{y}}\left[(\nabla_{\boldsymbol{\theta}}\boldsymbol{v}_l(\boldsymbol{x}))^{\top}(\mathbf{K}')^{-1}\mathbf{v}_l(\boldsymbol{y})\right] = \oint_{\partial V} q(\boldsymbol{y})\mathbb{E}_{\boldsymbol{x}}\left[[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{x})^{\top}]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}k(\boldsymbol{x},\boldsymbol{y})\right]\hat{\mathrm{u}}_l(\boldsymbol{y})ds + O_P(\hat{\varepsilon}_{m,2}), \tag{5.51}$$

where $\hat{\varepsilon}_{m,2}$ is the approximation error for the corresponding kernel regression on this term. When taking the sum, as it is in (5.42), (5.50) + (5.51),

$$\begin{aligned}\oint_{\partial V} q(\boldsymbol{x})\mathbb{E}_{\boldsymbol{y}}\left[k(\boldsymbol{x},\boldsymbol{y})[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{y})^{\top}]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}\right]\hat{\mathrm{u}}_l(\boldsymbol{x})ds + O_P(\hat{\varepsilon}_{m,1})& \\ + \oint_{\partial V} q(\boldsymbol{y})\mathbb{E}_{\boldsymbol{x}}\left[[\nabla_{\boldsymbol{\theta}}\psi_{p,l}(\boldsymbol{x})^{\top}]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}}k(\boldsymbol{x},\boldsymbol{y})\right]\hat{\mathrm{u}}_l(\boldsymbol{y})ds + O_P(\hat{\varepsilon}_{m,2}) = B + O_P(\hat{\varepsilon}_m),&\end{aligned}$$

where $\hat{\varepsilon}_m = \hat{\varepsilon}_{m,1} + \hat{\varepsilon}_{m,2}$. Therefore, when substituting all of (5.45), (5.50) and (5.51) back into (5.42), we have

$$\nabla_{\boldsymbol{\theta}}J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\star}} = O_P(\hat{\varepsilon}_m),$$

where $\hat{\varepsilon}_m$ is a decreasing function of m.

Now we show $\boldsymbol{\theta}^\star$ is the true density parameter, i.e., $p_{\boldsymbol{\theta}^\star} = q$. Let $J_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star) := J_{\mathrm{TKSD}}(\boldsymbol{\theta})^2|_{\boldsymbol{\theta}=\boldsymbol{\theta}^\star}$. First, we show that $L(\boldsymbol{\theta}^\star) = 0$. This is guaranteed as

$$\begin{aligned} L(\boldsymbol{\theta}^\star) &= \lim_{m\to\infty} J_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2 \\ &= \lim_{m\to\infty} \left\{ \sup_{\boldsymbol{g}\in\mathcal{G}^d_{0,m}} \mathbb{E}_q[\mathcal{T}_{p_{\boldsymbol{\theta}^\star}}\boldsymbol{g}] \right\}^2 \\ &= \lim_{m\to\infty} O_P(\varepsilon_m) = 0 \text{ (with high probability)}, \end{aligned}$$

for $\boldsymbol{g} \in \mathcal{G}^d_{0,m}$. The change from Stein discrepancy to $O_P(\varepsilon_m)$ is guaranteed by Theorem 5.6. Since we assume that the minimiser of the population objective $\boldsymbol{\theta}^\star$ is unique and our density model is identifiable, it shows that the unique solution of the population objective is the unique optimal parameter of the density model.

Second, we verify that $\|\nabla_{\boldsymbol{\theta}}\hat{L}(\boldsymbol{\theta}^\star)\| = O_P(\frac{1}{\sqrt{n}})$.

$$\begin{aligned} \|\nabla_{\boldsymbol{\theta}}\hat{L}(\boldsymbol{\theta}^\star)\| &= \|\nabla_{\boldsymbol{\theta}} \lim_{m\to\infty} \hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2\| \\ &= \lim_{m\to\infty} \|\nabla_{\boldsymbol{\theta}} \hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2\| \\ &= \lim_{m\to\infty} \|\nabla_{\boldsymbol{\theta}} \hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2 - \nabla_{\boldsymbol{\theta}} J_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2 + \nabla_{\boldsymbol{\theta}} J_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2\| \\ &\leq \lim_{m\to\infty} \left(\|\nabla_{\boldsymbol{\theta}} \hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2 - \nabla_{\boldsymbol{\theta}} J_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2\| + \|\nabla_{\boldsymbol{\theta}} J_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2\| \right) \\ &\leq \lim_{m\to\infty} \left(\|\nabla_{\boldsymbol{\theta}} \hat{J}_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2 - \nabla_{\boldsymbol{\theta}} J_{\mathrm{TKSD}}(\boldsymbol{\theta}^\star)^2\| + O_P(\hat{\varepsilon}_m) \right) \\ &\overset{(a)}{\leq} \lim_{m\to\infty} \left(O_P\left(\frac{1}{\sqrt{n}}\right) + O_P(\hat{\varepsilon}_m) \right) \\ &\overset{(b)}{=} O_P\left(\frac{1}{\sqrt{n}}\right), \end{aligned} \tag{5.52}$$

where the inequality in (a) is due to the convergence of U-statistics (Serfling, 2009), and the equality in (b) is due to $\lim_{m\to\infty} O_P(\hat{\varepsilon}_m) = 0$.

■

We have $\|\nabla_{\boldsymbol{\theta}}\hat{L}(\boldsymbol{\theta}^\star)\| = O_P(\frac{1}{\sqrt{n}})$, and assumption 5.10 provides $\lambda_{\min}\left[\nabla^2_{\boldsymbol{\theta}}\hat{L}(\boldsymbol{\theta})\right] \geq \Lambda_{\min} > 0$, then both conditions in lemma 5.12 are satisfied, and this concludes the proof of the consistency of $\hat{\boldsymbol{\theta}}_n$, i.e., $\|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^\star\| = O_P(\frac{1}{\sqrt{n}})$.

Chapter 6

Time Score Matching for Parameter Change Estimation and Changepoint Detection

6.1 Introduction

The change of a model parameter over time is in a lot of cases of more interest than the parameter itself. For instance, in changepoint detection, if the parameter change exceeds a certain threshold, a changepoint is declared (Aminikhanghahi and Cook, 2017) which segments the data into distinct regions. This can have meaningful intuition across applications such as climate change detection (Reeves et al., 2007), speech recognition (Bhandari et al., 2013), medical data segmentation (Malladi et al., 2013), amongst many others.

Most changepoint detection methods are designed to detect abrupt changes in discrete time, where an actual change could be more subtle and happen more gradually across a long time period. To this extent, estimating the *derivative* of the parameters that govern a distribution changing over time would inform fully about the trends of our data, as well as any changes that might occur.

Score matching (Hyvärinen, 2005, 2007) is a powerful density estimation method that does not require a tractable normalising constant, admitting a broad class of models. A recent development, called time score matching (Choi et al., 2022), can estimate the change of a distribution over time. Despite its huge potential in changepoint detection, time score matching has been understudied in prior works, limited only to use in density ratio estimation.

In this paper, we adapt the time score matching objective by incorporating elements of truncated score matching (Liu et al., 2022; Yu et al., 2021), and show its potential in estimating model parameter derivatives. We show that this score matching variant can also handle intractable models, which enables complex model parameterisations.

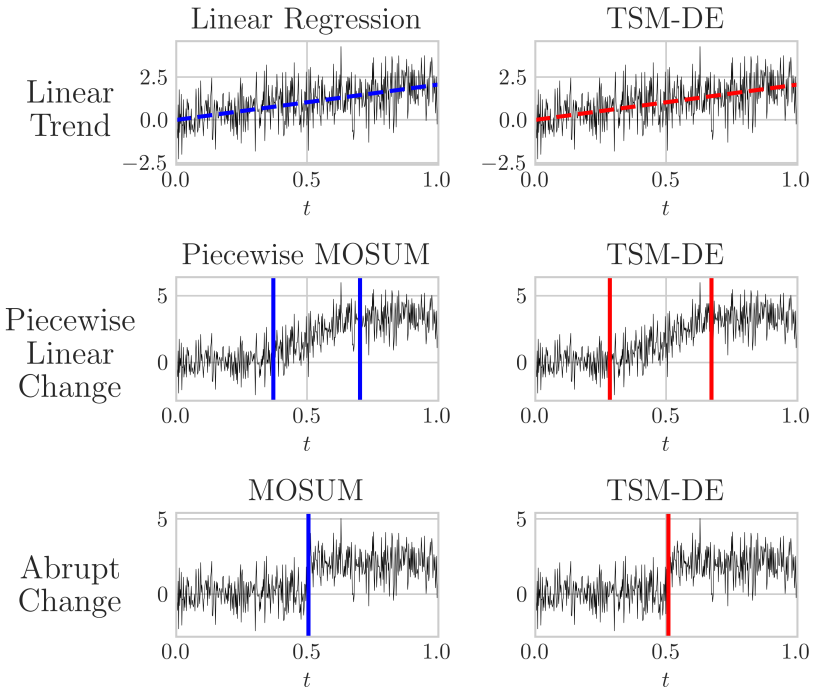


Figure 6.1. Usually, one would need to employ specific methods for these three distinct tasks, but our method, TSM-DE, estimates $\partial_t \boldsymbol{\theta}(t)$, and thus can capture *a linear trend* (top), detect piecewise linear changes (middle), and an abrupt change (bottom).

6.1.1 Method Overview and Motivation

Let $p_t = p(\boldsymbol{x}; \boldsymbol{\theta}(t))$ be a time-varying probability density model, where the time dependence is entirely through $\boldsymbol{\theta}(t)$. We seek to estimate changes in $\boldsymbol{\theta}(t)$ over time and develop a detection algorithm to identify locations of significant deviations from stationarity. In this work, we do not *a priori* assume the form of $\boldsymbol{\theta}(t)$: $\boldsymbol{\theta}(t)$ could be a piecewise constant function or a smooth changing function over time t.

One can estimate $\boldsymbol{\theta}(t)$ using time-varying density estimation, but it is not obvious how to measure a significant deviation from stationarity and therefore detect changes. It may also be impossible to estimate $\boldsymbol{\theta}(t)$ when p_t is an unnormalisable model. Furthermore, density estimation is not ideal if interest only lies in *the change of parameters*, as it would also involve modelling additional parameters that do not change over time. *Changepoint detection* relies on an *a priori* assumption on the type of change that occurs, e.g. abrupt, piecewise linear or otherwise. More importantly, these methods are not designed to learn changes in the parameters directly.

Our method, Time Score Matching Derivative Estimation (TSM-DE), estimates $\partial_t \boldsymbol{\theta}(t)$ directly, which is a natural quantification of changes and holds meaningful intuition. We also derive

an asymptotic distribution for $\partial_t \boldsymbol{\theta}(t)$ under a stationarity assumption, enabling TSM-DE to detect significant deviations from stationarity as changepoints. This flexible formulation makes TSM-DE versatile for changepoint detection across multiple scenarios, with only minimal assumptions required. figure 6.1 shows three examples of using TSM-DE for three distinct tasks, using the exact same TSM-DE algorithm.

6.2 Background

We review some fundamental prior works and methodologies that are relevant to our method. Firstly, score matching (section 2.1) plays a fundamental role, and we will begin by discussing an extension of score matching given by Choi et al. (2022).

6.2.1 Time Score Matching

Consider a data distribution whose probability density function changes over time $t \in [0, 1]$, denoted by $q_t(\boldsymbol{x})$. Choi et al. (2022) proposed the time score matching objective to estimate the time score, given by

$$J_{\text{time}}(\boldsymbol{\theta}) = \mathbb{E}_{q(t)}\mathbb{E}_{q_t(\boldsymbol{x})}\left[h(t)\|\dot{\psi}_{q_t}(\boldsymbol{x}) - s_0(\boldsymbol{x}, t)\|^2\right], \tag{6.1}$$

where $\dot{\psi}_{q_t}(\boldsymbol{x}) = \partial_t \log q_t(\boldsymbol{x})$ is the *time score* function, $s_0(\boldsymbol{x}, t)$ is a generic model intended to approximate $\dot{\psi}_{q_t}(\boldsymbol{x})$, $q(t)$ is a uniform distribution over timescales, and $h(t) : [0, 1] \to \mathbb{R}^+$ is a positive weighting function. Choi et al. (2022) propose to use a neural network as $s_0(\boldsymbol{x}, t)$ to flexibly model $\dot{\psi}_{q_t}(\boldsymbol{x})$. Under regularity conditions, this objective also can be expanded using integration by parts to a tractable form for use in practice.

6.2.2 Change Detection and Parameter Change Estimation

In this section, we review two lines of work that are relevant to our algorithm.

6.2.2.1 Changepoint Detection

Changepoint detection, or data segmentation, is a long developed and well studied field. In the traditional offline setting, we observe

$$x_t = \mu_t + \epsilon_t, \qquad \mu_t = \begin{cases} \mu_1, & \text{if } 0 < t < \tau \\ \mu_2, & \text{if } \tau \leq t \leq t_{\text{end}} \end{cases}$$

where μ_1 and μ_2 are constants, ϵ_t is a zero mean process, τ is a changepoint, and t_{end} is the length of the time series. Whilst not exhaustive, we will briefly outline the most noteworthy offline detection methods that are relevant to this work. Aminikhanghahi and Cook (2017) provide a more comprehensive overview of changepoint detection.

Most changepoint detection methods can be classified as moving window procedures, which scan over time series and, for a given window size, check for a statistic that deviates from the

nominal status significantly. For example, the moving sum (MOSUM) procedure (Bauer and Hackl, 1980; Eichinger and Kirch, 2018; Meier et al., 2021) checks for a significantly non-zero difference between moving sum statistics before and after a candidate changepoint, and is a simple and efficient algorithm for mean change detection. Moving window methodologies have been developed for time series that are autoregressive (Owens, 2024; Yau and Zhao, 2016), smoothly changing (Vogt and Dette, 2015), piecewise linear changing (Kim et al., 2024), high-dimensional and conditional Cho and Owens (2022) and even for changes in Gaussian process regression parameters (Caldarelli et al., 2022). A more general moving window framework has been studied by Kirch and Reckruehm (2022).

Other methods include the Pruned Exact Linear Time (PELT) method (Jackson et al., 2005; Killick et al., 2012), which partitions time series data by minimising a general statistical criterion function over all partitions in linear time, and non-parametric methods that detect changepoints in a kernel-transformed space (Arlot et al., 2019; Celisse et al., 2018).

6.2.2.2 Time-Varying Distributions

Suppose that p_t is the density of a time varying probability distribution. Assume p_t depends on t only through a parameter vector $\boldsymbol{\theta}(t)$. Classical time series modelling usually assumes that $\boldsymbol{\theta}(t)$ is a discrete-time autoregressive process (e.g. Box et al. (2015)), such that $\boldsymbol{\theta}(t)$ depends on one or several previous parameters. Other works, for example Kolar and Xing (2011) and Lu et al. (2022), estimate the time-dependent structure of specific density functions.

Another approach learns $\boldsymbol{\theta}(t)$ via conditional density estimation, for example, we assume $\boldsymbol{\theta}(t) = \boldsymbol{f}(t; \boldsymbol{\beta})$, where $\boldsymbol{f}$ is a linear or nonlinear function and is estimated by methods such as maximum likelihood estimation. Lindeløv (2020) detail a Bayesian framework that estimates $\boldsymbol{f}(t; \boldsymbol{\beta})$ whilst simultaneously estimating changepoints to partition the time series. This conditional density estimation approach can of course be extended to more flexible models of $\boldsymbol{\theta}(t)$, such as generalised additive models (Dominici et al., 2002; Hastie and Tibshirani, 1986) and reproducing kernel Hilbert space models (Lampert, 2015).

6.3 Estimating Parameter Change in the Exponential Family using Score Matching

Consider time series variables consisting of pairs $(\boldsymbol{x}, t)$, where $\boldsymbol{x} \sim q_t(\boldsymbol{x})$, a density function that changes over time t, and $t \in \mathcal{T}$. For convenience of notation, let $\mathcal{T} = [0, 1]$.

6.3.1 Density Change in the Exponential Family

Let us assume that the true data density function $q_t(\boldsymbol{x})$ is in the Exponential family, i.e. parameterised as

$$q_t(\boldsymbol{x}) = q(\boldsymbol{x}; \boldsymbol{\theta}^\star(t), \boldsymbol{\varphi}^\star) = \frac{\exp\left(\langle \boldsymbol{\theta}^\star(t), \boldsymbol{f}(\boldsymbol{x}) \rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x}) \rangle\right)}{Z(\boldsymbol{\theta}^\star(t), \boldsymbol{\varphi}^\star)}, \tag{6.2}$$

for an unknown normalising constant $Z(\boldsymbol{\theta}^\star(t), \boldsymbol{\varphi}^\star)$. We assume that $q_t(\boldsymbol{x}) = q(\boldsymbol{x}; \boldsymbol{\theta}^\star(t), \boldsymbol{\varphi}^\star)$ changes over time only through the parameter vector $\boldsymbol{\theta}^\star(t)$, but the entire distribution is parameterised by both $\boldsymbol{\theta}^\star(t)$ and $\boldsymbol{\varphi}^\star$, where $\boldsymbol{\varphi}^\star$ does not depend on t. We denote $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^k$ and $\boldsymbol{h} : \mathbb{R}^d \to \mathbb{R}^{k'}$ as the *feature functions*, for which choice of $\boldsymbol{f}$ and $\boldsymbol{h}$ induces different time-based distributions. For the remainder of the chapter, we will omit the dependence on $\boldsymbol{x}$ from $q_t(\boldsymbol{x})$ for brevity, and henceforth use $q_t = q_t(\boldsymbol{x})$.

Our method is motivated by the following observation:

Theorem 6.1. *Let q_t be defined as in equation (6.2), then the time score function for q_t can be written as*

$$\dot{\psi}_{q_t}(\boldsymbol{x}) = \partial_t \log q_t = \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x}) \rangle - \mathbb{E}_{q_t} \left[\langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x}) \rangle \right], \tag{6.3}$$

where we shorten $\boldsymbol{\theta}(t)$ to be $\boldsymbol{\theta}_t$.

For proof, see section 6.8.1. This result shows not only that the time score function $\dot{\psi}_{q_t}(\boldsymbol{x})$ needs no normalisation term, but it also only depends on the parameter derivative $\partial_t \boldsymbol{\theta}_t^\star$, and no longer involves $\boldsymbol{\varphi}^\star$ nor $\boldsymbol{h}$.

Since this formulation of $\dot{\psi}_{q_t}(\boldsymbol{x})$ no longer requires specification of either $\langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x}) \rangle$ nor the normalising constant $Z(\boldsymbol{\theta}^\star(t), \boldsymbol{\varphi}^\star)$, this admits a larger class of models. One only requires specification of the feature function $\boldsymbol{f}$, which is free to be chosen as flexibly as possible. For example, see section 6.6.3 where $\boldsymbol{f}$ is the output of a convolutional neural network.

Example If q_t is a univariate Gaussian density with time-dependent mean and variance, then equation (6.3) holds if $\boldsymbol{f}(x) = [x, x^2]$. Similarly, for fixed mean and time-dependent variance, $\boldsymbol{f}(x) = [x, x^2]$, and for fixed variance and time-dependent mean, $\boldsymbol{f}(x) = x$.

For proof, see section 6.8.2. Therefore, to model $\dot{\psi}_{q_t}(\boldsymbol{x})$ we propose to use the following

$$s(\boldsymbol{x}; t) \coloneqq \langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x}) \rangle - \mathbb{E}_{q_t} \left[\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x}) \rangle \right], \tag{6.4}$$

which requires learning only $\partial_t \boldsymbol{\theta}_t$. A general model of $\partial_t \boldsymbol{\theta}_t$ is discussed in section 6.3.4, and the choice of this model ultimately depends on the application, for which our proposals will be discussed in section 6.4.1.

6.3.2 Derivation of Score Matching Objective

Motivated by theorem 6.1, our method seeks to estimate the time score function $\dot{\psi}_{q_t}(\boldsymbol{x})$ via learning the parameter derivative $\partial_t \boldsymbol{\theta}_t$. Our approach in this section is based on time score matching (Choi et al., 2022). Whilst Choi et al. model $\dot{\psi}_{q_t}(\boldsymbol{x})$ as a neural network, we aim to model $\dot{\psi}_{q_t}(\boldsymbol{x})$ through the parameterisation given in equation (6.4), a function of $\partial_t \boldsymbol{\theta}_t$. Therefore we present a slight variation, given by

$$J_{\mathrm{C}}(\boldsymbol{\theta}_t) \coloneqq \int_{t=0}^{t=1} \mathbb{E}_{q_t} \left[g(t) \| \dot{\psi}_{q_t}(\boldsymbol{x}) - s(\boldsymbol{x}; t) \|^2 \right] dt \tag{6.5}$$

where $s(\boldsymbol{x}; t)$ is our parameterisation for the time score function, given by equation (6.4), and $g(t)$ is a weighting function chosen in advance such that $g(0) = g(1) = 0$, which is required in the integration by parts that follows to make this objective tractable. The choice of $g(t)$ is discussed later in section 6.4.6.

Theorem 6.2. *Let $g(t) = 0$ at $t = 0$ and $t = 1$. $J_{\mathrm{C}}(\boldsymbol{\theta}_t)$ can be written as*

$$J_{\mathrm{C}}(\boldsymbol{\theta}_t) = \int_{t=0}^{t=1} \mathbb{E}_{q_t}\Big[g_t s_t(\boldsymbol{x})^2 + 2\partial_t g_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x}) \right\rangle + 2 g_t \left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x}) \right\rangle \Big] dt + C, \tag{6.6}$$

where C is a constant independent of $s_t(\boldsymbol{x})$.

For proof, see section 6.8.3. For brevity, we have written $s(\boldsymbol{x}; t)$ as $s_t(\boldsymbol{x})$. The proof is based on two applications of integration by parts, and substituting our proposed model as given in equation (6.4).

This objective function depends on the parameters only through its derivatives, and serves as a natural objective for estimation of $\partial_t \boldsymbol{\theta}_t$ and $\partial_t^2 \boldsymbol{\theta}_t$. We title our method Time Score Matching for Derivative Estimation (TSM-DE), and denote equation (6.6) as the TSM-DE objective function.

6.3.3 Sample Objective

Equation (6.6) can be considered as a double expectation on a uniform distribution over time and on q_t. Both expectations can be approximated via a Monte-Carlo expectation. Given T timepoints, and n samples for each timepoint, i.e. the paired set of samples $\{(t_j, \{\boldsymbol{x}_{i,j}\}_{i=1}^n)\}_{j=1}^T$, where $t_j = (j-1)/T$ and $\boldsymbol{x}_{i,j} \sim q_{t_j}$, the sample objective is written as

$$\hat{J}_{\mathrm{C}}(\boldsymbol{\theta}_t) = \frac{1}{T}\sum_{j=1}^{T} \frac{1}{n}\sum_{i=1}^{n} g(t_j) s(\boldsymbol{x}, t_j)^2 + 2\partial_t g(t_j) \left\langle \partial_t \boldsymbol{\theta}(t_j), \boldsymbol{f}(\boldsymbol{x}_{i,j}) \right\rangle + 2 g(t_j) \left\langle \partial_t^2 \boldsymbol{\theta}(t_j), \boldsymbol{f}(\boldsymbol{x}_{i,j}) \right\rangle.$$

Choi et al. (2022) implement importance weighting to reduce variance in their estimator across time t. For our framework, and motivated by time series data often being defined across an equally spaced grid of time points, we propose to choose samples $\{t_j\}_{j=1}^T$ that are linearly spaced across the time interval $[0, 1]$, instead of randomly sampled.

6.3.4 Formulation of $\boldsymbol{\theta}_t$ and $\partial_t \boldsymbol{\theta}_t$

So far, the objective function given in equation (6.6) is not dependent on any particular formulation for $\partial_t \boldsymbol{\theta}_t$. Inspired by nonparametric modelling methods, we propose a modelling framework for $\boldsymbol{\theta}_t$ as $\boldsymbol{\theta}_t = \boldsymbol{\alpha}^\top \boldsymbol{\phi}(t)$, where $\boldsymbol{\alpha}$ is a $b \times d$ matrix, and the *time feature* function $\boldsymbol{\phi} : \mathbb{R} \to \mathbb{R}^b$ is at least twice differentiable. Here, b denotes the dimensionality of $\boldsymbol{\phi}$. Consequently, $\partial_t^{(l)} \boldsymbol{\theta}_t = \boldsymbol{\alpha}^\top (\partial_t^{(l)} \boldsymbol{\phi}(t))$, where $\partial_t^{(l)}$ denotes the l-th partial derivative with respect to time. This formulation is not a requirement for any of the results in section 6.3.1 or section 6.3.2, but is necessary for the derivation in the following section.

This proposed framework may lose some flexibility compared to other choices, but the complexity of $\boldsymbol{\phi}$ is not limited, and it can be a non-linear function. For practical use, our proposed choices for $\boldsymbol{\phi}$ and hence models for $\partial_t \boldsymbol{\theta}_t$ are described in section 6.4.1. This model observes strong empirical performance, as seen in sections 6.5 and 6.6.

6.3.5 Asymptotic Distribution of $\partial_t \widehat{\boldsymbol{\theta}}_t$

Consider a null hypothesis: the time series is stationary, i.e., $\partial_t \boldsymbol{\theta}_t^\star = \mathbf{0}$. The alternative hypothesis corresponds to a change happening at time t, i.e., $\partial_t \boldsymbol{\theta}_t^\star \neq \mathbf{0}$. The following theorem allows us to quantify the asymptotic null distribution of $\partial_t \widehat{\boldsymbol{\theta}}_t$ under this null hypothesis.

Assumption 6.3. *There exists a true parameter $\boldsymbol{\alpha}^\star$ such that $\boldsymbol{\theta}_t^\star = \boldsymbol{\alpha}^{\star\top}\boldsymbol{\phi}(t)$, and $J_{\mathrm{C}}(\boldsymbol{\theta}_t)$ is minimised at $\boldsymbol{\alpha}^\star$.*

Theorem 6.4. *Let assumption 6.3 hold, and let $\widehat{\boldsymbol{\alpha}}$ be a minimiser of equation* (6.7)*, and $\partial_t \widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\alpha}}^\top(\partial_t \boldsymbol{\phi}(t))$. As $n \to \infty$ and $T \to \infty$, the asymptotic distribution of $\partial_t \widehat{\boldsymbol{\theta}}_t$ under the null hypothesis that $\partial_t \boldsymbol{\theta}_t^\star = \mathbf{0}$ is given by*

$$\partial_t \widehat{\boldsymbol{\theta}}_t \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}_{\partial_t \widehat{\boldsymbol{\theta}}_t}\right),$$

where

$$\begin{aligned}
\boldsymbol{\Sigma}_{\partial_t \widehat{\boldsymbol{\theta}}_t} &= (\partial_t \boldsymbol{\phi}(t))^\top \boldsymbol{\Sigma}_A^{-1} \boldsymbol{\Sigma}_B \boldsymbol{\Sigma}_A^{-1} (\partial_t \boldsymbol{\phi}(t)), \\
\boldsymbol{\Sigma}_A &\coloneqq \sum_{j=1}^{T} \mathbb{E}_{q_t}\left[\nabla^2_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right], \\
\boldsymbol{\Sigma}_B &\coloneqq \sum_{j=1}^{T} \mathrm{Var}\left[\nabla_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right],
\end{aligned}$$

and

$$\begin{aligned}
\nabla_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star} &\coloneqq 2\left\langle \partial_t g_t(\partial_t \boldsymbol{\phi}(t))^\top + g_t(\partial_t^2 \boldsymbol{\phi}(t))^\top, \boldsymbol{f}(\boldsymbol{x}) \right\rangle, \\
\nabla^2_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star} &\coloneqq 2 g_t(\boldsymbol{f}(\boldsymbol{x}) - \mathbb{E}_{q_t}[\boldsymbol{f}(\boldsymbol{x})])(\boldsymbol{f}(\boldsymbol{x}) - \mathbb{E}_{q_t}[\boldsymbol{f}(\boldsymbol{x})])^\top (\partial_t \boldsymbol{\phi}(t))(\partial_t \boldsymbol{\phi}(t))^\top,
\end{aligned}$$

are the first and second derivatives of $m_{\boldsymbol{\alpha}}$ evaluated at $\boldsymbol{\alpha}^\star$.

For proof, see section 6.8.4. The proof revolves around first finding the asymptotic distribution for $\widehat{\boldsymbol{\alpha}}$ and transforming it to determine the asymptotic distribution for $\partial_t \widehat{\boldsymbol{\theta}}_t$. Whilst requiring evaluations at $\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star$, the asymptotic variance, $\boldsymbol{\Sigma}_{\partial_t \widehat{\boldsymbol{\theta}}_t}$, is in fact independent of $\boldsymbol{\alpha}^\star$. This asymptotic distribution allows us to quantify the variance of the estimator $\partial_t \widehat{\boldsymbol{\theta}}_t$ under the null hypothesis.

6.4 Practical Implementation

6.4.1 Proposed Models for $\partial_t \boldsymbol{\theta}_t$

Section 6.3.4 details the formulation of $\partial_t \boldsymbol{\theta}_t$ that enables the asymptotic distribution derived in theorem 6.4. Here, we propose some general frameworks to model $\partial_t \boldsymbol{\theta}_t$.

6.4.1.1 Examples of $\boldsymbol{\phi}(t)$

Linear Consider $\boldsymbol{\phi}(t) \coloneqq t$. Then estimating $\boldsymbol{\theta}(t) = \boldsymbol{\alpha} t$ corresponds to estimating a linear trend across time, and the model is estimating constant change. Therefore in modelling $\partial_t \boldsymbol{\theta}_t$, in practice we estimate $\boldsymbol{\alpha}$ and therefore a gradient term exclusively. This feature is used to estimate the linear trend in figure 6.1.

RBF Let $\boldsymbol{\phi}(t) \coloneqq [\phi_1(t), \dots, \phi_b(t)]^\top$, where $\phi_i(t)$ is a Radial Basis Function (RBF), given by

$$\phi_i(t) \coloneqq \exp\left\{ -\frac{\|t - \bar{t}_i\|^2}{2\sigma^2} \right\},$$

for all $i = 1, \dots, b$. Here, $\bar{t}_i$ is a centroid that is sampled uniformly across the time interval. This gives a more flexible parameterisation for $\partial_t \boldsymbol{\theta}_t$.

6.4.1.2 Sliding Window Linear Model

Inspired by the moving window methods for changepoint detection, consider partitioning the time series into a series of sliding window subsets. For a given time t, a subset is given by $\mathcal{S}_t = [t - w, t + w]$, for window size $0 < w < 1/2$. In each window, we assume a linear trend in the time series ($\boldsymbol{\phi}(t) \coloneqq t$) and estimate $\boldsymbol{\alpha}$ independently. Then, for all $t \in [w, 1 - w]$,

$$\boldsymbol{\theta}(t) = \boldsymbol{\alpha}_{\mathcal{S}_t} t,$$

where $\boldsymbol{\alpha}_{\mathcal{S}_t}$ denotes the parameter for subset $\mathcal{S}_t$.

6.4.2 Thresholding for Changepoint Detection

Theorem 6.4 allows us to quantify whether $\boldsymbol{\theta}(t)$ is stationary or non-stationary at time t, under our model $\partial_t \widehat{\boldsymbol{\theta}}_t$. We can use this result to identify periods of change across bounded time series data and hence for offline changepoint detection.

By modelling $\partial_t \boldsymbol{\theta}_t$, we obtain a natural quantity for changepoint detection which does not make any prior assumptions on the type of change encountered. Provided that $\partial_t \boldsymbol{\theta}_t$ is significantly large for some time t, then one can say there is a significant period of change at this time. The following corollary uses the asymptotic distribution of $\partial_t \boldsymbol{\theta}_t$, derived in section 6.3.5, to construct a detector to quantify when this significance level is reached.

Corollary 6.5. *Assume the same conditions as theorem 6.4, and define*

$$D(t) \coloneqq (\partial_t \widehat{\boldsymbol{\theta}}_t)^\top \boldsymbol{\Sigma}^{-1}_{\partial_t \widehat{\boldsymbol{\theta}}_t} (\partial_t \widehat{\boldsymbol{\theta}}_t). \tag{6.7}$$

then for any t*, as* $n \to \infty$ *and* $T \to \infty$*,* $D(t) \xrightarrow{d} \chi^2_k$.

The proof is straightforward and comes from properties of the quadratic form of multivariate normal variables. $D(t)$ is a one-dimensional measure that can test for violations of the null hypothesis.

Unlike traditional changepoint detection methods, we can define an entire *change period* as the set

$$\Delta_l \coloneqq \{t : D(t) > \chi^2_{k,1-\delta}\}, \tag{6.8}$$

for $l = 1, \ldots, \eta$, where η is the total number of change periods, and $\chi^2_{k,1-\delta}$ denotes the $(1-\delta)$-quantile of the χ^2_k distribution, for some significance level $\delta > 0$. Each period Δ_l surpasses the threshold independently and the time periods are disjoint. To detect a changepoint τ_l, we can take the local maxima of Δ_l, i.e.

$$\tau_l \coloneqq \underset{t \in \Delta_l}{\arg\max}\, D(t). \tag{6.9}$$

Further details of our changepoint detection process are described in the following section.

6.4.3 Changepoint Detection Algorithm Details

The detector, $D(t)$, given in equation (6.7), depends on a significance level $\delta > 0$. Smaller values of δ (e.g. $\delta = 0.01$) are likely to produce fewer false positives but risk not detecting the full period of change, whereas larger values (e.g. $\delta = 0.05$) will produce more false positives but will not risk losing potential true change periods. There is a trade-off between smaller and larger values of δ, which is standard in hypothesis testing. Choice of δ is up to the practitioner. In either case, we propose two additional frameworks to reduce the chance of false positives in the detection algorithm.

Small-peak criterion Setting the significance level δ means that for a given time t, there is a $1-\delta$ probability that the detector will exceed the threshold by chance. We propose to also disregard any change period Δ if

$$\max_t \Delta - \min_t \Delta < \varepsilon_{\text{SP}},$$

i.e. the total amount of time in the change period is less than ε_{SP}. Alternatively, this condition could be written as $|\Delta| < \varepsilon_{\text{SP}} T$.

Peak-proximity criterion We also implement an adapted version of the ε-criterion, as described in Meier et al. (2021). Given two periods of change Δ_1 and Δ_2 for which $t_1 \in \Delta_1$ and $t_2 \in \Delta_2$, this criterion classifies the two periods of change as the same period if

$$\min_{\substack{t_1 \in \Delta_1 \\ t_2 \in \Delta_2}} |t_1 - t_2| < \varepsilon_{\text{PP}},$$

i.e. the smallest difference between change period is less than ε_{PP}.

Choice of ε_{SP} and ε_{PP} depend greatly on the application. For example, for smaller δ, we recommend also choosing smaller ε_{SP} as one is less likely to observe false positives. Additionally, if changepoints are expected to be close together, we recommend smaller values of ε_{PP} to avoid mis-classifying two changepoints as one.

Throughout most of our experiments, we use $\varepsilon_{\text{SP}} = 0.01$ and $\varepsilon_{\text{PP}} = 0.02$ for significance level $\delta = 0.01$, with two exceptions. In our benchmarks in section 6.5.3, we expect fewer changepoints to be in close proximity, and therefore increase these values by a small amount, 0.05 each, and use $\varepsilon_0 = 0.015$ and $\varepsilon_1 = 0.025$. Section 6.6.2 contains three visually distinct changepoints, but the latter two changes would be misidentified as the same changepoint if $\varepsilon_{\text{PP}} = 0.02$. Using $\varepsilon_{\text{PP}} = 0.01$ correctly identifies these as distinct changes. In this case, these two changepoints are clearly visually separate from one another, and further, $\partial_t \widehat{\boldsymbol{\theta}}_t$ has a different sign for each change, highlighting that they are distinct.

6.4.4 ℓ_1 Regularisation

We observe that our estimator could be improved by instead performing the following optimisation

$$\min_{\boldsymbol{\alpha}} \hat{J}_{\text{C}}(\boldsymbol{\theta}_t) + \frac{\lambda}{T} \sum_{j=1}^{T} |\partial_t \boldsymbol{\theta}_t|, \tag{6.10}$$

i.e. apply a ℓ_1 regularisation penalty to the TSM-DE objective which penalises the absolute value of $\partial_t \boldsymbol{\theta}_t$. The strength of regularisation is controlled via the hyperparameter λ.

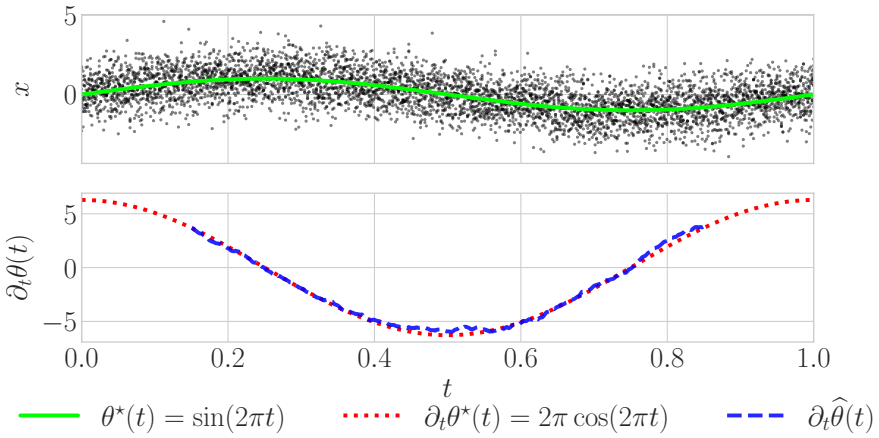
The ℓ_1 regularisation encourages $\partial_t \boldsymbol{\theta}_t$ to be zero when there is no change, making it suitable for changepoint detection. However, under this regularisation, the asymptotic variance of $\partial_t \widehat{\boldsymbol{\theta}}_t$ derived in section 6.3.5 (and subsequently section 6.4.2) is inaccurate. In experimental studies we have found that this asymptotic variance serves as a good approximation to the variance of the regularised estimator when λ is not too large.

In practice, we recommend larger values of λ to reduce the effect of spurious or insignificant changes when when the aim is to identify those changepoints with greater magnitude, and ignore smaller changes (e.g. outliers).

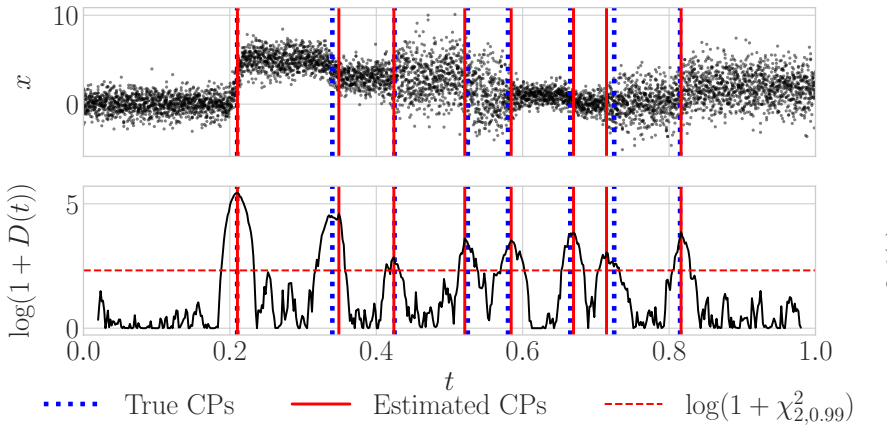
6.4.5 Improving the Finite Sample Expectation Approximation

The sample objective, as defined in equation (6.7), approximates the expectation using a finite set of samples of size n, where approximation accuracy improves as n increases. Time series data typically involves a single sample per time point; i.e. $n = 1$. In experiments, we find that TSM-DE performs well when n is small, however we present two approaches to address this minor limitation and enhance the usefulness of our methodology in this setting.

Binning One straightforward approach involves defining a bin size B, which partitions the time series into $\lfloor T/B \rfloor$ equally sized segments, so that each bin represents a set of samples at a single time point in the new time series.



(a) Estimating $\partial_t\theta(t)$ for a smoothly changing function.



(b) Changepoint detection for mean and variance.

Figure 6.2. Two illustrative examples of using TSM-DE for (a) parameter derivative estimation and (b) changepoint detection.

Nadarya-Watson estimator We can explore the weighting scheme known as Nadarya-Watson estimator (Nadaraya, 1964; Watson, 1964), which takes the form

$$\hat{\mathbb{E}}[a(\boldsymbol{x})|t = t_0] = \frac{\sum_{j=1}^{T} K(t_j, t_0)a(\boldsymbol{x})}{\sum_{j=1}^{T} K(t_j, t_0)}, \tag{6.11}$$

for any function a, where K is a kernel function such as the Gaussian kernel. This formulation resembles a form of weighted binning.

6.4.6 Choice of g Function

In Section 6.3.2, we specify the necessity for a function $g(t)$ with the condition that it equals zero at the boundaries of the time domain. For truncated score matching, Liu et al. (2022) propose a distance function as g, from which we take inspiration. Let t_{start} and t_{end} denote the start and end of the time domain respectively. We propose

$$g(t) \coloneqq -(t - t_{\text{start}})(t - t_{\text{end}}), \tag{6.12}$$

and consequently $\partial_t g_t = -(2t - t_{\text{start}} - t_{\text{end}})$. In experimental results, we have observed that the specific choice of g has minimal impact. In our setup so far, $t_{\text{start}} = 0$ and $t_{\text{end}} = 1$, but when using the sliding window linear model as described in section 6.4.1.2, t_{start} and t_{end} denote the beginning and end of each window.

6.4.7 Hyperparameter Selection

Depending on the model for $\partial_t \boldsymbol{\theta}_t$, there are a few hyperparameters that must be chosen for the TSM-DE method. These are

- ℓ_1-regularisation parameter λ.
- Nadarya-Watson estimator kernel and kernel hyperparameters. We always use a Gaussian kernel for the weighted expectation, which requires one bandwidth hyperparameter.
- For the *RBF time feature* (see section 6.4.1.1), the number of basis functions b, and kernel bandwidth parameter.
- For the *sliding window linear model* (see section 6.4.1.2), the window size w.

Provided that $n > 1$ (and if $n = 1$, then we can apply the binning approach described in section 6.4.5), we construct a validation set that spans all time points of size $n_{\text{val}} < n$ at each time point. Then, we perform a grid search over a specified set of hyperparameters, and choose those hyperparameters that minimise the testing loss, which is calculated on the sample objective, given in equation (6.7), which is un-penalised ($\lambda = 0$) and contains no weighted expectation via the Nadarya-Watson estimator.

In practice, we can apply this process to a single validation set across time. Alternatively, we can repeat this process multiple times across all possible validation sets, which corresponds to a form of cross-validation.

6.5 Numerical Experiments

Across all experiments, we choose $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x}$ for detecting changes in the mean only, and $\boldsymbol{f}(\boldsymbol{x}) = [\boldsymbol{x}, \boldsymbol{x}^2]$ for detecting changes in the variance, or both the mean and variance together. We also use significance level $\delta = 0.01$ across all experiments.

6.5.1 Other Method Implementation Details

In experiments in this section and the following section, we compare against three other methods, MOSUM, piecewise linear MOUSM and PELT, detailed in section 6.2.2.1. Our implementations of these methods are described below.

We implement MOSUM (Eichinger and Kirch, 2018; Meier et al., 2021) using the `mosum` package in python (Owens, 2023). We use bandwidth $G = T/6$ based on several factors. The first is that across simulations, we found this to be a well performing bandwidth. Secondly, the experiments considered in these experiments do not have close together changes, and we do not need a smaller bandwidth to cover that. Whilst the choice of bandwidth is important, we considered multiple options as different fractions of the time series length, and settled on this value as the consistently best performing one for experiments. For a more comprehensive overview of bandwidth selection, see Kirch and Reckruehm (2022).

We implement the Piecewise Linear MOSUM method described by Kim et al. (2024) for comparison, using the R implementation also provided by the authors, with the authors recommendation of multiple bandwidths as described in Kim et al. (2024).

To detect changes in the variance using MOSUM or piecewise linear MOSUM, we transform the time series $\{\boldsymbol{x}_i\}_{i=1}^T$ as follows

$$\widehat{\boldsymbol{x}}_i = \left(\boldsymbol{x}_i - \frac{1}{T}\sum_{i=1}^{T}\boldsymbol{x}_i\right)^2 \tag{6.13}$$

for all $i = 1, \ldots, T$, where T is the length of the time series. Then we apply MOSUM for mean-change detection on $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^T$.

To detect changes in both mean and variance is a combination of MOSUM for mean change detection and variance change detection. We run both MOSUM variations on the time series and combine the results afterwards.

We use PELT (Killick et al., 2012) with the RBF cost function for both mean and variance changes. The penalty on the number of changepoints we use is given by $2\log T$. We implement PELT with the `ruptures` package, see Truong et al. (2020) for details.

6.5.2 Illustrative Examples

figure 6.2 illustrates two examples of using TSM-DE for parameter derivative estimation and changepoint detection. Figure 6.2(a) shows Gaussian noise with smooth varying mean and we employ TSM-DE to estimate $\partial_t \mu(t)$ with the sliding window method. Figure 6.2(b) shows Gaussian noise over time with successive, quick jumps in the mean and variance. We use TSM-DE with the sliding window method to estimate the location of these changepoints using equation (6.9).

	Experiment Type		
	Mean	**Variance**	**Both**
MOSUM	0.994	-	-
MOSUM (Var)	-	0.995	-
MOSUM (Comb.)	-	-	0.993
PELT	0.999	0.999	0.997
TSM-DE (RBF)*	0.989	0.930	0.916
TSM-DE (SWL)*	0.950	0.963	0.945

Table 6.1. Mean covering metric values for the abrupt change benchmark, averaged over 64 trials, across various methods. Values closer to 1 indicate better performance. Our methods are indicated by asterisks. Methods not suited for detection for a given experiment are marked with a dash in the corresponding location.

	PWL Experiment Type		
	Mean	**Variance**	**Both**
PWL MOSUM	0.940	-	-
PWL MOSUM (Var)	-	0.798	-
PWL MOSUM (Comb.)	-	-	0.595
TSM-DE (RBF)*	0.903	0.612	0.779
TSM-DE (SWL)*	0.832	0.717	0.705

Table 6.2. Mean covering metric values for the piecewise linear change benchmark, averaged over 64 trials, see table 6.1 for notation.

6.5.3 Benchmark Comparison

To test the accuracy of our method for changepoint detection specifically, we benchmark it against other methods. We report the covering metric (Arbelaez et al., 2010), where values closer to 1 are more accurate.

The three experiments in table 6.1 are:

- **Mean** change experiment: The time series is Gaussian noise with mean $\mu(t)$ which is piecewise constant, with fixed $\sigma^2 = 1$. A changepoint at $t = 0.5$ shifts the mean from $\mu(t) = 0$ to $\mu(t) = 2$.
- **Variance** change experiment: The time series is Gaussian noise with variance σ_t^2 which is piecewise constant, with fixed $\mu = 0$. A changepoint at $t = 0.5$ shifts the variance from $\sigma(t)^2 = 1$ to $\sigma(t)^2 = 3$.
- Mean & Variance (**Both**) change experiment: The time series is Gaussian noise with mean $\mu(t)$ and variance σ_t^2, both piecewise constant. A changepoint at $t = 0.33$ shifts the mean

from $\mu(t) = 0$ to $\mu(t) = 2$, and a further changepoint at $t = 0.66$ shifts the variance from $\sigma(t)^2 = 1$ to $\sigma(t)^2 = 3$.

The three experiments in table 6.2 are:

- **Mean** change experiment: The time series is Gaussian noise with mean $\mu(t)$ given by

$$\mu(t) = \begin{cases} 0 & \text{if } t < 1/3, \\ 10(t - 1/3) & \text{if } 1/3 \leq t < 2/3, \\ 10/3 & \text{if } t \geq 1/3, \end{cases}$$

  i.e. the 'middle section', where $1/3 \leq t < 2/3$ is the smooth change. In this case, $\sigma^2 = 1$ is constant.
- **Variance** change experiment: The time series is Gaussian noise with variance $\sigma(t)$ given by

$$\sigma(t) = \begin{cases} 1 & \text{if } t < 1/3, \\ 6t - 1 & \text{if } 1/3 \leq t < 2/3, \\ 3 & \text{if } t \geq 1/3. \end{cases}$$

  In this case $\mu = 0$ is constant.
- Mean & Variance (**Both**) change experiment: The time series is Gaussian noise with $\mu(t)$ and $\sigma(t)$ given by

$$\mu(t) = \begin{cases} 0 & \text{if } t < 1/5, \\ 10(t - 1/5) & \text{if } 1/5 \leq t < 2/5, \\ 2 & \text{if } t \geq 2/5, \end{cases} \qquad \sigma(t) = \begin{cases} 1 & \text{if } t < 3/5, \\ 10t - 5 & \text{if } 3/5 \leq t < 4/5, \\ 3 & \text{if } t \geq 4/5. \end{cases}$$

We compare relevant prior works to TSM-DE with the RBF and the sliding window (SWL) model[1], where the hyperparameters in both cases were chosen via cross-validation, see section 6.4.7 for details.

Whilst not outperforming existing methods across all experiments, TSM-DE remains competitive. Further, TSM-DE is more flexible; it does not require *a priori* assumptions on the type of changepoint encountered, and has broad applicability to all of the considered changepoint detection problems. Overall, this suggests the method's potential in downstream applications.

6.6 Applications

6.6.1 Real Data: Temperature Anomalies (1850 - 2023)

Consider as an example the effect of climate change on global surface temperatures in the northern hemisphere (NCEA, 2024). Temperatures are increasing over time (IPCC, 2022), and

[1]The accuracy of TSM-DE will be artificially smaller due to the binning process described in section 6.4.5, as there is a less fine 'grid' to 'predict' changepoint locations on. We do not expect this disparity to be significant.

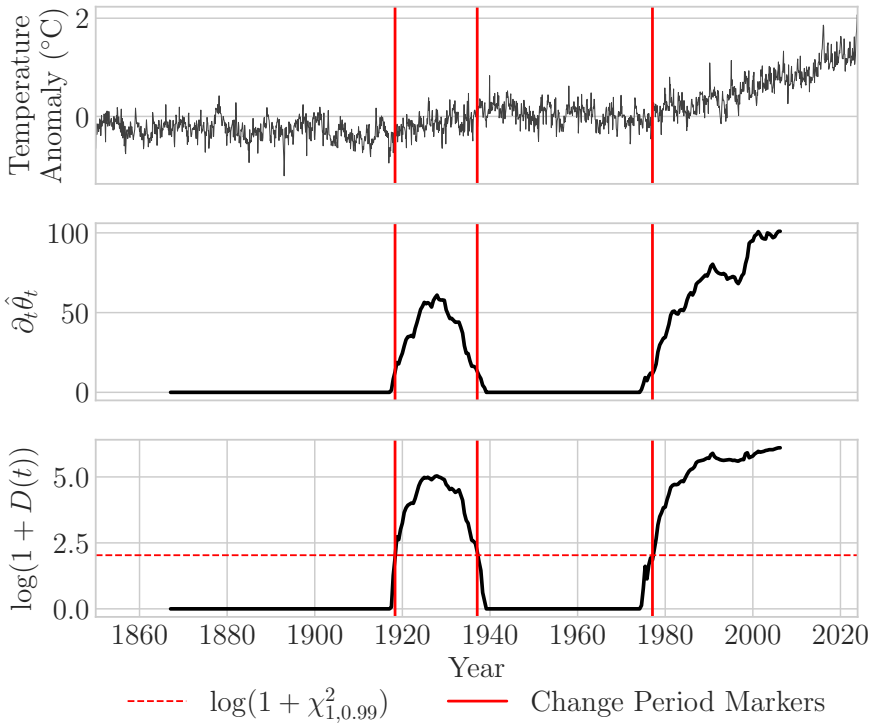
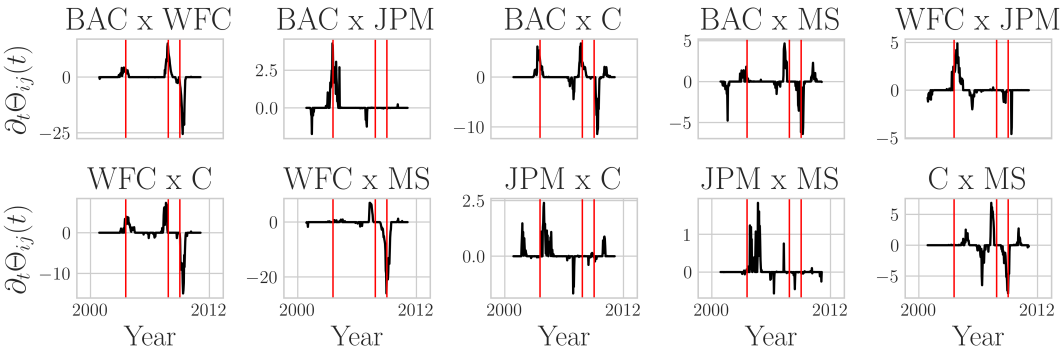


Figure 6.3. Changepoint detection (finding the beginning and end of the change periods) on temperature anomalies from 1850 to 2023, using TSM-DE with the sliding window method.

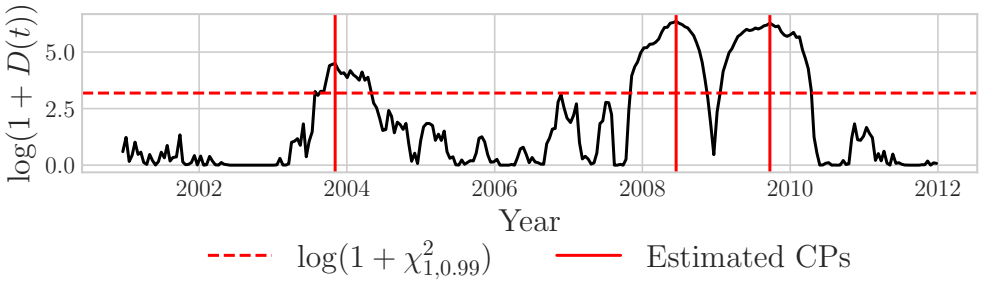
much interest lies not only in the detection of the changes of these temperatures, but in studying the rate at which the temperatures are changing. Figure 6.3 shows temperature anomalies; temperatures as a deviation from the average from 1901 to 2000, across the last 173 years, as well as the detector and detected change period locations from TSM-DE.

In this example, we assume that the change can be either gradual or abrupt. Our method does not make any assumptions on the type of change that is occurring. We use TSM-DE with the sliding window method, a small penalty parameter ($\lambda = 0.2$, to encourage periods of stationarity), and estimate the temperatures' change periods using equation (6.8). The first change period is between 1918 and 1937, and the second from 1977 to the present day. In both periods, we can interpret the value of $\partial_t \widehat{\boldsymbol{\theta}}_t$, which is positive, indicating that temperatures are increasing. Further, in the second period of change, $\partial_t \hat{\boldsymbol{\theta}}_t$ is also increasing steadily, indicating that the change in temperatures has been rising over the last 40 years, and continues to increase.

In a study on the changes of surface temperatures, Jones et al. (1999) find that the largest periods of increase are from 1925 to 1944, and from 1978 to 1977. These align with the change periods detected by TSM-DE, verifying our result. Further, in a summary on global temperature change, Lindsey and Dahlman (2024) show that temperatures start to increase rapdily from 1980 to the present day, which also matches our findings.



(a) Different estimates $\partial_t \Theta_{ij}(t)$ for each pair of stocks.



(b) Changepoint detection across all stocks.

Figure 6.4. Estimates of $\partial_t \boldsymbol{\theta}_t$ and changepoint detection for correlation changes across the global financial crisis time period, using TSM-DE with the sliding window method.

We compare our method to piecewise linear MOSUM (Kim et al., 2024), which find change periods between 1920 and 1943, and from 1977 onwards to the present day. These are very close to the change periods found using TSM-DE, and serve to also corroborate the findings of our method.

6.6.2 Real Data: Global Financial Crisis

We seek to detect changepoints in stock prices across various U.S. based companies, retrieved from the Wharton Research Data Service (WRDS, 2024), over the duration spanning the global financial crisis between 2007 and 2009. During this time, correlation measures between stock prices increased significantly as the market asynchronously exhibited a downward trajectory (Moldovan, 2011).

Inspired by Cho et al. (2023), we look at daily volatility measures, which is proportional to the squared difference between the highest and lowest log-price of a stock on a given day, across the time period spanning from 2001 to 2012. We take the $d = 5$ stocks with the highest trading volume over the time period, and consider their volatilities as a changing multivariate

Gaussian Graphical Model $\mathcal{N}(\mathbf{0}_d, \mathbf{\Theta}_d^{-1}(t))$, and detect changes in the off-diagonal of the inverse covariance matrix, $\mathbf{\Theta}_d(t)$. Any change in $\Theta_{ij}(t), \forall j < i$ indicates a change of interactions between the stock i and stock j. Figure 6.4 shows the results of this experiment using TSM-DE with the sliding window model. Figure 6.4(a) shows estimates of $\partial_t \Theta_{ij}(t)$ for each pair, and figure 6.4(b) shows $D(t)$ and the detected changepoints, calculated by equation (6.9). We select hyperparameters which minimise the testing loss on a validation set. Note that by modelling $\partial_t \mathbf{\Theta}_d(t)$ in our model, q_t is not restricted to a classic Gaussian graphical model. Since we only model the parameter change, q_t can contain other non-time dependent components.

Our method found three changepoints in 2004, the end of 2008 and the end of 2009. The changepoint at 2004 likely corresponds to the introduction of the *Net Capital Rule*, which has been referenced as one of the likely causes of the global financial crisis (Satow, 2008). The second changepoint is likely the bankruptcy of the Lehman brothers, which is claimed to have started the global financial crisis (Quax et al., 2013), and the third changepoint likely marks the end of the stock comovement, when the markets began to stabilise. Across most stock pairs, the sign of $\partial_t \Theta_{ij}(t)$ is positive for the 2008 change and negative for the 2009 change, encouraging the idea that our method has predicted that the correlations increased at the beginning of the crisis and decreased at the end.

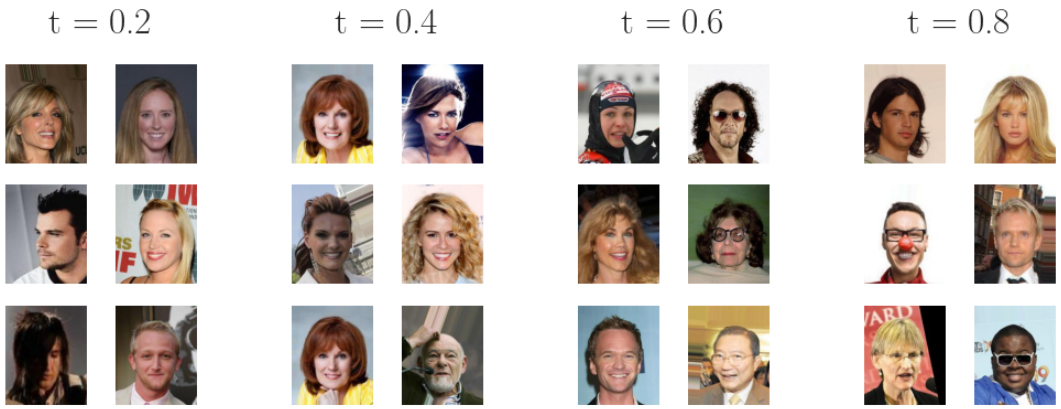
This example also highlights the interpretability of the method - the estimates of $\partial_t \Theta_{ij}(t)$ pictured in figure 6.4(a) correspond exactly to changes in the correlations between the volatility measures of the stock pairs.

Finally, we compare our method to MOSUM adapted for covariance change and PELT. MOSUM detected most changepoints around the end of 2004, the end of 2007 and midway through 2008. PELT detected changepoints at the end of 2003, as well as midway through 2007, 2008 and 2009. The changepoints detected by all methods remained largely consistent with the global financial crisis, which also serves to corroborate the findings from TSM-DE.

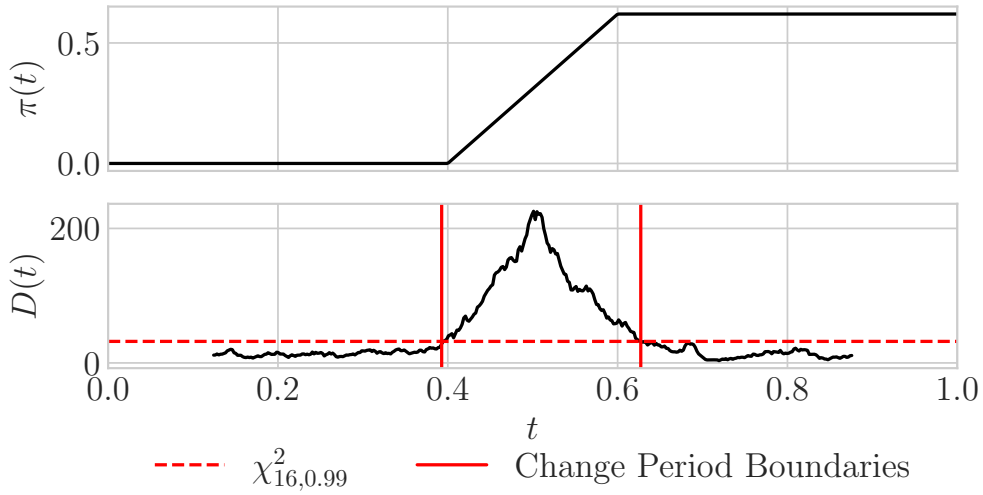
6.6.3 Incremental Concept Drift

Consider for one final example simulating the idea of concept drift (Widmer and Kubat, 1996), which describes the evolution of ideas over time, causing a shift in the underlying data distribution. It poses a significant challenge for models trained on outdated data, and may result in inaccurate predictions on modern data. Many works have studied detecting concept drift in the framework of changepoint detection, see for example Gama et al. (2014) and Lu et al. (2018). We simulate the idea of concept drift by training a basic convolutional neural network on the CelebA dataset (Liu et al., 2015) to classify whether an individual is smiling or not. Importantly, the neural network is *not* trained on any faces that are wearing glasses. Then, we gradually introduce samples including faces that *are* wearing glasses, and treat the neural network outputs as a time series.

More specifically, across $T = 500$ timepoints, we sample $n = 20$ images per timepoint. Each sample has probability $\pi(t)$ of being a face that is wearing glasses for a given time t, where $\pi(t)$ is a piecewise linear function that is increasing from $\pi(t) = 0$ at $t = 0.4$ to $\pi(t) = 0.62$ at



(a) Random samples of the concept drift simulated dataset for different time points. Note that glasses and sunglasses become more prevalent at larger time values.



(b) Probability of sampling a glasses wearer (top), and changepoint detection using TSM-DE (bottom).

Figure 6.5. Concept drift experiment. Detecting a shift in the distribution of a neural network representation as images of people wearing glasses gradually introduced to the dataset, using TSM-DE with the sliding window model.

$t = 0.6$, pictured in figure 6.5(b). Some image samples for different t can be seen in figure 6.5(a). The idea is to detect an underlying change in how the representation given by the network changes when out-of-distribution data is gradually introduced.

We use the sliding window method in TSM-DE, and let $\boldsymbol{f}$ be the output of the penultimate layer of the classification network, for which $d = 16$. Note that our choice of $\boldsymbol{f}$ induces an intractable normalisation term in the density model q_t. We choose hyperparameters by minimising our loss on a validation set. We estimate the change periods using equation (6.8): One change period is

detected spanning from $t = 0.399$ to $t = 0.623$, where the true change period spanned from $t = 0.4$ to $t = 0.6$.

We can apply other changepoint detection methods directly to the outputs, $\boldsymbol{f}(\boldsymbol{x})$, for comparison. Piecewise linear MOSUM detected no changes in the mean or variance. MOSUM detected a change in the mean at $t = 0.569$, and changes in the variance at $t = 0.451$ and $t = 0.527$. PELT detected one changepoint at $t = 0.502$. These methods are not directly suitable to this task, as they all detected changepoints at singular instances, not across the span of the changing trend. Detecting changes in the distribution of these neural network outputs is likely a more difficult task due to the added complexity in the dataset. These results suggest that TSM-DE is a better performing changepoint detection method when the problem is more complex.

6.7 Conclusion

We have presented TSM-DE, a novel methodology which estimates the derivative of a time-evolving density function, with minimal assumptions on the density itself. This makes it a more flexible approach to changepoint detection than previous methods. Experimental results show that TSM-DE is a comparable estimator to other changepoint detection methods, and performs better for a more complex example. Due to its flexibility, we believe this work has large scope for extension and future research in application to other, more complex changepoint detection problems, for example, a parameter change in time series regression, or in an autoregressive time series.

We recognise that the requirement of large n is difficult to obtain for most time series data, however experimental results are reassuringly accurate for small n. We presented two models: the RBF and sliding window model. Depending on the choice of model for $\phi(t)$, there are likely a number of hyperparameters to tune, for which we have implemented a basic selection method based on cross-validation. It is also possible that different models could improve the accuracy of TSM-DE further, which is an interesting direction for future work.

A complete implementation is available in Python at `https://github.com/tsmdepaper/tsmde/`.

6.8 Proofs

6.8.1 Proof of theorem 6.1

Recall that

$$q_t = q(\boldsymbol{x}; \boldsymbol{\theta}_t^\star, \boldsymbol{\varphi}^\star) = \frac{\exp\left(\langle \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x})\rangle\right)}{Z(\boldsymbol{\theta}_t^\star, \boldsymbol{\varphi}^\star)},$$

where $\boldsymbol{\varphi}^\star$ does not depend on t and therefore $\partial_t \boldsymbol{\varphi}^\star = 0$, and for simplicity we have written $\boldsymbol{\theta}_t = \boldsymbol{\theta}(t)$. Next, let us write the time score function as

$$\begin{aligned}
\dot{\psi}_{q_t}(\boldsymbol{x}) = \partial_t \log q_t &= \partial_t \left[\langle \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x})\rangle\right] - \partial_t \log Z(\boldsymbol{\theta}_t^\star, \boldsymbol{\varphi}^\star) \\
&= \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle - \frac{\partial_t Z(\boldsymbol{\theta}_t^\star, \boldsymbol{\varphi}^\star)}{Z(\boldsymbol{\theta}_t^\star, \boldsymbol{\varphi}^\star)} \\
&= \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle - \frac{\partial_t [\int_{\mathcal{X}} \exp\left(\langle \boldsymbol{\theta}^\star(t), \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x})\rangle\right) d\boldsymbol{x}]}{Z(\boldsymbol{\theta}_t^\star, \boldsymbol{\varphi}^\star)}
\end{aligned}$$

At this point let us note that the numerator on the right hand term can be written as

$$\begin{aligned}
&\partial_t \left[\int_{\mathcal{X}} \exp\left(\langle \boldsymbol{\theta}^\star(t), \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x})\rangle\right) d\boldsymbol{x}\right] \\
&= \int_{\mathcal{X}} \partial_t \left[\langle \boldsymbol{\theta}^\star(t), \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x})\rangle\right] \exp\left(\langle \boldsymbol{\theta}^\star(t), \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x})\rangle\right) d\boldsymbol{x} \\
&= \int_{\mathcal{X}} \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle \exp\left(\langle \boldsymbol{\theta}^\star(t), \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x})\rangle\right) d\boldsymbol{x}
\end{aligned}$$

and therefore

$$\begin{aligned}
\dot{\psi}_{q_t}(\boldsymbol{x}) &= \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle - \frac{\int_{\mathcal{X}} \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle \exp\left(\langle \boldsymbol{\theta}^\star(t), \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x})\rangle\right) d\boldsymbol{x}}{Z(\boldsymbol{\theta}_t^\star, \boldsymbol{\varphi}^\star)} \\
&= \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle - \int_{\mathcal{X}} \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle \frac{\exp\left(\langle \boldsymbol{\theta}^\star(t), \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{\varphi}^\star, \boldsymbol{h}(\boldsymbol{x})\rangle\right)}{Z(\boldsymbol{\theta}_t^\star, \boldsymbol{\varphi}^\star)} d\boldsymbol{x} \\
&= \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle - \int_{\mathcal{X}} \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle \, q(\boldsymbol{x}; \boldsymbol{\theta}_t^\star, \boldsymbol{\varphi}^\star) d\boldsymbol{x} \\
&= \langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle - \mathbb{E}_{q_t}\left[\langle \partial_t \boldsymbol{\theta}_t^\star, \boldsymbol{f}(\boldsymbol{x})\rangle\right],
\end{aligned}$$

as desired. All equalities either come from applications of the chain rule or simple rearrangements.

6.8.2 Proof of Gaussian Example

Let us group this example into three parts: fixed mean and time-dependent variance (μ and σ_t^2), time-dependent mean and fixed variance (μ_t and σ^2), and time-dependent mean and variance (μ_t and σ_t^2). Each one is detailed in distinct sections below.

Across all three cases we aim to verify the following equation holds

$$\partial_t \log q_t = \langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(x) - \mathbb{E}_{q_t}[\boldsymbol{f}(x)]\rangle, \tag{6.14}$$

which is our formulation given by theorem 6.1. For the exponential family, the natural parameterisation of the Gaussian distribution is given by

$$\boldsymbol{\theta} = \begin{bmatrix} \frac{\mu}{\sigma^2} \\ -\frac{1}{2\sigma^2} \end{bmatrix} \tag{6.15}$$

for a given μ and σ^2.

Fixed mean and time-dependent variance Let the fixed mean be written as μ and the time-dependent variance be written as σ_t^2. Firstly, according to this Gaussian distribution, the LHS of equation (6.14) is given by

$$\begin{aligned}\partial_t \log q_t &= \partial_t \left(\frac{-(x-\mu)^2}{2\sigma_t^2} \right) - \partial_t \log(\sqrt{2\pi}\,\sigma_t) \\ &= -\partial_t \left(\frac{x^2}{2\sigma_t^2} \right) + \partial_t \left(\frac{x\mu}{\sigma_t} \right) - \partial_t \left(\frac{\mu^2}{2\sigma_t^2} \right) - \partial_t \left(\log \sigma_t \right) \\ &= -x^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) + x\mu \partial_t \left(\frac{1}{\sigma_t} \right) - \mu^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - \frac{\partial_t \sigma_t}{\sigma_t} \end{aligned} \tag{6.16}$$

We aim to show that when $\boldsymbol{f}(x) = [x, x^2]$, the RHS of equation (6.14) is equal to this. We first write

$$\boldsymbol{\theta}_t = \begin{bmatrix} \frac{\mu}{\sigma_t^2} \\ -\frac{1}{2\sigma_t^2} \end{bmatrix}, \qquad \partial_t \boldsymbol{\theta}_t = \begin{bmatrix} \mu \partial_t \left(\frac{1}{\sigma_t^2} \right) \\ -\partial_t \left(\frac{1}{2\sigma_t^2} \right) \end{bmatrix}.$$

$$\begin{aligned}&\langle \partial_t \theta_t, \boldsymbol{f}(x) - \mathbb{E}_{q_t}[\boldsymbol{f}(x)] \rangle \\ &= \left\langle \begin{bmatrix} \partial_t \left(\frac{\mu}{\sigma_t^2} \right) \\ \partial_t \left(-\frac{1}{2\sigma_t^2} \right) \end{bmatrix}, \begin{bmatrix} x \\ x^2 \end{bmatrix} - \begin{bmatrix} \mathbb{E}_{q_t}[x] \\ \mathbb{E}_{q_t}[x^2] \end{bmatrix} \right\rangle \\ &= \left\langle \begin{bmatrix} \partial_t \left(\frac{\mu}{\sigma_t^2} \right) \\ \partial_t \left(-\frac{1}{2\sigma_t^2} \right) \end{bmatrix}, \begin{bmatrix} x \\ x^2 \end{bmatrix} - \begin{bmatrix} \mu \\ \sigma_t^2 + \mu^2 \end{bmatrix} \right\rangle \\ &= \left\langle \begin{bmatrix} \partial_t \left(\frac{\mu}{\sigma_t^2} \right) \\ \partial_t \left(-\frac{1}{2\sigma_t^2} \right) \end{bmatrix}, \begin{bmatrix} x - \mu \\ x^2 - \sigma_t^2 - \mu^2 \end{bmatrix} \right\rangle \\ &= \partial_t \left(\frac{\mu}{\sigma_t^2} \right) (x - \mu) - \partial_t \left(\frac{1}{2\sigma_t^2} \right) (x^2 - \sigma_t^2 - \mu^2) \\ &= x\mu \partial_t \left(\frac{1}{\sigma_t^2} \right) - \mu^2 \partial_t \left(\frac{1}{\sigma_t^2} \right) - x^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) + \sigma_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) + \mu^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) \\ &= x\mu \partial_t \left(\frac{1}{\sigma_t^2} \right) - \mu^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - x^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) + \sigma_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right). \end{aligned}$$

By the chain rule,

$$\sigma_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) = \frac{\sigma_t^2}{2} \partial_t \left(\sigma_t^{-2} \right) = -\frac{\sigma_t^2}{2} \frac{2}{\sigma_t^3} \partial_t \sigma_t = -\frac{\partial_t \sigma_t}{\sigma_t},$$

which, substituted into the equation above, leaves

$$x\mu \partial_t \left(\frac{1}{\sigma_t^2} \right) - \mu^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - x^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - \frac{\partial_t \sigma_t}{\sigma_t}$$

for which all terms match the terms in equation (6.16) as desired.

Since we are doing very similar operations across all examples, the following two derivations will be lighter on details but should be straightforward to follow.

Time-dependent mean and fixed variance Firstly, write μ_t and σ^2 as the mean and variance of this Gaussian distribution, respectively. The time score function, i.e. the LHS of equation (6.14) is given by

$$\begin{aligned}\partial_t \log q_t &= \partial_t \left(\frac{-(x-\mu_t)^2}{2\sigma^2} \right) - \partial_t \log(\sqrt{2\pi}\sigma) \\ &= \frac{2x\partial_t\mu_t - \partial_t(\mu_t^2)}{2\sigma^2}.\end{aligned} \tag{6.17}$$

The natural parameters are given by

$$\boldsymbol{\theta}_t = \begin{bmatrix} \frac{\mu_t}{\sigma^2} \\ -\frac{1}{2\sigma^2} \end{bmatrix}, \qquad \partial_t \boldsymbol{\theta}_t = \begin{bmatrix} \partial_t \left(\frac{\mu_t}{\sigma^2} \right) \\ \partial_t \left(-\frac{1}{2\sigma^2} \right) \end{bmatrix} = \begin{bmatrix} \frac{\partial_t \mu_t}{\sigma^2} \\ 0 \end{bmatrix},$$

and so we consider the first dimension only. The RHS of equation (6.14), when $\boldsymbol{f}(x) = x$ is given by

$$\begin{aligned}\langle \partial_t \theta_t, \boldsymbol{f}(x) - \mathbb{E}_{q_t}[\boldsymbol{f}(x)] \rangle &= \left\langle \frac{\partial_t \mu_t}{\sigma^2}, x - \mu_t \right\rangle \\ &= \frac{x\partial_t\mu_t - \mu_t\partial_t\mu_t}{\sigma^2} \\ &= \frac{x\partial_t\mu_t - \frac{\partial_t(\mu_t^2)}{2}}{\sigma^2} \\ &= \frac{2x\partial_t\mu_t - \partial_t(\mu_t^2)}{2\sigma^2}\end{aligned}$$

where the penultimate line is due to the chain rule. This matches equation (6.17) as desired.

Time-dependent mean and variance Firstly, write μ_t and σ_t^2 as the mean and variance of this Gaussian distribution, respectively. The time score function, i.e. the LHS of equation (6.14) is given by

$$\begin{aligned}\partial_t \log q_t &= \partial_t \left(\frac{-(x-\mu_t)^2}{2\sigma_t^2} \right) - \partial_t \log(\sqrt{2\pi}\sigma_t) \\ &= -x^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) + x\frac{\partial_t\mu_t}{\sigma_t^2} + x\mu_t\partial_t \left(\frac{1}{\sigma_t^2} \right) - \frac{\partial_t(\mu_t^2)}{2\sigma_t^2} - \mu_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - \frac{\partial_t\sigma_t}{\sigma_t}.\end{aligned} \tag{6.18}$$

The natural parameters are given by

$$\boldsymbol{\theta}_t = \begin{bmatrix} \frac{\mu_t}{\sigma_t^2} \\ -\frac{1}{2\sigma_t^2} \end{bmatrix}, \qquad \partial_t \boldsymbol{\theta}_t = \begin{bmatrix} \partial_t \left(\frac{\mu_t}{\sigma_t^2} \right) \\ \partial_t \left(-\frac{1}{2\sigma_t^2} \right) \end{bmatrix} = \begin{bmatrix} \frac{\partial_t \mu_t}{\sigma_t^2} + \mu_t \partial_t \left(\frac{1}{\sigma_t^2} \right) \\ -\partial_t \left(\frac{1}{2\sigma_t^2} \right) \end{bmatrix},$$

The RHS of equation (6.14) when $\boldsymbol{f}(x) = [x, x^2]$ is given by

$$
\begin{aligned}
&\langle \partial_t \theta_t, \boldsymbol{f}(x) - \mathbb{E}_{q_t}[\boldsymbol{f}(x)] \rangle \\
&= \left\langle \begin{bmatrix} \frac{\partial_t \mu_t}{\sigma_t^2} + \mu_t \partial_t \left(\frac{1}{\sigma_t^2} \right) \\ -\partial_t \left(\frac{1}{2\sigma_t^2} \right) \end{bmatrix}, \begin{bmatrix} x - \mu_t \\ x^2 - \sigma_t^2 - \mu_t^2 \end{bmatrix} \right\rangle \\
&= \left(\frac{\partial_t \mu_t}{\sigma_t^2} + \mu_t \partial_t \left(\frac{1}{\sigma_t^2} \right) \right) (x - \mu_t) - \partial_t \left(\frac{1}{2\sigma_t^2} \right) (x^2 - \sigma_t^2 - \mu_t^2) \\
&= x \frac{\partial_t \mu_t}{\sigma_t^2} + x \mu_t \partial_t \left(\frac{1}{\sigma_t^2} \right) - \mu_t \frac{\partial_t \mu_t}{\sigma_t^2} - \mu_t^2 \partial_t \left(\frac{1}{\sigma_t^2} \right) - x^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) + \sigma_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) + \mu_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) \\
&= x \frac{\partial_t \mu_t}{\sigma_t^2} + x \mu_t \partial_t \left(\frac{1}{\sigma_t^2} \right) - \mu_t \frac{\partial_t \mu_t}{\sigma_t^2} + \mu_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - x^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) + \sigma_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) \\
&= x \frac{\partial_t \mu_t}{\sigma_t^2} + x \mu_t \partial_t \left(\frac{1}{\sigma_t^2} \right) - \mu_t \frac{\partial_t \mu_t}{\sigma_t^2} + \mu_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - x^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - \frac{\partial_t \sigma_t}{\sigma_t} \\
&= x \frac{\partial_t \mu_t}{\sigma_t^2} + x \mu_t \partial_t \left(\frac{1}{\sigma_t^2} \right) - \frac{\partial_t (\mu_t^2)}{2\sigma_t^2} - \mu_t^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - x^2 \partial_t \left(\frac{1}{2\sigma_t^2} \right) - \frac{\partial_t \sigma_t}{\sigma_t},
\end{aligned}
$$

where the last three equalities, in their respective order, are due to collecting like terms, the chain rule on the last term, and the chain rule again on the third term. This matches equation (6.18), completing the proof.

6.8.3 Proof of theorem 6.2

Let us begin from the initial formulation of our time based score matching objective with weight function $g_t = g(t)$ for which $g(0) = g(1) = 0$, i.e. $g(t) = 0$ at the edges of our time domain $t \in [0, 1]$. The initial objective is given by

$$
\begin{aligned}
J_{\mathrm{C}}(\boldsymbol{\theta}_t) &= \int_{t=0}^{t=1} \mathbb{E}_{q_t} \left[g_t \| \dot{\psi}_{q_t}(\boldsymbol{x}) - s_t(\boldsymbol{x}) \|^2 \right] dt \\
&= \int_{t=0}^{t=1} \mathbb{E}_{q_t} \left[g_t s_t(\boldsymbol{x})^2 \right] dt - 2 \int_{t=0}^{t=1} \mathbb{E}_{q_t} \left[g_t s_t(\boldsymbol{x}) \dot{\psi}_{q_t}(\boldsymbol{x}) \right] dt + C,
\end{aligned}
$$

where $C = \int_{t=0}^{t=1} \mathbb{E}_{q_t} \left[g_t \dot{\psi}_{q_t}(\boldsymbol{x})^2 \right] dt$ is a constant with respect to the model $s_t(\boldsymbol{x})$. Continuing,

$$
\begin{aligned}
J_{\mathrm{C}}(\boldsymbol{\theta}_t) &= \int_{t=0}^{t=1} \mathbb{E}_{q_t} \left[g_t s_t(\boldsymbol{x})^2 \right] dt - 2 \int_{t=0}^{t=1} \int_{\mathcal{X}} q_t \dot{\psi}_{q_t}(\boldsymbol{x}) g_t s_t(\boldsymbol{x}) d\boldsymbol{x} dt + C \\
&= \int_{t=0}^{t=1} \mathbb{E}_{q_t} \left[g_t s_t(\boldsymbol{x})^2 \right] dt - 2 \int_{t=0}^{t=1} \int_{\mathcal{X}} \partial_t q_t g_t s_t(\boldsymbol{x}) d\boldsymbol{x} dt + C,
\end{aligned}
$$

where in the final line we have used $q_t\dot{\psi}_{q_t}(\boldsymbol{x}) = q_t\partial_t \log q_t = \partial_t q_t$ to simplify. We continue by expanding the middle term via integration by parts

$$\begin{aligned}
J_{\mathrm{C}}(\boldsymbol{\theta}_t) &= \int_{t=0}^{t=1} \mathbb{E}_{q_t}\left[g_t s_t(\boldsymbol{x})^2\right] dt - 2\left[\int_{\mathcal{X}} q_t g_t s_t(\boldsymbol{x}) d\boldsymbol{x}\right]_{t=0}^{t=1} + 2\int_{t=0}^{t=1}\int_{\mathcal{X}} q_t \partial_t (g_t s_t(\boldsymbol{x})) d\boldsymbol{x} dt + C \\
&= \int_{t=0}^{t=1} \mathbb{E}_{q_t}\left[g_t s_t(\boldsymbol{x})^2\right] dt + 2\int_{t=0}^{t=1}\int_{\mathcal{X}} q_t \partial_t (g_t s_t(\boldsymbol{x})) d\boldsymbol{x} dt + C \\
&= \int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})^2\right] dt + 2\int_{t=0}^{t=1} \partial_t g_t \mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})\right] dt + 2\int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[\partial_t s_t(\boldsymbol{x})\right] dt + C,
\end{aligned} \tag{6.19}$$

where the second equality is due to the boundary condition imposed on g, i.e. $\left[\int_{\mathcal{X}} q_t g_t s_t(\boldsymbol{x}) d\boldsymbol{x}\right]_{t=0}^{t=1} = 0$ as $g(0) = g(1) = 0$.

Let us now substitute the model given in equation (6.4), stated again here for completeness, given by

$$s_t(\boldsymbol{x}) = s(\boldsymbol{x}; t) = \langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle - \mathbb{E}_{q_t}\left[\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle\right].$$

We calculate $\mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})\right]$ and $\mathbb{E}_{q_t}\left[\partial_t s_t(\boldsymbol{x})\right]$ to substitute into the equation above. These are given by

$$\begin{aligned}
\mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})\right] &= \mathbb{E}_{q_t}\left[\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle - \mathbb{E}_{q_t}\left[\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle\right]\right] \\
&= \mathbb{E}_{q_t}\left[\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle\right] - \mathbb{E}_{q_t}\left[\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle\right] \\
&= 0 \\
\mathbb{E}_{q_t}\left[\partial_t s_t(\boldsymbol{x})\right] &= \mathbb{E}_{q_t}\left[\partial_t\left(\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle - \mathbb{E}_{q_t}\left[\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle\right]\right)\right] \\
&= \mathbb{E}_{q_t}\left[\left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle - \partial_t\left(\mathbb{E}_{q_t}\left[\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle\right]\right)\right] \\
&= \mathbb{E}_{q_t}\left[\left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle - \partial_t\left(\int_{\mathcal{X}'} q_t \langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle d\boldsymbol{x}'\right)\right] \\
&= \mathbb{E}_{q_t}\left[\left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle - \int_{\mathcal{X}'} \partial_t q_t \langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle d\boldsymbol{x}' - \int_{\mathcal{X}'} q_t \left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x}'\right] \\
&= \mathbb{E}_{q_t}\left[\left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle - \int_{\mathcal{X}'} \partial_t q_t \langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle d\boldsymbol{x}' - \mathbb{E}_{q_t}\left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x}'\right] \\
&= \mathbb{E}_{q_t}\left[\left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle - \int_{\mathcal{X}'} \partial_t q_t \langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle d\boldsymbol{x}'\right] - \mathbb{E}_{q_t}\left[\left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x}'\right] \\
&= \mathbb{E}_{q_t}\left[-\int_{\mathcal{X}'} \partial_t q_t \langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\rangle d\boldsymbol{x}'\right],
\end{aligned}$$

where the integral $\int_{\mathcal{X}'}(\dots)d\boldsymbol{x}'$ is the same integral as $\int_{\mathcal{X}}(\dots)d\boldsymbol{x}$ but written as such to make

them distinct from one another. Substituting these into equation (6.19) gives

$$\begin{aligned}
J_{\mathrm{C}}(\boldsymbol{\theta}_t) &= \int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})^2\right] dt - 2\int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[\int_{\mathcal{X}'} \partial_t q_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x}'\right] dt + C \\
&= \int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})^2\right] dt - 2\int_{t=0}^{t=1} g_t \int_{\mathcal{X}} q_t \left(\int_{\mathcal{X}'} \partial_t q_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x}'\right) d\boldsymbol{x} dt + C \\
&\overset{\text{(a)}}{=} \int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})^2\right] dt - 2\int_{t=0}^{t=1} g_t \left(\int_{\mathcal{X}} q_t d\boldsymbol{x}\right) \int_{\mathcal{X}'} \partial_t q_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x}' dt + C \\
&\overset{\text{(b)}}{=} \int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})^2\right] dt - 2\int_{t=0}^{t=1} g_t \int_{\mathcal{X}} \partial_t q_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x} dt + C.
\end{aligned}$$

In the equality denoted by (a), we have used the fact that inside the integral $\int_{\mathcal{X}} q_t(\dots)d\boldsymbol{x}$ the only variable dependent on $\boldsymbol{x}$ was q_t, as $\boldsymbol{x}$ and $\boldsymbol{x}'$ are independent. We also have that $\int_{\mathcal{X}} q_t d\boldsymbol{x} = 1$ by definition of q_t being a probability density function. The equality denoted by (b) contains a re-labelling of $\mathcal{X} = \mathcal{X}'$ and $\boldsymbol{x} = \boldsymbol{x}'$, as they are the same variable, only labelled differently originally to make them distinct. We use integration by parts one final time to obtain

$$\begin{aligned}
J_{\mathrm{C}}(\boldsymbol{\theta}_t) &= \int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})^2\right] dt - 2\left[\int_{\mathcal{X}} q_t g_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x}\right]_{t=0}^{t=1} \\
&\qquad + 2\int_{t=0}^{t=1} \int_{\mathcal{X}} q_t \partial_t \left(g_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle\right) d\boldsymbol{x} dt + C \\
&= \int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})^2\right] dt + 2\int_{t=0}^{t=1} \int_{\mathcal{X}} q_t \partial_t \left(g_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle\right) d\boldsymbol{x} dt + C,
\end{aligned}$$

using again that $g(0) = g(1) = 0$, and finally

$$\begin{aligned}
J_{\mathrm{C}}(\boldsymbol{\theta}_t) &= \int_{t=0}^{t=1} g_t \mathbb{E}_{q_t}\left[s_t(\boldsymbol{x})^2\right] dt + 2\int_{t=0}^{t=1} \int_{\mathcal{X}} q_t \partial_t g_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x} dt \\
&\qquad + 2\int_{t=0}^{t=1} \int_{\mathcal{X}} q_t g_t \left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle d\boldsymbol{x} dt + C \\
&= \int_{t=0}^{t=1} \mathbb{E}_{q_t}\left[g_t s_t(\boldsymbol{x})^2 + 2\partial_t g_t \left\langle \partial_t \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle + 2 g_t \left\langle \partial_t^2 \boldsymbol{\theta}_t, \boldsymbol{f}(\boldsymbol{x})\right\rangle\right] dt
\end{aligned}$$

which is the same as equation (6.6) in the main text.

6.8.4 Proof of theorem 6.4

Following section 6.3.4 and assumption 6.3, we have

$$\partial_t \boldsymbol{\theta}_t = \boldsymbol{\alpha}^\top (\partial_t \boldsymbol{\phi}(t)),$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{b \times k}$ and $\partial_t \boldsymbol{\phi}(t) \in \mathbb{R}^b$. Note that b refers to the dimension of the features given by $\boldsymbol{\phi}(t)$ which changes depending on the model for $\boldsymbol{\phi}(t)$. For brevity, let us shorten $\boldsymbol{\phi}(t)$ as $\boldsymbol{\phi}_t$ and $\boldsymbol{f}(\boldsymbol{x})$ as $\boldsymbol{f}$.

6.8.4.1 A Note on Matrix Sizes and Reshaping

Since $\boldsymbol{\alpha}$ is a $(b \times k)$-matrix, to derive its asymptotic distribution, we reshape $\boldsymbol{\alpha}$ to a separate vector $\widetilde{\boldsymbol{\alpha}} \in \mathbb{R}^{bk}$ without loss of generality. Further, we also define separate vectors $\partial_t \widetilde{\boldsymbol{\phi}}(t) \in \mathbb{R}^{bk \times k}$ and $\partial_t^2 \widetilde{\boldsymbol{\phi}}(t) \in \mathbb{R}^{bk \times k}$, defined such that $\widetilde{\boldsymbol{\alpha}}^\top (\partial_t \widetilde{\boldsymbol{\phi}}(t)) = \boldsymbol{\alpha}^\top (\partial_t \boldsymbol{\phi}(t))$ and $\widetilde{\boldsymbol{\alpha}}^\top (\partial_t^2 \widetilde{\boldsymbol{\phi}}(t)) = \boldsymbol{\alpha}^\top (\partial_t^2 \boldsymbol{\phi}(t))$. All these new variables are defined as follows:

$$\partial_t \widetilde{\boldsymbol{\phi}}(t) = \begin{bmatrix} \partial_t \boldsymbol{\phi}(t) & \mathbf{0}_b & \dots & \mathbf{0}_b \\ \mathbf{0}_b & \partial_t \boldsymbol{\phi}(t) & \dots & \mathbf{0}_b \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_b & \mathbf{0}_b & \dots & \partial_t \boldsymbol{\phi}(t) \end{bmatrix}, \qquad \partial_t^2 \widetilde{\boldsymbol{\phi}}(t) = \begin{bmatrix} \partial_t^2 \boldsymbol{\phi}(t) & \mathbf{0}_b & \dots & \mathbf{0}_b \\ \mathbf{0}_b & \partial_t^2 \boldsymbol{\phi}(t) & \dots & \mathbf{0}_b \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_b & \mathbf{0}_b & \dots & \partial_t^2 \boldsymbol{\phi}(t) \end{bmatrix},$$

$$\widetilde{\boldsymbol{\alpha}} = \begin{bmatrix} \alpha_{1,1}, & \dots, & \alpha_{b,1}, & \alpha_{1,2}, & \dots, & \alpha_{b,k} \end{bmatrix}^\top$$

The reason and motivation for this reshaping should hopefully become clear by the end of this section. In summary, it is so we can derive the asymptotic distribution for $\widetilde{\boldsymbol{\alpha}}$ as a vector, and the results will carry forward to $\boldsymbol{\alpha}$.

For convenience and brevity, for the remainder of this section we will *not* rename instances of $\boldsymbol{\alpha}$ as $\widetilde{\boldsymbol{\alpha}}$, nor $\partial_t \widetilde{\boldsymbol{\phi}}(t)$ or $\partial_t \widetilde{\boldsymbol{\phi}}(t)$, and they will remain as their original, non-tilde variants.

6.8.4.2 Asymptotic Distribution of $\sqrt{n}(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)$

Recall our sample objective function, given by

$$\hat{J}_{\mathrm{C}}(\boldsymbol{\theta}_t) = \frac{1}{T} \sum_{j=1}^{T} \frac{1}{n} \sum_{i=1}^{n} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t_j) = \frac{1}{T} \sum_{j=1}^{T} m_n(\boldsymbol{\alpha}, t_j), \tag{6.20}$$

where for convenience of notation we have written $m_n(\boldsymbol{\alpha}, t) = \frac{1}{n} \sum_{i=1}^{n} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t)$, and

$$\begin{aligned} m_{\boldsymbol{\alpha}}(\boldsymbol{x}, t) &= g(t) s_t(\boldsymbol{x})^2 + 2 \partial_t g(t) \left\langle \partial_t \boldsymbol{\theta}(t), \boldsymbol{f} \right\rangle + 2 g(t) \left\langle \partial_t^2 \boldsymbol{\theta}(t), \boldsymbol{f} \right\rangle \\ &= g(t) \left\langle \boldsymbol{\alpha}^\top (\partial_t \boldsymbol{\phi}_t), \boldsymbol{f} - \mathbb{E}_{q_t}[\boldsymbol{f}] \right\rangle^2 + 2 \left\langle \partial_t g(t) \boldsymbol{\alpha}^\top (\partial_t \boldsymbol{\phi}_t) + g(t) \boldsymbol{\alpha}^\top (\partial_t^2 \boldsymbol{\phi}_t), \boldsymbol{f} \right\rangle. \end{aligned} \tag{6.21}$$

Denote by

$$\begin{aligned} \boldsymbol{m}_n'(\boldsymbol{\alpha}^\star, t) &= \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t) \big|_{\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star}, \\ \boldsymbol{m}_n''(\boldsymbol{\alpha}^\star, t) &= \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\alpha}}^2 m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t) \big|_{\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star}, \\ \boldsymbol{m}_n'''(\boldsymbol{\alpha}^\star, t) &= \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\alpha}}^3 m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t) \big|_{\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star}, \end{aligned}$$

the first, second and third derivative, respectively, of $m_n(\boldsymbol{\alpha}, t)$ with respect to $\boldsymbol{\alpha}$, evaluated at $\boldsymbol{\alpha}^\star$. Let $\widehat{\boldsymbol{\alpha}}$ be the minimiser of the sample objective, given by equation (6.20), therefore

$$\frac{1}{T}\sum_{j=1}^{T} \boldsymbol{m}_n'(\widehat{\boldsymbol{\alpha}}, t_j) = \mathbf{0}, \tag{6.22}$$

Assumption 6.3 states that $\boldsymbol{\alpha}^\star$ is the true parameter value under the model $\partial_t \boldsymbol{\theta}_t^\star = \boldsymbol{\alpha}^{\star\top}(\partial_t \boldsymbol{\phi}_t)$, such that

$$\int_{t=0}^{t=1} \mathbb{E}\left[\nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}, t)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right] dt = \mathbf{0}. \tag{6.23}$$

Now consider the Taylor expansion of $\boldsymbol{m}_n'(\widehat{\boldsymbol{\alpha}}, t)$ around $\boldsymbol{\alpha}^\star$, given by

$$\begin{aligned}\boldsymbol{m}_n'(\hat{\boldsymbol{\alpha}}, t) &= \boldsymbol{m}_n'(\boldsymbol{\alpha}^\star, t) + (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)\boldsymbol{m}_n''(\boldsymbol{\alpha}^\star, t) + \frac{1}{2}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)^2 \boldsymbol{m}_n'''(\boldsymbol{\alpha}^\star, t) \\ &= \boldsymbol{m}_n'(\boldsymbol{\alpha}^\star, t) + (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)\boldsymbol{m}_n''(\boldsymbol{\alpha}^\star, t),\end{aligned}$$

where we have used that $\boldsymbol{m}_n'''(\boldsymbol{\alpha}, t) = 0$ due to $m_{\boldsymbol{\alpha}}(\boldsymbol{x}, t)$ being a quadratic function with respect to $\boldsymbol{\alpha}$. From the above, we can apply the sum on both sides to write

$$\frac{1}{T}\sum_{j=1}^{T} \boldsymbol{m}_n'(\hat{\boldsymbol{\alpha}}, t) = \frac{1}{T}\sum_{j=1}^{T} \boldsymbol{m}_n'(\boldsymbol{\alpha}^\star, t_j) + (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)\frac{1}{T}\sum_{j=1}^{T} \boldsymbol{m}_n''(\boldsymbol{\alpha}^\star, t_j) = 0,$$

where the final equality comes from equation (6.22). We can rearrange and then multiply by $\sqrt{n}$ on both sides to give

$$\sqrt{n}(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star) = \underbrace{\left(\frac{1}{T}\sum_{j=1}^{T} \boldsymbol{m}_n''(\boldsymbol{\alpha}^\star, t_j)\right)^{-1}}_{(a)} \underbrace{\left(\frac{1}{T}\sum_{j=1}^{T} \sqrt{n}\,\boldsymbol{m}_n'(\boldsymbol{\alpha}^\star, t_j)\right)}_{(b)}. \tag{6.24}$$

We can study the right hand side of equation (6.24) to understand its asymptotic distribution and therefore find an asymptotic distribution for $\sqrt{n}(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)$. In order, we will consider the asymptotic properties of both (a) and (b) as $n \to \infty$ and $T \to \infty$.

(a) By the law of large numbers,

$$\boldsymbol{m}_n''(\boldsymbol{\alpha}^\star, t) = \frac{1}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\alpha}}^2 m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star} \xrightarrow{P} \mathbb{E}_{q_t}\left[\nabla_{\boldsymbol{\alpha}}^2 m_{\boldsymbol{\alpha}}(\boldsymbol{x}, t)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right] \tag{6.25}$$

and therefore

$$\frac{1}{T}\sum_{j=1}^{T} \boldsymbol{m}_n''(\boldsymbol{\alpha}^\star, t_j) \xrightarrow{P} \frac{1}{T}\sum_{j=1}^{T} \mathbb{E}_{q_t}\left[\nabla_{\boldsymbol{\alpha}}^2 \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right],$$

because the dependence on t in equation (6.25) is not related to the convergence in probability.

(b) By the central limit theorem, when $n \to \infty$,

$$\sqrt{n}\ \boldsymbol{m}'_n(\boldsymbol{\alpha}^\star, t) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star} \xrightarrow{d} \mathcal{N}\left(\mathbb{E}\left[\nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right], \mathrm{Var}\left[\nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right]\right)$$

and hence

$$\frac{1}{T}\sum_{j=1}^{T} \sqrt{n}\, \boldsymbol{m}'_n(\boldsymbol{\alpha}^\star, t_j) \xrightarrow{d} \mathcal{N}\left(\frac{1}{T}\sum_{j=1}^{T} \mathbb{E}\left[\nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right], \frac{1}{T^2}\sum_{j=1}^{T} \mathrm{Var}[\nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}]\right).$$

Since we are studying in the case of $T \to \infty$, we can also write

$$\frac{1}{T}\sum_{j=1}^{T} \mathbb{E}\left[\nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right] \xrightarrow{P} \int_{t=0}^{t=1} \mathbb{E}\left[\nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right] dt = \mathbf{0}$$

by definition of $\boldsymbol{\alpha}^\star$ as given in equation (6.23). Therefore we can say, overall,

$$\frac{1}{T}\sum_{j=1}^{T} \boldsymbol{m}'_n(\boldsymbol{\alpha}^\star, t_j) \xrightarrow{d} \mathcal{N}\left(\mathbf{0}, \frac{1}{T^2}\sum_{j=1}^{T} \mathrm{Var}[\nabla_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}}(\boldsymbol{x}_i, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}]\right). \tag{6.26}$$

To combine both (a) and (b) together to determine the convergence in distribution of $\sqrt{n}(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)$, we use Slutsky's Theorem. Since (b) converges in probability and (a) converges in distribution, the product of the two converges in distribution which is given by

$$\sqrt{n}(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star) = \left(\frac{1}{T}\sum_{j=1}^{T} \boldsymbol{m}''_n(\boldsymbol{\alpha}^\star, t_j)\right)^{-1} \left(\frac{1}{T}\sum_{j=1}^{T} \sqrt{n}\, \boldsymbol{m}'_n(\boldsymbol{\alpha}^\star, t_j)\right) \xrightarrow{d} \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}}\right),$$

where

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} &= \left(\frac{1}{T}\boldsymbol{\Sigma}_E\right)^{-1} \left(\frac{1}{T^2}\sum_{j=1}^{T} \mathrm{Var}\left[\nabla_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right]\right) \left(\frac{1}{T}\boldsymbol{\Sigma}_E\right)^{-1} \\ &= \boldsymbol{\Sigma}_E^{-1} \left(\sum_{j=1}^{T} \mathrm{Var}\left[\nabla_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right]\right) \boldsymbol{\Sigma}_E^{-1}. \end{aligned} \tag{6.27}$$

and

$$\boldsymbol{\Sigma}_E = \sum_{j=1}^{T} \mathbb{E}\left[\nabla^2_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t_j)\big|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}\right].$$

Finally, we need to define $\nabla_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}^\star}(\boldsymbol{x}, t)$, $\nabla^2_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}^\star}(\boldsymbol{x}, t)$ and $\mathrm{Var}[\nabla_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}^\star}(\boldsymbol{x}, t_j)]$ for our case as given in equation (6.21), under the null hypothesis $\partial_t \boldsymbol{\theta}_t^\star = \boldsymbol{\alpha}^{\star\top}(\partial_t \boldsymbol{\phi}_t) = \mathbf{0}$. The first and second derivatives for a general $\boldsymbol{\alpha}$ are given by

$$\begin{aligned}
\nabla_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t) &= 2g(t)(\boldsymbol{f} - \mathbb{E}_{q_t}[\boldsymbol{f}])(\boldsymbol{f} - \mathbb{E}_{q_t}[\boldsymbol{f}])^\top \boldsymbol{\alpha}^\top (\partial_t \boldsymbol{\phi}_t)(\partial_t \boldsymbol{\phi}_t)^\top \\
&\qquad + 2\left\langle \partial_t g(t)(\partial_t \boldsymbol{\phi}_t)^\top + g(t)(\partial_t^2 \boldsymbol{\phi}(t))^\top, \boldsymbol{f} \right\rangle, \\
\nabla^2_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t) &= 2g(t)(\boldsymbol{f} - \mathbb{E}_{q_t}[\boldsymbol{f}])(\boldsymbol{f} - \mathbb{E}_{q_t}[\boldsymbol{f}])^\top (\partial_t \boldsymbol{\phi}_t)(\partial_t \boldsymbol{\phi}_t)^\top.
\end{aligned}$$

First note that $\nabla^2_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t)$ is independent of $\boldsymbol{\alpha}$ and therefore

$$\nabla^2_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t)|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star} = \nabla^2_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t)$$

for any $\boldsymbol{\alpha}$. Next,

$$\begin{aligned}
\nabla_{\boldsymbol{\alpha}} \boldsymbol{m}_{\boldsymbol{\alpha}}(\boldsymbol{x}, t)|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star} &= 2g(t)(\boldsymbol{f} - \mathbb{E}_{q_t}[\boldsymbol{f}])(\boldsymbol{f} - \mathbb{E}_{q_t}[\boldsymbol{f}])^\top \boldsymbol{\alpha}^{\star\top} (\partial_t \boldsymbol{\phi}_t)(\partial_t \boldsymbol{\phi}_t)^\top \\
&\qquad + 2\left\langle \partial_t g(t)(\partial_t \boldsymbol{\phi}_t)^\top + g(t)(\partial_t^2 \boldsymbol{\phi}(t))^\top, \boldsymbol{f} \right\rangle \\
&= 2\left\langle \partial_t g(t)(\partial_t \boldsymbol{\phi}_t)^\top + g(t)(\partial_t^2 \boldsymbol{\phi}(t))^\top, \boldsymbol{f} \right\rangle,
\end{aligned}$$

where the final equality is due to the initial assumption of the null hypothesis that $\partial_t \boldsymbol{\theta}_t^\star = \boldsymbol{\alpha}^{\star\top}(\partial_t \boldsymbol{\phi}_t) = \mathbf{0}$. Both of these terms are now independent of $\boldsymbol{\alpha}^\star$. Finally, we can work out $\mathrm{Var}[\boldsymbol{m}'_{\boldsymbol{\alpha}}(\boldsymbol{x}, t)|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}^\star}]$ using these calculations:

$$\begin{aligned}
&= \mathbb{E}_{q_t}\left[(2\left\langle \partial_t g(t)(\partial_t \boldsymbol{\phi}_t)^\top + g(t)(\partial_t^2 \boldsymbol{\phi}_t)^\top, \boldsymbol{f} \right\rangle - \mathbb{E}_{q_t}[2\left\langle \partial_t g(t)(\partial_t \boldsymbol{\phi}_t)^\top + g(t)(\partial_t^2 \boldsymbol{\phi}_t)^\top, \boldsymbol{f} \right\rangle])^2 \right] \\
&= \mathbb{E}_{q_t}\left[4(\left\langle \partial_t g(t)(\partial_t \boldsymbol{\phi}_t)^\top + g(t)(\partial_t^2 \boldsymbol{\phi}_t)^\top, \boldsymbol{f} \right\rangle - \left\langle \partial_t g(t)(\partial_t \boldsymbol{\phi}_t)^\top + g(t)(\partial_t^2 \boldsymbol{\phi}_t)^\top, \mathbb{E}_{q_t}[\boldsymbol{f}] \right\rangle)^2 \right] \\
&= \mathbb{E}_{q_t}\left[4(\left\langle \partial_t g(t)(\partial_t \boldsymbol{\phi}_t)^\top + g(t)(\partial_t^2 \boldsymbol{\phi}_t)^\top, \boldsymbol{f} - \mathbb{E}_{q_t}[\boldsymbol{f}] \right\rangle)^2 \right].
\end{aligned}$$

Combining all these terms together gives a tractable formulation for $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$ and hence a tractable asymptotic variance for $\sqrt{n}(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)$, which does not depend on knowing $\boldsymbol{\alpha}^\star$.

6.8.4.3 Asymptotic Distribution of $\partial_t \widehat{\boldsymbol{\theta}}_t$

Consider for a given t, taking the dot product between $\sqrt{n}(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)$ and $\partial_t \boldsymbol{\phi}_t$, which gives

$$\sqrt{n}(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^\star)^\top (\partial_t \boldsymbol{\phi}_t) = \sqrt{n}(\widehat{\boldsymbol{\alpha}}^\top (\partial_t \boldsymbol{\phi}_t) - \boldsymbol{\alpha}^{\star\top}(\partial_t \boldsymbol{\phi}_t)) = \sqrt{n}\ \widehat{\boldsymbol{\alpha}}^\top (\partial_t \boldsymbol{\phi}_t)$$

where the final equality is due to the initial assumption of the null hypothesis, that $\partial_t \boldsymbol{\theta}_t^\star = \boldsymbol{\alpha}^{\star\top}(\partial_t \boldsymbol{\phi}_t) = \mathbf{0}$. Then we have

$$\sqrt{n}\ \widehat{\boldsymbol{\alpha}}^\top (\partial_t \boldsymbol{\phi}_t) \xrightarrow{d} \mathcal{N}\left(\mathbf{0}, (\partial_t \boldsymbol{\phi}_t)^\top \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} (\partial_t \boldsymbol{\phi}_t)\right)$$

by the definition of multivariate Normal variables. This is the asymptotic distribution for $\partial_t \boldsymbol{\theta}_t$.

Chapter 7

Discussion

7.1 Contribution Overview

In this thesis, we have made a number of theoretical and methodological contributions towards the field of unnormalised modelling. Most notably, we have proposed a number of new methodologies for truncated density estimation under numerous settings. These methods are applicable to most truncated density estimation problems; such as bounded Euclidean domains, bounded Riemannian manifolds, and also where we only have access to samples from the truncation boundary as opposed to its exact formulation. Chronologically, in order of their presentation, these works build upon one another.

Chapter 3 In this chapter, we proposed Truncated Score Matching, or *TruncSM*, which is a general purpose method for truncated density estimation when the domain of truncation is non-trivial. It is applicable for any generic domain, and requires the specification of only a function measuring the distance from a given point to the boundary. Under a suite of numerical experiments, we have shown the power of the proposed method. *TruncSM* is broadly applicable, but the requirement of the specification of a distance function means it has some drawbacks, for example, when the functional form of the boundary is unknown.

Chapter 4 In this chapter, we extended *TruncSM* and Mardia et al. (2016) to where the domain is a manifold with boundary, called Truncated Manifold Score Matching (TMSM). This is not a straightforward extension, since the derivation in *TruncSM* to yield a tractable objective function requires an application of integration by parts which holds only under a Euclidean assumption on the domain. Instead, we have used a variation of Stokes' theorem to apply a similar process. We presented an application of TMSM to truncated domains on the sphere, by using distance functions also defined for the sphere, given by the Haversine distance and the

projected Euclidean distance. Across numerical experiments, we showed that by accounting for the manifold, we obtain more accurate results than *TruncSM* on the sphere.

Chapter 5 In this chapter, we presented a different way of performing truncated density estimation which does not rely on the specification of a distance function, unlike chapters 3 and 4. This is achieved by introducing a new type of Stein class, called an approximate Stein class. Truncated density estimation using this method requires only a set of points which are samples from the truncation boundary. We hypothesise that this is easier to obtain in practice than a distance function; for example, in geographical applications it is easier to acquire a readily available set of coordinates that define a domain, rather than a functional form of a boundary. We presented several numerical experiments which highlighted that even with this relaxed set of assumptions, this method performs on-par with *TruncSM*, and surpasses it for some specific examples.

Chapter 6 In this chapter, we presented an alternative method for unnormalisable density functions: by using time score matching, we develop a general purpose algorithm for changepoint detection that does not require any specific assumption on the type of changepoint expected. This flexibility is achieved by modelling the derivative of the density parameter with respect to time, which has broad applicability and meaningful intuition. Since the score matching objective does not require knowledge of the normalising constant, the range of data distributions which this method admits is extremely large in scope. We tested the method on a diverse set of problems, which showed that our method provides comparable results to changepoint detection methods that are designed specifically for a certain problem type. Notably, when the task is more complex, this method outperformed prior changepoint detection algorithms.

7.2 Related Work

Chapters 3 and 4 These works have built upon score matching (Hyvärinen, 2005), as well as its extensions by Hyvärinen (2007), Mardia et al. (2016), Yu et al. (2018), and Yu et al. (2019). We consider chapters 3 and 4 as further extensions of score matching that generalise its use to truncated domains and truncated domains on a manifold.

Previous methods for truncated density estimation have been limited to estimating Normal distributions only, (Daskalakis et al., 2018, 2019), whereas chapter 3 is the first method to our knowledge that is applicable to any generic truncated domain. However, a recent work by Lee et al. (2023) proposed an algorithm based on projected stochastic gradient descent that can learn a more general class of distributions in log-concave exponential families.

Whilst the methods presented in these chapters are for estimation in bounded domains, Yu et al. (2021) proposed an extension of chapter 3 to unbounded Euclidean domains, where the boundary, ∂V, is instead component-wise. A separate work by Scealy and Wood (2023) studied polynomially tilted pairwise interaction models for analysing compositional data, which also extended chapter 3 to manifolds with boundary using score matching, in a similar way to chapter 4, primarily focusing on the simplex (and by a simple transformation, the upper

hemisphere). Scealy and Wood proposed a distance function, inspired by chapter 3, given by the Euclidean distance on the simplex. This bears similarities to the projected Euclidean distance proposed in section 4.2.2.1.

Score-based methods, and score matching particularly, have seen recent developments in the application of generative modelling via diffusion processes. Recent works such as De Bortoli et al. (2022) have introduced an extension of diffusion models to Riemannian manifolds. Fishman et al. (2023) consider a further application of these models to bounded Riemannian manifolds by including a distance function which scales the score function, which was inspired by chapter 3.

Chapter 5 The work in this chapter was inspired by a number of unnormalised modelling methodologies, most notably Stein discrepancies (Gorham and Mackey, 2015; Stein, 1972), and kernelised Stein discrepancies (Chwialkowski et al., 2016; Liu et al., 2016). Since we propose the new framework of the approximate Stein class, we view this also as a more general extension of Stein classes that work in an approximate sense.

Several works have designed Stein discrepancies for which the Stein operator holds in an approximate sense. To assess the quality of graph generating functions, Xu and Reinert (2022a) introduce approximate Stein operators, which has some similarity to the approximate Stein class presented in this chapter. Later, Xu and Reinert (2022b) develop a non-parametric Stein operator which is an consistent estimator of the Langevin Stein operator, whose approximation quality increases with number of samples. It is likely that both of these methods fall under the approximate Stein class framework presented in this chapter.

Chapter 6 Similar to chapters 3 and 4, this work uses score matching (Hyvärinen, 2005), but instead of using the classical data score function, uses the time derivative of the log density function, called time score matching, which was introduced by Choi et al. (2022). Choi et al. extended telescoping density ratio estimation from discrete bridges to continuous paths via estimation of the time score function, and along the same lines, we believe that estimating the time score function in the changepoint detection setting has extended the classical moving-window based changepoint methods to continuous variations. This is due to these moving-window based methods, such as MOSUM (Eichinger and Kirch, 2018; Meier et al., 2021), measure the difference between a statistic (usually the mean) before and after a candidate changepoint. In essence, this can be seen as a form of estimating the derivative in a discrete setting, as it judges how much a quantity changes over some fixed timescale. This is similar to our method, but we estimate the derivative directly in a continuous fashion.

7.3 Future Work and Limitations

Chapter 3 The choice of the weighting function is a key point of interest in this work. Whilst the distance function naturally arises from maximising a Stein discrepancy, it is based on the assumption that the weighting function is Lipschitz continuous, which is not a flexible function class. A more adaptive choice of a weighting function could produce a better performing estimator, but would likely require larger computational overhead if the adaptive selection

procedure is based on an outer optimisation problem. Chapter 5 is one such way of adaptively choosing the distance function based on kernelised Stein discrepancies instead of score matching.

Section 3.4.2 detailed an experiment related to compensating for over-trimming in outlier detection. We showed that this method is able to recover the density of an uncorrupted dataset effectively, and this was tested successfully for synthetic data as well as latently embedded image data. In higher dimensions, such as higher-dimensional images, with more complicated truncation boundaries from more complex outlier detection methods, the distance function used in this method may be intractable or very computationally expensive to evaluate. We hypothesise that rather than using a distance function as the weighting function, we could instead directly use the discriminatory function that is output by the outlier detection method. More specifically, the outlier detection method outputs a function $f(\boldsymbol{x})$ for which $\boldsymbol{x}$ is classified as an outlier if $f(\boldsymbol{x}) < 0$, and it is an inlier if $f(\boldsymbol{x}) > 0$. This means that our truncation boundary is given by the level set where $f(\boldsymbol{x}) = 0$. This is exactly the conditions needed for the truncated score matching derivation in this chapter. Further study of using f as the weighting function could produce a better performing density estimator in this scenario, and is an interesting direction for future work.

Chapter 4 The estimator proposed in this chapter is applicable to any manifold for which Stokes' Theorem holds, and therefore a natural extension of this work is to consider other manifolds; we only discussed applications on the two-dimensional sphere, inspired by applications on the surface of the Earth. Other manifolds would require different distribution specification and different weighting functions. To generalise this work further, a more general purpose weighting function could be developed that would be applicable to any Riemannian manifold. For example, similarly to the projected Euclidean distance function, by projecting from any manifold into the Euclidean domain, or a more general geodesic distance function. Further extensions of this work could include generalising truncated score matching to other types of manifolds, such as Grassmanian manifolds.

Chapter 5 This method relies on a number of hyperparameters, and whilst this is typical in most kernel-based methodologies, there could be benefits to adaptively choosing these hyperparameters based on some selection criteria such as cross-validation. There is also a large computational expense; one needs to invert the kernel matrix of boundary points, which scales cubically with m, the number of samples from the boundary. One experiment has shown that the distribution of boundary points affects the estimation error, and therefore possible future work could include adaptively selecting the most relevant boundary points to include, possibly reducing the computational expense whilst only minimally affecting the performance of the method.

Chapter 6 We proposed a framework that is independent of the modelling choice for the time-varying derivative, $\partial_t \boldsymbol{\theta}_t$, but choice of this model ultimately impacts the performance of the method. We present a few choices for models, but it is possible that better choices of these models could hence produce a better performing estimator. An interesting example for future

work could be exploring when the model is an element of a reproducing Kernel hilbert space (RKHS), which would be extremely flexible and likely would have a closed form solution.

Since the approach presented in this chapter is quite flexible, due to estimation of the derivative of the data density's parameters directly, we suppose that there is a large amount of downstream application that this method could be applied to. More complicated changepoint detection problems, such as the concept drift example presented in section 6.6.3, may benefit from use of this method. Along the lines of that experiment, one could use a similar approach to detecting overall changes in a video feed embedded by a convolutional neural network.

Our changepoint detection method is an offline method, meaning that one needs to observe the full dataset to perform meaningful inference. It is possible that the method is adaptable to an online procedure, where detection happens iteratively and the detection procedure is updated as more data is streamed into the process. This would require redefining the timescale as well as the asymptotic distribution of the estimator with each data update, which would be an expensive computation in the current version of the method. We believe this is an interesting direction for future work, and would be a natural extension of our work.

Bibliography

Aminikhanghahi, S. and D. J. Cook (2017).
A survey of methods for time series change point detection.
Knowledge and information systems 51(2), 339–367.

Arbelaez, P., M. Maire, C. Fowlkes, and J. Malik (2010).
Contour detection and hierarchical image segmentation.
IEEE transactions on pattern analysis and machine intelligence 33(5), 898–916.

Arlot, S., A. Celisse, and Z. Harchaoui (2019).
A kernel multiple change-point algorithm via model selection.
Journal of Machine Learning Research 20(162), 1–56.

Atkinson, K. and W. Han (2005).
Theoretical numerical analysis, Volume 39.
Springer.

Bao, F., C. Li, K. Xu, H. Su, J. Zhu, and B. Zhang (2020).
Bi-level score matching for learning energy-based latent variable models.
Advances in Neural Information Processing Systems 33, 18110–18122.

Barp, A., F.-X. Briol, A. Duncan, M. Girolami, and L. Mackey (2019).
Minimum stein discrepancy estimators.
In *Advances in Neural Information Processing Systems*, Volume 32.

Bauer, P. and P. Hackl (1980).
An extension of the mosum technique for quality control.
Technometrics 22(1), 1–7.

Bhandari, G., R. Kawitkar, and M. Borawake (2013).
Audio segmentation for speech recognition using segment features.
International Journal of Computer Technology and Applications 4(2), 182–186.

Box, G. E., G. M. Jenkins, G. C. Reinsel, and G. M. Ljung (2015).
Time series analysis: forecasting and control.
John Wiley & Sons.

Boyd, S. and L. Vandenberghe (2004).
Convex optimization.
Cambridge university press.

Caldarelli, E., P. Wenk, S. Bauer, and A. Krause (2022).
Adaptive gaussian process change point detection.
In *International Conference on Machine Learning*, pp. 2542–2571. PMLR.

Celisse, A., G. Marot, M. Pierre-Jean, and G. Rigaill (2018).
New efficient algorithms for multiple change-point detection with reproducing kernels.
Computational Statistics & Data Analysis 128, 200–220.

Chen, H. and A. F. Murray (2003).
Continuous restricted boltzmann machine with an implementable training algorithm.
IEE Proceedings-Vision, Image and Signal Processing 150(3), 153–158.

Chen, L. H., L. Goldstein, and Q.-M. Shao (2011).
Normal approximation by Stein's method, Volume 2.
Springer.

Chen, Y. and T. Hanson (2014).
Bayesian nonparametric density estimation for doubly-truncated data.
Statistics and Its Interface 7(4), 455–463.

Chicago Police Department (2019).
Data: Crimes - 2001 to present.
Accessed 25 November 2019.

Cho, H., H. Maeng, I. Eckley, and P. Fearnhead (2023).
High-dimensional time series segmentation via factor-adjusted vector autoregressive modeling.
Journal of the American Statistical Association 0(0), 1–13.

Cho, H. and D. Owens (2022).
High-dimensional data segmentation in regression settings permitting heavy tails and temporal dependence.
arXiv preprint arXiv:2209.08892.

Choi, K., C. Meng, Y. Song, and S. Ermon (2022).
Density ratio estimation via infinitesimal classification.
In *International Conference on Artificial Intelligence and Statistics*, pp. 2552–2573. PMLR.

Chwialkowski, K., H. Strathmann, and A. Gretton (2016, 20–22 Jun).
A kernel test of goodness of fit.
In M. F. Balcan and K. Q. Weinberger (Eds.), *Proceedings of The 33rd International Conference on Machine Learning*, Volume 48 of *Proceedings of Machine Learning Research*, New York, New York, USA, pp. 2606–2615. PMLR.

Daskalakis, C., T. Gouleakis, C. Tzamos, and M. Zampetakis (2018, 09).
Efficient statistics, in high dimensions, from truncated samples.

Daskalakis, C., T. Gouleakis, C. Tzamos, and M. Zampetakis (2019, 25–28 Jun).
Computationally and statistically efficient truncated regression.
99, 955–960.

De Bortoli, V., E. Mathieu, M. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet (2022).
Riemannian score-based generative modelling.
Advances in Neural Information Processing Systems 35, 2406–2422.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019).
Bert: Pre-training of deep bidirectional transformers for language understanding.
In *North American Chapter of the Association for Computational Linguistics*.

Dominici, F., A. McDermott, S. L. Zeger, and J. M. Samet (2002).
On the use of generalized additive models in time-series studies of air pollution and health.
American journal of epidemiology 156(3), 193–203.

Eichinger, B. and C. Kirch (2018).
A MOSUM procedure for the estimation of multiple random change points.
Bernoulli 24(1), 526 – 564.

Fishman, N., L. Klarner, V. D. Bortoli, E. Mathieu, and M. J. Hutchinson (2023).
Diffusion models for constrained domains.
Transactions on Machine Learning Research.
Expert Certification.

Gama, J., I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia (2014).
A survey on concept drift adaptation.
ACM computing surveys (CSUR) 46(4), 1–37.

Gao, R., Y. Lu, J. Zhou, S.-C. Zhu, and Y. N. Wu (2018).
Learning generative convnets via multi-grid modeling and sampling.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9155–9164.

Gorham, J. and L. Mackey (2015).
Measuring sample quality with stein's method.
Advances in Neural Information Processing Systems 28.

Gorham, J., A. Raj, and L. Mackey (2020).
Stochastic stein discrepancies.
In *Advances in Neural Information Processing Systems*, Volume 33, pp. 17931–17942. Curran Associates, Inc.

Gutmann, M. and A. Hyvärinen (2010).
Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.
In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, Volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304. PMLR.

Gutmann, M. U. and J.-i. Hirayama (2011).
Bregman divergence as general framework to estimate unnormalized statistical models.
In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, pp. 283–290. AUAI Press.

Gutmann, M. U. and A. Hyvärinen (2012).
Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics.
Journal of Machine Learning Research 13(11), 307–361.

Hastie, T. and R. Tibshirani (1986).
Generalized Additive Models.
Statistical Science 1(3), 297 – 310.

Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman (2009).
The elements of statistical learning: data mining, inference, and prediction, Volume 2.
Springer.

He, X., R. S. Zemel, and M. A. Carreira-Perpinán (2004).
Multiscale conditional random fields for image labeling.
In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Volume 2, pp. II–II. IEEE.

Hinton, G. E. (2002).
Training products of experts by minimizing contrastive divergence.
Neural computation 14(8), 1771–1800.

Hjelm, R. D., A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio (2019).
Learning deep representations by mutual information estimation and maximization.
In *International Conference on Learning Representations*.

Ho, J., A. Jain, and P. Abbeel (2020).
Denoising diffusion probabilistic models.
In *Advances in Neural Information Processing Systems*, Volume 33, pp. 6840–6851. Curran Associates, Inc.

Hutchinson, M. F. (1989).
A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines.
Communications in Statistics-Simulation and Computation 18(3), 1059–1076.

Hyvärinen, A. (2005).
Estimation of non-normalized statistical models by score matching.
Journal of Machine Learning Research 6(Apr), 695–709.

Hyvärinen, A. (2007).
Some extensions of score matching.
Computational statistics & data analysis 51(5), 2499–2512.

IPCC (2022).
Climate Change 2022: Impacts, Adaptation and Vulnerability.
Summary for Policymakers. Cambridge, UK and New York, USA: Cambridge University Press.

Jackson, B., J. D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T. T. Tsai (2005).
An algorithm for optimal partitioning of data on an interval.
IEEE Signal Processing Letters 12(2), 105–108.

Jiang, W., J. Qin, L. Wu, C. Chen, T. Yang, and L. Zhang (2023, 23–29 Jul).
Learning unnormalized statistical models via compositional optimization.
In *Proceedings of the 40th International Conference on Machine Learning*, Volume 202 of *Proceedings of Machine Learning Research*, pp. 15105–15124. PMLR.

Jitkrittum, W., W. Xu, Z. Szabo, K. Fukumizu, and A. Gretton (2017).
A linear-time kernel goodness-of-fit test.
In *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Jones, P. D., M. New, D. E. Parker, S. Martin, and I. G. Rigor (1999).
Surface air temperature and its changes over the past 150 years.
Reviews of Geophysics 37(2), 173–199.

Jost, J. (2008).
Riemannian geometry and geometric analysis, Volume 42005.
Springer.

Kanagawa, H., W. Jitkrittum, L. Mackey, K. Fukumizu, and A. Gretton (2019).
A kernel stein test for comparing latent variable models.
arXiv preprint arXiv:1907.00586.

Kent, J. T. (1982).
The fisher-bingham distribution on the sphere.
Journal of the Royal Statistical Society: Series B (Methodological) 44(1), 71–80.

Killick, R., P. Fearnhead, and I. A. Eckley (2012).
Optimal detection of changepoints with a linear computational cost.
Journal of the American Statistical Association 107(500), 1590–1598.

Kim, J., H.-S. Oh, and H. Cho (2024).
Moving sum procedure for change point detection under piecewise linearity.
Technometrics (just-accepted), 1–19.

Kirch, C. and K. Reckruehm (2022).
Data segmentation for time series based on a general moving sum approach.
arXiv preprint arXiv:2207.07396.

Kolar, M. and E. P. Xing (2011, 11–13 Apr).
On time varying undirected graphs.
In G. Gordon, D. Dunson, and M. Dudík (Eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Volume 15 of *Proceedings of Machine Learning Research*, Fort Lauderdale, FL, USA, pp. 407–415. PMLR.

Kong, L., C. de Masson d'Autume, L. Yu, W. Ling, Z. Dai, and D. Yogatama (2020).
A mutual information maximization perspective of language representation learning.
In *International Conference on Learning Representations*.

Krishnamoorthy, A. and D. Menon (2013).
Matrix inversion using cholesky decomposition.
In *2013 signal processing: Algorithms, architectures, arrangements, and applications (SPA)*, pp. 70–72. IEEE.

Krizhevsky, A. (2009).
Learning multiple layers of features from tiny images.
University of Toronto.

Lampert, C. H. (2015).
Predicting the future behavior of a time-varying probability distribution.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 942–950.

Landsea, C. W. and J. L. Franklin (2013).
Atlantic hurricane database uncertainty and presentation of a new database format.
Mon. Wea. Rev., 141, 3576-3592. Date accessed: 10th May 2022.

Lee, J., A. Wibisono, and E. Zampetakis (2023).
Learning exponential families from truncated samples.
In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), *Advances in Neural Information Processing Systems*, Volume 36, pp. 34826–34851. Curran Associates, Inc.

Lee, J. M. (2013).
Smooth manifolds.
In *Introduction to Smooth Manifolds*, pp. 1–31. Springer.

Li, S. Z. (2009).
Markov random field modeling in image analysis.
Springer Science & Business Media.

Lindeløv, J. K. (2020).
mcp: An r package for regression with multiple change points.
OSF Preprints.

Lindsey, R. and L. Dahlman (2024).
Climate Change: Global Temperature — climate.gov.
https://www.climate.gov/news-features/understanding-climate/climate-change-global-temperature.
[Accessed 27-03-2024].

Liu, B., E. Rosenfeld, P. K. Ravikumar, and A. Risteski (2022).
Analyzing and improving the optimization landscape of noise-contrastive estimation.
In *International Conference on Learning Representations*.

Liu, Q., J. Lee, and M. Jordan (2016).
A kernelized stein discrepancy for goodness-of-fit tests.
In *International conference on machine learning*, pp. 276–284. PMLR.

Liu, S., T. Kanamori, and D. J. Williams (2022).
Estimating density models with truncation boundaries using score matching.
Journal of Machine Learning Research 23(186), 1–38.

Liu, Z., P. Luo, X. Wang, and X. Tang (2015).
Deep learning face attributes in the wild.
In *Proceedings of International Conference on Computer Vision (ICCV)*.

Lu, J., A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang (2018).
Learning under concept drift: A review.
IEEE transactions on knowledge and data engineering 31(12), 2346–2363.

Lu, Y., R. Maulik, T. Gao, F. Dietrich, I. G. Kevrekidis, and J. Duan (2022, 3).
Learning the temporal evolution of multivariate densities via normalizing flows.
Chaos: An Interdisciplinary Journal of Nonlinear Science 32(3).

Malladi, R., G. P. Kalamangalam, and B. Aazhang (2013).
Online bayesian change point detection algorithms for segmentation of epileptic activity.
In *2013 Asilomar conference on signals, systems and computers*, pp. 1833–1837. IEEE.

Mardia, K. V. and P. E. Jupp (2009).
Directional statistics.

Mardia, K. V., J. T. Kent, and A. K. Laha (2016).
Score matching estimators for directional distributions.
arXiv preprint arXiv:1604.08470.

Meier, A., C. Kirch, and H. Cho (2021).
mosum: A package for moving sums in change-point analysis.
Journal of Statistical Software 97, 1–42.

Meng, C., L. Yu, Y. Song, J. Song, and S. Ermon (2020).
Autoregressive score matching.
Advances in Neural Information Processing Systems 33, 6673–6683.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953).
Equation of state calculations by fast computing machines.
The journal of chemical physics 21(6), 1087–1092.

Mikolov, T., K. Chen, G. S. Corrado, and J. Dean (2013).
Efficient estimation of word representations in vector space.
In *International Conference on Learning Representations*.

Mnih, A. and K. Kavukcuoglu (2013).
Learning word embeddings efficiently with noise-contrastive estimation.
In *Neural Information Processing Systems*.

Mnih, A. and Y. W. Teh (2012).
A fast and simple algorithm for training neural probabilistic language models.
In *International Conference on Machine Learning*.

Moldovan, I. (2011).
Stock markets correlation: Before and during the crisis analysis.
Theoretical and Applied Economics 8(8), 111.

Moreira, C. and J. de Uña Álvarez (2012).
Kernel density estimation with doubly truncated data.
Electronic Journal of Statistics 6, 501–521.

Nadaraya, E. A. (1964).
On estimating regression.
Theory of Probability & Its Applications 9(1), 141–142.

National Climatic Data Center (2022).
Ncdc storm events database.
Date accessed: 10th May 2022.

NCEA (2024).
Global surface temperature anomalies.
National Centers for Environmental Information. https://www.ncei.noaa.gov/access/monitoring/global-temperature-anomalies/. Accessed 1 Feb 2024.

Oliver, J. E. (2008).
Encyclopedia of world climatology.
Springer Science & Business Media.

Owens, D. (2023).
Moving sum based procedures for changes in the mean.
https://pypi.org/project/mosum/0.2.0/.

Owens, D. (2024).
Data Segmentation and High Dimensional Time Series Analysis.
Ph. D. thesis, University of Bristol.
Unpublished thesis.

O'Shea, K. and R. Nash (2015).
An introduction to convolutional neural networks.
ArXiv abs/1511.08458.

Pang, T., K. Xu, C. Li, Y. Song, S. Ermon, and J. Zhu (2020).
Efficient learning of generative models via finite-difference score matching.
In *Advances in Neural Information Processing Systems*, Volume 33, pp. 19175–19188.

Parzen, E. (1962).
On Estimation of a Probability Density Function and Mode.
The Annals of Mathematical Statistics 33(3), 1065 – 1076.

Quax, R., D. Kandhai, and P. M. Sloot (2013).
Information dissipation as an early-warning signal for the lehman brothers collapse in financial time series.
Scientific reports 3(1), 1898.

Reeves, J., J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu (2007).
A review and comparison of changepoint detection techniques for climate data.
Journal of applied meteorology and climatology 46(6), 900–915.

Rosenblatt, M. (1956).
Remarks on Some Nonparametric Estimates of a Density Function.
The Annals of Mathematical Statistics 27(3), 832 – 837.

Satow, J. (2008).
Ex-SEC Official Blames Agency for Blow-Up of Broker-Dealers.
URL: https://www.nysun.com/article/business-ex-sec-official-blames-agency-for-blow-up.
Accessed 1 Feb 2024.

Scealy, J. L. and A. T. A. Wood (2023).
Score matching for compositional distributions.
Journal of the American Statistical Association 118(543), 1811–1823.

Schölkopf, B., R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt (1999).
Support vector method for novelty detection.
Advances in neural information processing systems 12.

Serfling, R. J. (2009).
Approximation theorems of mathematical statistics.
John Wiley & Sons.

Shi, J., C. Liu, and L. Mackey (2021).
Sampling with mirrored stein operators.
arXiv preprint arXiv:2106.12506.

Song, Y. and S. Ermon (2019).
Generative modeling by estimating gradients of the data distribution.
In *Advances in Neural Information Processing Systems*, Volume 32.

Song, Y., S. Garg, J. Shi, and S. Ermon (2020).
Sliced score matching: A scalable approach to density and score estimation.
In *Uncertainty in Artificial Intelligence*, pp. 574–584. PMLR.

Song, Y. and D. P. Kingma (2021).
How to train your energy-based models.
arXiv preprint arXiv:2101.03288.

Song, Y., J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole (2021).
Score-based generative modeling through stochastic differential equations.
In *International Conference on Learning Representations*.

Stein, C. (1972).
A bound for the error in the normal approximation to the distribution of a sum of dependent random variables.
In *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory*, pp. 583–602. University of California Press.

Steinwart, I. and A. Christmann (2008).
Support vector machines.
Springer Science & Business Media.

Stewart, J., D. K. Clegg, and S. Watson (2020).
Calculus: early transcendentals.
Cengage Learning.

Teh, Y. W., M. Welling, S. Osindero, and G. E. Hinton (2003).
Energy-based models for sparse overcomplete representations.
Journal of Machine Learning Research 4(Dec), 1235–1260.

Tieleman, T. (2008).
Training restricted boltzmann machines using approximations to the likelihood gradient.
In *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071.

Truong, C., L. Oudre, and N. Vayatis (2020).
Selective review of offline change point detection methods.
Signal Processing 167, 107299.

Turnbull, B. W. (1976).
The empirical distribution function with arbitrarily grouped, censored, and truncated data.
Journal of the royal statistical society series b-methodological 38, 290–295.

UCLA: Statistical Consulting Group (2022).
Truncated regression, stata data analysis examples.
Accessed March 17, 2023.

Vincent, P. (2011).
A connection between score matching and denoising autoencoders.
Neural computation 23(7), 1661–1674.

Vogt, M. and H. Dette (2015).
Detecting gradual changes in locally stationary processes.
The Annals of Statistics 43(2), 713 – 740.

Watson, G. S. (1964).
Smooth regression analysis.
Sankhyā: The Indian Journal of Statistics, Series A, 359–372.

Widmer, G. and M. Kubat (1996).
Learning in the presence of concept drift and hidden contexts.
Machine learning 23, 69–101.

Williams, D. J. and S. Liu (2022).
Score matching for truncated density estimation on a manifold.
In *Topological, Algebraic and Geometric Learning Workshops 2022*, pp. 312–321. PMLR.

Wood, A. T. (1994).
Simulation of the von mises fisher distribution.
Communications in statistics-simulation and computation 23(1), 157–164.

WRDS (2024).
Wharton Research Data Service. https://wrds-www.wharton.upenn.edu/. Accessed 1 Feb 2024.

Wu, S., E. Diao, K. Elkhalil, J. Ding, and V. Tarokh (2022).
Score-based hypothesis testing for unnormalized models.
IEEE Access 10, 71936–71950.

Xu, W. (2022, 28–30 Mar).
Standardisation-function kernel stein discrepancy: A unifying view on kernel stein discrepancy tests for goodness-of-fit.
Volume 151 of *Proceedings of Machine Learning Research*, pp. 1575–1597. PMLR.

Xu, W. and T. Matsuda (2021).
Interpretable stein goodness-of-fit tests on riemannian manifold.

In *Proceedings of the 38th International Conference on Machine Learning*, Volume 139 of *Proceedings of Machine Learning Research*, pp. 11502–11513. PMLR.

Xu, W. and G. D. Reinert (2022a).
Agrasst: Approximate graph stein statistics for interpretable assessment of implicit graph generators.
Advances in Neural Information Processing Systems 35, 24268–24279.

Xu, W. and G. D. Reinert (2022b).
A kernelised stein statistic for assessing implicit generative models.
Advances in Neural Information Processing Systems 35, 7277–7289.

Yang, J., Q. Liu, V. Rao, and J. Neville (2018).
Goodness-of-fit testing for discrete distributions via stein discrepancy.
In *Proceedings of the 35th International Conference on Machine Learning*, Volume 80 of *Proceedings of Machine Learning Research*, pp. 5561–5570. PMLR.

Yau, C. Y. and Z. Zhao (2016).
Inference for multiple change points in time series via likelihood ratio scan statistics.
Journal of the Royal Statistical Society. Series B (Statistical Methodology) 78(4), 895–916.

Yu, S., M. Drton, and A. Shojaie (2018, 09–11 Apr).
Graphical models for non-negative data using generalized score matching.
In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, Volume 84 of *Proceedings of Machine Learning Research*, pp. 1781–1790. PMLR.

Yu, S., M. Drton, and A. Shojaie (2019).
Generalized score matching for non-negative data.
Journal of Machine Learning Research 20(76), 1–70.

Yu, S., M. Drton, and A. Shojaie (2021, 01).
Generalized score matching for general domains.
Information and Inference: A Journal of the IMA.
iaaa041.