From Differential Equations to Differential Geometry: Aspects of Regularisation in Machine Learning

---

Dissertation

zur Erlangung des Grades des

Doktors der Naturwissenschaften

der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

vorgelegt von Dipl.-Phys.

Florian Steinke

Herrenberg, Februar 2009

Wissenschaftliches Kolloquium: 18. Mai 2009
Dekan: Prof. Dr. Joachim Weickert
Prüfungsausschuss:
Prof. Dr. Hans-Peter Seidel (Vorsitzender des Püfungsauschuss)
Prof. Dr. Matthias Hein (1. Berichterstatter)
Prof. Dr. Bernhard Schölkopf (2. Berichterstatter)
Prof. Dr. Jeff Bilmes (3. Berichterstatter)

Abstract

Machine learning requires the use of prior assumptions which can be encoded into learning algorithms via regularisation techniques. In this thesis, we examine in three examples how suitable regularisation criteria can be formulated, what their meaning is, and how they lead to efficient machine learning algorithms.

Firstly, we describe a joint framework for positive definite kernels, Gaussian processes, and regularisation operators which are commonly used objects in machine learning. With this in mind, it is then straightforward to see that linear differential equations are an important special case of regularisation operators. The novelty of our description is the broad, unifying view connecting kernel methods and linear system identification.

We then discuss Bayesian inference and experimental design for sparse linear models. The model is applied to the task of gene regulatory network reconstruction, where the assumed network sparsity improves reconstruction accuracy and our proposed experimental design setup outperforms prior methods significantly.

Finally, we examine non-parametric regression between Riemannian manifolds, a topic that has received little attention so far. We propose a regularised empirical risk minimisation framework, ensuring with the help of differential geometry that it does not depend on the representation of the input and output manifold. We apply our approach to several practical learning tasks in robotics and computer graphics.

Zusammenfassung

A priori Annahmen sind für das maschinelle Lernen unabdingbar, und eine Möglichkeit, diese Annahmen in Lernalgorithmen zu kodieren, ist die Regularisierung. In dieser Dissertation wird anhand von drei Beispielen untersucht, wie man sinnvolle Regularisierungskriterien formulieren kann und wie daraus effiziente Lernalgorithmen entstehen.

Zuerst werden Zusammenhänge zwischen positiv definiten Kernen, Gaußprozessen und Regularisierungsoperatoren, wie sie häufig im maschinellen Lernen verwendet werden, beschrieben. Dabei wird klar, dass lineare Differentialgleichungen einen wichtigen Spezialfall solcher Operatoren darstellen, und dass Kernmethoden daher eng mit der linearen Systemidentifikation verwandt sind.

Danach wird Bayessche Inferenz und Versuchsplanung in dünnbesetzten, linearen Modellen diskutiert. Das Modell wird auf die Rekonstruktion von genetischen Interaktionsnetzwerken angewendet. Durch die Annahmne, dass die zu schätzenden Vektoren dünnbesetzt sind, und durch die neuartige Versuchsplanungsmethode ergeben sich signifikante Verbesserungen der Rekonstruktion.

Schließlich wird nichtparametrische Regression zwischen Riemannschen Mannigfaltigkeiten mittels regularisierter, empirischer Risikominimierung untersucht. Es wird darauf geachtet, dass die Regularisierung unabhängig von der Darstellung der Mannigfalitgkeiten ist. Die vorgestellte Methode wird anhand verschiedener Beispiele aus der Robotik und der Computergraphik getestet.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisors Prof. Dr. Bernhard Schölkopf, Prof. Dr. Matthias Hein, and Dr. Matthias Seeger. Starting with my Master's thesis, Bernhard Schölkopf introduced me to the field of machine learning, and to the scientific world in general. I was always impressed by the wealth of ideas and proposals he offered to me, while leaving me any freedom to develop my own plans. Through his far-reaching network he brought me into contact with many interesting figures of current machine learning research. He offered me the chance to perform an internship at NICTA in Canberra, Australia, and to co-organise the machine learning summer school in Tübingen. Also concerning personal decisions, he was an invaluable source of help and advice. I cannot overstate the gratitude I feel towards his continued support. While I first had to adapt to Matthias Seeger's technical language, I grew to highly value his comments and advice. Sometimes I only realised after weeks that Matthias' instantaneous suggestion in the middle of a discussion was just the right idea. During the last part of my thesis, I collaborated closely with Matthias Hein. I was repeatedly impressed by the precision and depth of his work. I thank Matthias for officially supervising my thesis at Saarland University. All my supervisors were always open and available for questions and advice regarding all matters, and I am greatly thankful for this. I feel that I have developed a good personal relationship reaching well-beyond research with all of them.

I thank the Max Planck Society for providing me with financial support for writing this thesis. Travelling to conferences and workshops was always greatly encouraged. Furthermore, I was offered the opportunity to perform a two month internship at NICTA, Canberra, Australia, which I gratefully acknowledge.

I thank my further co-authors Volker Blanz, Koji Tsuda, Matthias Hofmann, and Jan Peters. The collaboration with them was exciting and fruitful. During our joint work, I learned a lot about computer graphics, bioinformatics, medical imaging, and robotics. Moreover, I found the atmosphere at the department of Empirical Inference at the MPI for Biological Cybernetics highly inspiring. The broad scope of the institute, the many guests and co-workers who gave stimulating talks, and the open atmosphere with frequent and interactive discussions reaching well beyond machine learning topics, they all kept me highly motivated and broadly interested. Without this stimulating interaction, my thesis would not have been possible. I want to mention specifically Yasemin Altun, Marc Deisenroth, Jan Eichhorn, Peter Gehler, Sebastian Gerwinn, Arthur Gretton, Frank Jäkel, Markus Maier, Hannes Nickisch, Sebastian Nowozin, Carl Rasmussen, Fabian Sinz, and Christian Walder, but many more could equally well be listed here. I also enjoyed organising the 9th Machine Learning Summer School 2007 in Tübingen together with Arthur Gretton, Gunnar Rätsch and Bernhard Schölkopf. It was as much fun and experience, as it was work to prepare. During my time in Australia, I lead several interesting discussions from which I learned a lot with Knut Hüper, Alex Smola, Markus Hegland, and Bob Williamson. I thank them for taking the time.

I also want to thank my parents, who have kept supporting me during this thesis in many ways, and last but not least, I thank my partner Sabine Roos. She has supported me throughout this thesis' work, not least by chasing me out of bed in the mornings ;-).

Contents

Chapter 1

Introduction

Many machine learning applications try to generalise example-based knowledge to new situations. We will argue in this introduction that this process critically requires the use of suitable prior knowledge.

A principled way of expressing and incorporating prior knowledge into learning algorithms is regularisation, that is, considering a large set of possible hypotheses but weighting them differently depending on their a priori plausibility. The three main chapters of this thesis present several different aspects of regularisation when applied for machine learning purposes. In particular, we consider connections between differential equations and regularisation in kernel methods, we use sparsity as a regularisation criterion in Bayesian network models, and we discuss appropriate smoothness criteria for learning between manifolds using differential geometric tools.

1.1 The Importance of Induction

The process of deriving general rules, models, or theories from a finite number of examples is known as *induction*. It not only lies at the heart of machine learning and statistics, but also forms the basis of the scientific method in general.

In machine learning, induction typically takes place in two steps. Given some set of observations that should be described, we first select a suitable model class, and then fit the parameters of the model to the observations, thereby minimising some appropriate error criterion. Instead of selecting the single best parameter set, as it is done in frequentist statistics, Bayesian statistics computes the full a posteriori probability distribution over the parameters. In either setting, the fitted model can then be used to make predictions about new observations, and may also help to better understand the underlying principles of the original dataset.

It are these capabilities of inductive modelling that are not only useful in the rather restricted scope of typical problems in machine learning and statistics, but that have a much broader appeal. In fact, induction is a key step of the general scientific method. Here, we also first select some theoretical framework that could potentially describe the observations, we adapt the parameters, and then exploit the fitted theory for predictive or explanatory purposes. For example, Newtonian mechanics is a mathematical set of rules about object movements, the gravitation constant is a free parameter that is fitted against the observations, and the system

of Newton's laws explains why and how apples fall from trees. This shows that induction is a critical component not only of formal statistics, but of our everyday reasoning about the world we live in. Our plans and decisions critically depend on the models and theories that we derive with the help of induction.

1.2 The Induction Problem

In this section we will argue that meaningful induction is only possible if we make non-trivial prior assumptions. This is to say before actually interpreting any observations we already need to have a rather concrete idea about what models or theories we consider, and these early guesses will have severe effects on what we will conclude from the observations. The problem is that the choice which prior assumptions to use is not always obvious.

To make the above statements clearer, let us look at the long history of this topic. In the philosophy of science, already [Hume, 1748] noted that having observed a certain event arbitrarily often does not logically imply that it is always true. A classic real-world example is that the observation of 100 white swans does not justify the universal statement that all swans are white. In contrast, observing a single black swan renders the general theory invalid. In some sense, induction is thus strongly asymmetric.

Another classic example highlighting the nature of induction is attributed to Laplace [Laplace, 1814]. It is centred on the question whether the sun will rise tomorrow or not. If we encode the sun rising on a particular day with a 1 and the sun not rising with a 0, then the set of possible models or "world theories" is isomorphic to the set of (infinite) binary strings. Restricting our attention to only three days, namely yesterday, today, and tomorrow, we can list all possibilities in a table.

	yesterday	today	tomorrow
not consistent	0	0	0
	0	0	1
	0	1	0
	0	1	1
	1	0	0
	1	0	1
consistent	1	1	0
	1	1	1

Without making any prior assumptions we cannot exclude certain theories a priori. Instead, we should assume that, before any observations are considered, each of these has equal opportunity of being true. If we then include our observations of having seen the sun rising yesterday and this morning, then 6 out of the 8 possible theories turn out to be inconsistent with the observations, and we thus do not need to consider them anymore. However, we are left with two consistent theories, one of which predicts 1 for tomorrow, that is, the sun will rise, and the other 0, that is, the world will end tonight. Since these theories cannot be distinguished based on past observations, we can not know what will happen tomorrow following this argument.

To make this argument even stronger, consider some possible objections. Surely, we know much more about the world than just whether the sun rose the last two days. In the history of

mankind, we have observed millions of sunrises before. We have also gathered many more observations, that are relevant for our reasoning about the world, from other independent sources. So let us include all these observations into the gedankenexperiment by adding many more additional binary positions coding for past observations. This would increase the number of possible theories dramatically. However, it would not change the fact that, after including all the actual observations, there would remain exactly two consistent theories with contradicting predictions for tomorrow.

One could also think that one could evade all this by turning to probability theory, which was the original setup of this experiment as discussed in [Laplace, 1814]. The idea is that maybe we cannot make definite statements about the future, but at least assign some non-uniform probabilities to certain outcomes. Unfortunately, the answer is negative. There are equally many consistent theories supporting each possible outcome. Thus, assuming a priori that all possible theories are equally likely, which is the only non-restrictive prior assumption, the predictive probability of the sun rising tomorrow is exactly 50%, which does not help us at all.

Note that the same argument carries much further than the binary prediction example discussed here. It applies to any kind of prediction problem. As long as we consider all possible theories, we have for each candidate theory many other candidates that are equal to the first, except that they predict each possible other outcome in the future. These theories cannot be distinguished from each other based on past observations. So, if one is consistent, then all the others are, too. If we do not a priori want to favour one or a group of them over the rest, we will again only be able to make trivial predictions, that is, state that something will happen with equal chances for each possible outcome.

The problem that induction without prior assumptions is under-determined is also the basis for Popper's theory of critical rationalism [Popper, 1934]. He states that all we can do in order to achieve scientific progress is to falsify proposed theories or models based on empirical observations, but that there are no means to corroborate a theory. In other words, while observations may help us to filter out some theories from the pool of all potential ones, they cannot help us to select among the remaining candidates.

In machine learning, the impossibility of induction without prior assumptions is commonly known as the no free lunch theorem [Wolpert, 1996]. The statement here is roughly that, if all possible prediction problems are considered, each classifier is on average as good as any other, specifically as good as random guessing.

The same flavour of results shows up in statistical learning theory, for an introduction see [Bousquet et al., 2004]. Here, one tries to bound the error of predictions – the test error – based on the performance of the model on the data used to determine the model – the training error. Such bounds, e.g. [Vapnik, 1995], always include a capacity term. This term measures in an appropriate way how many different sets of observations a model can describe, which is equivalent to measuring how many effectively different theories there are in the model under investigation. If the model is not restricted and no prior assumptions are made, many models could potentially describe all possible observations. In this case, the learning bounds will always become trivial, and nothing can be gained from them.

Note that statistical learning theory can actually give performance guarantees on the test error with high probability, if we are lucky enough to achieve a low training error with a low capacity model. Yet, such guarantees require that the observed data are an i.i.d. sample of the true underlying data distribution. In some tightly controlled cases this assumption seems

obvious, for example, for the tosses of a coin, and it is reassuring that at least in this situation we can make definite predictions. However, we often cannot be sure whether the given data are really the outcomes of i.i.d. random experiments. Such an assumption requires precise knowledge of the setting and the surroundings in which the data were recorded. Especially when considering science in general this is typically not the case. Moreover, the i.i.d. assumption is not a weak assumption, but it is heavily restrictive. It states that the joint probability density of all data points is the product of identical factors, which is a very special situation considering all possible joint distributions.

In sum, we have now collected many arguments showing that induction without prior assumptions or with uniform prior plausibility assigned to all possible models or theories is meaningless. At the same time, the above examples and arguments show that, if we actually do make correct, restrictive enough prior assumptions, then induction can be successful. We can then obtain non-trivial predictions, that is, we can be certain that a given outcome will happen in the future or at least assign a higher than random probability to it.

The need for non-uniform prior assumptions poses, of course, the question which prior assumptions we should use. In the following we will discuss three different regimes for induction where this question is problematic to a varying degree.

The first regime considers science as a whole, where the choice of the "right" prior assumptions is extremely problematic. By definition, prior assumptions cannot be tested experimentally in this case. Alternatively, one could rely on the common sense and say that a set of prior assumptions is good enough if at least most reasonable human beings would agree. But even if everyone would agree, how could we guarantee that mankind was right? Thus, when considering science as a whole we do not know how to choose the right prior assumptions. Since this choice heavily influences which models or theories we derive from our experiments, we can, as a result, never be sure about the general validity of scientific predictions or explanations.

A second regime concerns smaller, non-fundamental problems and questions that arise in science, in our everyday lives, or in technical domains. Here, we typically do not question the mainstream scientific theory about the world and how it works in general, but instead use it as given, fixed background knowledge, from which we can then derive meaningful prior assumptions for our problems at hand. Given that these prior assumptions are correct and precise enough, induction can then help us to derive useful explanations and predictions.

Many problems in machine learning or artificial intelligence, however, fall into a third regime, which lies somewhere in-between the other two. Here, we often have valid background knowledge available, but the complexity of the experimental setup may render the derivation of suitable prior assumptions for our problem at hand difficult. Moreover, consider the long term goal of artificial intelligence to build automatic inference machines. The above arguments make clear that such a machine will never be able to solve all possible induction problems. Yet, that does not say that it is impossible to automate induction for the subset of problems that actually occur in the real world. Determining the necessary abstract "world prior" for this task, however, is difficult.

In conclusion of this section, we should thus always be aware of the need for and the effects of restrictive prior assumptions when working on induction problems.

1.3 Induction and Regularisation

Prior assumptions can principally be included into machine learning algorithms in two ways: One choice is to restrict the set of possible theories or models right from the start. This is often done in classical statistics where it is assumed, for example, that the data are drawn from a Gaussian distribution and other alternatives are not considered when interpreting the observations.

A second option is *regularisation*, which is more common in machine learning and which is the focus of this thesis. Here, we consider all possible hypotheses, or at least a very large set of them, but we weight them differently according to their a priori plausibility. In frequentist statistics, we add an appropriate regularisation term to our fitting objective; in Bayesian treatments we use prior probability distributions over the hypothesis space to express our a priori assumptions.

Note that the first method to include prior knowledge, that is, restricting the set of possible models or theories right from the start, can actually also be expressed via regularisation principles. We just have to assign infinite penalties to the excluded models. As long as there exist models with finite penalties, the excluded ones will not be the minima of frequentist optimisations and they will have zero probability in Bayesian treatments. Thus, they are effectively ignored.

The regularisation principle has a long history and cannot be attributed to a specific piece of work or even a single community. The name "regularisation" originates from the theory of under-determined inverse problems. In so-called Tikhonov regularisation [Tikhonov, 1943] a quadratic Hilbert space norm penalty is used to obtain a unique, stable solution for otherwise under-determined integral equations.

1.4 Regularisation and Simplicity

In many machine learning tasks we do not know the underlying data-generating model precisely, and in this setting, it is not obvious how to determine suitable regularisation criteria. One commonly applied principle in this case is *Ockham's razor*, see for example [Maurer, 1984; Rasmussen and Williams, 2006]: "entia non sunt multiplicanda praeter necessitatem", which translates to "entities must not be multiplied beyond necessity". The idea is that, a priori, simple theories are better than more complicated ones.

One supporting argument for this "meta"-theory is that it is just easier to work with simple theories than with more complicated ones. Another may be that simple theories do not have that many features or "edges" that could potentially be falsified. It also often worked quite well when people adhered to this principle. Note, however, that as argued above none of these explanations guarantees that Ockham's razor is right or will lead to correct predictions in the future.

When applying Ockham's principle, one immediate problem is that simplicity is not easily defined precisely. The impression of what is simple or not is largely dependent on the observer's personal experience, knowledge, and beliefs, and may thus vary considerably between different people. Nevertheless, there are some basic aspects regarding simplicity that are shared amongst many people. Each of the chapters of this thesis can be seen as highlighting one specific such aspect of simplicity. This is described in more detail in the following.

1.4.1 Regularisation and Differential Equations

Kernel methods such as Support Vector Machines, Support Vector regression or Gaussian processes typically estimate functions using kernel-based regularisers which can be interpreted in terms of regularisation operators [Smola et al., 1998]. We will show in Chapter 2 that for many common kernel functions, namely all translation invariant ones, the corresponding regularisation operators are linear differential operators. We can thus interpret the preferred functions for many common kernel machines as (approximate) solutions to *linear differential equations*.

Differential equations are very flexible and simple regularisers. They constrain the *local* behaviour of the target function, for example, by enforcing certain smoothness or slow variation of a given form. At the same time they do *not* constrain the function *globally*, since small violations of the local equations can add up over longer distances, and thus do not lead to strong global restrictions.

1.4.2 Regularisation and Sparsity

Alternatively, we could say that a theory or model is simple if it can explain the observations with *only few causes*. For many common models that take the form of linearly parametrised function expansions, few causes correspond to few non-vanishing terms in the summations, that is, *sparse* coefficient vectors containing many zeros.

In Chapter 3, we will explore a problem where simplicity, but also a heap of independently gathered experimental evidence, suggests the appropriateness of a sparsity prior. When reconstructing genetic interaction networks from micro-array measurements, one can assume that not all genes are regulated by all others, but only by a few. We will show that suitable sparsity regularisation can actually improve the performance of network estimation algorithms dramatically, and we also show how to perform efficient experimental design in this setting.

Note that sometimes the interaction of vast number of different effects may in the end also lead to a simple model, think for example of diffusion models or the central limit theorem. However, such simple behaviour of a complex system typically requires strong additional symmetry principles, for example the i.i.d. assumption in the central limit theorem case.

1.4.3 Regularisation and Independence of Representation

Finally, we will examine non-parametric regression between two Riemannian manifolds in Chapter 4. One key characteristic of the manifold setting is that each manifold has *several different but equivalent representations*. For example, the sphere can be seen as a subset of $\mathbb{R}^3$, or also as a collection of spherical coordinate charts which fulfil certain overlap conditions.

One straightforward way to perform learning aiming at simple regression functions is to define simplicity with respect to a specifically chosen representation. For example, we could fit a set of data points on the sphere with straight lines in spherical coordinates, that is, straight lines in a two-dimensional "world map". However, when we map the lines back onto the true "globe" in $\mathbb{R}^3$, the lines would not be straight anymore, and describing them in 3D coordinate terms would be considerably more complicated.

Thus, if we aim at *average case simplicity*, then we should use regression mappings which have some characterisation that is independent of the features of a single specific representation. We therefore propose a regularisation framework for non-parametric regression between Riemannian manifolds, which is independent of the representation of the input and/or output manifold in terms of parametrisation or embedding, but which only depends on *intrinsic geometric properties*.

1.5 Conclusion

Induction is the core of machine learning and statistics, and furthermore also of science as a whole. For induction to be meaningful we have to use non-trivial prior assumptions, which can be incorporated into learning algorithms via regularisation. A well-accepted, though not provably correct, source of suitable regularisation criteria is Ockham's razor.

This thesis examines a number of different aspects of Ockham's simplicity principle. We describe several ways how simplicity can be formalised, how it can be included into statistical learning models via different regularisation schemes, and how we can efficiently work with the resulting models. Each regularisation setting is described in conjunction with one or more practical application examples, underlining its validity for a certain class of real-world problems.

1.6 Publication Record

Many parts of this thesis have been published before at conferences or in journals. The material of Chapter 2 was presented in [Steinke and Schölkopf, 2006, 2008], Chapter 3 in [Steinke et al., 2007b; Seeger et al., 2007], and Chapter 4 in [Steinke et al., 2008; Steinke and Hein, 2009; Steinke et al., 2009].

Other work (co-)authored during the work on this thesis that does not thematically fit this exposition is omitted here. Specifically, we do not present the work on 3D surface registration [Steinke et al., 2007a], psycho-physics [Cooke et al., 2005], or MR-based attenuation correction [Hofmann et al., 2008].

Chapter 2

Linking Kernels and Differential Equations

Many common machine learning methods such as Support Vector Machines or Gaussian process inference make use of positive definite kernels, reproducing kernel Hilbert spaces, Gaussian processes, and regularisation operators. In this chapter, we present these objects in a general, unifying framework, and interrelations are highlighted.
With this in mind we then show how linear stochastic differential equation models can be incorporated naturally into the kernel framework. And vice versa, many kernel machines can be interpreted in terms of differential equations. We focus especially on ordinary differential equations, also known as dynamical systems, and it is shown that standard kernel inference algorithms are equivalent to Kalman filter methods based on such models.
In order not to cloud qualitative insights with heavy mathematical machinery, we restrict ourselves to finite domains, implying that differential equations are treated via their corresponding finite difference equations.

2.1 Introduction

As depicted in Figure 2.1, Support Vector Machines can be thought of as follows [Schölkopf and Smola, 2002]. They first map the training and test input data into a potentially infinite dimensional feature space, a *reproducing kernel Hilbert space* (RKHS), and then classify the data with the help of a separating hyperplane. Since there are often many hyperplanes that separate the training data points, SVMs select the hyperplane with the largest *margin*, that is, the largest distance between the hyperplane and the data points. However, what is the intuitive meaning of distance in this feature space? One way to understand such distances is to explicitly choose a specific feature function Φ of which all components have some problem-dependent meaning. However, often the RKHS and its corresponding norm are only defined implicitly via the choice of a *kernel function* $k(x, y) = \Phi(x)^T \Phi(y)$. In this case, the interpretation is not as straightforward. It was noted by [Smola et al., 1998] that any kernel function is related to a specific *regularisation operator*. The present chapter explains this connection in a simple but very general form, and we show how it can help to better understand SVMs and other related kernel machines.

Furthermore, it turns out that for the commonly used Gaussian (RBF) kernel, the feature space is a subset of the space of all functions from the input domain to the real numbers, and

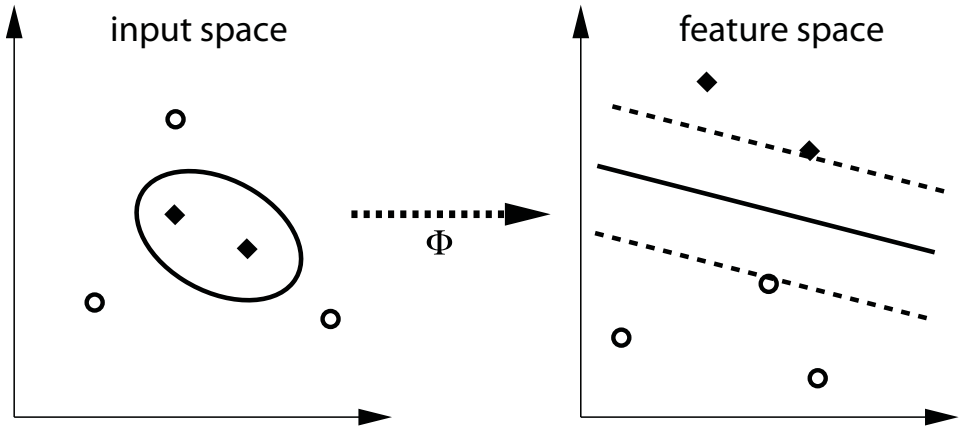


Figure 2.1: *Support Vector Machines map input data points via Φ into a potentially infinite dimensional feature space (Figure taken from [Schölkopf and Smola, 2002]). The classification then proceeds by finding the separating hyperplane with the largest margin between the classes. However, what is the meaning of distance in this feature space? Especially if the feature space is only defined implicitly via a kernel function $k(x, y) = \Phi(x)^T \Phi(y)$?*

the corresponding regularisation operator is an infinite sum of derivative operators [Girosi et al., 1993]. We generalise this result and show that all translation-invariant kernel functions are related to differential operators. The corresponding homogeneous differential equations are a useful tool for understanding the meaning of specific kernel functions. However, we could also exploit this relation in the inverse direction and construct kernels that are specifically adapted to problems involving differential equation models. To make this point clearer, let us consider a simple regression example from physics, which can be visualised easily and which we will thus use throughout the chapter. Assume that we have acquired measurements of a pendulum's position at given time instances, as depicted in Figure 2.2. We are then interested in two problems:

Firstly, we will discuss how to optimally reconstruct the full time course of the pendulum's position. The pendulum's dynamics can be described approximately by a simple linear differential equation, and estimating the full state trajectory from few measurements is equivalent to classical state estimation in linear dynamical systems. For this task one typically employs a variant of the Kalman filter. On the other hand, the problem of reconstructing a function from a finite number of measurements is also the goal of non-parametric regression techniques, such as the kernel-based methods Support Vector Machines / Support Vector Regression (SVR) or Gaussian process (GP) inference. In this chapter, we will show how the knowledge of a model differential equation can be included into kernel methods, and that these are closely related to Kalman filter-based approaches.

Secondly, we will explore how to learn about properties of the pendulum from the given measurements. In particular this will aim at determining parameters of the differential equation that characteristically describes the pendulum, a task that is commonly known as linear system identification. We will show how model selection methods for kernel methods such as cross-validation or marginal likelihood optimisation can be used for system identification purposes. As for state estimation, these machine learning-inspired approaches turn out to be equivalent to well-known system identification methods, such as prediction error methods.

Having these objectives in mind, we will first describe kernel methods in a relatively broad

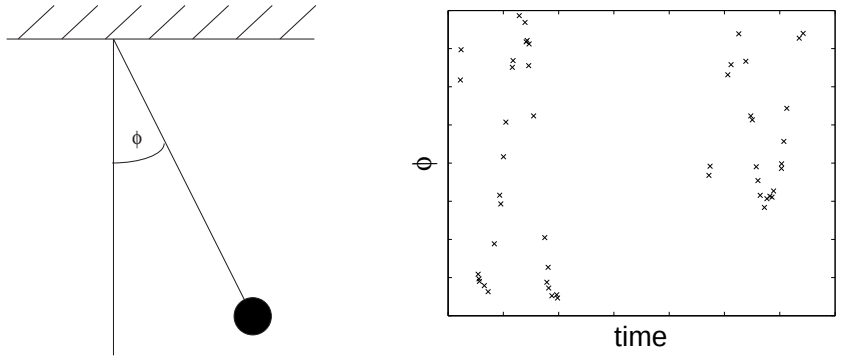


Figure 2.2: *(left) Schematic view of a pendulum, and (right)* 50 *noisy measurements of the pendulum's angle* $\phi(t_i)$ *at times* t_i*,* $i = 1, .., 50$*.*

way that is not specifically tailored towards differential equations. However, the developed framework will then allow us to straightforwardly understand the close links between linear differential equations and kernel methods as a special case. We mostly focus on ordinary linear differential equations, also known as dynamical systems, but will also give examples of linear partial differential equations. Other linear operator equations could also be dealt with similarly. By differential equations we will in this chapter always mean stochastic differential equations, since these can be nicely incorporated into kernel methods. Stochastic differential equations are a superset of normal differential equations, since any differential equation can be converted into a stochastic differential equation by adding a noise term with variance zero.

2.1.1 Finite Domains

The current chapter is formulated in terms of finite domains. Functions to be estimated are assumed to map finite domains to $\mathbb{R}$ or $\mathbb{R}^n$. In the pendulum example imagine time to be discretised into many small time steps. The use of finite domains thus means that whenever we speak of differential equations in this chapter we actually mean discretised versions thereof, that is, the corresponding finite difference equations.

In the authors' opinion, finite domains are just the right level of simplification needed for an easy, yet very far-reaching exposition of the matter. The restriction to finite domains simplifies the required mathematics dramatically. Functions on finite domains are finite dimensional vectors, requiring only simple linear algebra for analysis instead of more involved functional analysis. Existence and convergence of sums/integrals is trivial for finite domains, and point evaluations are described by inner products with unit vectors instead of functionals involving Dirac-delta distributions. Finite domains also allow one to define Gaussian densities for function-valued random variables. This is not possible for infinite dimensional functions, at least not with respect to the standard Lebesgue measure, which does not exist for infinite dimensional function spaces [Bogachev, 1998].

Despite these important simplifications, little qualitative expression power is lost. Most well-known results on kernels can be easily derived and motivated for finite domains. Reasonably smoothly varying functions can be approximated well by their finite dimensional piecewise-linear counterparts, which, in most cases, allow differential equations to be converted straightforwardly into qualitatively equivalent finite difference equations. Finally,

there are also some common settings for machine learning that naturally deal with finite domains, for example graph-based or transductive learning.

There are, of course, also certain shortcomings of a finite domain approach. Generally speaking, we cannot answer questions regarding the limiting behaviour for ever smaller discretisation steps. Note that while such limiting processes on continuous domains typically exist, see e.g. [Oksendal, 2002] for one-dimensional domains, they often have some additional surprising properties, some of which are at first sight in conflict with our understanding of the corresponding model for finite domains. For example, the sample paths of Brownian motion are continuous, yet nowhere differentiable [Oksendal, 2002]. This implies that the corresponding RKHS norm, defined below, is infinite for each sample path almost surely. While the RKHS is thus a null space under the measure of the continuous time process, the mean of non-parametric regression with a finite number of data points is nevertheless guaranteed to be an element of the RKHS, a very surprising fact. Also, if we define our models via discrete regularisation operators or inverse covariances as defined below and then take the limit of step size to zero, then the marginal distributions of these continuous processes are often not identical to the finite distributions. For example, for the linear difference equation $\boldsymbol{x}_i = (\mathbf{1} + \boldsymbol{A}\Delta t)\boldsymbol{x}_{i-1}$ the exact discretisation of the continuous analogue would be $\boldsymbol{x}_i = \exp(\boldsymbol{A}\Delta t)\,\boldsymbol{x}_{i-1}$. While these expressions are similar for small step sizes Δt they are not identical. This fact is sometimes important for computational reasons, since by construction the inverse covariance matrices of the discrete models often have some specific sparsity structure which is not, in general, preserved for the marginals.

The aim of this chapter is to offer a simple intuitive introduction to the kernel framework and to show its connections to differential equations. We thus concentrate solely on finite domains. Note that this means that when speaking of processes in this chapter, we just mean distributions over functions on a given *fixed* finite domain. We do not make statements about what happens if one or more points are added to the domain of the model, and the defined processes are not assumed to be marginals of their continuous analogues.

2.1.2 Overview

The remainder of the chapter is structured as follows: after introducing some notation in Section 2.2, we define in Section 2.3 a framework of basic objects used in kernel methods, and we explain how these objects are interrelated. Thereafter, we describe the use of these objects for SVR in Section 2.3.2, for GP regression in Section 2.3.3, and for vector-valued regression in Section 2.3.4. In Section 2.4, we discuss a typical kernel-machine regression model and show its relation to linear stochastic differential equations. We demonstrate how to develop kernel functions from linear state-space models or higher-order differential equations. We show that the resulting inference methods are equivalent to Kalman filter-based methods. The pendulum and other examples are presented in detail in Section 2.5. In Section 2.6 we discuss the practical implications of the link between kernel machines and linear stochastic differential equations. We summarise our conclusions of this chapter in Section 2.7.

For better readability, we have restricted the main part of the chapter to real-valued kernels, and postpone the more natural, slightly more technical treatment involving complex numbers to Additional Material 2.8.1. It will appear throughout the text that, with regularisation theory in mind, conditionally positive definite (cpd) kernels arise quite naturally. We have

transferred all parts dealing with cpd kernels to Additional Material 2.8.2, where we present an extension of the kernel framework to cpd kernels.

2.1.3 Related Work

Most of the mathematical results of this chapter are not the authors' original work, but have been mentioned in different contexts before. Our contribution is to reformulate them in a unified, easily understandable framework, the simple language of finite domains. Furthermore, we reinterpret them to highlight parallels between kernel methods and linear differential equations.

There is a large body of literature on kernels and differential equations in many different communities, and we only cite some relevant books containing overviews of their respective fields as well as further references. Many machine learning-related facts about kernels and regularisation methods are taken from [Schölkopf and Smola, 2002], as well as [Rasmussen and Williams, 2006] for the Bayesian interpretation. Sources in the statistics literature include [Wahba, 1990; Ramsay and Silverman, 2005], and in approximation theory [Wendland, 2005]. For an overview of linear stochastic dynamical systems and their estimation we refer to [Ljung, 1999; Oksendal, 2002].

The connection between stochastic processes and splines was first explored in [Kimeldorf and Wahba, 1970]. It is also well-known that thin-plate/cubic splines minimise the second derivative [Madych and Nelson, 1990; Wendland, 2005]. Connections between regularisation operators and kernel functions are explained in [Girosi et al., 1993; Smola et al., 1998], and general linear operator equations are solved with GPs in [Graepel, 2003]. A unifying survey of the theory of kernels, reproducing kernel Hilbert spaces, and GPs has been undertaken by [Hein and Bousquet, 2004]. However, they do not use finite domains, which complicates their study and they do not mention the link with differential or operator equations. Approaches that directly employ kernel methods towards the estimation of stochastic differential equation models are proposed in [Heckman and Ramsay, 2000] and [Steinke and Schölkopf, 2006].

2.2 Notation

We consider functions $f : \mathcal{X} \to \mathbb{R}$, where the domain $\mathcal{X}$ is a finite set, $|\mathcal{X}| = N$. When considering dynamical systems we will typically set $\mathcal{X}$ to be an evenly discretised interval and assume N to be large. Other examples of finite domains are discretised regions of higher dimensional spaces, but also finite sets of graphs, texts, or any other type of objects.

We denote by $\mathcal{H}$ the space of all functions $f : \mathcal{X} \to \mathbb{R}$. f is fully described by the $\mathbb{R}^N$-vector $\boldsymbol{f} = (f(x_1), ..., f(x_N))^T$. Vectors and matrices are denoted in bold font, but if an element of $\mathcal{H}$ is thought of as a function from $\mathcal{X}$ to $\mathbb{R}$, we use the corresponding normal font character. For points $x_i \in \mathcal{X}$ we define *location vectors/functions* by $\boldsymbol{\delta}_{x_i} = (\delta_{ij})_{j=1,..,N}$, where δ_{ij} is the Kronecker symbol. The inner product of these with a function $\boldsymbol{f} \in \mathcal{H}$ yields $\boldsymbol{\delta}_{x_i}{}^T \boldsymbol{f} = f(x_i)$. Thus, location vectors correspond to Dirac delta functions centred at the point x_i for continuous, infinite domains.

Linear operators $\boldsymbol{G} : \mathcal{H} \to \mathcal{H}$ are isomorphic to matrices in $\mathbb{R}^{N \times N}$. Therefore, any function $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ uniquely determines a linear operator $\boldsymbol{G} : \mathcal{H} \to \mathcal{H}$ through $\boldsymbol{G}_{ij} =$

$\boldsymbol{\delta}_{x_i}{}^T \boldsymbol{G} \boldsymbol{\delta}_{x_j} = g(x_i, x_j)$ and vice versa. The columns of $\boldsymbol{G}$ will be noted by $\boldsymbol{G}_{x_i} = \boldsymbol{G}\boldsymbol{\delta}_{x_i}$; they are real-valued functions on $\mathcal{X}$. For a set $X = \{x_i \mid i = 1, .., m\} \subseteq \mathcal{X}$ of points, $\boldsymbol{G}_X$ will denote the $m \times m$ sub-matrix of $\boldsymbol{G}$ corresponding to X.

2.3 The Kernel Framework

In non-parametric regression, we are given observations $(x_i, y_i) \in \mathcal{X} \times \mathbb{R}$, $i = 1, .., m$, $m \leq N$, and the goal is to predict the value y_* for arbitrary test points $x_* \in \mathcal{X}$. SVR estimates a prediction function $f : \mathcal{X} \to \mathbb{R}$, $y_* = f(x_*)$, as the minimiser of a functional like

$$\min_{\boldsymbol{f} \in \mathcal{H}} \|\boldsymbol{R}\boldsymbol{f}\|^2 + C \text{ Loss } (\{(x_i, y_i, f(x_i)) | i = 1, \ldots, m\}) . \tag{2.1}$$

On the one hand, f should be close to the observed data as measured through a *loss function* Loss : $(\mathcal{X} \times \mathbb{R} \times \mathbb{R})^m \to \mathbb{R}$. On the other hand, f should be *regular* as measured by the *regularisation operator* $\boldsymbol{R} : \mathcal{H} \to \mathcal{G}$, where $\mathcal{G}$ is any finite dimensional Hilbert space. These two objectives are relatively weighted through the *regularisation parameter* C.

Note that SVMs also use the same setting for binary classification. The classes are represented as $y = \pm 1$. First a real-valued function $f : \mathcal{X} \to \mathbb{R}$ is estimated and then thresholded to obtain the binary class predictions. Unlike radial basis function networks [Girosi et al., 1993, 1995], SVMs use the hinge loss $|yf(x) - 1|_+$ where $|x|_+ = x$ if $x > 0$ and $|x|_+ = 0$ otherwise.

Many questions arise around objective (2.1). How are $\|\boldsymbol{R}\boldsymbol{f}\|^2$ and the commonly used function space norm $\|\boldsymbol{f}\|_K^2$ related? This will lead to the notion of *reproducing kernel Hilbert spaces* (RKHS). The N-dimensional problem (2.1) can be solved using a smaller m-dimensional equivalent involving *kernel functions*. But how does $\boldsymbol{R}$ relate to the chosen kernel function? Can one interpret (2.1) in a Bayesian way? For example, with the help of *Gaussian processes*? The current section will answer the above questions in a simple, yet precise way for finite domains. We will furthermore show the interrelations between the terms mentioned above.

Throughout the main part of this chapter we assume that $\boldsymbol{R}$ is a one-to-one operator. This will lead to a framework with positive definite kernels. If $\boldsymbol{R}$ is not one-to-one, conditionally positive definite (cpd) kernels arise. All definitions and theorems derived for the positive definite case in the current section are extended to the cpd case in Additional Material 2.8.2.

2.3.1 Regularisation Operators, Kernels, RKHS, and Gaussian Processes

Figure 2.3 depicts the most common objects in the kernel framework. We will explain them below, starting with the covariance operator. The covariance operator is not commonly used in the kernel literature, but we introduce it as a useful abstraction in the centre of the framework. While it does not in itself have a special meaning, it helps us to unify the links between the other "leaf" objects. With the covariance operator in mind, the reader may then easily derive additional direct links.

Definition 2.1 (Covariance operator)**.** *A* covariance operator $\boldsymbol{K}$ *is a positive definite matrix of size* $N \times N$*, i.e. for all* $\boldsymbol{f} \in \mathcal{H}$*,* $\boldsymbol{f} \neq \boldsymbol{0}$*, it is* $\boldsymbol{f}^T \boldsymbol{K} \boldsymbol{f} > 0$.

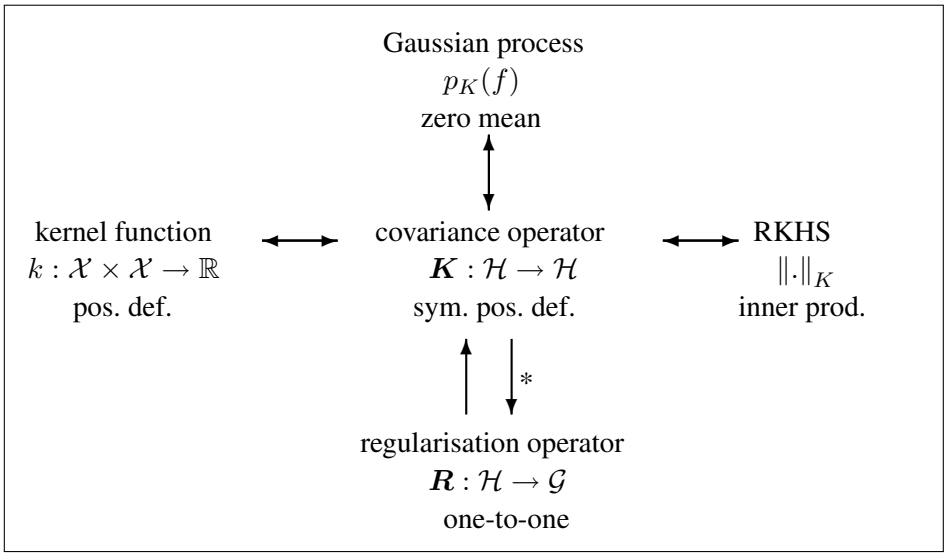


Figure 2.3: *Common objects in the kernel framework and their interrelations. Arrows denote that one can uniquely be determined from the other (the * denotes that this connection is not unique).*

A first interpretation of the covariance operator which gives $\boldsymbol{K}$ its name is given through its use in GPs.

Definition 2.2 (Gaussian processes (GP))**.** *A* Gaussian process *is a distribution over all functions $f : \mathcal{X} \to \mathbb{R}$ such that for any linear functional $w : \mathcal{H} \to \mathbb{R}$ the value $w(\boldsymbol{f}) = \boldsymbol{w}^T \boldsymbol{f}$ is a real-valued, normally distributed random variable.*

This definition taken from [Bogachev, 1998] is tailored to the case where f is infinite dimensional, and no Lebesgue density exists in $\mathcal{H}$. For finite $\mathcal{X}$, it simply implies that the distribution has a density $p_K(f)$ over the functions in $\mathcal{H}$, and that this density is a multivariate Gaussian. Note that this means that in the finite dimensional setting, distributions over functions can be described via standard multivariate Gaussian distributions. Given a covariance operator $\boldsymbol{K}$ we can define a special zero mean GP by

$$p_K(f) = N(0, \boldsymbol{K}) \propto \exp\left(-\frac{1}{2}\boldsymbol{f}^T \boldsymbol{K}^{-1} \boldsymbol{f}^2\right). \tag{2.2}$$

Conversely, given a GP, its covariance matrix is a valid positive definite covariance operator.

The covariance operator also allows one to define another well-known object.

Definition 2.3 (Kernel function)**.** *A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a* positive definite kernel function*, if for all subsets $X \subseteq \mathcal{X}$, $X = \{x_1, .., x_m\}$, $m \leq N$, and all $0 \neq \boldsymbol{\alpha} \in \mathbb{R}^m$, it holds that*

$$\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) = \boldsymbol{\alpha}^T \boldsymbol{K}_X \boldsymbol{\alpha} = \left(\sum_{i=1}^{m} \alpha_i \boldsymbol{\delta}_{x_i}\right)^T \boldsymbol{K} \left(\sum_{j=1}^{m} \alpha_j \boldsymbol{\delta}_{x_j}\right) > 0.$$

By definition, kernel functions give rise to a positive definite covariance operator $\boldsymbol{K}_{\mathcal{X}}$. Conversely, a covariance operator $\boldsymbol{K}$ defines a kernel function through $k(x_i, x_j) = \boldsymbol{K}_{ij} = {\boldsymbol{\delta}_{x_i}}^T \boldsymbol{K} \boldsymbol{\delta}_{x_j}$, since positive definiteness of $\boldsymbol{K}$ implies that $\boldsymbol{K}_X$, too, is positive definite for all $X \subseteq \mathcal{X}$.

Kernel functions naturally lead to the definition of specially adapted function spaces.

Definition 2.4 (Reproducing kernel Hilbert space (RKHS))**.** *A Hilbert space $(\mathcal{S}, (.,.)_S)$, $\mathcal{S} \subseteq \mathcal{H}$, of functions $f : \mathcal{X} \to \mathbb{R}$ is called a* reproducing kernel Hilbert space*, if the evaluation functionals $\delta_{x_i} : \mathcal{H} \to \mathbb{R}$ defined by $\delta_{x_i}(f) = \boldsymbol{\delta}_{x_i}^T \boldsymbol{f} = f(x_i)$ are continuous for all $x_i \in \mathcal{X}$, i.e., $|\delta_{x_i}(f)| \leq C \|\boldsymbol{f}\|_S$ for all $\boldsymbol{f} \in \mathcal{S}$.*

As for the definition of GPs, this formulation of the definition of RKHSs is tailored towards the continuous domain case. The definition ensures that point evaluations of functions in $\mathcal{S}$ are well-defined, which is not obvious for functions on continuous domains, for example, L_2 functions. Well-defined point evaluations are, of course, necessary for machine learning methods that deal with point-wise data measurements. In the finite domain setting, the definition of RKHSs is quite trivial. It implies that $\mathcal{H}$ with any inner product $(.,.)_S$ is an RKHS, also with the usual L_2 inner product. The proof is found in Additional Material 2.8.3, together with the proof of the following lemma which summarises some useful results about RKHSs.

Lemma 2.5. *The following statements hold for RKHS $(\mathcal{H}, (.,.)_S)$:*

1. *There exists a unique element $\boldsymbol{S}_{x_i} \in \mathcal{H}$ for each $x_i \in \mathcal{X}$, the* representer*, such that*

   $$\delta_{x_i}(f) = f(x_i) = (\boldsymbol{S}_{x_i}, \boldsymbol{f})_S$$

   for all $\boldsymbol{f} \in \mathcal{H}$. This property is called the reproducing property*.*

2. *The function $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined by $s(x_i, x_j) = (\boldsymbol{S}_{x_i}, \boldsymbol{S}_{x_j})_S$ is a positive definite kernel function in the sense of Definition 2.3.*

   Let the operator $\boldsymbol{S} : \mathcal{H} \to \mathcal{H}$ be defined by $\boldsymbol{S}_{ij} = s(x_i, x_j)$.

3. *Any inner product $(\boldsymbol{f}, \boldsymbol{g})_S$ can be uniquely expressed in the form $\boldsymbol{f}^T \boldsymbol{T} \boldsymbol{g}$ where $\boldsymbol{T}$ is a positive definite operator.*

4. *$s(x_i, x_j) = \boldsymbol{T}^{-1}_{ij}$ or equivalently $\boldsymbol{S} = \boldsymbol{T}^{-1}$.*

5. *The kernel s defines the inner product $(.,.)_S$ uniquely.*

The above lemma implies that for a given covariance operator $\boldsymbol{K}$ one can define an RKHS $(\mathcal{H}, (.,.)_K)$ by setting

$$(\boldsymbol{f}, \boldsymbol{g})_K \equiv \boldsymbol{f}^T \boldsymbol{K}^{-1} \boldsymbol{g}.$$

Then the representer of this RKHS is identical with the kernel function $\boldsymbol{K}\boldsymbol{\delta}_{x_i}$ derived from $\boldsymbol{K}$ via $k(x_i, x_j) = \boldsymbol{K}_{ij}$. Since the relation between kernel and inner product is unique, one could also construct a unique valid covariance operator from a given RKHS.

The definitions so far have been purely technical, but we can give them a practical meaning when considering them in conjunction with a regularisation operator as used in the SVR objective (2.1).

Definition 2.6 (Regularisation operator)**.** *A* regularisation operator $\boldsymbol{R} : \mathcal{H} \to \mathcal{G}$ *is a one-to-one linear operator. Here,* $\mathcal{G}$ *is any finite dimensional Hilbert space.*

If we use $\boldsymbol{K} = (\boldsymbol{R}^T\boldsymbol{R})^{-1}$, then by Lemma 2.5 it is

$$\|\boldsymbol{f}\|_K^2 = \boldsymbol{f}^T\boldsymbol{K}^{-1}\boldsymbol{f} = \boldsymbol{f}^T\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{f} = \|\boldsymbol{R}\boldsymbol{f}\|^2 .$$

That means that if $\|\boldsymbol{R}\boldsymbol{f}\|$ measures the *regularity* of $f : \mathcal{X} \to \mathbb{R}$, then the RKHS norm exactly equals the regularity measure. In the SVR objective (2.1) regular functions are thus preferred over less regular ones. Furthermore, the related GP is

$$p_K(f) = N(0, \boldsymbol{K}) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{R}\boldsymbol{f}\|^2\right),$$

implying that under this distribution regular functions are more likely than less regular ones. The most likely functions are those which exactly fulfil the *regularity/model equation*

$$\boldsymbol{R}\boldsymbol{f} = 0.$$

Note that since $\boldsymbol{R}$ is assumed to be one-to-one, only the zero function can fulfil the model equation exactly. Non-vanishing functions violate this equation by an amount that is determined by the structure of $\boldsymbol{R}$. If non-trivial functions are to be considered fully regular, that is, $\|\boldsymbol{R}\boldsymbol{f}\| = 0$, then $\boldsymbol{R}$ cannot be one-to-one. This case is discussed in Additional Material 2.8.2.

Given a covariance operator $\boldsymbol{K}$, we can compute an associated regularisation operator $\boldsymbol{R}$ as $\boldsymbol{R} = \sqrt{\boldsymbol{K}^{-1}}$. However, note that if we transform $\boldsymbol{R} \to \boldsymbol{K} \to \boldsymbol{R}$ in this way we will not necessarily recover the same regularisation operator we started from. The original $\boldsymbol{R}$ does not have to be quadratic and even if it is, taking the root would set all originally negative eigenvalues of $\boldsymbol{R}$ to positive.

The objects of the kernel framework and their interrelations are summarised in Table 2.1.

2.3.2 Support Vector Machines

With the above definitions the SVR objective (2.1) can be rewritten as

$$\min_{\boldsymbol{f}\in\mathcal{H}} \|\boldsymbol{f}\|_K^2 + C\,\text{Loss}\left(\{(x_i, y_i, f(x_i))|i = 1, \ldots, m\}\right). \tag{2.3}$$

This optimisation problem over the whole function space $\mathcal{H}$, i.e. over N variables where N is potentially large, can be reduced to a typically much smaller m-dimensional optimisation problem using kernel functions. To see this, we will derive the famous representer theorem in two steps. The proofs are found in Additional Material 2.8.3.

The first step, which is interesting in itself, shows a general property of RKHSs: Any function in an RKHS can be decomposed into a set of kernel functions and its $\mathcal{H}$-orthogonal complement. If the complement is understood as a function from $\mathcal{X}$ to $\mathbb{R}$, then it has function value zero at all kernel centres.

Lemma 2.7. *Given distinct points* $X = \{x_i \mid i = 1, .., m\}$, $m \leq N$, *any* $\boldsymbol{f} \in \mathcal{H}$ *can be uniquely written as* $\boldsymbol{f} = \sum_{i=1}^m \alpha_i \boldsymbol{K}_{x_i} + \boldsymbol{\rho}$, $\boldsymbol{\alpha} \in \mathbb{R}^m$, $\boldsymbol{\rho} \in \mathcal{H}$, *where* $\boldsymbol{\rho}$ *satisfies the conditions* $\rho(x_i) = (\boldsymbol{K}_{x_i}, \boldsymbol{\rho})_K = 0, \quad i = 1, .., m.$

entity	symbol	relations
kernel function	$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ $\boldsymbol{K}_{x_i} : \mathcal{X} \to \mathbb{R}$	$k(x_i, x_j) = \boldsymbol{K}_{i,j} = {\boldsymbol{\delta}_{x_i}}^T \boldsymbol{K}_{x_j}$ $k(x_i, x_j) = (\boldsymbol{K}_{x_i}, \boldsymbol{K}_{x_j})_K$ $k(x_i, x_j) = {\boldsymbol{\delta}_{x_i}}^T (\boldsymbol{R}^T \boldsymbol{R})^{-1} \boldsymbol{\delta}_{x_j}$ $k(x_i, x_j) = \mathrm{Cov}_{\boldsymbol{f} \sim p_K}(f(x_i), f(x_j))$
covariance op.	$\boldsymbol{K} : \mathcal{H} \to \mathcal{H}$	$\boldsymbol{K}_{i,j} = k(x_i, x_j)$, $\boldsymbol{K} = (\boldsymbol{R}^T \boldsymbol{R})^{-1} = \mathrm{Cov}_{\boldsymbol{f} \sim p_K}(\boldsymbol{f}, \boldsymbol{f})$
RKHS	$(.,.)_K : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ $\|.\|_K . : \mathcal{H} \to \mathbb{R}$	$(\boldsymbol{f}, \boldsymbol{g})_K = \boldsymbol{f}^T \boldsymbol{K}^{-1} \boldsymbol{g} = \boldsymbol{f}^T \boldsymbol{R}^T \boldsymbol{R} \boldsymbol{g}$ $\|\boldsymbol{f}\|_K = (\boldsymbol{f}, \boldsymbol{f})_K^{1/2} = \|\boldsymbol{R}\boldsymbol{f}\|$
Gaussian process	$p_K : \mathcal{H} \to \mathbb{R}$	$p_K(f) = N(0, \boldsymbol{K})$ $p_K(f) \propto \exp\left(-\frac{1}{2} \|\boldsymbol{f}\|_K^2\right)$ $p_K(f) \propto \exp\left(-\frac{1}{2} \|\boldsymbol{R}\boldsymbol{f}\|^2\right)$
regularisation op.	$\boldsymbol{R} : \mathcal{H} \to \mathcal{G}$	($\boldsymbol{R} = \sqrt{\boldsymbol{K}^{-1}}$, not unique)

Table 2.1: *Summary of the objects of the positive definite kernel framework and their interrelations.* $\mathrm{Cov}_{\boldsymbol{x} \sim p(\boldsymbol{x})}(x_i, x_j)$ *denotes the covariance between* x_i *and* x_j *under a distribution of* $\boldsymbol{x}$ *with density* $p(\boldsymbol{x})$*. If the arguments are vectors, the corresponding covariance matrix is meant.*

The second step then is as follows.

Theorem 2.8 (Representer theorem)**.** *Given* $m \leq N$ *distinct points* $X = \{x_i \mid i = 1, .., m\}$ *and labels* $\{y_i \mid i = 1, .., m\} \subseteq \mathbb{R}$ *the minimiser* $\boldsymbol{f}$ *of (2.3) has the form* $\boldsymbol{f}_{\boldsymbol{\alpha}} = \sum_{i=1}^m \alpha_i \boldsymbol{K}_{x_i}$*,* $\boldsymbol{\alpha} \in \mathbb{R}^m$*, where* $\boldsymbol{\alpha}$ *minimises*

$$\boldsymbol{\alpha}^T \boldsymbol{K}_X \boldsymbol{\alpha} + C \, Loss\left(\{(x_i, y_i, f_{\boldsymbol{\alpha}}(x_i)) | i = 1, \ldots, m\}\right). \tag{2.4}$$

If the loss is convex, $\boldsymbol{\alpha}$ *is determined uniquely.*

Remark: f can also be expanded in another function system, say $\boldsymbol{f} = \sum_{j=1}^L c_j \boldsymbol{\phi}_j$. Then $\min_{\boldsymbol{c} \in \mathbb{R}^L} \; \boldsymbol{c}^T \boldsymbol{M} \boldsymbol{c} + C \, \mathrm{Loss}\left(\{(x_i, y_i, f_{\boldsymbol{c}}(x_i)) | i = 1, \ldots, m\}\right)$ with $\boldsymbol{M}_{ij} = {\boldsymbol{\phi}_i}^T \boldsymbol{R}^T \boldsymbol{R} \boldsymbol{\phi}_j$ is the optimisation problem corresponding to (2.1), see e.g. [Ramsay and Silverman, 2005; Walder et al., 2006]. This is also a convex problem and can sometimes be solved very efficiently if, for example, compactly supported basis functions are used [Walder et al., 2006]. However, one only finds the optimal solution within the span of the selected basis functions. A globally optimal solution in $\mathcal{H}$ would, in general, require $L = N$ basis functions. Furthermore, $\boldsymbol{M}_{ij} = {\boldsymbol{\phi}_i}^T \boldsymbol{R}^T \boldsymbol{R} \boldsymbol{\phi}_j$ has to be computed for all i, j which could be challenging.

2.3.3 Gaussian Process Inference

The SVR objective (2.1) can also be interpreted from a Bayesian perspective. Assume a two step-model where firstly a latent function $f : \mathcal{X} \to \mathbb{R}$ is drawn from the *GP prior* $p_K(f)$ with covariance operator $\boldsymbol{K}$, and where subsequently the measurements are determined from this function as described by a *local likelihood* $p(\boldsymbol{y}|f) = p(\boldsymbol{y}|\boldsymbol{f}_X)$, where $\boldsymbol{y} = (y_1, \ldots, y_m)^T$ and $X = \{x_1, \ldots, x_m\}$. A common example of a local likelihood is the i.i.d. likelihood, that is, $p(\boldsymbol{y}|f) = \prod_i p(y_i|f(x_i))$. The posterior for local likelihoods is

$$p(f|\boldsymbol{y}, X) \quad \propto \quad p(\boldsymbol{y}|f)p_K(f) \propto p(\boldsymbol{y}|\boldsymbol{f}_X) \exp\left(-\frac{1}{2}\|\boldsymbol{R}\boldsymbol{f}\|^2\right),$$

and the maximum a posteriori (MAP) estimate is

$$\operatorname*{argmax}_{\boldsymbol{f}\in\mathcal{H}} p(f|\boldsymbol{y}, X) \quad = \quad \operatorname*{argmin}_{\boldsymbol{f}\in\mathcal{H}} \frac{1}{2}\|\boldsymbol{R}\boldsymbol{f}\|^2 - \log p(\boldsymbol{y}|\boldsymbol{f}_X).$$

So if one can identify $-\log p(\boldsymbol{y}|\boldsymbol{f}_X)$ with Loss $(\{(x_i, y_i, f(x_i))|i = 1, \ldots, m\})$, which is possible, for example, for the common squared loss, then SVR is just a MAP estimate of a GP model. Note, however, that in some well-known cases such as, for example, the hinge loss, this identification is in a strict sense not possible. The resulting likelihood would not be normalisable with respect to $\boldsymbol{y}$. Nevertheless, if one is willing to work with unnormalised models, the equivalence holds in general. The qualitative meaning of the prior is the same in any case.

Bayesian statistics is typically not only interested in the maximum a posteriori estimate of $f(x_*)$ but in the full predictive distribution,

$$p(f(x_*)|\boldsymbol{y}, X) \quad \propto \quad \int p(\boldsymbol{y}|\boldsymbol{f}_X)\, p_K(f) d\boldsymbol{f}_{\mathcal{X}\setminus x_*}.$$

Here, we have used the notation that for every set $I = \{x_{i_1}, .., x_{i_k}\} \subseteq \mathcal{X}$, $d\boldsymbol{f}_I$ means $df(x_{i_1})...df(x_{i_k})$. Because of the local likelihood we can then split the $N-1$ dimensional integral as follows,

$$p(f(x_*)|\boldsymbol{y}, X) \quad \propto \quad \int p(\boldsymbol{y}|\boldsymbol{f}_X) \underbrace{\left(\int p_K(f) d\boldsymbol{f}_{\mathcal{X}\setminus X\cup x_*}\right)}_{=p_K(\boldsymbol{f}_{X\cup x_*})} d\boldsymbol{f}_X.$$

So if an analytic expression of the marginal $p_K(\boldsymbol{f}_{X\cup x_*})$, which is independent of the data, could be computed, then only an m-dimensional integral would have to be solved for inference. Such an expression is given in the following theorem, which just expresses a standard property of Gaussian distributions. Since it reduces the work from N dimensions to m dimensions similar to the representer theorem 2.8, one could call it the *Bayesian representer theorem*.

Theorem 2.9. *Given $m \leq N$ distinct points $X = \{x_1, ..., x_m\} \subseteq \mathcal{X}$ the GP $p_K(f)$ has the marginals*

$$p_K(\boldsymbol{f}_X) = \frac{1}{\sqrt{(2\pi)^m |\boldsymbol{K}_X|}} \exp\left(-\frac{1}{2}\boldsymbol{f}_X^T \boldsymbol{K}_X^{-1} \boldsymbol{f}_X\right) = N(0, \boldsymbol{K}_X).$$

This property is often used to construct GPs: Given a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ one stores the values corresponding to X into a square matrix $\boldsymbol{K}_X$ and sets $p(\boldsymbol{f}_X) = N(0, \boldsymbol{K}_X)$. Using standard formulas for conditioning Gaussian distributions and block-partitioned matrix inversion one can show that this construction is *consistent*, i.e. for all $X' \subseteq \mathcal{X}, X \cap X' = \emptyset$ it holds that $p(\boldsymbol{f}_X) = \int p(\boldsymbol{f}_{X \cup X'}) d\boldsymbol{f}_{X'}$. By Kolmogorov's extension theorem, or by simply using $X = \mathcal{X}$ in our finite dimensional case, this yields a GP on all of $\mathcal{X}$.

2.3.4 Vector-Valued Regression

Consider now regression from $\mathcal{X}$ to $\mathbb{R}^n$, $n > 1$. We will show that the kernel framework explained above can be easily extended to this case. The function space of all functions $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^n$ will be denoted by $\mathcal{H}^n$. We can represent such a function as a vector $\underline{\boldsymbol{f}}$ in $\mathbb{R}^{nN}$. Denoting the component functions by $f^i : \mathcal{X} \to \mathbb{R}$ it is $\underline{\boldsymbol{f}} \equiv \left(\boldsymbol{f}^{1T} \dots \boldsymbol{f}^{nT}\right)^T$. The standard inner product in $\mathcal{H}^n$ is $\underline{\boldsymbol{f}}^T \underline{\boldsymbol{g}} = \sum_{j=1}^n \boldsymbol{f}^{jT} \boldsymbol{g}^j$. The unit vector $\underline{\boldsymbol{\delta}}^j_{x_i}$, i.e. the location vector for location x_i and the j-th component, then has the j-th component equal to $\boldsymbol{\delta}_{x_i}$ and all others equal to zero. It is $\boldsymbol{\delta}^{j\,T}_{x_i} \underline{\boldsymbol{f}} = f^j(x_i)$. Linear operators $\underline{\boldsymbol{A}} : \mathcal{H}^n \to \mathcal{H}^n$ are isomorphic to $\mathbb{R}^{(Nn)\times(Nn)}$ matrices.

Theorem 2.10. *The function space $\mathcal{H}^n$ is isomorphic to the space $\tilde{\mathcal{H}}$ of all functions from $\tilde{\mathcal{X}} = \mathcal{X} \times \{1, .., n\}$ to $\mathbb{R}$.*

This obvious theorem includes all we need in order to work with vector-valued functions: As $\mathcal{X}$ is a finite set, so is $\tilde{\mathcal{X}}$. All the above theory on kernels, regularisation operators, and GPs applies. For example, using the regularisation operator $\underline{\boldsymbol{R}} : \mathcal{H}^n \to \mathcal{G}$, the corresponding kernel function is

$$k(x_i, x_j)^{lm} = k((x_i, l), (x_j, m)) = \boldsymbol{\delta}^{l\,T}_{x_i} (\underline{\boldsymbol{R}}^T \underline{\boldsymbol{R}})^{-1} \boldsymbol{\delta}^m_{x_j}. \tag{2.5}$$

To construct a sensible regulariser $\underline{\boldsymbol{R}}$, a similarity measure between points in $\tilde{\mathcal{X}}$ is needed. Since in many applications it is not clear how to compare different components of $\underline{\boldsymbol{f}}$, it is common to use a block-diagonal regulariser $\underline{\boldsymbol{R}} = \mathrm{diag}(\boldsymbol{R}^1, .., \boldsymbol{R}^n)$, i.e. regularising each component separately. The corresponding kernel function then has the vector form

$$\underline{\boldsymbol{K}}^j_{x_i} = \left(0, \dots, 0, \boldsymbol{K}^{j\,T}_{x_i}, 0, \dots, 0\right)^T,$$

with the individual kernel functions $\boldsymbol{K}^j_{x_i} = (\boldsymbol{R}^{j,T} \boldsymbol{R}^j)^{-1} \boldsymbol{\delta}_{x_i}$ in the corresponding components. The joint covariance matrix $\underline{\boldsymbol{K}}$ is block-diagonal in this case. If the loss/likelihood term does not imply a dependency between different components, such as, for example, the quadratic loss, then each dimension can be treated separately. However, there are also numerous situations where a joint regularisation makes sense. Examples are shown in the next section.

The theory as described here was mentioned in [Hein and Bousquet, 2004]. [Micchelli and Pontil, 2005] have introduced a slightly different formalism employing operator-valued kernel functions in this context. However, the derived representer theorem is equivalent to the simple approach presented here.

Note that one could also reorder the entries in $\underline{\boldsymbol{f}}$; for example, we could define $\underline{\boldsymbol{f}} = \begin{pmatrix}\boldsymbol{f}(x_1)^T & \dots & \boldsymbol{f}(x_N)^T\end{pmatrix}^T$. While in this section we have used a special notation for vector-valued functions in order to highlight the differences, we will from now on use normal vector notation also for vector-valued functions to keep the notation simple.

2.3.5 Inhomogeneous Regularisation

As shown in the next section, there are numerous cases where one would like to have $\|\boldsymbol{R}\boldsymbol{f} - \boldsymbol{u}\|$, $\boldsymbol{u} \neq 0$, as the regulariser in the SVR objective (2.1) or equivalently use non-zero means for GPs.

Since for $\boldsymbol{f} = 0$, $\|\boldsymbol{R}\boldsymbol{f} - \boldsymbol{u}\| = \|\boldsymbol{u}\| \neq 0$, $\|\boldsymbol{R}\boldsymbol{f} - \boldsymbol{u}\|$ cannot be used as a norm in an RKHS. To circumvent this problem, note that since $\boldsymbol{R}$ is assumed to be one-to-one $\boldsymbol{R}^{-1}\boldsymbol{u}$ exists uniquely and can be computed without regard to the measurement data. We can then base any inference on $\tilde{\boldsymbol{f}} = \boldsymbol{f} - \boldsymbol{R}^{-1}\boldsymbol{u}$, adapting the loss term appropriately. The regularisation term then reads $\left\|\boldsymbol{R}\tilde{\boldsymbol{f}}\right\| = \|\boldsymbol{R}\boldsymbol{f} - \boldsymbol{u}\|$, which represents a true norm for $\tilde{\boldsymbol{f}}$. The kernel framework can now be applied as described above.

2.4 Kernels and Differential Equations

SVR and GP inference both use an *a priori model* that can be expressed in the form

$$\boldsymbol{R}\boldsymbol{f} \approx 0, \tag{2.6}$$

Functions $f : \mathcal{X} \to \mathbb{R}$ which fulfil eq. (2.6) to a high degree as measured by $\|\boldsymbol{R}\boldsymbol{f}\|$, the two-norm of the residual, are preferred to functions that significantly violate the equation.

In this section we discuss a common choice for $\boldsymbol{R}$, namely linear stochastic differential equations (DEs). If the input domain is one-dimensional, one speaks of ordinary differential equations (ODEs) or dynamical systems, and for multivariate input these are partial differential equations (PDEs). Since this chapter is restricted to finite domains, the term differential equation should be understood as meaning finite difference equations throughout. In most cases, the differences are negligible for discretisation steps that are sufficiently small.

Linking differential equations and kernel machines is useful both from a machine learning perspective as well as from a perspective focused primarily on work with differential equations.

From a machine learning point of view, stochastic differential equations can be seen as an ideal prior model. They describe *local* properties of the function f, that is, how the function value at one point relates to function values in the neighbourhood. On a *global* level, stochastic differential equations do not constrain the function very much, because small local noise contributions can add up over longer distances. Thus, this prior is well-suited to situations where we a priori do not know much about the global structure of the target function, but we assume that locally it should not vary too much or only in a certain predefined manner.

From a differential equation point of view, it is useful to have all the machinery of kernel methods at hand. With these, one can estimate the *state/trajectory* of the DE model, that

is, the function described by the differential equation. One can also estimate the DE or its parameters, a task commonly known as *system identification*. Both problems are ubiquitous throughout natural science, statistics and engineering.

2.4.1 Linear State-Space Models

Linear state-space models are the most common models in the class of ODEs, or dynamical systems [Ljung, 1999]. They are classically given as

$$\boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{x}_{i-1} + \boldsymbol{B}\boldsymbol{u}_i + \boldsymbol{\epsilon}_i^{(P)},, \quad i = 1, .., N-1 \tag{2.7}$$

$$\boldsymbol{y}_i = \boldsymbol{C}\boldsymbol{x}_i + \boldsymbol{D}\boldsymbol{u}_i + \boldsymbol{\epsilon}_i^{(M)}, \quad i = 1, .., N-1. \tag{2.8}$$

The *model* equation (2.7) states that the hidden *states* $\boldsymbol{x}_i \in \mathbb{R}^n$ follow a stochastic difference equation with external user-defined control $\boldsymbol{u}_i \in \mathbb{R}^k$ and i.i.d. *process noise* $\boldsymbol{\epsilon}_i^{(P)}$, which is Gaussian-distributed with mean zero and covariance $\boldsymbol{\Sigma}_P$. The *likelihood* of the *measurements* $\boldsymbol{y}_i \in \mathbb{R}^m$ is defined via eq. (2.8). The measurements are linear combinations of the state and the control with additive i.i.d. Gaussian *measurement noise* $\boldsymbol{\epsilon}_i^{(M)}$ with mean zero and covariance $\boldsymbol{\Sigma}_M$. The *initial state* $\boldsymbol{x}_0$ is independently Gaussian-distributed with mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$.

Note that the assumption that the process noise is Gaussian-distributed is in fact a very natural one if the finite difference equations ought to be discretisations of a continuous stochastic model. In this case, the distribution of a finite difference model should not depend on the discretisation step size. Suppose we split one interval into M smaller steps; then the joint process noise in this interval is $\sum_{i=1}^{M} \epsilon_i^{(P)}$, where the $\epsilon_i^{(P)}$ are i.i.d. random variables. If the variance of the $\epsilon_i^{(P)}$ is finite, then the sum will have a Gaussian distribution for large M, regardless of the distribution of the $\epsilon_i^{(P)}$. Thus, if the process noise has finite variance, the only valid distribution that can be refined on an ever smaller grid is the Gaussian distribution.

We now interpret the state-space model in terms of the kernel framework.

Theorem 2.11. *The linear state-space model (2.7) defines a GP over trajectories* $\boldsymbol{x} : \mathcal{X} \to \mathbb{R}^n$, $\mathcal{X} = \{0, .., N-1\}$. *Mean and covariance for* $i, j \in \mathcal{X}$ *are given as*

$$\boldsymbol{\mu}_i = \mathbb{E}(\boldsymbol{x}_i) = \boldsymbol{A}^i\boldsymbol{\mu}_0 + \sum_{l=1}^{i} \boldsymbol{A}^{i-l}\boldsymbol{B}\boldsymbol{u}_l, \tag{2.9}$$

$$\boldsymbol{K}_{i,j} = \mathbb{E}((\boldsymbol{x}_i - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_j)^T) = \boldsymbol{A}^i\boldsymbol{\Sigma}_0\boldsymbol{A}^{j,T} + \sum_{l=1}^{\min(i,j)} \boldsymbol{A}^{i-l}\boldsymbol{\Sigma}_P\boldsymbol{A}^{j-l,T}. \tag{2.10}$$

Proof. **[Dynamical systems view]** Since all (conditional) distributions of the $\boldsymbol{x}_i$ are Gaussian, so is the joint distribution of $\boldsymbol{x} : \mathcal{X} \to \mathbb{R}^n$, i.e. it is a GP. Furthermore, it is

$$\boldsymbol{x}_i = \boldsymbol{A}^i\boldsymbol{x}_0 + \sum_{l=1}^{i} \boldsymbol{A}^{i-l}\left(\boldsymbol{B}\boldsymbol{u}_l + \boldsymbol{\epsilon}_l^{(P)}\right).$$

Using the independence assumptions, eq. (2.9) and eq. (2.10) follow. □

Proof. **[Kernel View]** Equation (2.7) can be written equivalently as

$$\underbrace{\begin{pmatrix} \boldsymbol{\Sigma}_0^{-1/2} & & & \\ & \boldsymbol{\Sigma}_P^{-1/2} & & \\ & & \dots & \\ & & & \boldsymbol{\Sigma}_P^{-1/2} \end{pmatrix} \begin{pmatrix} \mathbf{1} & & & \\ -\boldsymbol{A} & \mathbf{1} & & \\ & \ddots & \ddots & \\ & & -\boldsymbol{A} & \mathbf{1} \end{pmatrix}}_{=\boldsymbol{R}} \underbrace{\begin{pmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{N-1} \end{pmatrix}}_{=\boldsymbol{x}} - \underbrace{\begin{pmatrix} \boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\mu}_0 \\ \boldsymbol{\Sigma}_P^{-1/2}\boldsymbol{B}\boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{\Sigma}_P^{-1/2}\boldsymbol{B}\boldsymbol{u}_{N-1} \end{pmatrix}}_{=\boldsymbol{u}} = \boldsymbol{\epsilon},$$

where the deviations $\boldsymbol{\epsilon} \in \mathbb{R}^{Nn}$ are i.i.d. Gaussian-distributed with mean zero and covariance one. Since, for any initial state $\boldsymbol{x}_0$ there exists exactly one solution of the system, i.e. one trajectory $\boldsymbol{x}$ that follows eq. (2.7), the $\boldsymbol{R}$ thus constructed is one-to-one and defines a valid regularisation operator. Using the theory from Section 2.3, the model is then equivalent to a GP with mean $\boldsymbol{\mu} = \boldsymbol{R}^{-1}\boldsymbol{u}$ and covariance $\boldsymbol{K} = (\boldsymbol{R}^T\boldsymbol{R})^{-1}$. Formulas (2.9) and (2.10) can be verified by checking that $\boldsymbol{R}\boldsymbol{\mu} = \boldsymbol{u}$ and $\boldsymbol{K}(\boldsymbol{R}^T\boldsymbol{R}) = (\boldsymbol{R}^T\boldsymbol{R})\boldsymbol{K} = \mathbf{1}$. □

The GP equivalent to (2.7) has the density

$$p(x) \propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{R}\boldsymbol{x} - \boldsymbol{u}\right\|^2\right). \tag{2.11}$$

This expression has a nice, simple interpretation: trajectories x that follow the model differential equation (2.7) are a priori the most likely functions $\boldsymbol{x} : \mathcal{X} \to \mathbb{R}^n$, and deviations from the equation are penalised quadratically.

So far, we have shown that linear state-space models define GP distributions on trajectories $\boldsymbol{x} : \mathcal{X} \to \mathbb{R}^n$. Whether any GP can be written as a linear state-space model depends on whether the reader considers models with state dimension N — or infinite state dimension in the continuous case — as valid state-space models. An introduction to infinite dimensional systems can be found in [Curtain and Zwart, 1995]. Imagine an arbitrary GP $p(z) = N(\boldsymbol{\mu}, \boldsymbol{K})$ for $z : \mathcal{X} \to \mathbb{R}$. One could simply set $\boldsymbol{x}_0 = \boldsymbol{z}$, i.e. $\boldsymbol{\mu}_0 = \boldsymbol{\mu}$, $\boldsymbol{\Sigma}_0 = \boldsymbol{K}$, and then propagate with $\boldsymbol{A} = \mathbf{1}$, $\boldsymbol{u}_i = 0$, and $\boldsymbol{\Sigma}_P = \mathbf{0}$. Alternatively, one could use the decomposition $p(z) = p(z_0)p(z_1|z_2)...p(z_{N-1}|z_0, .., z_{N-2})$ to formulate a state-space model. Since for arbitrary covariances $\boldsymbol{K}$, we cannot assume special Markov properties, we would need again an N-dimensional state-space to represent the GP. For special $\boldsymbol{K}$, however, this construction may allow one to exploit Markov properties of the GP, and thus a representation with a much lower state dimension.

2.4.2 Linear Differential Equations and the Fourier Transform

Kernel methods are often motivated via regularisation in the Fourier domain [Schölkopf and Smola, 2002]. At the same time, derivative operators reduce to simple multiplications in the Fourier domain. This leads us to examine more closely the connection between differential equations and Fourier space penalisation in this section.

Assume $\mathcal{X}$ to be the discretised real line, i.e. $\mathcal{X} = \{\frac{i}{h} | i = 1, .., N\}$, $h > 0$, and let $L(\lambda) = \sum_{i=0}^{n} a_i \lambda^i$ be an n-th order polynomial. Consider the linear ODE

$$L(\boldsymbol{D})\boldsymbol{f} = \sum_{i=0}^{n} a_i \boldsymbol{D}^i \boldsymbol{f} = 0, \tag{2.12}$$

where $\boldsymbol{D}$ is the first derivative operator and $f : \mathcal{X} \to \mathbb{R}$.

For the remainder of the chapter we will assume periodic boundary conditions, allowing the use of the discrete Fourier transform to express the derivative operator. Periodic systems are in general not causal, since random events in the future could propagate forward to influence the past. However, for stable linear systems these effects can be neglected for large enough domains, because the contribution of any state onto future state values decays to zero eventually. The natural formulation of the Fourier transform in terms of complex exponentials requires the use of complex-valued linear algebra. For ease of presentation we have omitted this so far, however, all definitions and theorems can also be formulated with complex numbers, as sketched in Additional Material 2.8.1. We will also assume that $L(\boldsymbol{D})$ is one-to-one. Unfortunately, there are common examples where this is not the case, e.g. for the second derivative used for thin-plate splines. Regularisation with non-one-to-one operators requires the use of the cpd kernels as described in Additional Material 2.8.2.

For discrete $\mathcal{X}$, a straightforward approximation of the continuous derivative is the approximate derivative operator $\boldsymbol{D}$ given as follows in the case of periodic boundary conditions,

$$D = \frac{1}{h}\begin{pmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \\ 1 & & & -1 \end{pmatrix}. \tag{2.13}$$

$\boldsymbol{D}$ can be diagonalised in the Fourier basis, $\boldsymbol{D} = \sum_{k=1}^{N} \boldsymbol{u}_k w_k \boldsymbol{u}_k{}^T$, where $w_k = \frac{1}{h}(\exp\left(i\frac{2\pi}{N}k\right) - 1)$ and $\boldsymbol{\delta}_{x_j}{}^T \boldsymbol{u}_k = \exp\left(i\frac{2\pi}{N}jk\right)$. It is well-known that functions of $\boldsymbol{D}$ can be computed by applying equivalent operations to the eigenvalues of w_k. In particular, the corresponding kernel function then is

$$\begin{aligned} k(x_l, x_m) &= (\overline{L(\boldsymbol{D})}^T L(\boldsymbol{D}))^{-1}_{lm} = \boldsymbol{\delta}_{x_l}{}^T (\overline{L(\boldsymbol{D})}^T L(\boldsymbol{D}))^{-1} \boldsymbol{\delta}_{x_m} && (2.14) \\ &= \sum_{k=1}^{N} \boldsymbol{\delta}_{x_l}{}^T \boldsymbol{u}_k \frac{1}{\overline{L(w_k)} L(w_k)} \boldsymbol{u}_k{}^T \boldsymbol{\delta}_{x_m} && (2.15) \\ &= \sum_{k=1}^{N} \frac{1}{|L(w_k)|^2} \exp\left(i\frac{2\pi}{N}k(l-m)\right). && (2.16) \end{aligned}$$

Thus, the kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the (discrete) Fourier transform of $g(w_k) = \frac{1}{|L(w_k)|^2}$. Since g is real-valued, the Fourier transform of it is also real and additionally symmetric. The corresponding kernel function then is real-valued and only depends on the distance between x_l and x_m, $d = |l - m|$, that is, it is translation-invariant.

Let us motivate eq. (2.12) from a regularisation point of view. High derivatives are described by polynomials $L(\lambda)$ of high order, in which case $\|L(\boldsymbol{D})\boldsymbol{f}\|^2 = \sum_k \boldsymbol{f}^T \boldsymbol{u}_k |L(w_k)|^2 \boldsymbol{u}_k{}^T \boldsymbol{f}$ strongly penalises high frequencies. The corresponding kernel then contains few high frequency components and is thus relatively smooth.

One can also discuss the reverse derivation from a translation-invariant kernel function on $\mathcal{X}$ to a differential regularisation operator. Translation-invariance implies that the covariance operator $\boldsymbol{K}$ is diagonal in the Fourier basis. In order to derive a differential equation, invert the eigenvalues of $\boldsymbol{K}$, take the square root, and interpolate the result by a polynomial L of

at most degree N. Eq. (2.12) then yields the model that is implicitly used when performing regression with this kernel.

A famous example is the Gaussian kernel, $k(x_i, x_j) \propto \exp\left(-\frac{1}{2\sigma^2}|i-j|^2\right)$. The discrete Fourier transform is difficult to compute analytically in this case, so we approximate it with its continuous counterpart for large N and small step sizes. The continuous Fourier transform of a Gaussian is again a Gaussian with variance σ^{-2}. Inverting and taking the square root, we derive a function $\exp\left(\frac{\sigma^2}{4}w^2\right)$, whose Taylor expansion is $L(w) = \sum_{n=0}^{\infty} \frac{\sigma^{2n}}{2^{2n}n!}w^{2n}$. Replacing w by the derivative ∂_x, we re-derive the result of [Girosi et al., 1993]. They state that the Gaussian kernel is equivalent to regularisation with derivatives of all (even) orders,

$$\boldsymbol{R} = \sum_{n=0}^{\infty} \frac{\sigma^{2n}}{2^{2n}n!}\boldsymbol{D}^{2n}. \tag{2.17}$$

A larger σ leads to a stronger penalisation of high derivatives, i.e., to smoother functions.

The introduction of the Fourier transform above also leads to a discrete version of Bochner's theorem [Bochner, 1933]. While the original theorem in continuous domains deals with positive semi-definite functions, we can make a stronger statement involving positive definiteness for finite domains: A translation-invariant function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $k(x_i, x_j) = \phi(i-j)$, is positive definite if and only if the (discrete) Fourier transform of ϕ is positive. Since the Fourier transform of ϕ is identical with the eigenvalues of $\boldsymbol{K}$, and we do not have to be concerned with the existence and regularity of Fourier transforms in finite domains, the result, in our case, is trivial.

2.4.3 Linear Stochastic PDEs

A general form of discrete stochastic linear PDEs for $f : \mathcal{X} \to \mathbb{R}$ is

$$f(x_i) = \sum_{x_j \in \mathcal{N}_i} a_{ij} f(x_j) + \epsilon_i^{(P)}, \quad x_i \in \mathcal{X}, \tag{2.18}$$

where $\mathcal{N}_i \subset \mathcal{X}$ is the set of neighbours of x_i, $a_{ij} \in \mathbb{R}$, and $\boldsymbol{\epsilon}^{(P)}$ is i.i.d. zero mean Gaussian noise with covariance $\boldsymbol{K}_i$. Since eq. (2.18) is a linear equation system in $\boldsymbol{f}$, it is a valid kernel model equation (2.6). If the x_i are placed on a regular grid and periodic boundary conditions are assumed, the Fourier transform methods from the previous section can also be applied for this multivariate setting.

Note that apart from being a discretised stochastic PDE, eq. (2.18) is also one form of writing Gaussian Markov random fields. Additionally, graph-based learning involving the graph Laplacian can be written in this form. This noteworthy fact implies that multiple methods in physics, control theory, image processing, PDE theory, machine learning, and statistics all use the same underlying model.

2.4.4 State Estimation and System Identification Using Kernels

Both GP and SVR regression can be interpreted as optimal state estimators if the kernel is chosen with respect to a differential equation as described above. Both methods try to

minimise the deviation of the estimated trajectory from the differential equation $\boldsymbol{Rf} = 0$ and at the same time try to minimise the distance to the measured data points, where the distance is measured either through a loss function in the SVR case or through a likelihood in the probabilistic setting. An optimal trade-off between these potentially contradicting targets is obtained.

Furthermore, SVR and GP regression can both be used for system identification. In SVR one typically chooses the kernel to minimise the cross validation error on the training set. In GP regression one tries to find the kernel function that maximises the marginal likelihood, that is, the complete likelihood of the training data and latent function $f : \mathcal{X} \to \mathbb{R}$ marginalised over the latent variables. Since each DE can be related to a specific kernel function, optimising for the best kernel in a class of kernels derived from DEs is equivalent to choosing the most appropriate DE model for the given data set. More formally, assume, for example, that we are interested in a DE model of the form

$$L_{\boldsymbol{\theta}}(\boldsymbol{D})\boldsymbol{f} = \sum_{i=0}^{\theta_0} \theta_{i+1} \boldsymbol{D}^i \boldsymbol{f} = 0. \tag{2.19}$$

Optimising for the best parameters $\boldsymbol{\theta}$ of the corresponding kernel function $\boldsymbol{K_\theta} = (L_{\boldsymbol{\theta}}(\boldsymbol{D})^T L_{\boldsymbol{\theta}}(\boldsymbol{D}))^{-1}$ is equivalent to determining the best differential model of the above form.

The possibility of using kernel machines to estimate the state and the parameters of differential equations has been noticed by [Heckman and Ramsay, 2000] in a spline context, and by [Steinke and Schölkopf, 2006] who use SVR and cross-validation.

Before discussing the practical implications of this matter, we present some examples highlighting the kernel framework and its connections to differential equations.

2.5 Examples

2.5.1 The Pendulum – State Estimation

Consider again the pendulum in Figure 2.2. According to Newton's third law, the free motion dynamics of the angle of the pendulum is *approximately* described by the second-order linear differential equation

$$ml^2\ddot{\phi}(t) + \lambda\dot{\phi}(t) + mgl\phi(t) = 0, \tag{2.20}$$

where m is the mass of the pendulum, l the length, g the gravitational constant, and $\lambda > 0$ a damping factor. Equation (2.20) is only approximately correct for two qualitatively different reasons. Firstly, it is only the linearisation around the rest position of a truly nonlinear differential equation. The true gravitational effect is $mgl\sin(\phi(t))$ which for small $\phi(t)$ is similar to $mgl\phi(t)$. Secondly, there may be many, potentially random influences on the pendulum which are not known or cannot in principle be observed. For example, the viscosity of the surrounding air could change slightly due to local temperature changes, or more drastically a by-passer could simply hit the pendulum. Both model mismatch and stochastic influences can be modelled as process noise in a stochastic differential equation system, rendering this a versatile model.

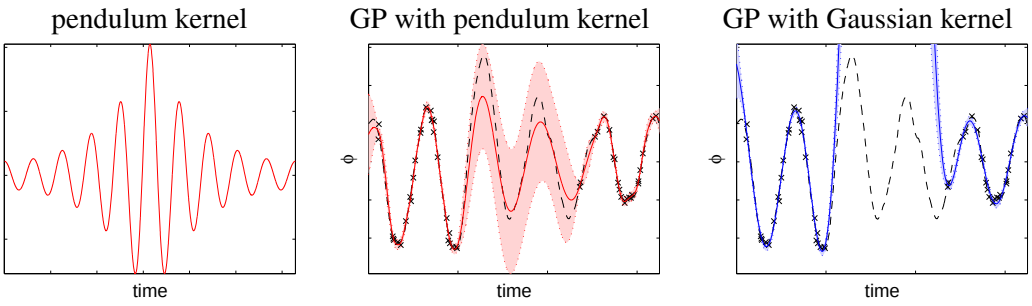


Figure 2.4: *(left) Kernel function $k(x_i, .)$ derived from the differential equation (2.20) describing a pendulum. Fourier space transforms with periodic boundary conditions were used. The resulting kernel is translation invariant, x_i is chosen in the middle of the interval. (middle) The 50 data points from Figure 2.2, denoted by black crosses, are regressed using a GP with the pendulum kernel, left, and a Gaussian i.i.d. likelihood. The solid red line denotes the mean of the posterior GP, the shaded area plus-minus one marginal standard deviation of the function values. The dashed black line shows the true sample path from which the data points were generated. (right) GP regression as in the middle figure, however, with a Gaussian kernel.*

Fourier space method The pendulum equation (2.20) can be written in the operator form

$$L(\partial_x)f(x) = (\partial_x^2 + c_1\partial_x + c_2 I)f(x) = 0, \tag{2.21}$$

where $I : \mathcal{H} \to \mathcal{H}$ it the identity operator. We discretise an input interval into $N = 4096$ steps and apply the Fourier framework from Section 2.4.2 to derive a translation invariant kernel $k(x_i, x_j) = (L(\boldsymbol{D})^T L(\boldsymbol{D}))^{-1}_{ij}$. The resulting kernel and a GP regression with this kernel for the pendulum data in Figure 2.2 (right) is shown in Figure 2.4.

Observe that the GP regression with the kernel adapted to the pendulum is able to nicely follow the true sample path (middle). While a GP regression with a standard Gaussian kernel yields comparable results in regions where many data points are observed, it performs much worse in the middle where no observations are recorded. This can be explained as follows. Since the a priori model of $\boldsymbol{f}$ in terms of a stochastic differential equation, $\boldsymbol{Rf} = \boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{1})$, allows violations of the exact differential equation $\boldsymbol{Rf} = 0$, multiple observations can override the model. However, in regions with no observations the prior is more important. Since the Gaussian kernel encodes for the wrong prior model (2.17) its predictions are especially bad in these regions.

State-space view The pendulum equation (2.20) can equally be written as a state-space model with a two-dimensional state, $n = 2$. Then it is

$$\boldsymbol{A} = h\begin{pmatrix} 0 & 1 \\ -\lambda/ml^2 & -g/l \end{pmatrix} + \mathbf{1},\ \boldsymbol{C} = \begin{pmatrix} 1 & 0 \end{pmatrix},\ \boldsymbol{B} = \boldsymbol{D} = 0,$$
$$\boldsymbol{K} = \begin{pmatrix} 0 & \\ & \sigma^{(P),2} \end{pmatrix},\ \boldsymbol{H} = \sigma^{(M),2},$$

where we used $N = 4096$, $h = 0.003$, $\mu_0 = (0.2, 0.1)^T$, $\Sigma_0 = 10^{-5}\mathbf{1}$, $\lambda/ml^2 = 25$, $g/l = 1$, $\sigma^{(P)} = 0.085$, and $\sigma^{(M)} = 0.02$. The data samples for the pendulum – see Figure 2.2 (right) – were drawn from this model.

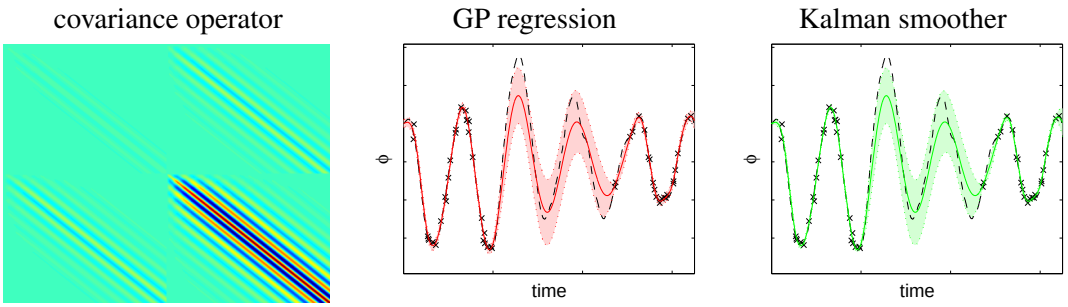


Figure 2.5: *(left) The covariance matrix derived from the differential equation describing a pendulum (2.20) using a state-space formulation with initial condition. Since the state-space is two dimensional the kernel function has for each position pair i, j four entries. Two entries describe the covariance within each component, the two others the cross-covariances. (middle) GP regression using the kernel from the left figure and the 50 data points from Figure 2.2. The solid red line denotes the mean of the posterior GP, the shaded area plus-minus one marginal standard deviation for the function values. The dashed black line is the original sample path. (right) Equivalent results produced by a Kalman smoother.*

The covariance operator for this state-space model computed by eq. (2.10) is colour-coded in Figure 2.5 (left). Observe the oscillations when fixing a row or column which corresponds to fixing a kernel centre x_i and observing the kernel function $\boldsymbol{K}_{x_i}$. Figure 2.5 (middle) shows the marginal posterior mean and variances when performing GP regression using the kernel from the left figure and the data from Figure 2.2 (right). Note that the results are up to numerical errors identical to the solution of a Kalman smoother [Kalman, 1960], as shown in Figure 2.5 (right). This fact is discussed in more detail in Section 2.6.

2.5.2 The Pendulum – Parameter Estimation

In Figure 2.6 we show results from a simple system identification task, i.e. determining the parameter c_2 of the pendulum model (2.21). We use the pendulum kernel in Figure 2.4 and maximise the marginal likelihood of a GP regression model for the optimal value of c_2, where c_1 is assumed to be known. The maximum is attained for a value c_2 close to the true model. We also computed the marginal likelihood for GP regression with a Gaussian kernel. The maximal marginal likelihood for a Gaussian kernel with automatically chosen parameters is 20 orders of magnitude smaller than for the pendulum kernel. In a Bayesian interpretation the data thus strongly prefers a pendulum-adapted model over the standard Gaussian kernel model.

2.5.3 Two-Dimensional PDEs

In this section we discuss kernels for two-dimensional domains. We show how the harmonic and the thin-plate spline regulariser that both build on derivatives and can be interpreted as stochastic PDEs can be incorporated into the kernel framework.

Next, we show examples of harmonic and thin-plate spline regularisation in the kernel framework.

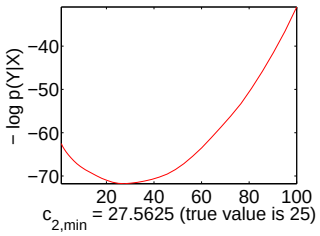


Figure 2.6: *The negative log marginal likelihood of a GP regression for the pendulum data set in Figure 2.2. Different parameters c_2 are used for the pendulum adapted kernel in Figure 2.4. The minimum of the negative log marginal likelihood is obtained for $c_{2,min} = 27.5$, the true value is $c_{2,true} = 25$.*

As mentioned in Section 2.4.3, the Fourier transform can also be applied for functions on higher-dimensional domains, and derivative operators can also be translated into multiplications in this setting. Consider a rectangular grid with $N^2 = 256^2$ points and periodic boundary conditions. The discrete derivative $\boldsymbol{D}^1$ in the first direction and the derivative $\boldsymbol{D}^2$ in the second direction are both diagonal in the tensor Fourier basis $\boldsymbol{u}_{k^1} \otimes \boldsymbol{u}_{k^2}$, where $(\boldsymbol{\delta}_{x_l} \otimes \boldsymbol{\delta}_{x_m})^T \boldsymbol{u}_{k^1} \otimes \boldsymbol{u}_{k^2} = \exp\left(i\frac{2\pi}{N}(lk^1 + mk^2)\right)$, and the eigenvalues are $w_{k^1 \otimes k^2} = w_{k^1} w_{k^2}$, $k^1, k^2 = 1, .., N$.

Harmonic regularisation results from penalising the Jacobian of $f : \mathcal{X} \to \mathbb{R}$, that is, all first derivatives,

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{D}^1 \\ \boldsymbol{D}^2 \end{pmatrix}.$$

This results in $\|\boldsymbol{R}\boldsymbol{f}\|^2 = \boldsymbol{f}^T \boldsymbol{\Delta} \boldsymbol{f}$, where $\boldsymbol{\Delta} = \boldsymbol{D}^{1^T}\boldsymbol{D}^1 + \boldsymbol{D}^{2^T}\boldsymbol{D}^2$ is the (discrete) Laplace operator. Functions minimising this expression, the so-called harmonic energy, effectively minimise the graph's area and are thus very common in many fields of research, especially computer graphics [Floater and Hormann, 2005]. Since constant functions are not penalised by $\boldsymbol{R}$, the cpd framework for non one-to-one $\boldsymbol{R}$ has to be used in this case, see Additional Material 2.8.2. Postponing a more detailed discussion, the most important change here is to use the pseudoinverse instead of the inverse for deriving the kernel, $\boldsymbol{K} = (\boldsymbol{R}^T\boldsymbol{R})^+$. This operation is easily performed using the two-dimensional fast Fourier transform.

The thin-plate splines energy penalise the Hessian of $f : \mathcal{X} \to \mathbb{R}$, that is, all second derivatives,

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{D}^1\boldsymbol{D}^1 \\ \boldsymbol{D}^1\boldsymbol{D}^2 \\ \boldsymbol{D}^2\boldsymbol{D}^1 \\ \boldsymbol{D}^2\boldsymbol{D}^2 \end{pmatrix}.$$

The energy leaves linear functions unpenalised, thus we again have to use the cpd framework and correspondingly the pseudoinverse.

In Figure 2.7, we show the resulting kernels for harmonic and thin-plate spline regularisation. Furthermore, we show results of approximating 5 randomly chosen data points with a GP regression with the respective kernels. Note that the harmonic kernel is sharply peaked, but the regression output stays in the convex hull of the training output values, the famous mean value property of harmonic maps. The thin-plate spline solution is much smoother, but occasionally overshoots the training values.

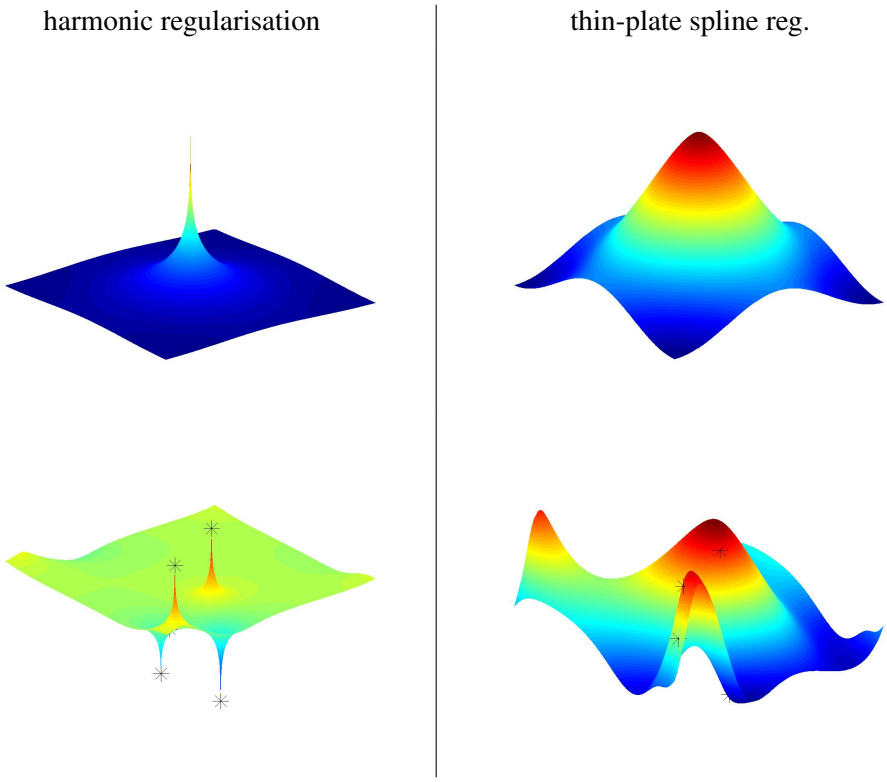


Figure 2.7: *For a two-dimensional domain $\mathcal{X}$ with periodic boundary conditions, the kernel functions $\boldsymbol{R}_{x_i}$ for harmonic and thin-plate spline regularisation are shown in the top row. x_i is chosen in the middle of $\mathcal{X}$. Below we show the mean of a GP regression with these kernels and 5 data points, denoted as black stars.*

2.5.4 Graph Laplacian

Since graph domains are naturally finite, graph-based learning is a good example of where the finite domain kernel framework directly applies without the need for discretisation.

The graph Laplacian is an approximation of the true Laplacian $\boldsymbol{\Delta}$ on graphs [Hein et al., 2007]. Kernels on graphs based on the graph Laplacian are described by [Smola and Kondor, 2003]; they are used for semi-supervised learning by [Zhu et al., 2003]. [Tipping and Bishop, 2003] use them in GPs on finite image domains for image super-resolution. The graph Laplacian $\boldsymbol{\Delta}_G$ for a graph $G = (E, \mathcal{X})$ with edges E and vertices $\mathcal{X}$ is given by $\boldsymbol{\Delta}_G = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{W}_{ij}$ is the weight of edge $(i,j) \in E$, 0 if $(i,j) \notin E$, and the degree matrix $\boldsymbol{D}$ is diagonal with entries $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$. We use an ϵ-neighbourhood graph constructed from 40 random points in $[0,1]^2$, $\epsilon = 0.2$, i.e. $(i,j) \in E$ if and only if $\|x_i - x_j\| < \epsilon$. Edge weights $\boldsymbol{W}_{ij}$ are set as $\boldsymbol{W}_{ij} = \exp\left(-\frac{1}{\epsilon^2}\|x_i - x_j\|^2\right)$.

As in the above section, setting $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{\Delta}_G$ leads to the problem that $\boldsymbol{\Delta}_G$ is not one-to-one. Functions $\boldsymbol{f}$ constant on a connected component have $\boldsymbol{f}^T\boldsymbol{\Delta}_G\boldsymbol{f} = 0$, a fact commonly used in spectral clustering [von Luxburg, 2007]. Thus, in order to derive a kernel we again use the pseudoinverse. For more details see Additional Material 2.8.2.

Figure 2.8 shows the resulting kernel function $\boldsymbol{K}_{x_i}$. The closer a point is to x_i the larger its corresponding kernel values. Equivalently, under the corresponding GP prior the correlation

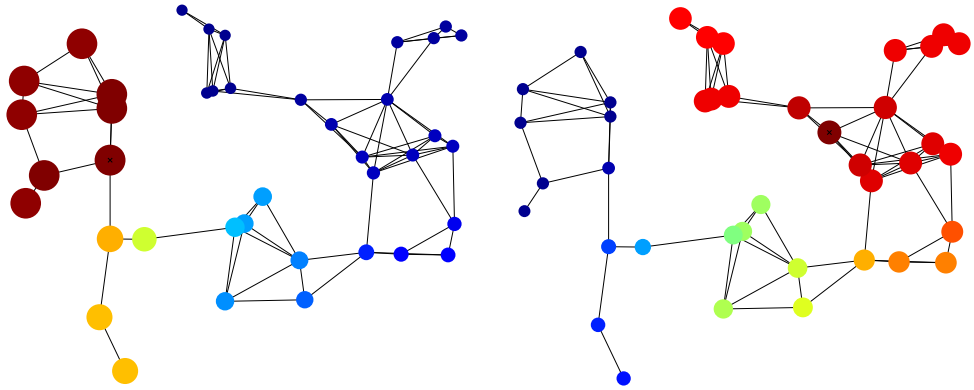


Figure 2.8: *Kernel corresponding to a graph Laplacian as regulariser* $\boldsymbol{R}^T\boldsymbol{R}$*. The kernel functions* $\boldsymbol{R}_{x_i}$ *are encoded in the colour and the size of the nodes. Vertex* x_i *is marked with a black cross, the edges of the graph are shown in black.*

of the function value at a certain point with the function value at x_i is the stronger the closer the point is to x_i. Note that the distance is measured in terms of the geodesic distance intrinsic to the graph, not the Euclidean distance of the embedding space.

2.6 Discussion

We have shown that common linear differential equation models can be flawlessly integrated into the kernel framework and that trajectory/state estimation and system identification can both be performed with kernel machines such as SVR or GP regression. However, there are already many well-established algorithms for state estimation and system identification. In this section, we discuss how kernel methods relate to these standard methods, and when one should prefer which type of algorithm.

State estimation in the linear state-space model described in Section 2.4.1 is classically dominated by the Kalman filter/smoother [Kalman, 1960] and its variants [Ljung, 1999]. For such models the Kalman filter algorithm is also equivalent to graphical model message-passing algorithms [Jordan et al., 1999]. Since all these models perform optimal state estimation in the state-space model as do kernel methods such as GP regression or SVR, the results of the two types of methods are identical. The Kalman filter can be interpreted as just an efficient way of computing GP regression exploiting the special features of (low-dimensional) linear state-space models. SVR is slightly different in that it typically uses an ϵ-insensitive linear loss function [Schölkopf and Smola, 2002] which corresponds to a different likelihood model. For a quadratic loss, however, the output of an SVR will be identical to the mean estimate of a Kalman smoother. It is interesting to note that even without considering equivalence of the underlying model assumptions, kernel methods can be related to Kalman filter-like algorithms. For dynamical systems, the matrix $\boldsymbol{R}^T\boldsymbol{R}$, whose inverse yields the covariance operator, is block-tridiagonal. [Huang and McColl, 1997] propose an algorithm to invert such matrices in linear time using a forward-backward scheme that is closely reminiscent of the Kalman smoother algorithm.

Considering system identification for linear ODEs, there exist many different algorithms in the control community such as subspace identification, Fourier space methods, or prediction

error methods [Ljung, 1999]. Statisticians classically use Expectation Maximisation (EM), which maximises the marginal likelihood of the model, that is, the likelihood of the observed outputs given the parameters with the hidden states integrated out. The marginal likelihood can be efficiently computed using a Kalman smoother. As for the case of state estimation, all these methods are at least qualitatively equivalent to kernel machine model selection algorithms. The marginal likelihood is also used in GP regression for kernel selection. The cross validation error can be seen as an approximation of the negative marginal likelihood or the prediction error, which also links SVR regression to this picture.

Since we have argued above that kernel methods are largely equivalent to standard algorithms for treating differential equations, we might ask in which context may one benefit from using kernel methods. Kernel methods are to be understood here as algorithms that explicitly compute the kernel function and that perform batch inference by minimising/integrating an expression of the dimension m, where m is the number of measured data points. Conversely, all classical algorithms work sequentially, performing inference without explicitly computing the kernel function.

For one-dimensional problems, that is, ordinary differential equations or dynamical systems, Kalman filter or graphical model-based methods concentrate on the chain-like structure of the model. They give rise to many $O(N)$ algorithms for computing marginal means, marginal variances, or the marginal likelihood, where N is the number of discretisation steps. If only m measurements, $m \ll N$, are given, this effort can be reduced to $O(m)$ with a little pre-computation, summarising many small steps without observations into one large step. In contrast, kernel-based methods working with the full covariance matrix typically scale around $O(m^3)$ for regression or computing the marginal likelihood. Furthermore, such methods have to compute the kernel function for the given dynamical system. Using the Fourier framework described in Section 2.4.2, the fast Fourier transform takes $O(N \log N)$ time, and using the state-space model, the kernel is given explicitly by eq. (2.10). One advantage of the kernel view for dynamical systems is that it yields direct access to all pairwise marginal distributions, even for non-neighbouring points, which is not obvious with sequential algorithms.

For multidimensional problems, that is, in partial differential equations, the kernel method's view on the joint problem is more useful in practical terms, since message-passing is difficult due to many loops and is not guaranteed to yield the optimal solution [Jordan et al., 1999]. However, in this case, too, the kernel cannot be computed analytically but has to be derived either through a fast Fourier transform or, in the worst case, through matrix inversion, which scales like $O(N^3)$. If one aims at estimating the whole latent function $f : \mathcal{X} \to \mathbb{R}$, then direct optimisation of problem (2.1) may be advantageous in comparison with computing the kernels first and then optimising the kernelised problem. For example, in graph-based learning one typically solves the estimation problem directly in the so-called primal. However, if the graph were given in advance and the labels of the nodes were only uncovered at a later time, it would be advantageous to precompute the kernel functions, since regression to yield all of $f : \mathcal{X} \to \mathbb{R}$ could then be performed in $O(m^3)$ instead of $O(N^3)$.

In sum, one could say that the connection between kernels and differential equations will typically not yield faster or better algorithms, except in a few special cases. However, it may help to gain deeper theoretical understanding of both kernel methods and differential equations. For example, the connection presented shows that given a state-space model

and measurements, the posterior covariances between states at different time points are not dependent on the observations; they are simply given through the covariance matrix $\boldsymbol{K}$. This insight is not obvious from looking at the Kalman update equations. Conversely, the existence of an $O(N)$ inversion algorithm for tridiagonal matrices is not surprising when formulating the inversion in terms of a Kalman filter state estimation problem.

2.6.1 Nonlinear Extensions

This chapter has so far solely focused on linear differential equations or equivalently on linear regularisation operators. However, there is great interest in nonlinear models in many fields, and it is natural to ask whether any of the insights presented above carry over to such a situation.

The disappointing answer is that most of the results are critically dependent on the linearity assumption. If $\boldsymbol{R}$ is not a linear operator, then $\|\boldsymbol{Rf}\|$ does not define a norm. Also, interpreting the kernel as the Green's function of $\boldsymbol{R}^T\boldsymbol{R}$, that is, the solution of $\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{K}_{x_i} = \boldsymbol{\delta}_{x_i}$, does not make sense, since the solution of nonlinear differential problems $\boldsymbol{Rf} = \boldsymbol{u}$ can not in general be represented as a linear sum of such Green's functions as in the linear case. Also, corresponding probability distributions over functions $f : \mathcal{X} \to \mathbb{R}$ are then, in general, not Gaussian any more, and often can not be described through an analytic expression at all.

Kernel methods are sometimes used for nonlinear systems, typically in the form that $\boldsymbol{x}_{i+1} = \boldsymbol{f}(\boldsymbol{x}_i)$, where $f : \mathbb{R}^n \to \mathbb{R}^n$ is described by a kernel regression. However, such kernel methods should not be mixed up with the type of kernels we discussed here, since in this chapter the kernels were functions of time, not of the preceding state. Furthermore, such one-step-ahead prediction with kernels is not associated with a GP over trajectories in $\mathcal{H}$, nor does it yield an SVR problem of type (2.1) over trajectories.

While these are strong negative statements, the dual view of differential equations — either in terms of local conditional distributions or more kernel-like as joint distributions over whole functions — may still help to shape intuitions for the nonlinear case and may help to develop new approximate inference algorithms. For example, [Archambeau et al., 2007] investigate the joint N-dimensional state distribution of a nonlinear differential equation, and approximate it using an N-variate GP distribution corresponding to a low order linear differential equation. Their key calculation is motivated in finite dimensions and is then extended to continuous domains. Conversely, one could also ask whether sequential inference schemes for nonlinear differential equations such as the extended Kalman filter, the unscented Kalman filter [Julier and Uhlmann, 1997], or sequential Monte Carlo methods [Doucet et al., 2001] can be transferred to other, potentially multivariate, nonlinear kernel-like problems.

2.7 Conclusion

We have presented a joint framework for kernels, RKHSs, GPs, and regularisation operators. All these objects are closely related to each other. Given the theoretical framework, it is natural to see stochastic linear differential equations as important examples of regularisation operators. We have discussed ordinary as well as partial linear differential equations.

While the exposition is kept simple through the use of the finite domain assumption, note that most results also hold for infinite/continuous domains and we hope the readers will be able to realise this when making comparisons with existing work. An exact treatment for infinite, continuous domains often requires advanced mathematical machinery [Bogachev, 1998; Oksendal, 2002; Wendland, 2005], and we have thus concentrated on the finite dimensional case, which mostly yields qualitatively similar results.

A good understanding of all the mentioned interrelations between different methods and communities will help the readers to select suitable algorithms for specific problems and may guide their intuition in developing new methods, for example, for dealing with nonlinear differential equations. One potential future application may be to explore the meaning of kernel PCA [Smola et al., 1998] for kernels derived from dynamical systems, which to our knowledge has not yet been studied.

2.8 Additional Material

2.8.1 Complex-Valued Functions and Kernels

For finite domains $\mathcal{X}$, complex-valued functions $f : \mathcal{X} \to \mathbb{C}$ are isomorphic to elements in $\mathbb{C}^N = \mathcal{H}$. Some basics of linear algebra in $\mathbb{C}^N$ are as follows: Set $\boldsymbol{f}^* = \overline{\boldsymbol{f}^T}$. The standard inner product in $\mathbb{C}^N$ is $\boldsymbol{f}^*\boldsymbol{g} = \sum_i \overline{f(x_i)} g(x_i)$ and thus satisfies $\boldsymbol{f}^*\boldsymbol{g} = \overline{\boldsymbol{g}^*\boldsymbol{f}}$. A matrix $\boldsymbol{A}$ is called symmetric or hermitian, if $\boldsymbol{A}^* = \overline{\boldsymbol{A}^T} = \boldsymbol{A}$. Hermitian matrices have real eigenvalues λ_i and an orthogonal basis of eigenfunctions $\{\boldsymbol{u}_i\}_{i=1,..,N}$, thus, $\boldsymbol{f}^*\boldsymbol{A}\boldsymbol{f}$ is real for any $\boldsymbol{f} \in \mathcal{H}$.

Complex-valued algebra does not interfere with the kernel framework. All definitions, theorems, and proofs of Section 2.3 hold if the functions are understood as complex-valued and the appropriate inner product is used. For example the positive definite kernel condition then states that $\sum_{i,j} \overline{\alpha_i}\alpha_j k(x_i, x_j) > 0$, where the sum is real-valued, since $\boldsymbol{K}$ is a hermitian matrix by assumption.

We will not be more explicit here, but just state the following theorem, that shows that the complex-valued theory consistently reduces to the real-valued one described in Section 2.3, if all involved entities are in fact real.

Theorem 2.12. *With the notation of the SVR objective (2.3) and the representer theorem 2.8 the following holds: if the observation values $\{y_i \mid i = 1, .., m\}$ and the kernel $\boldsymbol{K}$ are real-valued and the loss term is a non-decreasing function of $|f_{\boldsymbol{\alpha}}(x_i) - y_i|$, then the function $f_{\boldsymbol{\alpha}} : \mathcal{X} \to \mathbb{C}$ minimising (2.3) is real-valued and additionally all coefficients $\boldsymbol{\alpha}$ in Theorem 2.8 are real.*

Proof. Assume $f = \boldsymbol{f}^{\Re} + i\boldsymbol{f}^{\Im} \in \mathcal{H}$, $\boldsymbol{f}^{\Re}, \boldsymbol{f}^{\Im} \in \mathbb{R}^N$. Then

$$\|\boldsymbol{f}\|_K^2 = \left\|\boldsymbol{f}^{\Re}\right\|_K^2 + \left\|\boldsymbol{f}^{\Im}\right\|_K^2 + 2\underbrace{\Im\left(\boldsymbol{f}^{\Im T}\boldsymbol{K}^{-1}\boldsymbol{f}^{\Re}\right)}_{=0, \text{ as } \boldsymbol{K} \text{ is real}} \tag{2.22}$$

is minimised for $\boldsymbol{f}^{\Im} = 0$. Similarly, the loss term is minimised for $\boldsymbol{f}^{\Im} = 0$, since the loss of $|f(x_i) - y_i|^2 = ({\boldsymbol{\delta}_{x_i}}^T\boldsymbol{f}^{\Re} - y_i)^2 + ({\boldsymbol{\delta}_{x_i}}^T\boldsymbol{f}^{\Im})^2$ is by assumption larger that the loss of $\left|f^{\Re}(x_i) - y_i\right|^2$. Thus the combined minimum is attained for $\boldsymbol{f}^{\Im} = 0$. It is $\boldsymbol{f}_X = \boldsymbol{K}_X\boldsymbol{\alpha}$ and $\boldsymbol{K}_X$ is real and positive definite, thus one-to-one. It follows that $\boldsymbol{f}_X \in \mathbb{R}^m$ requires $\boldsymbol{\alpha} \in \mathbb{R}^m$. □

2.8.2 The CPD World

Regularisation operators $\boldsymbol{R}^c$ which are not one-to-one motivate the use of the conditionally positive definite (cpd) framework. For example, regularising with the first derivative yields zero penalty for all constant functions, thus $\boldsymbol{R}^c$ cannot be one-to-one in this case.

Most kernel results in Section 2.3 can be extended to cpd kernels. However, special care has to be taken of the null space of the regularisation operator. The description in this section will use the complex-valued setting as introduced in Additional Material 2.8.1 above.

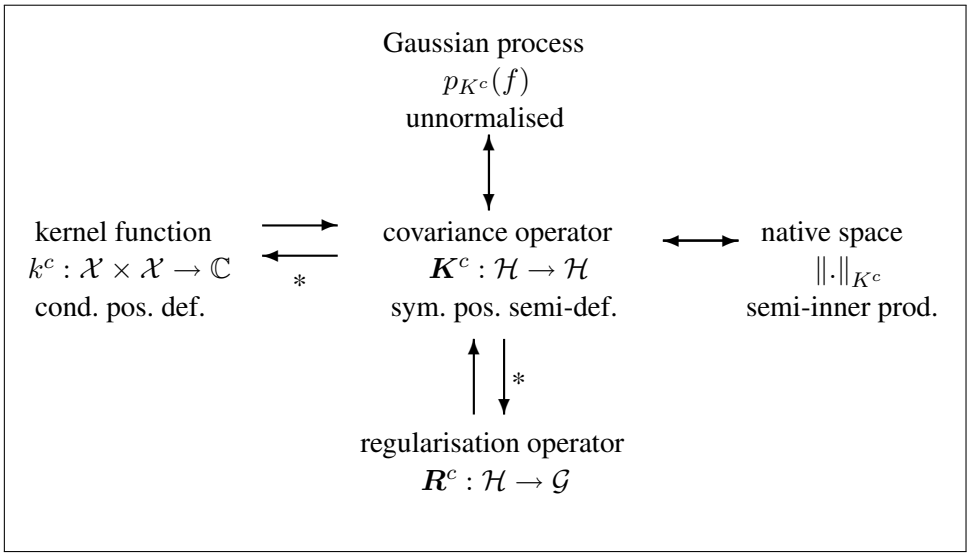


Figure 2.9: *Common objects in the cpd kernel framework and their interrelations. Arrows denote that one can uniquely be determined from the other (the * denotes that this connection is not unique). A semi-inner product is an inner product which is only positive semi-definite.*

The Pseudoinverse

Consider a hermitian matrix $\boldsymbol{A}$ with orthonormal eigendecomposition $\boldsymbol{A} = \sum_i \boldsymbol{u}_i \lambda_i \boldsymbol{u}_i^*$. If $\boldsymbol{A}$ is not one-to-one, i.e. $\exists i : \lambda_i = 0$, then we can define the (Moore-Penrose) pseudoinverse of $\boldsymbol{A}$ by

$$\boldsymbol{A}^+ = \sum_{i=1, \lambda_i \neq 0}^{N} \boldsymbol{u}_i \frac{1}{\lambda_i} \boldsymbol{u}_i^*.$$

Lemma 2.13. *For $\boldsymbol{A}$ as above and $\boldsymbol{P} = \sum_{\{i|\lambda_i=0\}} \boldsymbol{u}_i \boldsymbol{u}_i^*$ the orthogonal projection from $\mathcal{H}$ to the null space $\mathcal{N}$ of $\boldsymbol{A}$, we have*

1. $(\boldsymbol{A}^+)^* = \boldsymbol{A}^+$
2. $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}$, $\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{A}^+$, *and* $\boldsymbol{A}^+\boldsymbol{A} = \mathbf{1}_{\mathcal{N}^\perp}$
3. $[\boldsymbol{P}, \boldsymbol{A}] = 0$ *where* $[\boldsymbol{A}, \boldsymbol{P}] = \boldsymbol{A}\boldsymbol{P} - \boldsymbol{P}\boldsymbol{A}$
4. *If* $(\mathbf{1} - \boldsymbol{P})\boldsymbol{A}(\mathbf{1} - \boldsymbol{P})$ *is positive definite on* $\mathcal{N}^\perp$, *then* $(\mathbf{1} - \boldsymbol{P})\boldsymbol{A}^+(\mathbf{1} - \boldsymbol{P})$ *is also positive definite on that subspace.*

The CPD Kernel Framework

Figure 2.9 depicts the most common objects for the cpd setting in parallel to Figure 2.3. The structures and interrelations are very similar to the positive definite case, see Section 2.3.1, but a non-empty null space of $\boldsymbol{R}^c$ requires a few changes.

Throughout this section we will assume that the *regularisation operator* $\boldsymbol{R}^c : \mathcal{H} \to \mathcal{G}$ is an arbitrary operator from $\mathcal{H}$ to some linear space $\mathcal{G}$. We do *not* assume that it is one-to-one.

We denote its null space of dimension $0 \leq M \leq N$ as $\mathcal{P}$ and let $\boldsymbol{P}$ be the orthogonal projection from $\mathcal{H}$ to $\mathcal{P}$.

If $\boldsymbol{R}^c$ is not one-to-one, neither is $\boldsymbol{R}^{c*}\boldsymbol{R}^c$, and we cannot define the covariance operator as the inverse of this matrix. Instead, we redefine the *covariance operator* $\boldsymbol{K}^c$ to be a symmetric positive semi-definite matrix, i.e.

$$\boldsymbol{f}^*\boldsymbol{K}^c\boldsymbol{f} \geq 0 \quad \forall \boldsymbol{f} \in \mathcal{H}. \tag{2.23}$$

The covariance operator is then related to the regularisation operator $\boldsymbol{R}^c$ as

$$\boldsymbol{K}^c = (\boldsymbol{R}^{c*}\boldsymbol{R}^c)^+. \tag{2.24}$$

Note that the null space of $\boldsymbol{K}^c$ is also $\mathcal{P}$. The corresponding *Gaussian process* $p_{K^c}(f)$ has the form

$$p_{K^c}(f) = N^U(0, \boldsymbol{K}^c) \propto \exp\left(-\frac{1}{2}\left\|\boldsymbol{R}^c\boldsymbol{f}\right\|^2\right), \tag{2.25}$$

where $N^U(.,.)$ is an unnormalised Gaussian density. If the dimension M of the null space $\mathcal{P}$ is greater than zero, then $p_{K^c}(f)$ cannot be normalised since the density is constant in the directions of $\mathcal{P}$, $\|\boldsymbol{R}^c\boldsymbol{p}\| = 0$ for $\boldsymbol{p} \in \mathcal{P}$. However, an unnormalisable prior may nevertheless be useful and lead to a valid posterior, if the likelihood constrains possible functions f enough.

We define a semi-inner product $(.,.)_{K^c}$ by

$$(\boldsymbol{f}, \boldsymbol{g})_{K^c} = \boldsymbol{f}^T\boldsymbol{R}^{c*}\boldsymbol{R}^c\boldsymbol{g} = \boldsymbol{f}^T\boldsymbol{K}^{c+}\boldsymbol{g}. \tag{2.26}$$

A semi-inner product is an inner product which is also only positive semi-definite, the corresponding semi-norm $\|.\|_{K^c}$ is only positive semi-definite. The tuple $(\mathcal{H}, (.,.)_{K^c})$ then is not a Hilbert space, we follow [Wendland, 2005] and call it a *native space*.

$(\mathcal{H}, (.,.)_{K^c})$ can be converted into an RKHS in two ways: firstly, by restricting the function space to $\left(\mathcal{P}^\perp, (.,.)_{K^c}\right)$. The second alternative is to extend the inner product to $(\boldsymbol{f}, \boldsymbol{g})_S = (\boldsymbol{f}, \boldsymbol{g})_{K^c} + \boldsymbol{f}^*\boldsymbol{P}\boldsymbol{g}$, such that $(\mathcal{H}, (.,.)_S)$ is an RKHS.

When discussing *cpd kernel functions* there are some additional subtleties not encountered in the positive definite case.

Definition 2.14. *A symmetric function $k^c : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ is called conditionally positive definite with respect to the linear space $\mathcal{P} \subseteq \mathcal{H}$, if for all distinct points $x_1, .., x_m \in \mathcal{X}$, $m \leq N$, and all $0 \neq \boldsymbol{\alpha} \in \mathbb{C}^m$ with*

$$\sum_{j=1}^m \alpha_j \overline{p(x_j)} = \sum_{j=1}^m \alpha_j \boldsymbol{p}^*\boldsymbol{\delta}_{x_j} = 0, \quad \forall \boldsymbol{p} \in \mathcal{P} \tag{2.27}$$

we have that

$$\sum_{i=1}^m \sum_{j=1}^m \overline{\alpha_i}\alpha_j k^c(x_i, x_j) = \boldsymbol{\alpha}^*\tilde{\boldsymbol{K}^c}_X\boldsymbol{\alpha} = \left(\sum_{i=1}^m \alpha_i\boldsymbol{\delta}_{x_i}\right)^* \tilde{\boldsymbol{K}^c}\left(\sum_{j=1}^m \alpha_j\boldsymbol{\delta}_{x_j}\right) > 0, \tag{2.28}$$

where $\tilde{\boldsymbol{K}^c}$ is the operator given as $\tilde{\boldsymbol{K}^c}_{ij} = k^c(x_i, x_j)$.

In other words, if $\boldsymbol{f} = \sum_{i=1}^m \alpha_i\boldsymbol{\delta}_{x_i}$, $\boldsymbol{\alpha} \neq 0$, and $\boldsymbol{f}^*\boldsymbol{p} = 0$ $\forall \boldsymbol{p} \in \mathcal{P}$, then $\boldsymbol{f}^*\tilde{\boldsymbol{K}^c}\boldsymbol{f} > 0$. Or equivalent but shorter, $\tilde{\boldsymbol{K}^c}$ is positive definite on $\mathcal{P}^\perp$.

It is important to note, that the operator $\tilde{\boldsymbol{K}}^c$ which is composed form the cpd kernel function values is not necessarily equal to the covariance operator $\boldsymbol{K}_c$, and there exists famous counter examples, e.g. thin-plate spline kernel functions [Wendland, 2005]. The definition of a cpd kernel function with respect to $\mathcal{P}$ just implies that $\tilde{\boldsymbol{K}}^c$ be positive definite on $\mathcal{P}^\perp$, it does *not* make any claim about the behaviour on $\mathcal{P}$. For example, thin-plate spline kernels yield matrices $\tilde{\boldsymbol{K}}^c$ which have $\boldsymbol{f}^* \tilde{\boldsymbol{K}}^c \boldsymbol{f} < 0$ for some $\boldsymbol{f} \in \mathcal{P}$. This contradicts the positive semi-definiteness assumption of the covariance operator $\boldsymbol{K}^c$, which was enforced since surely $\|\boldsymbol{f}\|^2_{K^c} = \|\boldsymbol{R}_c \boldsymbol{f}\|^2 \geq 0$ for all $\boldsymbol{f} \in \mathcal{H}$.

This problem can be circumvented by setting

$$\boldsymbol{K}^c = (\mathbf{1} - \boldsymbol{P}) \tilde{\boldsymbol{K}}^c (\mathbf{1} - \boldsymbol{P}). \tag{2.29}$$

Due to the projection step the assignment of a cpd kernel function to a covariance operator is not unique. If $\{\boldsymbol{p}_i\}_{i=1,..,M}$is an orthonormal basis of $\mathcal{P}$, then eq. (2.29) implies that

$$\begin{aligned} {\boldsymbol{K}^c}_{ij} &= \boldsymbol{\delta}_{x_i}{}^* (\mathbf{1} - \boldsymbol{P}) \tilde{\boldsymbol{K}}^c (\mathbf{1} - \boldsymbol{P}) \boldsymbol{\delta}_{x_j} \\ &= k^c(x_i, x_j) - \sum_l p_l(x_i) \left(\boldsymbol{p}_l{}^* {\tilde{\boldsymbol{K}}^c}_{x_j} \right) \\ &\quad - \sum_m \left({\tilde{\boldsymbol{K}}^c}^*_{x_i} \boldsymbol{p}_m \right) p_m(x_j) + \sum_{l,m} p_l(x_i) \left(\boldsymbol{p}_l{}^* \boldsymbol{p}_m \right) p_m(x_i). \end{aligned} \tag{2.30}$$

Note that above we have made an important assumption that does not in general hold for infinite domains and thus requires a slightly different formalism when extended to this setting. We have assumed that an L_2-type inner product exists in $\mathcal{H}$. While we could restrict the space of functions $\mathcal{H}$ to $L_2(\mathcal{X})$ for infinite domains, this is not natural for our purposes. Since we aim at regularising with $\|\boldsymbol{R}^c \boldsymbol{f}\|$ we only need this expression to be well-defined. We do not need that $\boldsymbol{f}$ itself has a finite L_2 norm, it could be an element of a larger space than $L_2(\mathcal{X})$. For example, using $\mathcal{X} = \mathbb{R}$ and regularising with the first derivative we could include constant functions into $\mathcal{H}$ even though an L_2-type inner product between two linear functions on $\mathbb{R}$ does not exist. While for finite domains it is trivially $\mathcal{H} \subseteq L_2(\mathcal{X})$, [Wendland, 2005] gives an account for more general function spaces $\mathcal{H}$ and infinite domains. Specifically, he uses a slightly different projection for relating the covariance operator with the kernel function in eq. (2.29) and eq. (2.30).

The results of this section are summarised in Table 2.2.

Support Vector Machines

Employing regularisation operators which are not necessarily one-to-one leads to Support Vector Regression (SVR) which is slightly different from the positive definite case. As in Section 2.3.2 Lemma 2.7, we first present a useful decomposition of an arbitrary function in $\mathcal{H}$ and then the representer theorem follows.

Definition 2.15. *A set $X = \{x_i \mid i = 1, .., m\} \subseteq \mathcal{X}$, $m \leq N$, of points is called* unisolvent *with respect to the linear space $\mathcal{P} \subseteq \mathcal{H}$, $\dim(\mathcal{P}) \leq m$, if the only solution for $p(x_i) = 0$ with $\boldsymbol{p} \in \mathcal{P}$, $i = 1, .., m$ is $\boldsymbol{p} = 0$.*

entity	symbol	relations
cpd kernel func.	$k^c : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$	$k^c(x_i, x_j) = \tilde{\boldsymbol{K}^c}_{ij}$
covariance op.	$\boldsymbol{K}^c : \mathcal{H} \to \mathcal{H}$	$\boldsymbol{K}^c = (\mathbf{1} - \boldsymbol{P})\tilde{\boldsymbol{K}^c}(\mathbf{1} - \boldsymbol{P})$ $\boldsymbol{K}^c = (\boldsymbol{R}^{c*}\boldsymbol{R}^c)^+$
native space	$(.,.)_{K^c} : \mathcal{H} \times \mathcal{H} \to \mathbb{C}$ $\|.\|_{K^c} . : \mathcal{H} \to \mathbb{R}$	$(\boldsymbol{f}, \boldsymbol{g})_{K^c} = \boldsymbol{f}^* \boldsymbol{K}^{c+} \boldsymbol{g} = \boldsymbol{f}^* \boldsymbol{R}^{c*} \boldsymbol{R}^c \boldsymbol{g}$ $\|\boldsymbol{f}\|_{K^c} = (\boldsymbol{f}, \boldsymbol{f})_{K^c}^{1/2} = \|\boldsymbol{R}^c \boldsymbol{f}\|$
Gaussian process	$p_{K^c} : \mathcal{H} \to \mathbb{R}$	$p_{K^c}(f) = N^U(0, \boldsymbol{K}^c)$ $p_{K^c}(f) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{f}\|_{K^c}^2\right)$ $p_{K^c}(f) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{R}^c \boldsymbol{f}\|^2\right)$
regularisation op.	$\boldsymbol{R}^c : \mathcal{H} \to \mathcal{G}$	($\boldsymbol{R}^c = \sqrt{\boldsymbol{K}^{c+}}$, not unique)

Table 2.2: *Summary of the objects of the conditionally positive definite kernel framework and their interrelations.*

Lemma 2.16. *Given distinct points $X = \{x_i \mid i = 1, .., m\}$, $m \leq N$, which are unisolvent with respect to $\mathcal{P}$, any $\boldsymbol{f} \in \mathcal{H}$ can be written like*

$$\boldsymbol{f} = \sum_{i=1}^{m} \alpha_i \boldsymbol{K}^c{}_{x_i} + \sum_{j=1}^{M} \beta_j \boldsymbol{p}_j + \boldsymbol{\rho}. \tag{2.31}$$

where $\{\boldsymbol{p}_j\}_{j=1,..,M}$ is a basis of $\mathcal{P}$ and $\boldsymbol{\alpha} \in \mathbb{C}^m$, $\boldsymbol{\beta} \in \mathbb{C}^M$, and $\boldsymbol{\rho} \in \mathcal{H}$ are uniquely determined and satisfy the following conditions

$$\sum_{i=1}^{m} \alpha_i \overline{p_j(x_i)} = \boldsymbol{p}_j{}^* \left(\sum_{i=1}^{m} \alpha_i \boldsymbol{\delta}_{x_i} \right) = 0, \quad j = 1, .., M, \tag{2.32}$$

$$\boldsymbol{\rho}(x_i) = 0, \quad i = 1, .., m \tag{2.33}$$

Furthermore, $\|\boldsymbol{f}\|_{K^c}^2$ can then be written as $\|\boldsymbol{f}\|_{K^c}^2 = \boldsymbol{\alpha}^* \boldsymbol{K}^c{}_X \boldsymbol{\alpha} + \|\boldsymbol{\rho}\|_{K^c}^2$.

Note that condition (2.32) ensures that $\sum_{i=1}^m \alpha_i \boldsymbol{\delta}_{x_i} \in \mathcal{P}^\perp$. Furthermore, it is $\sum_{i=1}^m \alpha_i \boldsymbol{K}^c{}_{x_i} = \boldsymbol{K}^c (\sum_{i=1}^m \alpha_i \boldsymbol{\delta}_{x_i})$, and $\boldsymbol{K}^c{}_{x_i}$ and $\tilde{\boldsymbol{K}}^c{}_{x_i}$ just differ by an element of $\mathcal{P}$. Thus, one could replace $\boldsymbol{K}^c{}_{x_i}$ in eq. (2.31) by $\tilde{\boldsymbol{K}}^c{}_{x_i}$ without changing the expression. Practically that means that we can work directly with the cpd kernel function when performing SVR regression and do not have to use the more complicated expression (2.30) which includes projections.

Proof. The theorem states that $f(x_i) = \sum_{j=1}^m \alpha_j \boldsymbol{K}^c{}_{ji} + \sum_{j=1}^M \beta_j p(x_i)$, $i = 1, .., m$, where

$\sum_{i=1}^m \alpha_i \overline{p_j(x_i)} = 0$, $j = 1, .., M$. In matrix notation this is

$$\boldsymbol{K^c}_{\text{ext}} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} \equiv \begin{pmatrix} \boldsymbol{K^c}_X & \boldsymbol{T} \\ \boldsymbol{T}^* & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_X \\ 0 \end{pmatrix} \tag{2.34}$$

with $\boldsymbol{T} \in \mathbb{C}^{m \times M}$ defined by $\boldsymbol{T}_{ij} = p_j(x_i)$. This system is uniquely solvable for $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ because of the following argument due to [Wendland, 2005, p.117]: Suppose that $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ lies in the null space of $\boldsymbol{K^c}_{\text{ext}}$. Then we have

$$\begin{aligned} \boldsymbol{K^c}_X \boldsymbol{\alpha} + \boldsymbol{T}\boldsymbol{\beta} &= 0, \\ \boldsymbol{T}^* \boldsymbol{\alpha} &= 0. \end{aligned}$$

$\boldsymbol{K^c}_X$ is positive definite for all $\boldsymbol{\alpha}$ that satisfy the second equation. Multiplying the first equation by $\boldsymbol{\alpha}^*$ yields $0 = \boldsymbol{\alpha}^* \boldsymbol{K^c}_X \boldsymbol{\alpha} + (\boldsymbol{T}^* \boldsymbol{\alpha})^* \boldsymbol{\beta} = \boldsymbol{\alpha}^* \boldsymbol{K^c}_X \boldsymbol{\alpha}$. Due to positive definiteness, we can conclude that $\boldsymbol{\alpha} = 0$ and thus $\boldsymbol{T}\boldsymbol{\beta} = 0$. Since X is a unisolvent set of points, this implies $\boldsymbol{\beta} = 0$.

Returning to the inhomogeneous system (2.34) it can be shown [Wahba, 1990] using block matrix inversion theorems that

$$\boldsymbol{\alpha} = (\boldsymbol{K^{c+}}_X - \boldsymbol{K^{c+}}_X \boldsymbol{T} (\boldsymbol{T}^* \boldsymbol{K^{c+}}_X \boldsymbol{T})^+ \boldsymbol{T}^* \boldsymbol{K^{c+}}_X) \boldsymbol{f}_X, \tag{2.35}$$

$$\boldsymbol{\beta} = (\boldsymbol{T}^* \boldsymbol{K^{c+}}_X \boldsymbol{T})^+ \boldsymbol{T}^* \boldsymbol{K^{c+}}_X \boldsymbol{f}_X. \tag{2.36}$$

Finally, set $\boldsymbol{\rho} = \boldsymbol{f} - \sum_{i=1}^m \alpha_i \boldsymbol{K^c}_{x_i} + \sum_{j=1}^M \beta_j \boldsymbol{p}_j$. □

Using this decomposition, the representer theorem for cpd kernels is straight-forward as in the positive definite case.

Theorem 2.17 (Representer Theorem)**.** *Given distinct, unisolvent points* $X = \{x_i \mid i = 1, .., m\} \subseteq \mathcal{X}$*,* $m \leq N$*, and labels* $\{y_i \mid i = 1, .., m\} \subseteq \mathbb{C}$*,* $C \in \mathbb{R}$*, the minimiser of*

$$\|\boldsymbol{f}\|^2_{K^c} + C \; \textit{Loss}\left(\{(x_i, y_i, f(x_i)) | i = 1, \ldots, m\}\right) \tag{2.37}$$

has the form $\boldsymbol{f}_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \sum_{i=1}^m \alpha_i \boldsymbol{K^c}_{x_i} + \sum_{j=1}^M \beta_j \boldsymbol{p}_j$*.* $\boldsymbol{\alpha} \in \mathbb{C}^m$*,* $\boldsymbol{\beta} \in \mathbb{C}^M$ *minimise the expression*

$$\boldsymbol{\alpha}^* \boldsymbol{K^c}_X \boldsymbol{\alpha} + C \; \textit{Loss}\left(\{(x_i, y_i, f_{\alpha,\beta}(x_i)) | i = 1, \ldots, m\}\right). \tag{2.38}$$

subject to the conditions

$$\sum_{i=1}^m \alpha_i \overline{p_j(x_i)} = 0 \quad j = 1, .., M. \tag{2.39}$$

Gaussian Process Inference

The decomposition in Lemma 2.16 is also the key to compute the marginals of an unnormalised GP. As in Section 2.3.3 we will call this the *GP representer theorem* for the conditionally positive definite case.

Theorem 2.18. *For $X \subseteq \mathcal{X}$ unisolvent with respect to $\mathcal{P}$, the marginal distribution $p_{K^c}(\boldsymbol{f}_X) \propto N^U(0, \boldsymbol{M}^+)$ under the joint GP $p_{K^c}(f) \propto N^U(0, \boldsymbol{K}^c)$ is given by*

$$\boldsymbol{M} = \boldsymbol{K}^{c+}_X - \boldsymbol{K}^{c+}_X \boldsymbol{T}(\boldsymbol{T}^* \boldsymbol{K}^{c+}_X \boldsymbol{T})^+ \boldsymbol{T}^* \boldsymbol{K}^{c+}_X \tag{2.40}$$

where $\{\boldsymbol{p}_j\}_{j=1,..,M}$ is a basis of $\mathcal{P}$ and $\boldsymbol{T}_{ij} = p_j(x_i)$.

Proof. By Lemma 2.16 any $\boldsymbol{f} \in \mathcal{H}$ can be written as $\boldsymbol{f} = \sum_{i=1}^m \alpha_i \boldsymbol{K}^c{}_{x_i} + \sum_{j=1}^M \beta_j \boldsymbol{p}_j + \boldsymbol{\rho}$ where $\rho(x_i) = 0,\ i = 1, ..., m$. Therefore $\boldsymbol{\rho}$ is independent of $\boldsymbol{f}_X$. Furthermore with eq. (2.35) it is

$$\begin{aligned} \|f\|^2_{K^c} &= \boldsymbol{\alpha}^* \boldsymbol{K}^c{}_X \boldsymbol{\alpha} + \|\boldsymbol{\rho}\|^2_{K^c} \\ &= \boldsymbol{f}^*_X (\boldsymbol{K}^{c+}_X - \boldsymbol{K}^{c+}_X \boldsymbol{T}(\boldsymbol{T}^* \boldsymbol{K}^{c+}_X \boldsymbol{T})^+ \boldsymbol{T}^* \boldsymbol{K}^{c+}_X) \boldsymbol{f}_X + \|\boldsymbol{\rho}\|^2_{K^c} \\ &= \boldsymbol{f}^*_X \boldsymbol{M} \boldsymbol{f}_X + \|\boldsymbol{\rho}\|^2_{K^c}\,. \end{aligned}$$

From that it follows that

$$\begin{aligned} p(\boldsymbol{f}_X) &\propto \int \exp\left(-\frac{1}{2}\|\boldsymbol{R}^c \boldsymbol{f}\|^2\right) d\boldsymbol{f}_{\mathcal{X}\setminus X} \\ &\propto \exp\left(-\frac{1}{2}\boldsymbol{f}^*_X \boldsymbol{M}^+_X \boldsymbol{f}_X\right) \underbrace{\int \exp\left(-\frac{1}{2}\|\boldsymbol{\rho}\|^2_{K^c}\right) d\boldsymbol{f}_{\mathcal{X}\setminus X}}_{=const} \\ &\propto \exp\left(-\frac{1}{2}\boldsymbol{f}^*_X \boldsymbol{M}^+_X \boldsymbol{f}_X\right) \end{aligned}$$

□

Transitions Between the CPD and the Positive Definite Worlds

Imagine a family of regularisation operators $\boldsymbol{R}_\theta : \mathcal{H} \to \mathcal{G}$ continuously parametrised by $\theta \in U$ where $U \subseteq \mathbb{R}$ is an open neighbourhood of 0. Assume that $\boldsymbol{R}_\theta$ is one-to-one for all θ except for $\theta = 0$. Thus, for $\theta = 0$ we have to use the cpd framework, for $\theta \neq 0$ we should use the positive definite scheme. However, the limit of $\boldsymbol{K}_\theta$ for $0 \neq \theta \to 0$ is not equal to $\boldsymbol{K}^c{}_{\theta=0}$. The limit does not even exist since in the positive definite case the kernel is the inverse of $\boldsymbol{R}^* \boldsymbol{R}$ which diverges for $\theta \to 0$. On the other hand, the Support Vector Regression objective function

$$V(\theta, \boldsymbol{f}) \equiv \|\boldsymbol{R}_\theta \boldsymbol{f}\|^2 + C \text{ Loss}\left(\{(x_i, y_i, f(x_i)) | i = 1, \ldots, m\}\right) \tag{2.41}$$

depends continuously on θ. Thus one might hope that the minimiser also depends continuously on θ.

The following theorem which is novel to our knowledge shows that this apparent problem of continuity can be resolved. It shows especially that, while the kernel is diverging for $\theta \to 0$, the SVR solution for $\theta \neq 0$ converges for $\theta \to 0$, and that the limiting element is equal to the cpd SVR solution for $\theta = 0$.

Theorem 2.19. *Let $\boldsymbol{R}_\theta : \mathcal{H} \to \mathcal{G}$ depend continuously differentiable on $\theta \in U$, $U \in \mathbb{R}^d$ an open neighbourhood of 0 and let $\boldsymbol{R}_\theta$ be one-to-one if and only if $\theta \neq 0$. Let $\mathcal{P}$ be the null space of $\boldsymbol{R}_{\theta=0}$. Furthermore, let $X = \{x_i \mid i = 1, .., m\} \subseteq \mathcal{X}$, $m \leq N$, be a set of distinct*

points unisolvent with respect to $\mathcal{P}$ with corresponding observations $\{y_i \mid i = 1, .., m\} \subseteq \mathbb{C}$. The minimiser

$$\boldsymbol{f}_\theta = \operatorname*{argmin}_{\boldsymbol{f} \in \mathcal{H}} V(\theta, \boldsymbol{f})$$

depends continuously on θ, if Loss $(\{(x_i, y_i, f(x_i))|i = 1, \ldots, m\})$ is strictly convex and twice continuously differentiable with respect to the $f(x_i)$.

Proof. As a first step note that $V(\theta, \boldsymbol{f})$ is strictly convex in $\boldsymbol{f}$ for all $\theta \in U$. Both $\|\boldsymbol{R}_\theta \boldsymbol{f}\|^2$ and Loss $(\{(x_i, y_i, f(x_i))|i = 1, \ldots, m\})$ are convex with respect to $\boldsymbol{f}$ for all θ. If $\theta \neq 0$ then $\|\boldsymbol{R}_\theta \boldsymbol{f}\|^2$ is strictly convex and so is the sum ("strictly convex + convex = strictly convex "). If $\theta = 0$ then $\|\boldsymbol{R}_\theta \boldsymbol{f}\|^2$ is constant in the direction of vectors $\boldsymbol{p} \in \mathcal{P}$. However, for these $\boldsymbol{p}$ at least one of the $p(x_i)$, $i = 1, .., m$, is not equal to zero since X is unisolvent. Thus, the loss term is strictly convex with respect to ϵ where $\boldsymbol{f}_\epsilon = \boldsymbol{f} + \epsilon \boldsymbol{p}$, and so is the whole objective function.

Since $V(\theta, \boldsymbol{f})$ is strictly convex in $\boldsymbol{f}$ and continuously differentiable, the unique minimum for given θ is determined by

$$F(\theta, \boldsymbol{f}) \equiv \frac{\partial}{\partial \boldsymbol{f}} V(\theta, \boldsymbol{f}) = 0.$$

By assumption $F : U \times \mathbb{C}^N \to \mathbb{C}^N$ is continuously differentiable and $\frac{\partial}{\partial \boldsymbol{f}} F(\theta, \boldsymbol{f}) = \frac{\partial^2}{\partial \boldsymbol{f}^2} V(\theta, \boldsymbol{f})$ is invertible since the objective is strictly convex. Using the implicit function theorem [Heuser, 1991] there exists a continuous function $f_\theta : U \to \mathcal{H}$ with $F(\theta, \boldsymbol{f}_\theta) = 0$. □

Given this theorem one could argue that the cpd framework is unnecessary: if the goal is to regularise with a non one-to-one operator $\boldsymbol{R}$ one could just use a slightly perturbed version of $\boldsymbol{R}$ which actually is one-to-one and for which one could use the positive definite framework. The solution of a SVR would then not differ very much from the unperturbed result. However, if $\boldsymbol{R}^* \boldsymbol{R}$ is nearly singular the corresponding covariance operator $\boldsymbol{K} = (\boldsymbol{R}^* \boldsymbol{R})^{-1}$ will have some large values. Computations with such a kernel will then be numerically unstable, and it is better to use the cpd framework instead.

2.8.3 Additional Proofs

In the finite domains, $\mathcal{H}$ with any inner product $(.,.)_S$ is an RKHS, also with the usual L_2 inner product. To see this note that in $\mathbb{R}^N$ all norms are equivalent and $|\delta_{x_i}(f)| = |f(x_i)| \leq \|\boldsymbol{f}\|_1 \leq C \|\boldsymbol{f}\|_S$.

Lemma 2.5. 1. Riesz's theorem.

2. Since the functionals δ_{x_i} are linearly independent, so are their representers $\boldsymbol{S}_{x_i}$. Then for $\boldsymbol{\alpha} \neq 0$ it is $\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j s(x_i, x_j) = \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j (\boldsymbol{S}_{x_i}, \boldsymbol{S}_{x_j})_S = \|\sum_{i=1}^m \alpha_i \boldsymbol{S}_{x_i}\|_S^2 > 0$.

3. Set $\boldsymbol{T}_{ij} = (\boldsymbol{\delta}_{x_i}, \boldsymbol{\delta}_{x_j})_S$. Then for any $\boldsymbol{f} = \sum_i f(x_i) \boldsymbol{\delta}_{x_i}$, $\boldsymbol{g} = \sum_i g(x_i) \boldsymbol{\delta}_{x_i}$, it is $(\boldsymbol{f}, \boldsymbol{g})_S = \sum_{i,j} f(x_i) g(x_j) (\boldsymbol{\delta}_{x_i}, \boldsymbol{\delta}_{x_j})_S = \sum_{i,j} f(x_i) g(x_j) \boldsymbol{T}_{ij} = \boldsymbol{f}^T \boldsymbol{T} \boldsymbol{g}$.

4. Using the reproducing property on $\boldsymbol{\delta}_{x_i}$, $\delta_{ij} = (\boldsymbol{S}_{x_i}, \boldsymbol{\delta}_{x_j})_S = {\boldsymbol{\delta}_{x_i}}^T \boldsymbol{S}\boldsymbol{T}\boldsymbol{\delta}_{x_j}$, and $\delta_{ij} = (\boldsymbol{\delta}_{x_i}, \boldsymbol{S}_{x_j})_S = {\boldsymbol{\delta}_{x_i}}^T \boldsymbol{T}\boldsymbol{S}\boldsymbol{\delta}_{x_j}$ for all $x_i, x_j \in \mathcal{X}$ implies the claim.

5. Since necessarily $\boldsymbol{S} = \boldsymbol{T}^{-1}$ and $\boldsymbol{T}$ uniquely defines the inner product, the last claim follows.

□

Lemma 2.7. $\boldsymbol{f}$ is the sum of a part $\boldsymbol{f}_{\boldsymbol{\alpha}}$ in the span of the $\boldsymbol{K}_{x_i}$, $x_i \in X$, and the $\boldsymbol{K}$-orthogonal complement $\boldsymbol{\rho}$. The orthogonality condition $(\boldsymbol{K}_{x_i}, \boldsymbol{\rho})_K = 0$ implies $\rho(x_i) = 0$. Since $\boldsymbol{K}$ is positive definite, so is the submatrix $\boldsymbol{K}_X$. Therefore the system $\boldsymbol{f}_X = \boldsymbol{K}_X\boldsymbol{\alpha}$ is uniquely solvable for $\boldsymbol{\alpha} \in \mathbb{R}^m$. □

Theorem 2.8. Following Lemma 2.7, and $\boldsymbol{f} \in \mathcal{H}$ can be written as $\boldsymbol{f} = \boldsymbol{f}_{\boldsymbol{\alpha}} + \boldsymbol{\rho}$ with $(\boldsymbol{f}_{\boldsymbol{\alpha}}, \boldsymbol{\rho})_K = 0$. The objective can then be written as

$$\boldsymbol{\alpha}^T \boldsymbol{K}_X \boldsymbol{\alpha} + \|\boldsymbol{\rho}\|_K^2 + C \text{ Loss } (\{(x_i, y_i, f_{\boldsymbol{\alpha}}(x_i)) | i = 1, \ldots, m\})$$

The loss term is independent of $\boldsymbol{\rho}$ because $\rho(x_i) = 0,\ i = 1, .., m$, and thus the objective is minimised for $\boldsymbol{\rho} = 0$. Convexity of the loss and the uniqueness of the map between $\boldsymbol{f}_\alpha$ and $\boldsymbol{\alpha}$, Lemma 2.7, imply that the whole objective here is convex in $\boldsymbol{\alpha}$. Thus, the minimum is unique in this case. □

Chapter 3

Experimental Design for the Identification of Gene Regulatory Networks: Inference in the Sparse Linear Model

Identifying large gene regulatory networks is an important task, where the acquisition of data through perturbation experiments (*e.g.*, gene switches, RNAi, heterozygotes) is expensive. It is thus desirable to use an identification method that effectively incorporates available prior knowledge — such as the sparse connectivity of gene regulatory networks — and that allows to design experiments such that maximal information is gained from each one.

The main contributions of this chapter are twofold. Firstly, we develop a method for consistent inference of network structure, incorporating prior knowledge about sparse connectivity. The algorithm is time efficient and robust to violations of model assumptions. Moreover, we show how to use that network reconstruction algorithm for optimal experimental design, reducing the number of required experiments substantially.

We employ sparse linear models, and show how to perform full Bayesian inference for these. We not only estimate a single maximum likelihood network, but compute a posterior distribution over networks, using a novel variant of the expectation propagation method. The representation of uncertainty enables us to perform effective experimental design in a standard statistical setting: experiments are selected such that the experiments are maximally informative.

Few methods have addressed the design issue so far. Compared to the most well-known one [Tegnér et al., 2003], our method is more transparent, and is shown to perform qualitatively superior. In [Tegnér et al., 2003], hard and unrealistic constraints have to be placed on the network structure for mere computational tractability, while such are not required in our method. We demonstrate reconstruction and optimal experimental design capabilities on tasks generated from realistic nonlinear network simulators.

3.1 Introduction

Retrieving a gene regulatory network from experimental measurements and biological prior knowledge is a central issue in computational biology. The DNA micro-array technique allows to measure expression levels of hundreds of genes in parallel, and many approaches to identify network structure from micro-array experiments have been proposed. Models include dynamical systems based on ordinary differential equations (ODEs) [Yeung et al., 2002; Kholodenko et al., 2002; Tegnér et al., 2003; Sontag et al., 2004; Schmidt et al., 2005], Bayesian networks [Hartemink et al., 2002; Friedman et al., 2000], or Boolean networks [Shmulevich et al., 2002]. We focus on the ODE setting, where one or few expression levels are perturbed by external means, such as RNA interference [Fire et al., 1998], gene toggle switches (plasmids) [Gardner et al., 2000], or using diploid heterozygotes, and the network structure is inferred from changes in the system response. So far only few studies investigate the possibility of designing experiments *actively*. In an active setting, *experimental design* is used to choose an order of perturbations (from a set of feasible candidates) such that maximum novel information about the underlying network is obtained in each experiment. Multi-gene perturbations are becoming increasingly popular, yielding more informative data, and automated data-driven design technologies are required to deal with the combinatorial number of choices which can be opaque even for a human expert.

Identifying (linear) ODE systems from observations and experimental design are well developed within the control community [Ljung, 1999]. However, in the systems biology context, only very few measurements are available compared to the dimension of the system (*i.e.* number of genes), and experiments leading to such observations are severely restricted. Biological measurements are noisy, and time resolution is low, so that in practice only steady states of a system may be accurately measurable. On the other hand, there are no real-time requirements in biological control applications, and more advanced models and analysis can be used. A large body of biological knowledge can be used to counter the small number of observations, for example by specifying a prior distribution within a Bayesian treatment. The standard system identification and experimental design solutions of control theory may therefore not be well-suited for biology.

We propose a full Bayesian framework for network recovery and optimal experimental design. Given many observed genes and rather few noisy measurements, the recovery problem is highly under-determined, and a prior distribution encoding biological knowledge about the connectivity matrix does have a large impact. One of the key assumptions is network sparsity, which holds true for all known regulatory networks. We adopt the linear model frequently used in the ODE setting [Yeung et al., 2002; Kholodenko et al., 2002; Sontag et al., 2004; Peeters and Westra, 2004; Schmidt et al., 2005], but use a sparsity-enforcing prior on the network matrix. The sparse linear model is the basis of the *Lasso* [Tibshirani, 1996], previously applied to the gene network problem in [Peeters and Westra, 2004]. However, they simply estimate the single network maximising the posterior probability from passively acquired data, and do not address experimental design. We closely approximate the Bayesian posterior distribution over connectivity matrices, allowing us to compute established design criteria such as the information gain, which cannot be done using maximum a posteriori (MAP) estimation. The posterior distribution cannot be computed in closed form, and obtaining an accurate approximation efficiently is challenging. We apply a novel variant of the recent expectation propagation algorithm towards this end.

Many other approaches for sparse network recovery have been proposed. In [Yeung et al.,

2002], the space of possible networks (as computed by a singular value decomposition) is scanned for the sparsest solution. A sparse Bayesian model is proposed in [Rogers and Girolami, 2005], see also [Tipping, 2001]. While there is some work on experimental design for boolean networks [Ideker et al., 2000] and Bayesian causal networks [Yoo and Cooper, 2003], none of the above mentioned methods have been used towards this goal. Experimental design remains fairly unexplored in the sparse ODE setting, with the notable exception of [Tegnér et al., 2003]. We compare our approach to theirs, finding our method to perform recovery with significantly less experiments and running much faster. Our method is more robust to observation noise frequently present for biological experiments, and somewhat more transparent and in line with statistical practice. Finally, their method consists of a combinatorial search and is therefore only applicable to networks with uniformly small in-degree, an assumption invalid for many known regulatory networks, *e.g.*[Cokus et al., 2006].

The remainder of the chapter is structured as follows. In Section 3.2, we give an overview of our network reconstruction model and the experimental design method. The key ingredient for both, the novel approximate inference scheme is presented thereafter in Section 3.3. Describing some additional issues that are important to understand the capabilities of our method in Section 3.4, we continue with an extensive experimental evaluation of the proposed approach in Section 3.5. We conclude this chapter in Section 3.6.

3.2 Methodological Overview

3.2.1 Our Model

We start with the common linearised ODE model: expression levels $\boldsymbol{x}(t) \in \mathbb{R}^N$ of N measured genes at time t are modelled by the stochastic dynamical system

$$d\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t))dt - \boldsymbol{u}(t)dt + d\boldsymbol{W}(t). \tag{3.1}$$

Here, $\boldsymbol{f} : \mathbb{R}^N \to \mathbb{R}^N$ describes the nonlinear system dynamics, $\boldsymbol{u}(t)$ is a user-applied disturbance, and $d\boldsymbol{W}(t)$ is white noise. With $\boldsymbol{u}(t) \equiv \boldsymbol{0}$, we assume that the system settles in a steady state, and we linearise the system around that point. In this setting, a perturbation experiment consists of applying a constant disturbance $\boldsymbol{u}(t) \equiv \boldsymbol{u}$ to the system, then measuring the difference $\boldsymbol{x}$ between new and undisturbed steady state. Under the linearity assumption, we have that

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}, \tag{3.2}$$

where $\boldsymbol{A}$ is the *system matrix* with entries a_{ij}, the non-zero a_{ij} describing the gene regulatory network. The noise $\boldsymbol{\epsilon}$ is assumed to be i.i.d. Gaussian with variance σ^2. We focus on steady state differences, as in [Tegnér et al., 2003]. Time course measurements are modelled linearly in [Sontag et al., 2004; Schmidt et al., 2005], and our method can easily be formulated in their setup as well.

We assume that the disturbances $\boldsymbol{u}$ do not drive the system out of the linearity region around the unperturbed steady state. While this seems a fairly strong assumption, our simulation experiments show that effective network recovery is possible even if it is partly violated.

Our contribution to this standard linear regression formulation is a Bayesian model, incorporating prior information about $\boldsymbol{A}$, namely its sparsity. The unknown matrix $\boldsymbol{A}$ is inferred via

a posterior distribution, rather than merely estimated, allowing us to perform experimental design within a statistically optimal framework.

Observations are denoted $\boldsymbol{X} = (\boldsymbol{x}_1 \dots \boldsymbol{x}_m)$, $\boldsymbol{U} = (\boldsymbol{u}_1 \dots \boldsymbol{u}_m)$, and the Bayesian posterior is

$$P(\boldsymbol{A}|\boldsymbol{U}, \boldsymbol{X}) \propto P(\boldsymbol{U}|\boldsymbol{A}, \boldsymbol{X})P(\boldsymbol{A}), \tag{3.3}$$

where the likelihood is $P(\boldsymbol{U}|\boldsymbol{A}, \boldsymbol{X}) = \prod_{j=1}^{m} N(\boldsymbol{u}_j; \boldsymbol{A}\boldsymbol{x}_j, \sigma^2\mathbf{1})$, owing to (3.2). Here, $N(\boldsymbol{u}_j; \boldsymbol{A}\boldsymbol{x}_j, \sigma^2\mathbf{1})$ denotes the multi-variate normal distribution for $\boldsymbol{u}_j$ with mean $\boldsymbol{A}\boldsymbol{x}_j$ and variance $\sigma^2\mathbf{1}$.

Note that typically $m < N$, certainly in early stages of experimental design, and $\boldsymbol{U} = \boldsymbol{A}\boldsymbol{X}$ has no unique solution for $\boldsymbol{A}$. In this situation, the encoding of knowledge in the prior $P(\boldsymbol{A})$ is of large importance. True biological networks are known to be sparsely connected, so we would expect sparse network matrices $\boldsymbol{A}$. The prior should force as many entries of $\boldsymbol{A}$ close to zero as possible, at the expense of allowing for fairly large values of a few components. It should be a *sparsity prior*.

We employ a *Laplace* prior distribution

$$P(\boldsymbol{A}) = \prod_{i,j} P(a_{ij}), \quad P(a_{ij}) = \frac{\tau}{2}\mathrm{e}^{-\tau|a_{ij}|}. \tag{3.4}$$

It is instructive to compare the Laplace against the Gaussian distribution, which is commonly used as prior in the linear model. The Laplace puts much more weight close to zero than the Gaussian, while still having higher probabilities for large values. The implications are depicted in Figure 3.1, see also [Tipping, 2001]. In fact, the Gaussian prior is used with the linear model mostly for convenience, since the posterior is Gaussian again and can be computed easily [O'Hagan, 1994]. Even within our framework, computations with a Gaussian prior are significantly more efficient than with a Laplace. However, our results prove that theoretical arguments in favour of the Laplace prior do have real practical weight, in that the computational advantages with the Gaussian are paid for by a much worse predictive accuracy, and identification needs significantly more measurements than for the Laplace.

The bi-separation characteristic of the Laplace prior into few large and many small parameters (which is not present for the Gaussian) is embodied even more strongly in other sparsity priors, such as "spike-and-slab" (mixture of narrow and wide Gaussian), Student-t, or distributions based on α-norms, $\|x\|_\alpha^\alpha = \sum_i |x_i|^\alpha$, with $\alpha < 1$, see also Figure 3.1. However, among these only the Laplace distribution is log-concave, i.e. has a log-concave density function, leading to a posterior whose log density is a concave function, thus has a single local maximum. This simplifies accurate inference computations significantly. For a non-log-concave prior, posteriors are usually multi-modal, spreading their mass among many isolated bumps, and the inference problem is in general at least as hard as the combinatorial problem of testing all possible sparse graphs. For such posteriors, all known methods for approximate Bayesian inference tend to either perform poorly or require an excessive amount of time. Furthermore, they tend to be algorithmically unstable, and the approximation quality is hard to assess. Robustness of the inference approximation is important for experimental design, since decisions should not be based on numerical instability artefacts of the method, but on the data alone. These points motivate our choice of a Laplace sparsity prior.

Note that the Laplace prior does not imply any strict constraints on the graph structure, *i.e.* the sparsity pattern of $\boldsymbol{A}$, in contrast to other combinatorial approaches which can be run

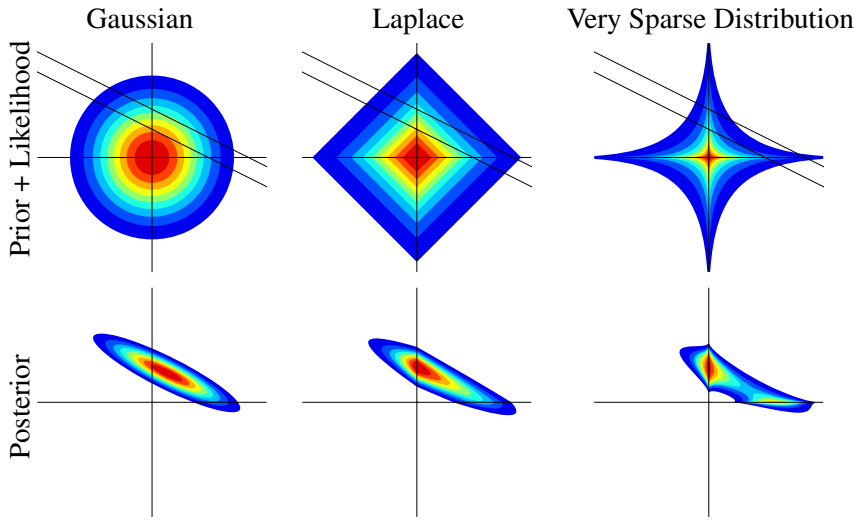


Figure 3.1: *Three prior distribution candidates over network matrix coefficients: Gaussian, Laplace, and "very sparse" distribution ($P(a_{ij}) \propto \exp(-\tau|a_{ij}|^{0.4})$). We show contour plots of density functions over two entries, coloured areas contain the same probability mass for each of the distributions. Upper row: prior distributions (unit variance), and likelihood for single measurement (linear constraint with Gaussian uncertainty). Lower row: corresponding posterior distributions. The Gaussian is spherically distributed, the others shift probability mass towards the axes, giving more mass to sparse tuples (≥ 1 entry close to 0). This effect is clearly visible in the posterior distributions. For the Gaussian prior, the area close to the axes has rather low mass. The Laplace-posterior is skewed: more mass is concentrated close to the vertical axis. Both posteriors are log-concave (and unimodal). The "very sparse"-posterior is shrunk towards the axes more strongly, sparsity is enforced stronger than for the Laplace prior. But it is bimodal, giving two different interpretations for the single observation. This multimodality increases exponentially with the number of dimensions, rendering accurate inference very difficult. The Laplace prior therefore is a good compromise between computational tractability and suitability of the model.*

affordably only after placing hard constraints on the in-degree of all network nodes [Tegnér et al., 2003]. The Laplace prior $P(\boldsymbol{A})$ and the resulting posterior have densities, so that the probability of a matrix $\boldsymbol{A}$ having entries exactly equal to zero vanishes. Sparsity priors with point masses on zero have been used in statistics, but approximate Bayesian inference for such is very hard in general (such priors are certainly not log-concave). We predict discrete network graphs from our posterior as follows. For a small threshold δ_e, we take a_{ij} to represent an edge $i \leftarrow j$ iff $|a_{ij}| > \delta_e$. Moreover, the marginal posterior probability of $\{|a_{ij}| > \delta_e\}$ is used to rank potential edges $i \leftarrow j$.

The posterior for the sparse linear model with Laplace prior does not fall into any standard multivariate distribution family, and it is not known how to do computations with it analytically. On the other hand, experimental design requires at least a good approximation to the posterior, which can be updated efficiently in order to score an experiment. Denote the observations (experiments) obtained so far by D. From (3.3) and (3.4), we see that the posterior factorises w.r.t. rows of $\boldsymbol{A}$, in that

$$P(\boldsymbol{A}|D) = P(D)^{-1} P(D|\boldsymbol{A}) P(\boldsymbol{A}) = \prod_i P(\boldsymbol{A}_{i,\cdot}^T|D),$$

where $\boldsymbol{A}_{i,\cdot}^T$ is the i-th row of $\boldsymbol{A}$. The factors are joint distributions over N variables. We noted above that these factors are log-concave, and thus have a single local maximum and convex upper level sets (see Figure 3.1). These features motivate approximating them by Gaussian factors, so that a posterior approximation is obtained as $Q(\boldsymbol{A}) = \prod_i Q(\boldsymbol{A}_{i,\cdot}^T)$ with multivariate Gaussians $Q(\boldsymbol{A}_{i,\cdot}^T)$. The approximate inference method we use is a novel variant of *expectation propagation* (EP) [Opper and Winther, 2000a; Minka, 2001]. Our approach deals correctly with very underdetermined models ($m \ll N$ in our setup), where previous EP variants would fail due to severe numerical instability. Our framework for computing approximate posterior distributions and its specialisations to the under-determined case are explained in detail in Section 3.3.

3.2.2 Experimental Design

In our setup, an experiment consists of applying a constant disturbance $\boldsymbol{u}$ to the system, then measuring the new steady state. With current technology, such an experiment is expensive and time-consuming, especially if $\boldsymbol{u}$ is to be controlled fairly accurately. The goal of sequential experimental design is to choose the next experiment among a set of candidates (of about the same cost), with the aim of decreasing the uncertainty in $\boldsymbol{A}$ using as *few experiments as possible*. A successful design methodology allows to obtain the same conclusion with less cost and time, compared to doing experiments at random or even following an exhaustive coverage. To this end, an information value score is computed for each candidate, and the maximiser is chosen.

Different costs of experiments can be considered by multiplying the information value score with the costs. However, note that if the costs are extremely different, experiment design is often not necessary since the costs alone determine what should be done next.

A straightforward choice of an information value score is the expected decrease in uncertainty. In general, experimental design thus cannot be done without a representation of uncertainty in $\boldsymbol{A}$, and the Bayesian framework maintains such a representation at its core, namely the posterior. Methods based solely on maximum likelihood or maximum a posteriori estimation (such as Lasso) fail to represent uncertainties.

Denote the current posterior by $Q(\boldsymbol{A}) = Q(\boldsymbol{A}|D)$. If $(\boldsymbol{u}_*, \boldsymbol{x}_*)$ is the outcome of an experiment, let $Q'(\boldsymbol{A}) = Q'(\boldsymbol{A}|D \cup \{(\boldsymbol{u}_*, \boldsymbol{x}_*)\})$ be the posterior including the additional observation. Different information value scores have been proposed for experimental design, see [Chaloner and Verdinelli, 1995] for an overview. A measure for the amount of uncertainty in Q is the differential entropy $\mathrm{E}_Q[-\log Q]$, so a convenient score would be the entropy difference $\mathrm{E}_Q[-\log Q] - \mathrm{E}_{Q'}[-\log Q']$. A related score is the *information gain*

$$S(\boldsymbol{u}_*, \boldsymbol{x}_*|D) = \mathrm{D}[Q' \,\|\, Q] = \mathrm{E}_{Q'}[\log Q' - \log Q],$$

where $\mathrm{D}[Q' \,\|\, Q]$ is the relative entropy (or Kullback-Leibler divergence). $\mathrm{D}[Q' \,\|\, Q]$ is a common measure for the "cost" (in terms of information) of replacing Q' by Q, and the inclusion of a new experiment leads precisely to the replacement $Q \to Q'$. Unlike the entropy difference, the information gain is also sensitive to a shift in the mean of the distribution, so the information gain is well-motivated in our setup.

While scores such as information gain or entropy difference are hard to compute for general distributions Q, Q', this can be done straightforwardly for Gaussians. If $Q(\boldsymbol{a}) =$

$N(\boldsymbol{a}; \boldsymbol{h}, \boldsymbol{\Sigma})$, $Q'(\boldsymbol{a}) = N(\boldsymbol{a}; \boldsymbol{h}', \boldsymbol{\Sigma}')$ and $\boldsymbol{a} = \boldsymbol{A}_{i,\cdot}^T$, the information gain is

$$\frac{1}{2}\Big(\log|\boldsymbol{M}| + \operatorname{tr}\boldsymbol{M}^{-1} - N + (\boldsymbol{h}' - \boldsymbol{h})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{h}' - \boldsymbol{h})\Big), \tag{3.5}$$

with $\boldsymbol{M} = (\boldsymbol{\Sigma}')^{-1}\boldsymbol{\Sigma}$, which can be computed very efficiently in our framework, see Section 3.3.4.

The outcome $(\boldsymbol{u}_*, \boldsymbol{x}_*)$ of an experiment is of course not completely known before it is performed. The central idea of Bayesian sequential design is to compute the distribution over outcomes of the experiment, based on all observations so far, with which to average the score $S(\boldsymbol{u}_*, \boldsymbol{x}_*|D)$. Thus, some experimental candidate e is represented by a distribution $Q_e(\cdot|D)$ over $(\boldsymbol{u}_*, \boldsymbol{x}_*)$. In the setting of this chapter, $\boldsymbol{u}_*$ is completely known, say $\boldsymbol{u}_* = \boldsymbol{u}^{(e)}$ for candidate e, although in an extended setting, e might only specify a distribution over $\boldsymbol{u}_*$. In general, the information value for candidate e is then given as $S(e|D) = \mathrm{E}_{Q_e}[S(\boldsymbol{u}_*, \boldsymbol{x}_*|D)]$. In our setup, it is $Q_e(\boldsymbol{u}_*, \boldsymbol{x}_*|D) = \mathrm{I}_{\{\boldsymbol{u}_* = \boldsymbol{u}^{(e)}\}} Q(\boldsymbol{x}_*|D, \boldsymbol{u}_*)$ and we obtain

$$S(\boldsymbol{u}^{(e)}|D) = S(\boldsymbol{u}_*|D) = \mathrm{E}_{Q(\boldsymbol{x}_*|D,\boldsymbol{u}_*)}[\mathrm{D}[Q' \parallel Q]].$$

The expectation above can be computed easily via sampling: We first draw $\boldsymbol{A} \sim Q(\boldsymbol{A}|D)$, and then $\boldsymbol{x}_* = \boldsymbol{A}^{-1}(\boldsymbol{u}_* - \boldsymbol{\epsilon}_*)$, $\boldsymbol{\epsilon}_* \sim N(\boldsymbol{\epsilon}_*; \boldsymbol{0}, \sigma^2\boldsymbol{1})$.

3.3 Approximate Bayesian Inference

In the setup described above, network reconstruction requires the marginal distributions of the posterior, experimental design additionally the information gain between two consecutive posteriors. Since the posterior distribution factors with respect to the rows of $\boldsymbol{A}$, the problem can be decomposed, and it is enough to compute these quantities for any row $\boldsymbol{a} = \boldsymbol{A}_{i,\cdot}$ separately. However, the remaining task is still difficult. The posterior distribution for each row $P(\boldsymbol{a}|D) \propto N(\boldsymbol{U}_{i,\cdot}; \boldsymbol{X}^T\boldsymbol{a}, \sigma^2\boldsymbol{1}) \prod_j t_j(a_j)$ with *sites* $t_j(a_j) = \exp(-\tau|a_j|)$ does not fall into an analytically tractable family of distributions and, thus, the marginals and the information gain have to be computed via numerical integration which is infeasible for N-dimensional integrals, $N \gg 1$.

The idea of approximate Bayesian inference to solve this problem is to approximate the posterior with an element of a simpler, tractable family of distributions, for which the marginals and the information gain can then be computed analytically. Since the logarithm of the posterior density is concave in our setup implying that the posterior is unimodal, we choose the Gaussian distributions here. The goal is thus to find that Gaussian $Q(\boldsymbol{a}) = N(\boldsymbol{a}; \mu, \Sigma)$ for which the Kullback-Leibler (KL) divergence to the true posterior $\mathrm{D}(P(\boldsymbol{a}|D)\|Q(\boldsymbol{a}))$ is minimised, see Figure 3.2.

At first, this approximation problem in terms of the KL divergence looks easy since it can be solved analytically. The optimal values for μ and Σ are just the mean and the covariance of the true posterior, that is, minimising the KL divergence is equivalent to moment matching. However, computing such moments for the posterior requires again the computation of high-dimensional, not analytically tractable integrals, rendering the approximation no less complicated than the original problem of directly computing the marginals and the information gain for the posterior.

But note that unlike arbitrary posterior distributions, the posterior $P(\boldsymbol{a}|D)$ has a special form in our setup. It consists of one *global*, "simple" Gaussian distribution

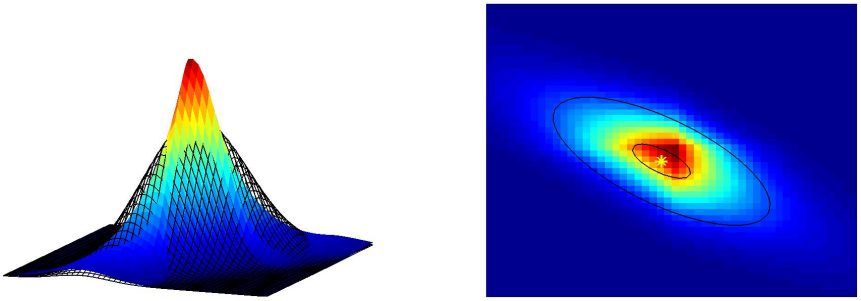


Figure 3.2: *A 3D and a contour plot of an example two-dimensional posterior distribution $P(\boldsymbol{a}|D)$ (colour-coded) and an approximating Gaussian $Q(\boldsymbol{a})$ (black) which is optimally close to $P(\boldsymbol{a}|D)$ with respect to the KL divergence. The yellow star in the right figure denotes the mean of the posterior $P(\boldsymbol{a}|D)$.*

$N(\boldsymbol{U}_{i,:}; \boldsymbol{X}^T\boldsymbol{a}, \sigma^2\mathbf{1})$, which couples all components of $\boldsymbol{a}$, and many *local* sites $t_j(a_j)$, which just depend on a single component. We will show in the following that this characteristic is the basis for the EP algorithm which splits the one high-dimensional approximation problem into a series of smaller one-dimensional sub-problems, that can be solved in an efficient and numerically robust way.

In the following, we give a derivation of EP that is tailored to our setup at hand. The focus is on conveying the important steps and their plausibility, full algorithmic and implementation details are given in [Seeger et al., 2006, 2007; Seeger, 2008]. EP was originally introduced in [Minka, 2001; Opper and Winther, 2000b], a good general overview is given in [Seeger, 2005]. Before describing EP, however, we first review some relevant facts about Gaussian distributions.

3.3.1 Some Facts about Gaussian Distributions

Gaussian distributions can be parametrised in two ways. Classically, they are defined via their so-called *mean parameters*,

$$N(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})).$$

Another way to represent the same distribution is via the *natural parameters*,

$$N'(\boldsymbol{x}; \boldsymbol{b}, \boldsymbol{\Pi}) \propto \exp(-\frac{1}{2}\operatorname{tr}(\boldsymbol{\Pi}\boldsymbol{x}\boldsymbol{x}^T) + \boldsymbol{b}^T\boldsymbol{x}).$$

The two sets of parameters can be converted into each other via the identities $\boldsymbol{b} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and $\boldsymbol{\Pi} = \boldsymbol{\Sigma}^{-1}$.

The representation via the natural parameters is especially useful when multiplying and dividing Gaussian distributions. Since the exponent is linear in the natural parameters, these operations amount to simply adding or subtracting the respective natural parameters. This concept of linearity of the exponent with respect to the parameters is the defining property for the so-called *exponential families*, *e.g.*[Seeger, 2005; Canu and Smola, 2006], the Gaussian distributions being just one example thereof.

The usefulness of having available both representations for Gaussians becomes even more obvious when observing their "dual" behaviour under conditioning and marginalisation: if $\boldsymbol{x} \sim N(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = N'(\boldsymbol{x}; \boldsymbol{b}, \boldsymbol{\Pi})$ and if we split $\boldsymbol{x}$ and the corresponding parameter vectors and matrices like $\boldsymbol{x} = (\boldsymbol{x}_1^T \boldsymbol{x}_2^T)^T$, then we have,

$$\begin{aligned} \text{conditioning,} \quad p(\boldsymbol{x}_1|\boldsymbol{x}_2) &= N(\boldsymbol{x}_1; \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}) \\ &= N'(\boldsymbol{x}_1; \boldsymbol{b}_1 + \boldsymbol{\Pi}_{12}\boldsymbol{x}_2, \boldsymbol{\Pi}_{11}), \qquad (3.6) \\ \text{marginalisation,} \quad p(\boldsymbol{x}_1) &= N(\boldsymbol{x}_1; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) \\ &= N'(\boldsymbol{x}_1; \boldsymbol{b}_1 + \boldsymbol{\Pi}_{12}\boldsymbol{\Pi}_{22}^{-1}\boldsymbol{b}_2, \boldsymbol{\Sigma}_{11} - \boldsymbol{\Pi}_{12}\boldsymbol{\Pi}_{22}^{-1}\boldsymbol{\Pi}_{21}). \qquad (3.7) \end{aligned}$$

Thus, in the mean parameters marginalisation is trivial, but conditioning involves matrix inversion, whereas for the natural parameters the roles are exchanged.

3.3.2 The Idea of Expectation Propagation

Our derivation of EP is based on a decomposition of the global, intractable KL divergence into smaller, local parts which can actually be computed. By combining the resulting local terms in an appropriate iterative algorithm, we can then efficiently compute that Gaussian distribution that approximately minimises the KL divergence to the true posterior.

Proposition 3.1. *For any probability densities $p(\boldsymbol{a})$, $q(\boldsymbol{a})$, $\boldsymbol{a} \in \mathbb{R}^N$, and local terms $t(a_i)$, it is*

$$\mathrm{D}(p(\boldsymbol{a})t(a_i) \,\|\, q(\boldsymbol{a})) = \mathrm{D}(p(a_i)t(a_i) \,\|\, q(a_i)) + \mathbb{E}_{a_i \sim p(a_i)t(a_i)} \left[\mathrm{D}(p(\boldsymbol{a}_{\setminus i}|a_i) \,\|\, q(\boldsymbol{a}_{\setminus i}|a_i))\right], \qquad (3.8)$$

where $\boldsymbol{a}_{\setminus i}$ denotes all components of $\boldsymbol{a}$ except a_i.

Proof.

$$\begin{aligned} \mathrm{D}(p(\boldsymbol{a})t(a_i) \,\|\, q(\boldsymbol{a})) &= \int p(\boldsymbol{a})t(a_i) \log \frac{p(\boldsymbol{a})t(a_i)}{q(\boldsymbol{a})} d\boldsymbol{a} \\ &= \int p(a_i)p(\boldsymbol{a}_{\setminus i}|a_i)t(a_i) \left[\log \frac{p(a_i)t(a_i)}{q(a_i)} + \log \frac{p(\boldsymbol{a}_{\setminus i}|a_i)}{q(\boldsymbol{a}_{\setminus i}|a_i)}\right] d\boldsymbol{a}_{\setminus i} da_i \\ &= \mathrm{D}(p(a_i)t(a_i) \,\|\, q(a_i)) + \int p(a_i)t(a_i)\mathrm{D}(p(\boldsymbol{a}_{\setminus i}|a_i) \,\|\, q(\boldsymbol{a}_{\setminus i}|a_i)) da_i. \square \end{aligned}$$

For distributions with a certain local/global structure the proposition, Proposition 3.1 allows us to split the global KL divergence between two N-dimensional distributions into a divergence between the one-dimensional marginal distributions and an expression for the $(N-1)$-dimensional conditionals. Applying the proposition to the posterior $P(\boldsymbol{a}|D)$ then suggests the following iterative procedure for approximating the posterior with the Gaussian $Q(\boldsymbol{a})$ with minimal KL divergence: We start with $Q^{(0)} = N(\boldsymbol{U}_{i,\cdot}; \boldsymbol{X}^T\boldsymbol{a}, \sigma^2\boldsymbol{1})$, then for each site $t(a_i)$ we update

$$Q^{(i)} = \underset{Q \text{ Gaussian}}{\operatorname{argmin}} \; \mathrm{D}(Q^{(i-1)}(\boldsymbol{a})t(a_i) \,\|\, Q(\boldsymbol{a})). \qquad (3.9)$$

This first algorithm is known as assumed density filtering [Kushner and Budhiraja, 2000]. Note that minimising the KL divergence iteratively is *not* equivalent to globally searching

the best approximating Gaussian w.r.t. the KL divergence in one step. The decomposition into several consecutive, local steps is only an approximation, since one approximation is built onto the other.

Moreover, note that each minimisation of type (3.9) can be computed efficiently considering only one-dimensional integrals. This is because the second term on the right hand side of (3.8) vanishes, if the conditional distribution of Q matches that of $Q^{(i-1)}$, and the first term is minimised if the (one-dimensional) moments of the marginal $Q(a_i)$ match the moments of $Q^{(i-1)}(a_i)t(a_i)$. The iterative procedure thus reduces the computation of one high-dimensional integral into a series of one-dimensional integrals. Given that the requirements for multi-dimensional numerical integration scale approximately exponential in the number of dimensions of the integral, this linear time iterative approach is the key to reducing an infeasible problem into one, which can actually be solved. Note that in our setup where the sites $t(a_i)$ have exponential form the necessary one-dimensional integrals can even be solved analytically, allowing for a very efficient implementation, see [Seeger et al., 2006].

The fact that only the marginal $Q(a_i)$ changes in each update, but the conditional distribution of the approximate posterior $Q(\boldsymbol{a})$ stays the same suggests to use a representation for $Q(\boldsymbol{a})$, which allows for an efficient implementation of these steps. In the last section, we have shown that accessing the conditional distribution of a Gaussian distribution represented in its natural parameters is trivial. We therefore parametrise $Q(\boldsymbol{a})$ as,

$$Q(\boldsymbol{a}) = Q^{(0)}(\boldsymbol{a}) \prod_{j=1}^{N} \tilde{t}(a_j), \tag{3.10}$$

where $\tilde{t}(a_j) = N'(a_j; b_j, \pi_j)$. This parametrisation only has $2N$ free *site parameters* b_j, π_j, not $N(N+1)$ which would be required for an arbitrary Gaussian distribution. Nevertheless, this form can describe each minimiser of (3.9) exactly, since in each update the conditional $Q(\boldsymbol{a}_{\setminus i}|a_i)$ stays constant for all a_i implying that only the parameters b_i, π_i need to be adapted when the marginal distribution is changing, see (3.6). This means that the only approximation towards computing the global KL divergence in assumed density filtering is the split into an iterative setting, but that the representation does not pose any additional limitations. Moreover, this also shows that the algorithm is highly efficient since, while each update step requires a certain computational effort for computing the marginal distribution $Q^{(i-1)}(a_i)$, see (3.7), the parameter updates are local.

The inclusion of one site $t(a_j)$ after the other is strongly reminiscent of the Bayesian inclusion of evidence, *i.e.* likelihood terms, into the posterior. The conceptual difference is, that we here start from the likelihood and add one term after the other of the prior. Algorithmically, however, this does not make a difference. Furthermore, note that assumed density filtering (3.9) is not equivalent to simply approximating all the site $t(a_i)$ with the best fitting one-dimensional Gaussian $\tilde{t}(a_i)$. In each update, all the previous information is taken into account through the use of the previous marginal distribution $Q^{(i-1)}(a_i)$. Also, each update has a non-trivial effect on all other marginals, not just the marginal of index i.

Assumed density filtering (3.9) may, however, lead to rather disappointing approximations of the true posterior. While we start with an exact term $Q^{(0)}$, we afterwards build one approximation onto the other, thereby accumulating small errors in each approximation. This can be avoided through the following trick which leads to the final EP algorithm: we keep

including terms until the approximation $Q^{(i)}$ converges; however, since we cannot include sites $t(a_i)$ twice, we have to divide out the corresponding contributions $\tilde{t}(a_i)$ before including $t(a_i)$ for the second time. In detail, we define the *cavity distributions*

$$Q_c^{(i-1)}(\boldsymbol{a}) = Q^{(i-1)}(\boldsymbol{a})\tilde{t}(a_j)^{-1}$$

and set

$$Q^{(i)} = \operatorname*{argmin}_{Q \text{ Gaussian}} \mathrm{D}(Q_c^{(i-1)}(\boldsymbol{a})t(a_i) \,\|\, Q(\boldsymbol{a})),$$

which can be iterated through all indices $i = 1, .., N$ in arbitrary order until convergence. As above each update here only requires to update two site parameters b_i, π_i. But now, errors that are made in one of the early site inclusions can be corrected for later on, if the remaining sites provided helpful information.

The exact conditions under which EP converges are so far not known, and it is also not known how far the result of this iterative procedure deviates from the minimiser of the true, global KL divergence. However, $P(\boldsymbol{a}|D)$ being unimodal suggests that approximating it with a Gaussian distribution will be well-behaved. This is what we observed in all of our experiments, where EP always converged within few iterations.

3.3.3 Special Adaptations

In the under-determined case $m < N$ that we are principally interested in here, the standard application of EP fails. In this case $Q^{(0)}(\boldsymbol{a}) = N(\boldsymbol{U}_{i,:}; \boldsymbol{X}^T\boldsymbol{a}, \sigma^2\mathbf{1})$ cannot be normalised, and only the sites $\tilde{t}(a_j)$ ensure finite variances of the approximate posterior $Q^{(i)}(\boldsymbol{a})$. If these factors are divided out to obtain the cavity distributions $Q_c^{(i-1)}(\boldsymbol{a})$, the resulting unnormalisable Gaussians cause numerical problems during the marginal moment matching.

Therefore, we propose to use a variant of fractional or Power EP [Minka, 2004]. The idea is to split the sites $t(a_i)$ into several identical copies $t'(a_i) = (t(a_i))^{1/q}$, $q = 2, 3, ..$, and include them separately into the posterior. This guarantees that when dividing out a term $\tilde{t}'(a_i)$, another copy of the same will keep the variances of the cavity distribution finite. In order not to end up with too many parameters, we couple the parameters b_i, π_i for all copies of the same site.

Note that technically this constitutes a different approximation of the global KL divergence for each q. However, we did not experimentally observe significant differences for different $q \geq 2$.

3.3.4 Efficient Scoring of Candidates

Returning to experimental design, the information gain score $S(\boldsymbol{u}_*, \boldsymbol{x}_*|D)$ for an experimental outcome $(\boldsymbol{u}_*, \boldsymbol{x}_*)$ is $\mathrm{D}[Q' \,\|\, Q]$, where $Q' = Q(\boldsymbol{A}|D \cup \{(\boldsymbol{u}_*, \boldsymbol{x}_*)\})$ and $Q = Q(\boldsymbol{A}|D)$. Note that two things happen in $Q \to Q'$. Firstly, $(\boldsymbol{u}_*, \boldsymbol{x}_*)$ is included, which modifies the Gaussian coupling factor in Q. Secondly, all site parameters b_i, π_i are updated by EP. For the purpose of scoring, early trials showed that the second step can be skipped in scoring without much loss in performance. Doing so, we see that $\boldsymbol{M}$ in equation (3.5) has the form $\mathbf{1} + \boldsymbol{x}_*\boldsymbol{u}_*^T$, and $S(\boldsymbol{u}_*, \boldsymbol{x}_*|D)$ can be computed very efficiently using a rank one matrix update in our representation of $Q(\boldsymbol{a})$. For more details see [Seeger et al., 2006].

3.3.5 Running Time

The running time for a naive implementation of our method (Laplace prior, experimental design) is $O(N^5)$, if N experiments are done. Namely, after each experiment, we need to update N posterior representations, one for each row of $\boldsymbol{A}$. For each, we require at least N EP updates, one at each Laplace site, and each such update costs $O(N^2)$ for computing the marginal distribution $Q(a_i)$ (at least once m, the number of experiments so far, is close to N).

This scaling behaviour can be improved by noting that especially during later stages, it will not be necessary to do EP updates for all N^2 sites after each new experiment. For a row $\boldsymbol{a}$, we can compute the change in marginal moments of each $Q(a_i)$ upon including the new observation into the likelihood $P^{(0)}$ only. We then do EP updates for $O(1)$ sites only, namely the ones with most significantly changed marginals. This cuts the scaling to $O(N^4)$.

This concludes the current outline of the EP algorithm. The full algorithmic details are given in [Seeger et al., 2006], our implementation is available at `http://www.kyb.tuebingen.mpg.de/sparselinearmodel/`.

3.4 Further Topics

We continue with discussing some more aspects of how the formal model relates to the biological problem setting.

3.4.1 Unobserved Variables

We have so far focused on modelling mRNA levels, which can be measured easily and cost-effectively. However, protein and metabolite concentrations also play important roles in any regulatory pathway, and a concise ODE explanation of a system cannot be formulated if they are ignored. In this section, we discuss how the unobserved elements of the network influence our network inference, showing that our method allows to identify *effective networks* between the genes.

For simplicity, we will term all unobserved quantities as proteins in this section. Denote the observed mRNA concentrations by $\boldsymbol{x}(t) \in \mathbb{R}^N$ as before, unobserved protein concentrations by $\boldsymbol{y}(t) \in \mathbb{R}^M$. Furthermore, let $\boldsymbol{u}(t) \in \mathbb{R}^N$ be a perturbation vector, which does not affect the proteins. The biological system would now realistically be described by a joint (nonlinear) ODE system for $(\boldsymbol{x}, \boldsymbol{y})$, which we can again linearise around its steady state. If time constant perturbations are used, the difference between new and old steady state follows again a linear equation (up to noise),

$$\begin{pmatrix} \boldsymbol{u} \\ 0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix}.$$

From this, we deduct $\boldsymbol{u} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})\boldsymbol{x}$. Thus, given only the $\boldsymbol{u}$ and $\boldsymbol{x}$ our algorithm will not recover $\boldsymbol{A}$, but $\tilde{\boldsymbol{A}} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$.

We show that $\tilde{\boldsymbol{A}}$ encodes an *effective* gene network in the following sense. If $\tilde{\boldsymbol{A}}_{ij} \neq 0$, then there exists either a direct link from gene j to gene i or there is a path from gene j to gene i which also passes through some proteins in the full gene regulatory network, but not

through other observed genes. This is logically equivalent to the statement, that if there is no such path from j to i, then $\tilde{\boldsymbol{A}}_{ij} = 0$. However, $\tilde{\boldsymbol{A}}_{ij} = 0$ does *not* imply that there is no (indirect) connection between i and j. It could be for example that two protein pathways from j to i are equally strong, but of opposite influence on gene i, and thus cancel each other.

To prove that $\tilde{\boldsymbol{A}}$ encodes such an effective network, we first need the following lemma.

Lemma 3.2. *Let $\boldsymbol{W} \in \mathbb{R}^{n,n}$ be the weighted adjacency matrix of a directed graph, in that $i \leftarrow j$ has weight w_{ij}, and the edge is present iff $w_{ij} \neq 0$. Assume that $\boldsymbol{W}$ is nonsingular. The following holds: if $(\boldsymbol{W}^{-1})_{ij} \neq 0$, then there exists some directed path $j \rightarrow i$.*

Proof. We prove the logical converse. For $i = j$, there is always a path of length 0 from i to i, so the lemma makes no statement. For $i \neq j$, assume that there is no directed path from j to i. Let J be the set of all nodes reachable by j (note that $j \in J$), and let I be its complement. $i \in I$ by our assumption. Without loss of generality, assume that $J = \{1, \ldots, |J|\}$, noting that this can always be achieved by renaming nodes, without changing the network. Now,

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}_J & \boldsymbol{W}_{J,I} \\ \boldsymbol{0} & \boldsymbol{W}_I \end{pmatrix}.$$

If $\boldsymbol{W}_{I,J}$ was not zero, there would be some element in I reachable from J, therefore from j, so $I \cap J \neq \emptyset$, a contradiction. From the special form of $\boldsymbol{W}$ we have that $|\boldsymbol{W}| = |\boldsymbol{W}_J||\boldsymbol{W}_I|$, so that both $\boldsymbol{W}_J$, $\boldsymbol{W}_I$ are nonsingular. Now,

$$\boldsymbol{W}^{-1} = \begin{pmatrix} \boldsymbol{W}_J^{-1} & \boldsymbol{R} \\ \boldsymbol{0} & \boldsymbol{W}_I^{-1} \end{pmatrix},$$

with $\boldsymbol{R} = -\boldsymbol{W}_J^{-1}\boldsymbol{W}_{J,I}\boldsymbol{W}_I^{-1}$. This proves the lemma. □

Back to the effective gene network, we have that $\tilde{\boldsymbol{A}}_{ij} = \boldsymbol{A}_{ij} - \sum_{k,l} \boldsymbol{B}_{ik}(\boldsymbol{D}^{-1})_{kl}\boldsymbol{C}_{lj}$. Suppose there is no path from $j \rightarrow i$ passing through ≥ 0 proteins only in the full network. Then, $\boldsymbol{A}_{ij} = 0$ (no direct gene-gene link). Furthermore, $\boldsymbol{B}_{ik}(\boldsymbol{D}^{-1})_{kl}\boldsymbol{C}_{lj} \neq 0$ for some k, l would mean a path from gene j to protein l, then to protein k via potentially other proteins (apply the lemma above with $\boldsymbol{W} = \boldsymbol{D}$), then to gene i. Therefore, all terms in the sum are zero, and $\tilde{\boldsymbol{A}}_{ij} = 0$.

The fact that our reconstruction methods thus can recover a meaningful effective network in the presence of hidden variables is reassuring, since all regulatory networks between genes are nothing else but effective networks of larger partially unobserved systems. Note, however, that the knowledge of $\tilde{\boldsymbol{A}}$ does not uniquely determine $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, or $\boldsymbol{D}$, or in fact even the number M of unobserved variables.

3.4.2 Incorporating Additional Biological Prior Knowledge

In our method presented so far, we assumed that nothing is known about the network, apart from it being sparse. However, much biological prior knowledge about the (effective) regulatory network may already be available before any experiments are done. In this section, we show how some types of such prior knowledge can be incorporated into our method, if it can be formulated in terms of the system matrix $\boldsymbol{A}$. This will generally help to obtain a faster and more accurate identification of the network.

In general, our method can be extended by using additional *sites* beyond the $t_j(a_{ij}) = \frac{\tau}{2}e^{-\tau|a_{ij}|}$ coming from the Laplace prior. Such sites must have the form $f(\boldsymbol{w}^T \boldsymbol{A}_{i,\cdot}^T)$, where $\boldsymbol{w} \in \mathbb{R}^N$ and $f(\cdot)$ is log-concave.

First, suppose that mRNA degradation rates for some genes are roughly known from independent experiments, say r_i for gene i. We could either fix $a_{ii} = -r_i$ and eliminate this variable, or we could use the factor

$$P(a_{ii}) = \frac{\tau}{2}e^{-\tau|a_{ii}+r_i|}$$

with smaller τ than usual, which would allow for errors in the knowledge of r_i. Using such off-centre factors is of course possible in our framework with very minor changes.

Next, suppose that partial connectivity knowledge is available. For example, if there is no influence $j \to i$, then $a_{ij} = 0$, and the corresponding variable can simply be eliminated. If it is known that $j \to i$ is an activating influence, this means that $a_{ij} > \epsilon$ for some $\epsilon \geq 0$. We can incorporate a site $\mathrm{I}_{\{a_{ij}>\epsilon\}}$ into our method, noting that this is log-concave as an indicator function of a convex set (ϵ, ∞). A better option is to assume that $a_{ij} - \epsilon$ has an exponential prior distribution, which also gives rise to a log-concave site.

3.5 Experiments

In the literature, there are some small networks with known dynamics, *e.g.* the Drosophila segment polarity network [von Dassow et al., 2000]. However, a thorough evaluation of our method requires significantly larger systems for which the dynamics are known, so that disturbance experiments can be simulated, and the predictions of our method can be verified. We are not aware of such models having been established for real biological networks yet, the DREAM project [DREAM, 2006] aims at providing such data in the future. We therefore concentrate on realistic "in-silico" models, applying our method to many randomly generated instances with different structures and dynamics in order to obtain a robust evaluation and comparison.

We simulate the whole network identification process. First, we generate a biologically inspired ground-truth network together with parameters for a numerical simulator of nonlinear dynamics. We feed our method with a number of candidate perturbations $\{\boldsymbol{u}_*\}$, among which it can choose the experiments to be done. If some $\boldsymbol{u}_*$ is selected, the corresponding $\boldsymbol{x}_*$ is obtained from the simulator, and $(\boldsymbol{u}_*, \boldsymbol{x}_*)$ is included into the posterior as new observation. We score the current posterior $Q(\boldsymbol{A})$ against the true network after each inclusion, comparing our method against variants in different settings. Free hyperparameters (τ, σ^2) are selected individually for each of the methods to be compared. We also compare against the experimental design method proposed in [Tegnér et al., 2003], and finally show results on the real, but small Drosophila segment polarity network [von Dassow et al., 2000].

3.5.1 Network Simulation

Common computational models of sparse regulatory networks often build on the *scale-free* or the *small-world* assumption [Watts and Strogatz, 1998]. In small world networks the average path length is much shorter than in a uniform random network. We sample such small-world networks with $N = 50$ nodes (unless otherwise said), see Figure 3.3 for an

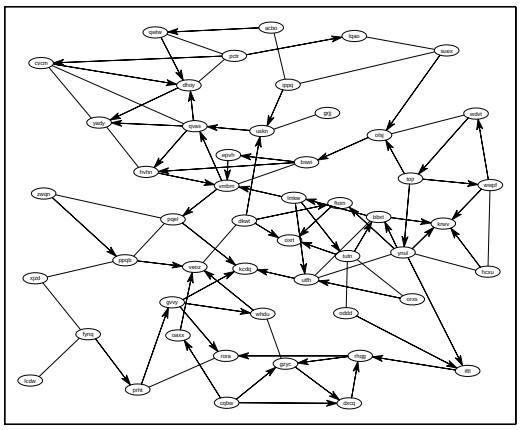


Figure 3.3: *Small-world network of $N = 50$ nodes. Arrowless edges are bi-directional. "Gene names" are randomly drawn. Some nodes have rather high in-degree, characteristic of real biological networks, e.g.[Cokus et al., 2006].*

example. Further details about network generation and properties are given in Additional Material 3.7.1.

For a given network structure, we sample plausible interaction dynamics using Hill-type kinetics, inspired by the model in [Kholodenko et al., 2002]. The nonlinear function in (3.1) is

$$f_i(\boldsymbol{x}) = -V_{di}\frac{x_i}{d_i + x_i} + V_{si}\prod_{j\in\mathcal{A}_i}\frac{1 + A_{ij}\left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}{1 + \left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}}\prod_{j\in\mathcal{I}_i}\frac{1}{1 + \left(\frac{x_j}{\kappa_{ij}}\right)^{n_{ij}}},$$

where $\mathcal{A}_i$ ($\mathcal{I}_i$) are the activating (inhibitory) parents of gene i. The parameters in (3.11) and the way they are randomly sampled are described in Additional Material 3.7.2. Proposed system equations are subject to the condition, that the model produces dynamics with a reasonable stable steady state.

Each observation $(\boldsymbol{u}, \boldsymbol{x})$ consists of a constant disturbance $\boldsymbol{u}$ and its effect $\boldsymbol{x}$, being the difference between a new (perturbed) and the old (unperturbed) steady state. Disturbance candidates were restricted to a small number r of non-zero entries, since experimental techniques for disturbing many genes in parallel by tightly controlled amounts are not yet available. All non-zero u_j are in $\{\pm\nu\}$, where the sign is random, so $\|\boldsymbol{u}\|$ is the same for all $\boldsymbol{u}$. We measure $\|\boldsymbol{u}\|$ in units given by the average relative change in steady state when such disturbances $\boldsymbol{u}$ are applied. We use a pool of 200 randomly generated candidates. The SDE simulator can be used with different levels of noise, measured in terms of the signal-to-noise ratio (SNR), *i.e.* the ratio of $\|\boldsymbol{u}\|$ and the standard deviation of the resulting $\boldsymbol{\epsilon}$ in (3.2).

All results are averaged over 100 runs with independently drawn networks. In the comparative plots presented below, the different methods all see the same data in each run.

3.5.2 Evaluation Criterion

The output from a regulatory network identification method most relevant to a practitioner is a ranking of all possible links, ordered by the probability that they are true edges. With this in mind, we choose the following evaluation score, based on ROC analysis.

At any time, our method provides a posterior $Q(\boldsymbol{A})$, of which at present we only use the marginal distributions $Q(a_{ij})$. We produce a ranking of the edges according to the posterior probabilities $Q(\{|a_{ij}| > \delta_e\})$, where $\delta_e = 0.1$ in all experiments. δ_e was calibrated against average component sizes $|a_{ij}|$, which are roughly given through the dominant time scales in the dynamical system. The predicted rankings are robust against moderate changes of δ_e.

In a standard ROC analysis, the true positive rate (TPR) is plotted as a function of the false positive rate (FPR), and the area under this curve (AUC) is measured. This is not useful in our setting, because only very small FPRs are acceptable at all (there are N^2 potential edges). Our *iAUC* score is obtained by computing AUC only up to a number of FP equal to the number of edges in the true network, normalised to lie in $[0, 1]$. For $N = 50$, the "baseline" of outputting a random edge ranking has an expected iAUC of 0.02.

Furthermore, on average about 25% of the true edges are "undetectable" by any method using the linearised ODE assumption: although present in the nonlinear system, their corresponding entries a_{ij} are very close to zero, and they do not contribute to the dynamics within the linearisation region. Such edges were excluded from the computation of iAUC, for all competing methods.

3.5.3 Setting Free Parameters

We need to adjust two free parameters: the noise variance σ^2, and the scale τ of the Laplace prior. Given some substantial amount of observations, these could be estimated by empirical Bayesian techniques, but this is not possible for experimental design, where we start with very few observations. One may be able to correct initial estimates of σ^2, as more observations are made, and a method for doing so is subject to future work.

There are two sources of noise, *i.e.* non-zero $\boldsymbol{\epsilon}$ for observations $(\boldsymbol{u}, \boldsymbol{x})$ and true linearisation matrix $\boldsymbol{A}$. First, the ODE of our simulator is stochastic, and measurement errors are made for $\boldsymbol{u}$, $\boldsymbol{x}$. Second, we have systematic deviations between the true nonlinear dynamics to ones of the linearisation. It is possible to estimate the variance of errors of the first kind without knowing the true $\boldsymbol{A}$ or performing specific disturbance experiments, by observing fluctuations around the undisturbed steady state. This is not possible for errors of the second kind. However, it is reasonable to assume that a good value for σ^2 does not change too much between networks with similar biological attributes, so that we can transfer it from a system whose dynamics are known, or for which sufficiently many observations are already available. This transfer was simulated in our experiments by generating 50 networks with data as mentioned above, then estimating σ^2 from the size of the $\boldsymbol{\epsilon}$ residuals. Note that these additional networks were only used to determine σ^2, for the other experiments we used independent samples from our network generator.

The scale parameter τ determines the *a priori* expected number of edges in the network. It could be determined similar to σ^2, but a simple heuristic worked just as well in most setups we looked at (the exception was very high noise situations). We need a rough guess of the average node in-degree $\bar{d}$. Then, under the Laplace prior, we expect $\bar{d}$ to be $N\mathrm{e}^{-\tau\delta_e}$ *a priori*. Solving for τ, we obtain

$$\tau = -\frac{1}{\delta_e} \log \frac{\bar{d}}{N}.$$

We found in practice that our method is quite robust to moderate changes in τ and σ^2, as long as the correct order of magnitude is chosen.

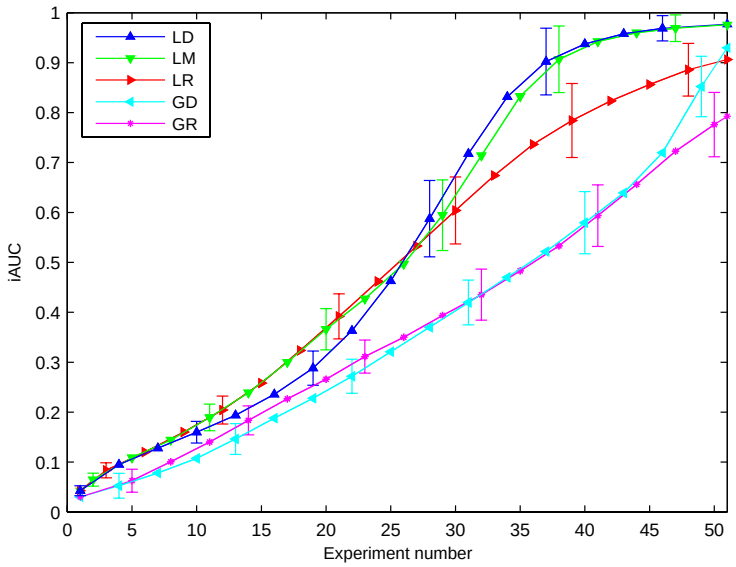


Figure 3.4: *Reconstruction curves for experiments (gene expression changes of* 1%*, SNR* 100*,* $r = 3$ *non-zeros per* $\boldsymbol{u}$*).* **LD***: Laplace prior, experimental design.* **LR***: Laplace prior, random experiments.* **GD***: Gaussian prior, experimental design.* **GR***: Gaussian prior, random experiments.* **LM***: Laplace prior, mixed selections (first* 20 *random, then designed). Error bars show one standard deviation over runs. All visually discernible differences in mean curves of different methods are significant under the* t*-test at level* 1%*.*

3.5.4 Discussion

In Figure 3.4, we present reconstruction curves for our method versus competing techniques, lacking novelties of our approach (optimal experimental design, Laplace sparsity prior). Very clearly, optimal design helps to save on costly and time-consuming experiments. The effect is more pronounced for the Laplace than for the Gaussian prior. The former is a better prior for the task, and it is well known that the advantage of designed versus random experiments scales with the appropriateness of the model. In this case, the iAUC level 0.9 is attained after 36 experiments with designed disturbances, yet only after 50 measurements with randomly chosen ones, thus saving 30% of the experiments.

In general, the model with Laplace prior does significantly better than with a Gaussian one (τ of the Laplace and the variance of the Gaussian prior were of course selected independently). The difference is most pronounced at times when significantly less than N experiments have been done and the linear system (3.2) is strongly under-determined. This confirms our arguments in favour of the Laplace prior.

The systematic underperformance of the most direct variant LD of our method, up to about $N/2$ observations, is not yet completely understood. One should be aware that aggressive experimental design based on very little knowledge can perform worse than a random choice. This is a variant of the well-known "explore-exploit" trade-off [Daw et al., 2006], which can be countered by either specifying prior knowledge more explicitly, or by doing a set of random inclusions (explore) before starting the active design (exploit). This is done in the LM variant.

In Figure 3.5, experimental design is compared to the random experiment choice setting,

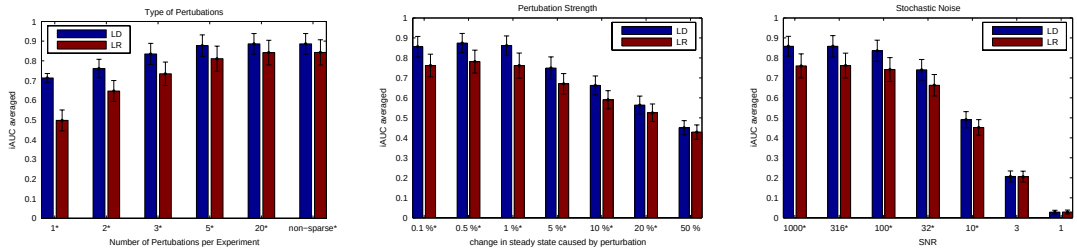


Figure 3.5: *Comparison between* **LD** *(Laplace, design) and* **LR** *(Laplace, random experiments) under different conditions. Score is average iAUC after* $25, \ldots, 50$ *experiments. (Left): Number* r *of non-zero* $\boldsymbol{u}$ *coefficients in each disturbance varied, keeping* $\|\boldsymbol{u}\|$ *constant. (Middle): Norm* $\|\boldsymbol{u}\|$ *of disturbances varied, while keeping* $r = 3$ *and low noise level. (Right): Stochastic noise in the data (3.1) varied, for constant* $\|\boldsymbol{u}\|$, $r = 3$. *Settings marked with* $*$*: LD is significantly superior to LR, according to* t*-test at level* 1%.

both with a Laplace prior. In the left panel, we vary the number r of non-zero entries in the disturbances $\boldsymbol{u}$. Recall that large r are in fact unrealistic in experimental techniques available today, but may well become accessible in the future. The less constraints there are on $\boldsymbol{u}$, the more information one may obtain about $\boldsymbol{A}$ in each experiment, and the better our method performs. This is in line with linear systems theory, where *persistent excitations* [Ljung, 1999] (*i.e.* full $\boldsymbol{u}$'s) are known to be most effective for exploring a system. The edge of experimental design is diminished with larger r. This is plausible, in that the informativeness of each $\boldsymbol{u}$ increases strongly with more non-zeros, thus the relative differences between $\boldsymbol{u}$'s are smaller. Experimental design can outperform random choices only if there are clear advantages in doing certain experiments over others.

The middle panel in Figure 3.5 explores effects of different sizes $\|\boldsymbol{u}\|$, *i.e.* different perturbation strengths (here, $r = 3$, and the noise in the SDE is very small). For larger $\|\boldsymbol{u}\|$, the real nonlinear dynamics deviate more and more from the linearised ones, thus decreasing recovery performance above about 5%. On the other hand, larger $\|\boldsymbol{u}\|$ would result in a better SNR for each experiment, given that nonlinear effects could be modelled as well. This is not yet done in our method, but these shortcomings are shared by all other methods relying on a linearisation assumption. It is, however, encouraging that our method is quite robust to the fact that even at smaller $\|\boldsymbol{u}\|$, the residuals $\boldsymbol{\epsilon}$ behave distinctly non-Gaussian (occasional large values).

The right panel in Figure 3.5 shows how increasing stochastic noise in (3.1) influences network recovery. We keep $r = 3$ and set $\|\boldsymbol{u}\|$ to generate steady state deviations of 1%. Good performance is obtained at SNRs beyond 10. With a SNR of 1, one cannot expect any decent recovery with less than N measurements. At all SNRs shown, the network was recovered eventually with more and more experiments, but this is probably not an option one has in current biological practice.

3.5.5 Comparison to Tegner et.al.

The method proposed in [Tegnér et al., 2003] is state-of-the-art for experimental design applied to gene network recovery, and in this section, we compare our method against theirs. Their approach can be interpreted in Bayesian terms as well, this is detailed in Additional Material 3.7.3.

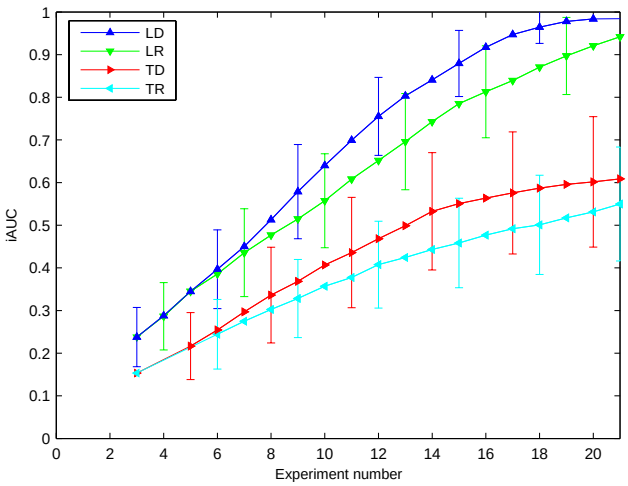


Figure 3.6: *Network recovery performance, comparing our method (Laplace, design) with [Tegnér et al., 2003]. Networks of size* $N = 20$*,* $r = 1$ *non-zeros in* $\boldsymbol{u}$*, perturbation size* 1%*, SNR* 100*. Three initial random experiments, to reduce memory requirements in [Tegnér et al., 2003] method.* **TD***: [Tegnér et al., 2003], experimental design.* **TR***: [Tegnér et al., 2003], random experiments.* **LD***: Our method, Laplace prior, experimental design.* **LR***: Our method, Laplace prior, random experiments.*

In contrast to our method, they discretise the space of possible matrices $\boldsymbol{A}$. Observations are used to sieve out candidates which are not "consistent" with all measurements so far. They have to restrict the maximum node in-degree for each gene to 3 in order to arrive at a procedure of reasonable cost. To our knowledge, the code used in [Tegnér et al., 2003] has not been released. We implemented it, following all details in their paper carefully (some details of our re-implementation are given in Additional Material 3.7.3). In general, the diagonal of $\boldsymbol{A}$ (self-decay rates) is assumed to be known in [Tegnér et al., 2003]. For the comparison, we modified our method to accept a fixed known $\mathrm{diag}\,\boldsymbol{A}$ and changed the iAUC score not to depend on self-edges.

Results of a direct comparison are shown in Figure Figure 3.6 with and without the proposed optimal design methods. Due to the high resource requirements of the method in [Tegnér et al., 2003], we use networks of size $N = 20$ (simulated as above), restricted to in-degrees at most 3. In general, our method performs much better in recovering the true network. This difference is robust even to significant changes in the ground truth simulator. We find that their method is very sensitive to measurement and system noise, or to violations of the linearisation assumption, whereas our technique is markedly more robust w.r.t. all these. We give some arguments why this might be the case. Firstly, their "consistency" sieve of $\boldsymbol{A}$ candidates in light of measurements is impractical. After every experiment a number of inconsistent $\boldsymbol{A}$ is rejected from consideration, and noisy experiments may well lead to a wrong decision. Any future evidence for such a rejected solution is, however, not considered any more. At the same time, an experiment does not help to discriminate between matrices which are still consistent afterwards. Another severe problem with their approach lies in the discretisation of $\boldsymbol{A}$ entries. A histogram of values of a_{ij} from our simulator reveals a very non-uniform (and also non-Gaussian) distribution: many values close to zero, but also a substantial number of quite large values. At the very least, their quantisation would have to be chosen non-uniformly and adaptively, such that each bin has about equal mass under

this distributions. However, it is quite likely that the best quantisation depends on details of the true system which are not known *a priori*. Statistics with continuous variables, as we employ, is a classical way of avoiding such quantisation issues. Furthermore, our Laplace prior seems to capture features of the a_{ij} distribution favourably.

In Table 3.1, we compare running times. Even though they restrict the node in-degree to 3, which is often unrealistic for known biological networks [Cokus et al., 2006], the required running times are orders of magnitude larger than for our method. Also, their memory requirements are huge, so that networks sizes beyond $N = 50$ could not be dealt with on a unit with 4 GB RAM. Both are clearly consequences of their quantisation approach, which we circumvent completely by applying a continuous model.

N	20	30	40	50	100	150	200
Our method	0.02	0.08	0.2	0.5	8	52	175
Tegnér *et.al.*[Tegnér et al., 2003] *	0.8	5	16	55	-	-	-

Table 3.1: *Running time for full network recovery, comparing our method (Laplace, design) with [Tegnér et al., 2003] In minutes; 2 GHz Opteron processor,* 1.5 *GB RAM.* *: *We allowed* 4 *GB RAM for [Tegnér et al., 2003], but this failed due to even higher demand for* $N > 50$.

3.5.6 Drosophila Segment Polarity Network

In [von Dassow et al., 2000], von Dassow *et.al.* describe a realistic model of the Drosophila segment polarity network. We tested our algorithm on a single cell submodule, using the equations and parameters as described in [Tegnér et al., 2003, Supplement], who also used this model. The Drosophila network not only contains mRNA levels but also 5 proteins which play an important role in the regulatory network. As described in Section 3.4.1, we thus focus on identifying the *effective* network between the genes.

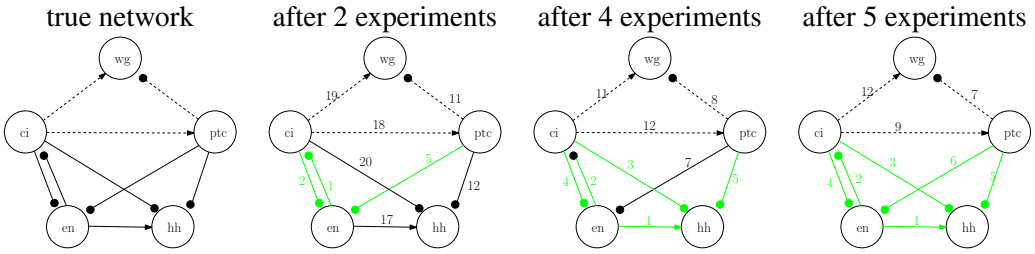


Figure 3.7: *The left figure shows the effective single cell model with five genes of the Drosophila segment polarity network [von Dassow et al., 2000]. Lines with circles denote inhibitory, arrows activating influence, functionally weak links are dashed. The figures on the right show the ranks that our algorithm assigns to each of the edges after* n *experiments* ($n = 2, 4, 5$). *There are* 6 *rel. strong edges with* $\tilde{\boldsymbol{A}}_{ij} \neq 0$ *in the network, and we assume that an edge is correctly identified if its rank is among the top* 6. *These edges are coloured green.*

As shown in Figure 3.7 the network contains 9 inter-gene regulatory pathways, apart from the self-links that are dominated by the respective self-decay rates. Three of the inter-gene links are functionally weak (i.e. $\tilde{\boldsymbol{A}}_{ij} \approx 0$). We simulated single gene perturbation experiments with an ordering chosen by our algorithm (Laplace prior distribution, perturbation

size 1%, SNR 100). After each experiment we ranked potential edges according to their probability. Resulting ranks after $2, 3, 5$ experiments for the true network edges are shown in Figure 3.7. All significant network edges are recovered after 5 experiments ($iAUC = 1$). Even weak links are assigned low ranks compared to a maximal rank 20, which places them among the first that would have to be examined more closely.

3.6 Conclusions

We have presented a Bayesian method for identifying gene regulatory networks from microarray measurements in perturbation experiments (*e.g.*, RNAi, toggle-switch, heterozygotes), and shown how to use optimal design in order to reconstruct networks with a minimum number of such experiments. The approach proves robust and efficient in a realistic nonlinear simulation setting. Our main improvements over previous work consist of employing a Laplace prior instead of a simpler Gaussian one, encoding the key property of sparse connectivity of regulatory networks within the model, and of actively designing rather than randomly choosing experiments. Both features are shown to lead to significant improvements. When it comes to experimental design, our method outperforms the most prominent instance of previous work significantly, both in higher recovery performance and in smaller resource requirements. Our application of the recent expectation propagation technique to the under-determined sparse linear model is novel, and variants may be useful for other models in bioinformatics.

Throughout the chapter we have assumed that $\boldsymbol{u}_*$ is known for an experiment, *i.e.* the disturbance levels of the r targeted genes can be controlled or at least predicted in advance, before the experiment is actually done. For example, a study trying to model the efficacy of RNAi experiments is given in [Vert et al., 2006]. In the context of experiment design, we can only hope to compute the expected decrease in uncertainty for a specific experiment, and thus rank potential experiments according to their expected value, if the experimental outcome is predictable to some degree. In our method, the outcome $\boldsymbol{x}_*$ for a given $\boldsymbol{u}_*$ is inferred through the current posterior, *i.e.* the information gain from $(\boldsymbol{u}_*, \boldsymbol{x}_*)$ is averaged over $Q(\boldsymbol{x}_*|\boldsymbol{u}_*, D)$. This can be extended to uncertain $\boldsymbol{u}_*$, if distributions $Q_e(\boldsymbol{u}_*|D)$ specific to each experiment e can be specified. For experimental biology, this means that not only do we need experimental techniques which deliver quantitative measurements, but furthermore the parameters distinguishing between different experiments ($\boldsymbol{u}$ in our case) either have to be fairly tightly controlled (our assumption in this chapter), or their range of outcome has to be characterised well by a mathematical model.

There are several other setups of formulating the network recovery problem in terms of a sparse linear model. Time-course mRNA measurements with unknown, yet time-constant disturbances $\boldsymbol{u}$ are used in [Schmidt et al., 2005] and [Sontag et al., 2004]. Relative rather than absolute changes in expression levels are employed in [Kholodenko et al., 2002]. Within all these setups, our general efficient Bayesian framework for the sparse linear model could be beneficial, and could lead to improvements due to the Laplace sparsity prior.

The linearised ODE assumption is frequently done [Yeung et al., 2002; Tegnér et al., 2003; Kholodenko et al., 2002; Peeters and Westra, 2004; Sontag et al., 2004; Schmidt et al., 2005], yet it is certainly problematic. For disturbances which change steady state expression levels by more than about 5%, our simulator showed a behavior which cannot directly be captured by a linearised approach. But such perturbation levels may be necessary to achieve

a useful SNR in the presence of typically high measurement noise. An important point for future work is the extension of the model by simple nonlinear effects of relevance to biological systems. For example, our model can directly be extended to higher-order Taylor expansions of nonlinear dynamics, since these are still linear in the parameters.

3.7 Additional Material

3.7.1 Sampling Small-World Networks

Following the description in [Albert and Barabási, 2002] we generate our random *small-world* networks using two steps: first we generate a network with nodes equally distributed on the unit circle and connect each node randomly to 50% of its 4 nearest neighbours. Then we create long range edges by randomly connecting any two nodes. In order to get a directed graph we orient edges with equal probabilities.

For our most commonly used networks of size $N = 50$ nodes showed in-degrees (excluding self-edges) in the range $\{0, ..., 6\}$ (average 2.3).

3.7.2 Dynamics of the Simulator

A review of potential dynamics for gene regulatory networks is given in [Smolen et al., 2000]. Here, the form of the nonlinear dynamic model and the parameter ranges were designed in similarity to the system described in [Kholodenko et al., 2002, Supporting Table 2].

Parameters were drawn randomly, see Table 3.2, subject to the model producing dynamics with a stable steady state with values in $[0, 10]$. Typical linearisation matrices $\boldsymbol{A}$ obtained at the unperturbed steady state have non-vanishing entries with mean zero and standard deviation 1.1, yet some quite large values do occur.

Parameter	Description	Range
V_{di}	Max. enzyme rate for degradation	$U[150..500]$
d_i	Max. degradation level	$U[20..70]$
κ_{ij}	Half-saturation / Michaelis constant	$U[20..70]$
n_{ij}	Hill coefficient	$U[1..2]$
V_{si}	Basal rate of expression	$U[3..5]$
A_{ij}	Max. over-expression factor	$U[2..5]$

Table 3.2: *Parameters of the nonlinear simulator. $U[a..b]$ is the uniform distribution between a and b.*

3.7.3 The Method of Tegner et.al.

We first describe the approach of [Tegnér et al., 2003] in Bayesian terms, which facilitates a comparison to ours. They start by discretising the space of possible matrices $\boldsymbol{A}$, having a finite number of bins for values of a_{ij}, one of them symmetric around 0. This results in a finite (but large) number of hypotheses for $\boldsymbol{A}$, and they put a uniform prior on allowable matrices: for each gene i, only up to three non-zero a_{ij} are allowed. In other words, the

node in-degree is limited to three in their, and also in our comparative experiments here. Their likelihood is an indicator distribution, in that $\boldsymbol{A}$ is *consistent* with the observations iff $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}$ is fulfilled up to a bounded error $\boldsymbol{\epsilon}$, across all measurements taken. Their posterior is therefore uniform over all (discretised) $\boldsymbol{A}$ consistent with the data and of node in-degree at most three. Experimental design in their method works by next perturbing the gene j for which the variance of a_{ij}'s (outgoing edges) is maximal, under this posterior.

We now give details of our implementation of their method. As [Tegnér et al., 2003] do not explicitly define what a *consistent* solution is, we will state the criterion that we used, in order to make our implementation of their method comparable.

Let us just consider one row of $\boldsymbol{A}$, namely $\boldsymbol{A}_{*,:}$. We assumed that the maximal in-degree is $k = 3$, i.e. there are at most 3 non-zero entries in $\boldsymbol{A}_{*,:}$ apart from the diagonal entry a_{**}. The non-zero entries are quantised into bins of equal width $\Delta_{\boldsymbol{A}}$ and with means $\bar{a}_j$ (j being the index of the bin). Symmetric around zero an interval of width $2\Delta_{\boldsymbol{A}}$ is excluded, for these entries are assumed to be zero and do not represent edges. $\boldsymbol{A}_{*,:}$ is then fully described by up to three tuples of one bin index j and one column index i each, *i.e.* by $D_* = \{(j(k), i(k))\}_{k \leq 3}$. We will assume that the measurement error of any component of $\boldsymbol{x}$ is at most Δ_x, that the maximal absolute value of $\boldsymbol{x}$ is x_{max}, and that the diagonal entry a_{**} is known exactly. We consider the row $\boldsymbol{A}_{*,:}$ given through a descriptor D_* as consistent with a measurement $(u_*, \boldsymbol{x}) \in \mathbb{R} \times \mathbb{R}^N$ if the value u_* falls into the following range

$$a_{**}\left(x_* \pm \Delta_x\right) \pm \Delta_x + \sum_{k=1}^{|D_*|} \left(\bar{a}_{j(k)} \left(x_{i(k)} \pm \Delta_x \right) \pm \frac{\Delta_{\boldsymbol{A}}}{2} x_{i(k)} \right) \pm (3 - |D_*|)\Delta_{\boldsymbol{A}} x_{max}.$$

This considers quantisation errors in the matrix entries of $\boldsymbol{A}$ and measurement errors in $\boldsymbol{x}$ and $\boldsymbol{u}$. The last term helped to improve results, and accounts for entries in $\boldsymbol{A}$ that are smaller than $\Delta_{\boldsymbol{A}}$ but may still represent an edge.

Given this criterion our implementation was quite simple: after the first random experiments, all possible row descriptors are checked whether they are consistent, and if so, were stored in an array. After each inclusion, only this array is parsed to detect row descriptors which have become inconsistent through the last experiment.

Chapter 4

Non-Parametric Regression between Riemannian Manifolds

In this chapter, we study non-parametric regression between Riemannian manifolds based on regularised empirical risk minimisation. We define and analyse a general family of regularisation functionals for mappings between manifolds which respect the geometry of input and output manifold and which are independent of the specific representation of the manifolds in terms of parametrisation or embedding. We then focus on the three most simple functionals of this family, namely the harmonic, the biharmonic and the novel Eells energy. We compare the energies against each other and show some of their properties. In particular, we will show that the Eells energy is a generalisation of the thin-plate spline energy to the case where input and output are Riemannian manifolds.

Following the theoretical analysis, we present a flexible numerical scheme for solving the resulting optimisation problems, and discuss several application examples. Specifically, we examine interpolation on the sphere, we compute regressions to surfaces of 3D objects, and we demonstrate the usefulness of the proposed approach for correspondence computations, task-space tracking, and colour image compression.

We conclude the chapter with characterising some interesting and sometimes counterintuitive implications and new open problems that are specific to learning between Riemannian manifolds and are not encountered in multivariate regression in Euclidean space.

4.1 Introduction

In machine learning, manifold structure has so far been mainly used in manifold learning [Belkin and Niyogi, 2004], to enhance learning methods especially in semi-supervised learning. The setting we want to discuss in this chapter is rather different, and has not been addressed yet in the machine learning community. Namely, we want to predict a mapping between *known* Riemannian manifolds based on input/output example pairs. We focus on a non-parametric regression setting, which subsumes interpolation, extrapolation, and smoothing as special cases.

In the statistics literature [Mardia and Jupp, 2000], this problem is treated for certain special output manifolds in directional statistics, where the main applications are to predict angles (circle), directions (sphere) or orientations (set of orthogonal matrices). Similarly, human

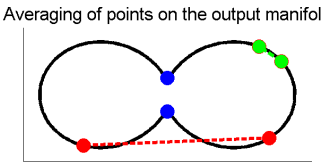


Figure 4.1: *The black line depicts a 1D-manifold in $\mathbb{R}^2$. The average of the red points in $\mathbb{R}^2$ does not lie on the manifold. Averaging of the green points which are close with respect to the geodesic distance is still reasonable. However, the blue points which are close with respect to the Euclidean distance are not necessarily close in geodesic distance and therefore averaging can fail.*

perception of colour values can be modelled via a colour circle [Shepard, 1980], and circular structure is also found for interpherometric measurements in SAR images [Massonnet et al., 1993]. More complex manifolds appear naturally in signal processing [Srivastava, 2000; Rahman et al., 2005], image processing [Tenenbaum et al., 2000], computer graphics [Mémoli et al., 2004; Hofer and Pottmann, 2004], and robotics [Noakes and Popiel, 2007; Steinke et al., 2008]. Impressive results in shape processing have recently been obtained [Davis et al., 2007; Kilian et al., 2007] by imposing a Riemannian metric on the set of shapes, so that shape interpolation is reduced to the estimation of a smooth curve in the manifold of all shapes. Moreover, note that almost any regression problem with differentiable equality constraints can also be seen as an instance of manifold-valued learning.

The regression problem where input and output domain are Riemannian manifolds is quite distinct from standard multivariate regression between Euclidean spaces. One fundamental problem of using traditional regression methods for manifold-valued regression is that most standard regression schemes assume that the output space is linear. It thus makes sense to linearly combine simple basis functions, since the addition of function values is still an element of the target space. While this approach still works for manifold-valued input, it is no longer feasible if the output space is a manifold, as general Riemannian manifolds do not have linear structure. This problem is demonstrated with an example in Figure 4.1.

One way how one can still learn manifold-valued mappings using standard regression techniques is to learn mappings directly into charts of the manifold. Another one is to use an embedding of the manifold in Euclidean space and utilise back-projections onto the manifold. While both approaches yield manifold-valued mappings, the solution will depend on the chart or embedding respectively, and in particular will not respect the geometric local relationships of the manifold, since close points in Euclidean space need not be close in the geometry of the manifold.

Here, we propose an approach for regression between manifolds that is based on regularised empirical risk minimisation, directly influencing the smoothness of the learned mapping via a suitable regulariser. We describe the construction of a family of general regularisation functionals for mappings between Riemannian manifolds and discuss in more detail three specific functionals, namely the harmonic, biharmonic, and the novel Eells energy, which can be seen as a generalisation of the thin-plate-spline energy. One important property of a regularisation functional is its null space, the set of mappings which are not penalised. Interestingly, in the case of the Eells energy the null space turns out to be the set of totally geodesic maps which can be seen as a proper generalisation of the set of linear mappings to the case of Riemannian manifolds.

From a computational perspective, the proposed regularisation functionals are quite complicated when expressed in coordinates of the manifolds. However, if input and output manifold can be embedded isometrically in Euclidean spaces, we will show that the regularisa-

tion functionals can be rewritten in an equivalent but much simpler extrinsic form. Using this formulation we then construct a relatively simple, yet very versatile implementation. We demonstrate regression between manifolds for several applications. First, we show the differences of the three regularisers for two interpolation tasks on the sphere, and then continue to apply the presented framework in a more realistic surface registration problem. Furthermore, we demonstrate an application for task-space tracking in robotics and animation, and lastly show how our ideas could be used for colour image compression. We conclude the chapter by discussing some challenging, yet very interesting new mathematical and statistical questions which arise due to the non-Euclidean structure of input and/or output space.

The general learning setup is described in Section 4.2. In Section 4.3 we define regularisation functionals for manifold-valued mappings, followed by a discussion of their properties in Section 4.4. In Section 4.5 we provide extrinsic expressions of the regularisation functionals which turn out to be crucial for for an efficient implementation, which is described in Section 4.6. Experimental results are shown in Section 4.7, interesting aspects and open problems in learning between Riemannian manifolds are discussed in Section 4.8, and we conclude in Section 4.9. The additional material in Section 4.10 features besides the proofs of this chapter a step-by-step introduction to the pull-back connection which is needed in the construction of parametrisation invariant differential regularisers for mappings between Riemannian manifolds.

4.1.1 Related Work

Riemannian manifolds are commonly used in so-called manifold learning, where either only the input domain is considered to be a manifold [Belkin and Niyogi, 2004] or where a description of the manifold itself is learnt [Tenenbaum et al., 2000; Lawrence and Quiñonero-Candela, 2006]. In both cases the manifold is unknown and only a sample of points from this manifold is given. Instead, the focus in this work is to learn a predictor from given pairs of input/output examples lying on *known* input and output manifolds.

For regression with manifold-valued output there are classic methods for spherical data [Fisher et al., 1993], and recently a k-nearest neighbour [Karcher, 1977; Buss and Fillmore, 2001], a Nadaraya-Watson type [Davis et al., 2007] and a wavelet [Rahman et al., 2005] type estimator have been adapted for this task. In contrast, our work is based on differential energies for mappings between general Riemannian manifolds. It unifies and extends previous such approaches in various ways. The harmonic [Eells and Sampson, 1964; Urakawa, 1993; Nishikawa, 2002] and biharmonic [Montaldo and Oniciuc, 2005] energy have been studied extensively in the differential geometry community, but less so in a learning context. Close to our setting are [Gabriel and Kajiya, 1985; Noakes et al., 1989; Machado et al., 2006; Camarinha et al., 1995]. All of these consider the problem of learning a curve in the output manifold, that is, in contrast to our approach the input domain is constrained to be one dimensional and Euclidean. Interpolation is performed in [Gabriel and Kajiya, 1985; Noakes et al., 1989] with a regulariser that penalises second-order derivatives, whereas [Camarinha et al., 1995] proposes regularisation functionals of arbitrary order. Approximation is analysed in [Machado et al., 2006], but only a first order regulariser is used. All these approaches fix start and endpoints of the curve. The closest in spirit to our approach is [Mémoli et al., 2004], where the harmonic energy is used in an approximation setting.

4.1.2 Notation

Throughout the article we will use the following notation. M is always the input manifold, N the target manifold, and $\phi : M \to N$ is the mapping from input to target manifold. The dimensions of M and N are m and n, and x and y are coordinates in M and N. Moreover we will use the Einstein summation convention and Penrose's abstract index notation, see [Wald, 1984, Ch 2.4]. "Abstract" indices indicate only the tensor type, they should not be mixed up with the indices for the components. For example a two-times covariant tensor h is written as h_{ab} and the coordinate representation would be $h_{ab} = h_{\mu\nu}\, dx^\mu_a \otimes dx^\nu_b$. In general, we use Greek letters for components (α,β,γ for components in M and μ,ν,ρ for components in N) and Latin ones for abstract indices (a, b, c for indices in M and r,s,t in N). We denote by g_{ab}, h_{ab} the metrics on M and N, by ${}^M\nabla$ and ${}^N\nabla$ the Levi-Civita connections on M and N with corresponding Christoffel symbols ${}^M\Gamma^\alpha_{\beta\gamma}$ and ${}^N\Gamma^\rho_{\nu\mu}$. We follow [Lee, 1997] and define the Riemannian curvature tensor $R : \otimes^3 TM \otimes T^*M \to \mathbb{R}$ as $\nabla_a \nabla_b Z^c - \nabla_b \nabla_a Z^c = R_{abd}{}^c\, Z^d$. As usual, $\otimes$ denotes the tensor product. For the reader's convenience we have summarised all symbols used in this chapter in a table in Additional Material 4.10.5.

4.2 Regularised Empirical Risk Minimisation for Manifold-Valued Regression

Given a set of K training pairs (X_i, Y_i) with $X_i \in M$ and $Y_i \in N$ we would like to learn a mapping $\phi : M \to N$. This learning problem reduces to standard multivariate regression if M and N are both Euclidean spaces $\mathbb{R}^m$ and $\mathbb{R}^n$ and to regression on a manifold if at least N is Euclidean. We propose to use regularised empirical risk minimisation, which can be formulated in our setting as

$$\underset{\phi \in C^\infty(M,N)}{\arg\min} \frac{1}{K} \sum_{i=1}^{K} L\big(Y_i, \phi(X_i)\big) + \lambda\, S(\phi), \tag{4.1}$$

where $C^\infty(M, N)$ denotes the set of smooth mappings ϕ between M and N, $L : N \times N \to \mathbb{R}_+$ is the loss function, $\lambda \in \mathbb{R}_+$ the regularisation parameter, and $S : C^\infty(M, N) \to \mathbb{R}$ the regularisation functional. The regularisation functional should measure the complexity of the mapping ϕ; the proper definition of such a functional will be the topic of the next section. Note, that for simplicity we constrain ϕ to be smooth, an issue that is discussed in more detail in Section 4.8.1.

In multivariate regression, $f : \mathbb{R}^m \to \mathbb{R}^n$, the most common loss function is the squared Euclidean distance of $f(X_i)$ and Y_i, $L\big(Y_i, f(X_i)\big) = \|Y_i - f(X_i)\|^2_{\mathbb{R}^n}$. A direct generalisation to a loss function on a Riemannian manifold N is to use the squared geodesic distance in N, $L\big(Y_i, \phi(X_i)\big) = d^2_N\big(Y_i, \phi(X_i)\big)$. The correspondence to the multivariate case can be seen from the fact that $d_N(Y_i, \phi(X_i))$ is the length of the shortest path between Y_i and $\phi(X_i)$ in N, as the norm $\|f(X_i) - Y_i\|$ is the length of the shortest path, namely the length of the straight line, between $f(X_i)$ and Y_i in $\mathbb{R}^n$. Naturally, taking the p-th power of the geodesic distance as well as any other function $\Theta : \mathbb{R}_+ \to \mathbb{R}_+$ of the geodesic distance is also possible.

Generalising multivariate loss functions which are not isotropic in $\mathbb{R}^n$ is more difficult. For example for $p \neq 2$, the $l_p(\mathbb{R}^n)$ loss depends not only on the length of the vector $Y_i - f(X_i)$,

but also on the angles relative to a fixed global coordinate system. Difference vectors and global coordinate systems are, however, not well-defined in the general case of Riemannian manifolds. An appropriate generalisation of l_p losses could be defined via so called Finsler manifolds. Whereas for Riemannian manifolds one has an inner product in the tangent space, for Finsler manifolds only a norm is defined in each tangent space. For simplicity, we will in this chapter only consider loss functions based on the geodesic distance of the *Riemannian* manifold N.

In general, we assume to be in a statistical setting (however, the framework also works also if this is not the case), where the given input/output pairs (X_i, Y_i) are i.i.d. samples from a probability measure P on $\mathcal{X} \times \mathcal{Y}$. The setting we have in mind is that our data is perturbed by noise in the output space. In multivariate regression it is well known that using the squared Euclidean distance as loss function, $L(Y_i, f(X_i)) = \|Y_i - f(X_i)\|^2_{\mathbb{R}^n}$, the Bayes optimal predictor f^*, that is, the function f^* minimising,

$$f^* = \underset{f \text{ measurable}}{\arg\min} \; \mathbb{E}\,\|Y - f(X)\|^2 = \underset{f \text{ measurable}}{\arg\min} \; \mathbb{E}_X\, \mathbb{E}_{Y|X}[\|Y - f(x)\|^2 \mid X],$$

is given by the conditional mean $f^*(x) = \mathbb{E}[Y|X = x]$, usually denoted as the regression function. The regression function $f^*(x)$ is uniquely determined (almost everywhere) since the risk functional is strictly convex in $f(x)$.

Naturally, the question arises which is the Bayes optimal mapping $\phi^* : M \to N$ for regression between manifolds; that is, using the squared geodesic distance in N as a loss measure, which map ϕ^* minimises the expected loss,

$$\phi^* := \underset{\phi \text{ measurable}}{\arg\min} \; \mathbb{E}\, d^2_N(Y, \phi(X)) = \underset{\phi \text{ measurable}}{\arg\min} \; \mathbb{E}_X\, \mathbb{E}_{Y|X}[d^2_N(Y, \phi(X)) \,|\, X].$$

Here, we have used in the second step the result of [Blackwell and Maitra, 1984] that a joint probability measure on the product of two separable metric spaces can always be factorised into a conditional probability measure and the marginal, and we assume that $\mathbb{E}\, d^2_N(Y, \phi(X)) < \infty$ for some measurable $\phi : M \to N$. Note, that every Riemannian manifold is a metric space and since we assume that M and N are finite dimensional they are separable. This factorisation allows us to find the Bayes optimal mapping pointwise,

$$\phi^*(x) = \underset{p \in N}{\arg\min}\, \mathbb{E}[d^2_N(Y, p) \,|\, X = x] = \underset{p \in N}{\arg\min} \int_N d^2(y, p)\, d\mu_x(y),$$

where $d\mu_x$ is the conditional probability measure of Y given $X = x$. The global minimiser of the functional,

$$F(p) = \underset{p \in N}{\arg\min} \int_N d^2(y, p)\, d\mu_x(y),$$

is called the Frechét mean or Karcher mean[1]. It is the direct generalisation of a mean in Euclidean space to a general metric space. Unfortunately, it needs no longer to be unique as in the Euclidean case. A simple example is the sphere as the output space together with a uniform probability measure on it. In this case every point p on the sphere attains the same value $F(p)$ and thus the global minimum is non-unique. We refer to [Karcher, 1977; Kendall, 1990; Bhattacharya and Patrangenaru, 2003] for more information under which conditions one can prove uniqueness of the global minimiser.

[1] In some cases the set of all local minimisers is denoted as the Frechét mean set and the mean is called unique if there exists only one global minimiser.

4.3 Regularisation Functionals for Mappings Between Riemannian Manifolds

We would like to define regularisation functionals,

$$S : C^\infty(M, N) \to \mathbb{R}_+,$$

for mappings between two Riemannian manifolds M and N measuring the smoothness of the mapping $\phi : M \to N$. Two objectives should hold for the regularisation functional:

1. independence of the representation of the manifolds M and N,
2. dependence only on ϕ and the geometry of M and N.

There are basically two ways to represent manifolds. The first one is via a collection of local charts or parametrisations. There are many different ways to choose these charts and, obviously, our energy should not depend on this arbitrary choice, e.g., the energy of curves on the sphere should be the same if we represent the sphere in spherical or stereographic coordinates. A second way to represent many manifolds is via an isometric embedding in Euclidean space, that is, the manifold is defined as a subset of some ambient space and the metric of the manifold corresponds locally to the distance in the embedding space. Examples of embedded manifolds are the sphere S^2 in $\mathbb{R}^3$ or SO_3 in $\mathbb{R}^{3\times 3}$. Again, our energy should not depend on this choice of representation since it is also not unique. We will show in Section 4.5.3 that the penalisation of components in the ambient space (extrinsic quantities) leads to a notion of smoothness for manifold-valued mappings which contradicts our intuitive expectations. Instead, the energy should only depend on the map $\phi : M \to N$ and how it relates invariant *intrinsic* geometric properties of the manifolds M and N with each other. These dependence/independence properties can be achieved by formulating the energy in the covariant language of differential geometry.

The remainder of this section requires some technical notions from differential geometry, in particular the one of a pull-back connection. For the sake of a clear presentation we have moved the exact definition of this term to Additional Material 4.10.1. The basic properties can be understood also without this knowledge.

Before we discuss general regularisation functionals penalising derivatives of arbitrary order let us begin with the most simple energy functional for manifold-valued mappings. The differential or Jacobian $d\phi^r_a : T_xM \to T_{\phi(x)}N$ of a mapping $\phi : M \to N$ evaluated at x is given as

$$d\phi^r_a(x) = \frac{\partial \phi^\mu}{\partial x^\alpha} dx^\alpha_a \Big|_x \otimes \frac{\partial^r}{\partial y^\mu}\Big|_{\phi(x)} \tag{4.2}$$

It measures the change of the output $\phi(x) \in N$ as one varies x in the input manifold M. This 1-1-tensor can be used to define the most simple differential energy, the so called harmonic energy.

Definition 4.1. *The **harmonic energy** $S_{\text{harmonic}}(\phi)$ of a mapping $\phi : M \to N$ is defined as*

$$\begin{aligned} S_{\text{harmonic}}(\phi) &= \int_M \|d\phi\|^2_{T^*_xM \otimes T_{\phi(x)}N}\, dV(x) && (4.3)\\ &= \int_M g^{ab}(x) h_{rs}(\phi(x)) d\phi^r_a d\phi^s_b dV(x) && (4.4)\\ &= \int_M g^{\alpha\beta} h_{\mu\nu} \frac{\partial \phi^\mu}{\partial x^\alpha} \frac{\partial \phi^\nu}{\partial x^\beta} dV(x), \end{aligned}$$

where $dV = \sqrt{\det g}\, dx$ *is the volume element of* M.

For standard regression, that is $M = \mathbb{R}^m$ and $N = \mathbb{R}$, the harmonic energy reduces to

$$S_{\text{harmonic}}(\phi) = \int_{\mathbb{R}^m} \|\nabla\phi\|^2 \, dx.$$

For $m = 1$ this functional in turn reduces to the energy functional of linear splines, and using this energy in approximation or interpolation as in objective (4.1) leads to piecewise linear solutions which are non-differentiable at the mapped data points $\phi(X_i)$. A similar behaviour can be observed for curves on manifolds, that is, for $M = [a, b]$ and N a Riemannian manifold, where

$$S_{\text{harmonic}}(\phi) = \int_a^b \left\|\dot{\phi}\right\|^2 dt$$

with $\dot{\phi}(t) = \frac{d\phi}{dt}(t)$. In this case, minimisers of (4.1) are piecewise geodesic [Machado et al., 2006].

Since we are generally interested in solutions which have higher smoothness, we have to use higher order derivatives in the regulariser. In the Euclidean case this is typically done e.g. using the thin-plate spline energy $\int_{\mathbb{R}^m} \|Hf\|_F^2 \, dx$, where Hf is the Hessian of $f : \mathbb{R}^m \to \mathbb{R}$ and $\|.\|_F$ the Frobenius norm. Another alternative is the biharmonic regulariser, $\int_{\mathbb{R}^m} \left|\Delta f\right|^2 dx$, where $\Delta f = \text{trace}(Hf)$.

For the generalisation of regularisers of this type to the case of mappings between manifolds we have to define the second derivative of mappings between Riemannian manifolds, that is, the covariant derivative of the differential $d\phi_a^r$. The problem is here that $d\phi$ "lives" in the cotangent and tangent space, T_x^*M and $T_{\phi(x)}N$, of two different manifolds. Thus we cannot simply use the connection ${}^M\nabla$ of M. The solution is to use the pull-back connection defined in Additional Material 4.10.1, which yields a notion of the derivative of a vector field on N with respect to a variation in M, where M and N are connected via $\phi : M \to N$. We then use the pull-back connection for derivatives of vector-fields in the target manifold N plus the connection on M for derivatives on the input manifold together in a so-called tensor product connection, see also Additional Material 4.10.1. The p-th order covariant derivative of the differential $d\phi$ will yield the tensor field

$$\nabla'_{a_1} \ldots \nabla'_{a_p} d\phi^r_{a_{p+1}} \;\in\; \otimes^{p+1} T^*M \otimes \phi^{-1}TN,$$

where $\phi^{-1}TN$ is the so-called pull-back bundle, see Definition 4.16. This derivative is by definition invariant with respect to parametrisation and respects the intrinsic geometry of M and N. Note that for a function $\phi : \mathbb{R}^m \to \mathbb{R}^n$ the p-th order covariant derivative equals

$$\frac{\partial^{p+1}\phi^\mu}{\partial x^{\alpha_1} \ldots \partial x^{\alpha_{p+1}}} dx^{\alpha_1}_{a_1} \otimes \ldots \otimes dx^{\alpha_{p+1}}_{a_{p+1}} \otimes \frac{\partial^r}{\partial y^\mu}.$$

In this form the Euclidean $p+1$-order derivative is covariant, that is invariant under coordinate changes.

We are now ready to define higher order differential energies. In order to obtain a real-valued regularisation functional, we have to define an operation $\Theta : \otimes^{p+1} T^*M \otimes \phi^{-1}TN \to \mathbb{R}_+$. The function Θ usually consists of two steps. First one takes traces in some entries and

then the norm or some power of the norm of the resulting tensor. This yields the general regularisation functional, $S : C^\infty(M, N) \to \mathbb{R}_+$, defined as

$$S(\phi) = \int_M \Theta\Big(\nabla'_{a_1} \dots \nabla'_{a_p} d\phi^r_{a_p+1}\Big)\, dV. \tag{4.5}$$

We will illustrate this for second order differential energies ($p = 1$). The tensor field $\nabla'_b d\phi^r_a$ is given in coordinates, see Additional Material 4.10.1, as

$$\nabla'_b d\phi^r_a = \Big[\tfrac{\partial^2\phi^\mu}{\partial x^\beta \partial x^\alpha} - \tfrac{\partial\phi^\mu}{\partial x^\gamma}{}^M\Gamma^\gamma_{\beta\alpha} + \tfrac{\partial\phi^\rho}{\partial x^\alpha}\tfrac{\partial\phi^\nu}{\partial x^\beta}{}^N\Gamma^\mu_{\nu\rho}\Big] dx^\beta_b \otimes dx^\alpha_a \otimes \tfrac{\partial^r}{\partial y^\mu}. \tag{4.6}$$

Note that non-vanishing Christoffel symbols of M keep the expression linear in ϕ, whereas non-zero Christoffel symbols of N render the second-order differential a non-linear operator. This illustrates again, why manifold-valued input is easier to handle than manifold-valued output.

For the tensor field $\nabla'_b d\phi^r_a$ we can either first take the trace in b and a and then use the squared norm in $T_{\phi(x)}N$, which yields the biharmonic energy.

Definition 4.2. *The **biharmonic energy** $S_{\text{biharmonic}}(\phi)$ is defined as*

$$\begin{aligned} S_{\text{biharmonic}}(\phi) &= \int_M \Big\| g^{ba} \nabla'_b d\phi^r_a \Big\|^2_{T_{\phi(x)}N} dV(x) \\ &= \int_M g^{ba}\, g^{cd}\, h_{rs}\, \nabla'_b d\phi^r_a\, \nabla'_c d\phi^s_d\, dV(x). \end{aligned} \tag{4.7}$$

Another possibility is to use directly the squared norm in $T^*_x M \otimes T^*_x M \otimes T_{\phi(x)}N$.

Definition 4.3. *The **Eells energy** $S_{\text{Eells}}(\phi)$ is defined as*

$$\begin{aligned} S_{\text{Eells}}(\phi) &= \int_M \big\| \nabla'_b d\phi^r_a \big\|^2_{T^*_x M \otimes T^*_x M \otimes T_{\phi(x)}N}\, dV(x) \\ &= \int_M g^{ac}\, g^{bd}\, h_{rs}\, \nabla'_b d\phi^r_a\, \nabla'_d d\phi^s_c\, dV(x). \end{aligned} \tag{4.8}$$

While the biharmonic energy has been discussed in the differential geometry community, see [Montaldo and Oniciuc, 2005], the Eells energy has to our knowledge not been studied in differential geometry or elsewhere before. We have named it after James Eells who pioneered the study of harmonic maps between Riemannian manifolds [Eells and Sampson, 1964] and recently passed away.

The Eells energy reduces to the thin-plate spline energy in the Euclidean case. If M and N are Euclidean we obtain

$$S_{\text{Eells}}(\phi) = \int_M g^{\alpha\beta} g^{\gamma\delta} h_{\mu\nu} \frac{\partial^2\Phi^\mu}{\partial x^\alpha \partial x^\gamma} \frac{\partial^2\Phi^\nu}{\partial x^\beta \partial x^\delta}\, dV(x),$$

where g and h are the Riemannian metrics corresponding to Euclidean space. This is the parametrisation independent form of the thin-plate spline energy. In Cartesian coordinates we have $g^{\alpha\beta} = \delta^{\alpha\beta}$ and $h_{\mu\nu} = \delta_{\mu\nu}$ where δ is the Kronecker symbol. The Eells energy thus reduces to the standard form of the thin-plate spline energy:

$$S_{\text{Eells}}(\phi) = \sum_{\mu=1}^{n} \int_M \sum_{\alpha,\gamma=1}^{m} \Big(\frac{\partial^2\Phi^\mu}{\partial x^\alpha \partial x^\gamma}\Big)^2 dx. \tag{4.9}$$

For curves ϕ in a manifold N, that is $M = [a, b]$, the Eells energy and the biharmonic energy are identical,

$$S_{\text{Eells}}(\phi) = S_{\text{biharm.}}(\phi) = \int_a^b \left\| \nabla_{\dot{\phi}(t)} \dot{\phi}(t) \right\|^2_{T_{\phi(t)}N} dt, \tag{4.10}$$

where $\dot{\phi}(t) = \frac{\partial}{\partial t}\phi(t)$. Using this energy we recover the interpolation problem of cubic splines on curved spaces proposed by [Gabriel and Kajiya, 1985; Noakes et al., 1989] in our framework (4.1) for $\lambda \to 0$.

Note, that in the three examples of regularisation functionals above we restricted ourselves to the squared norm of the differentials. However, in order to construct a regularisation functional which resembles the total variation regulariser, $\int_{\mathbb{R}^m} \|\nabla\phi\| \, dx$ for $\phi : \mathbb{R}^m \to \mathbb{R}$ often used in image processing, see e.g. [Aubert and Kornprobst, 2006], one just takes the norm of $d\phi^r_a$.

Definition 4.4. *The **total variation energy** $S_{\text{TotalVar}}(\phi)$ of a mapping $\phi : M \to N$ is defined as*

$$\begin{aligned} S_{\text{TotalVar}}(\phi) &= \int_M \|d\phi\|_{T^*_x M \otimes T_{\phi(x)}N} \, dV(x) \\ &= \int_M \sqrt{g^{\alpha\beta}(x) h_{\mu\nu}(\phi(x)) \frac{\partial \phi^\mu}{\partial x^\alpha} \frac{\partial \phi^\nu}{\partial x^\beta}} dV(x). \end{aligned} \tag{4.11}$$

4.4 Properties of the Regularisation Functionals

In this section we describe and compare general properties of the harmonic, biharmonic, and Eells energy and their use as regularisers for regression between two general Riemannian manifolds. We start by describing the null-space of the different functionals, which characterises the mappings which are *not* penalised, continue with an analysis of the difference between biharmonic and Eells energy, and end with a discussion why second-order energies are useful in modelling physical systems.

4.4.1 The Null Space

The null space of a regularisation functional $S(\phi)$ is the set $\{\phi \,|\, S(\phi) = 0\}$, which is interesting out of two reasons. The first one is that the null space consists of the mappings which are not penalised and therefore defines a set of mappings which we are free to fit the data with. In standard regression these are usually linear mappings or polynomials of small degree. The other reason is that, as the regularisation parameter λ tends to infinity, the regularised empirical risk minimisation problem in Eq. (4.1) reduces to

$$\underset{\phi \in C^\infty(M,N)}{\arg\min} \frac{1}{K} \sum_{i=1}^{K} L(Y_i, \phi(X_i)), \quad s.t. \quad S(\phi) = 0. \tag{4.12}$$

Thus, in this limit the only feasible set of mappings is the null space of S.

The harmonic energy The null space of the harmonic energy $S_{\text{harmonic}}(\phi)$ consists of the constant maps $\phi \equiv y$, $y \in N$, see [Eells and Lemaire, 1983], that is all input points in M are mapped to a single point y in N. The property that the harmonic energy penalises deviations from a constant mapping has severe consequences for the learning task. Namely, if the image of the boundary ∂M is not fixed, then the harmonic energy can always be reduced by contracting the mapping as much as the trade-off between loss and regulariser allows. It is often not easy to know a priori how to fix the image of the boundary ∂M such that no big distortions arise. One example of the negative contraction effects resulting from this problem can be seen in Figure 4.8 (c), another in [Mémoli et al., 2004, Fig. 4].

It is interesting to note that for the squared geodesic distance loss, the learning problem in (4.12) reduces to a classical problem in differential geometry: the task to find the mean of a set of points on a Riemannian manifold, the so-called Karcher mean [Karcher, 1977]. The Karcher mean is only unique given that the data points Y_i are sufficiently close in N. In the case of $M = \mathbb{R}^m$ and $N = \mathbb{R}^n$, problem (4.12) corresponds to the prediction of the usual mean $\frac{1}{K}\sum_{i=1}^{K} Y_i$.

The Eells energy We have shown in the last section that the Eells energy reduces to the classical thin-plate spline energy if input and output manifold are Euclidean. For the thin-plate spline energy it is well-known that the null space consists of the linear mappings between input and output space. Thus in the Euclidean case we are free to fit the data with a linear map but any deviation from linearity will be penalised. The concept of linearity breaks down in the manifold setting since input and output space have no linear structure. An interesting question is if there exists a proper generalisation of linear mappings to the case where input and output space are Riemannian manifolds. A key observation towards a natural generalisation of the concept of linearity is that linear maps map straight lines to straight lines. Now a straight line between two points in Euclidean space corresponds to a path of shortest length and is thus a geodesic between the two points. In analogy to the Euclidean case we will therefore consider in Riemannian manifolds mappings which map geodesics to geodesics as the proper generalisation of linear maps.

The following proposition taken from [Eells and Lemaire, 1983] defines this concept and characterises these mappings. The proof is presented in Additional Material 4.10.2.

Proposition 4.5. *[Eells and Lemaire, 1983] A map $\phi : M \to N$ is totally geodesic if ϕ maps geodesics of M linearly to geodesics of N, i.e. the image of any geodesic in M is also a geodesic in N though potentially with a different constant speed. The following three properties are equivalent:*

1. *ϕ is totally geodesic,*
2. *ϕ preserves the connection, i.e.*

$$ {}^{N}\nabla_{d\phi(X)} d\phi(Y) = d\phi({}^{M}\nabla_{X} Y), $$

   where $d\phi$ is the differential of ϕ and X, Y are smooth vector fields on M,
3. $\nabla'_a d\phi^r_b = 0$.

Proposition 4.5 immediately characterises the null space of the Eells energy as the set of totally geodesic maps. This is one more argument why the Eells energy can be seen as

the valid generalisation of thin-plate splines to the case where input and output spaces are Riemannian manifolds.

Linear maps encode a very simple relation in the data: the local relative changes between input and output are the same everywhere. This is the simplest relation a non-trivial mapping can encode between input and output, and totally geodesic mappings encode the same "linear" relationship even though the input and output manifold are nonlinear. However, note that as linear maps, totally geodesic maps are *not* necessarily distortion-free, but every distortion-free (isometric) mapping is totally geodesic. Furthermore, given "isometric" training points,

$$d_M(X_i, X_j) = d_N(Y_i, Y_j), \quad i, j = 1, \dots, k,$$

then among all minimisers of (4.1), there will be an isometry fitting the data points, given that such an isometry exists. With this restriction in mind, one can see the Eells energy also as a measure of distortion of the mapping ϕ. This makes the Eells energy an interesting candidate for a variety of geometric fitting problems, for example, for surface registration as demonstrated in the experimental section.

Despite the similarity of linear and totally geodesic maps it should be noted that there are certain circumstances in which they show completely different behaviour. One important example is discussed in Section 4.8.3.

In contrast to the harmonic energy, the Eells energy does not lead to contraction effects. Imagine the situation of only two given training points in a regression problem from the real line to the sphere. While the solution for the harmonic energy tends to contract and would only for $\lambda \to \infty$ pass exactly through the points, the solution for the Eells energy would yield a geodesic which exactly fits the given training data points for any value of λ. It would also extrapolate "linearly", whereas the harmonic solution which minimises the change of the prediction function has no reason to extrapolate at all beyond the first and last training point. These effects are demonstrated in Figure 4.6 and Figure 4.8.

The biharmonic energy The null space of the biharmonic energy is a superset of the null space of the Eells energy, since here only the trace of the "Hessian" of ϕ has to vanish, not all its components. Apart from totally geodesic mappings, the null space of the biharmonic energy also contains all stationary maps of the harmonic energy, see Theorem 4.29 below. Although this sounds reasonable at first, the null space may thus be too big for some applications. This can already be seen from an example in Euclidean space. Consider the mapping $\phi : \mathbb{R}^2 \to \mathbb{R}$ with $\phi(x_1, x_2) = x_1^2 - x_2^2$, which is clearly non-linear and intuitively not very smooth, nevertheless the biharmonic energy of this mapping is zero. While the variational equation of the biharmonic energy which involves the iterated Laplacian (see Theorem 4.29) is often easy to implement, we recommend the Eells energy due to its better interpretation as a smoothness measure.

4.4.2 Difference of Biharmonic and Eells Energy

One can show, see Theorem 4.6 below, that in Euclidean spaces the biharmonic and the Eells/thin-plate spline energy only differ by a boundary term. In the literature they are therefore often considered as equivalent, see for example [Duchamp and Stuetzle, 2003]. However, even in Euclidean space, this is only justified given that one can guarantee that the first or second derivative of the function one wants to learn vanishes on the boundary of

the domain or decay to zero at infinity. Furthermore, if either M or N is non-Euclidean, the two energies are due to curvature effects different even when one neglects boundary terms. Interestingly, this difference even holds for simple real-valued functions on a non-Euclidean Riemannian manifold M, that is, for $N = \mathbb{R}$. The proof of the following theorem is found in Additional Material 4.10.2.

Theorem 4.6. *The biharmonic and Eells energy are related in the following way,*

$$S_{\textit{biharmonic}}(\phi) = S_{\textit{Eells}}(\phi) + \int_M h_{rs}\, g^{ab}\, g^{cd}\, d\phi^r_c \left(R^{M}_{adb}{}^{e}\, d\phi^s_e - R^{N}_{tuv}{}^{s}\, d\phi^t_a\, d\phi^u_d\, d\phi^v_b \right) dV + \int_{\partial M} N^b h_{rs}\, g^{cd} \left(d\phi^r_b\, \nabla'_c d\phi^s_d - d\phi^r_c\, \nabla'_b d\phi^s_d \right) d\tilde{V},$$

where $R^{M}_{adb}{}^{e}$, $R^{N}_{tuv}{}^{s}$ is the Riemannian curvature tensor of M, N, and $d\tilde{V}$ the volume form of the boundary ∂M.

4.4.3 Physical Interpretation of Intrinsic Second-Order Energies

In [Marsden and Ratiu, 1999] it is shown that classical mechanics can be understood in a differential geometric way. Namely, one considers the set of possible configurations of a system as a manifold N. The standard example is the rigid body which has configuration space $\mathbb{R}^3 \times SO(3)$, that is, position plus orientation. The manifold of configurations is then given a geometric structure by using the kinetic energy as Riemannian metric. Using this formulation on can write Newton's equation for the time-dependent state γ of the physical system, $\gamma : [a, b] \to N$, as

$$\nabla_{\dot\gamma(t)} \dot\gamma(t) = \tau(t, \gamma(t)),$$

where τ are the external forces acting upon the system. Noting that it is exactly this acceleration $\nabla_{\dot\gamma(t)} \dot\gamma(t)$ which is penalised in the biharmonic/Eells energy of curves (4.10), one can interpret the corresponding smoothing problem (4.1) as a trade-off between passing through the set of training points and following free motion as much as possible. Since the acceleration is directly related to the external forces acting on the state, the biharmonic/Eells energy also penalises the amount of external forces which have to act on a physical system to follow a certain trajectory. Thus, for applications like animation or robot control where a real physical system is lying beneath the learning problem, the biharmonic/Eells energy will provide an optimal solution.

4.5 From Intrinsic to Extrinsic Representation

One can deduce from the equation for the second derivative of ϕ (4.6) that the representation of the Eells energy in coordinates of M and N is quite complicated and not easily accessible for its optimisation. Moreover, the use of local coordinate systems introduces the additional complication that the mapped point $\phi(x)$ can be in different coordinate systems during the optimisation.

In this section we show that these difficulties can be circumvented elegantly if M and N are assumed to be isometrically embedded sub-manifolds in Euclidean spaces $\mathbb{R}^k$ and $\mathbb{R}^l$ respectively. We show that in this case the first and second order differential energies presented

above have *equivalent* but much simpler forms in terms of the derivatives with respect to the embedding spaces. Expressing the regularisation functionals in terms of an embedding of the output also allows to use only one global coordinate system for the output, which reduces the algorithmic overhead dramatically.

The assumption of the existence of an isometric embedding into Euclidean space is not very restrictive. Any compact manifold can be isometrically embedded into Euclidean space $\mathbb{R}^k$ for large enough k, see [Nash, 1956]. For a huge class of manifolds an isometric embedding in Euclidean space is known. Often the manifold is even defined as a constrained set in $\mathbb{R}^k$ or given just as a point cloud in $\mathbb{R}^k$, where in both cases the metric is induced from $\mathbb{R}^k$ and the isometric embedding is trivial.

Below, quantities which are defined on M or N are called intrinsic, whereas quantities related to the embedding spaces $\mathbb{R}^k$, $\mathbb{R}^l$ are called extrinsic. The goal will be to represent the above introduced intrinsic expressions with simpler computable extrinsic ones. *We have to stress that in doing this we neither lose the invariance with respect to parametrisation nor do we change the regulariser.*

For simplicity of presentation we split the discussion below. We first consider the case where N is a general Riemannian manifold isometrically embedded in $\mathbb{R}^l$, afterwards the case where M is a general manifold embedded in $\mathbb{R}^k$. The proofs of all theorems are found in Additional Material 4.10.3.

4.5.1 Computation of the Energies for General Output Manifolds

Assume the output manifold N can be isometrically embedded into $\mathbb{R}^l$, and let $i : N \to \mathbb{R}^l$ be the embedding map. Denote by $\Psi : M \to \mathbb{R}^l$ the composition $\Psi = i \circ \phi$. Let z^μ be standard Cartesian coordinates in $\mathbb{R}^l$. Then the differential of Ψ is given as $d\Psi^r_a = \frac{\partial \Psi^\mu}{\partial x^\alpha} dx^\alpha_a \otimes \frac{\partial^r}{\partial z^\mu}$. In order to define derivatives of the differential $d\Psi^r_a$ we again, see Additional Material 4.10.1, need an pull-back connection $\tilde{\nabla} : TM \otimes \Psi^{-1}T\mathbb{R}^l \to \Psi^{-1}T\mathbb{R}^l$ for the mapping Ψ,

$$\tilde{\nabla}_{\frac{\partial}{\partial x^\alpha}} \frac{\partial^r}{\partial z^\mu} := {}^{\mathbb{R}^l}\nabla_{d\Psi(\frac{\partial}{\partial x^\alpha})} \frac{\partial^r}{\partial z^\mu} = 0,$$

which is trivial due to the flatness of the connection of $\mathbb{R}^l$. Because of this property the expressions for the corresponding covariant derivatives expression will simplify significantly. However, note that the coordinate vector $\frac{\partial^r}{\partial y^\mu}$ of N has the derivative $\tilde{\nabla}_{\frac{\partial}{\partial x^\alpha}} \left(di(\frac{\partial^r}{\partial y^\mu}) \right) = \frac{\partial^2 i^\rho}{\partial y^\nu \partial y^\mu} \frac{\partial \phi^\nu}{\partial x^\alpha} \frac{\partial^r}{\partial z^\rho}$. The following theorem shows how intrinsic expressions in ϕ can be expressed in terms of the extrinsic ones in Ψ.

Theorem 4.7. *The following equivalences between intrinsic and extrinsic objects hold,*

$$d\phi^r_a = d\Psi^r_a, \qquad \nabla'_c d\phi^r_a = \left(\tilde{\nabla}_c d\Psi^r_a \right)^\top, \tag{4.13}$$

where $^\top$ *denotes the projection onto the tangent space* $T_{\Psi(x)}N$ *of* N.

The statement of Theorem 4.7 about the connection between the intrinsic and the extrinsic second derivative is visualised in Figure 4.2. For the case where M is a domain in $\mathbb{R}^m$ the above theorem allows to derive a dramatic simplification of the energy expressions.

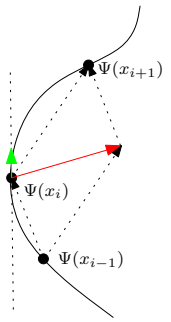


Figure 4.2: *Comparison of extrinsic and intrinsic second derivative: Suppose $\phi : \mathbb{R} \to N$, N the black curve on the left. Thus, $\Psi : \mathbb{R} \to \mathbb{R}^2$, but $\Psi(x) \in N$. If the images $\Psi(x_i)$ of equidistant points x_i in the input manifold $M = \mathbb{R}$ are also equidistant on the output manifold, then Ψ has no acceleration in terms of N, i.e. its intrinsic second derivative in N should be zero. However, the extrinsic second derivative of Ψ in the ambient space, which is marked red in the left figure, is not vanishing in this case. The Eells energy only penalises the intrinsic acceleration, that is, only the component parallel to the tangent space at $\Psi(x_i)$, the green arrow.*

Theorem 4.8. *Let $M \subset \mathbb{R}^m$ and x^α be Cartesian coordinates, then*

$$S_{\text{harmonic}}(\Psi) = \int_M \sum_{\mu=1}^{l} \sum_{\alpha=1}^{m} \left(\frac{\partial^2 \Psi^\mu}{\partial x^\alpha} \right)^2 dx, \tag{4.14}$$

$$S_{\text{biharmonic}}(\Psi) = \int_M \sum_{\mu=1}^{l} \sum_{\alpha=1}^{m} \left[\left(\frac{\partial^2 \Psi^\mu}{\partial x^\alpha \partial x^\alpha} \right)^\top \right]^2 dx, \tag{4.15}$$

$$S_{\text{Eells}}(\Psi) = \int_M \sum_{\mu=1}^{l} \sum_{\alpha,\beta=1}^{m} \left[\left(\frac{\partial^2 \Psi^\mu}{\partial x^\alpha \partial x^\beta} \right)^\top \right]^2 dx. \tag{4.16}$$

4.5.2 Computation of the Eells Energy for General Input Manifolds

Now assume that the input manifold M is isometrically embedded in $\mathbb{R}^k$. This will allow us to construct local parametrisations of M, for which the evaluation of the Christoffel symbols ${}^M\Gamma^\gamma_{\alpha\beta}$ in the second derivative (4.6) is particularly easy. These parametrisations are based on local second order approximations of M around given points $p \in M$.

Proposition 4.9. *Let $x^1, \dots, x^m$ be the coordinates associated with an orthonormal basis of the tangent space at T_pM. Then in Cartesian coordinates z of $\mathbb{R}^k$, the manifold can be approximated up to second order as*

$$z(x) = \big(x^1, \dots, x^m, f^{m+1}(x), \dots, f^k(x)\big),$$

where $f^i(x) = \sum_{\alpha,\beta=1}^m \Pi^i_{\alpha\beta} x^\alpha x^\beta$ and $\Pi^i_{\alpha\beta}$ is the second fundamental form of M at p. If M is a hypersurface in $\mathbb{R}^k$ ($k = m+1$), then we have

$$f^k(x) = \sum_{\alpha=1}^{m} \kappa_\alpha (x^\alpha)^2,$$

if the coordinates x^α are aligned with the principal directions and κ_α are the principal curvatures of M at p.

A simple example of a second-order approximation of a hypersurface is given in Figure 4.3. The principal curvature, also called *extrinsic curvature*, quantifies how much the input manifold bends with respect to the ambient space. Local second-order approximations allow us to compute the second derivative in (4.6) efficiently as the next proposition shows.

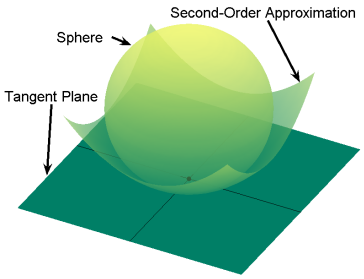


Figure 4.3: *Second-order approximation of a sphere at the south pole on the left. Note, that the principal curvature, also called* extrinsic curvature*, quantifies how much the manifold bends with respect to the ambient space.*

Proposition 4.10. *Given a second-order approximation of M at p as in Proposition 4.9, then for the coordinates x we have that*

$$g_{\alpha\beta}(0) = \delta_{\alpha\beta}, \qquad {}^{M}\Gamma^{\alpha}_{\beta\gamma}(0) = 0.$$

Furthermore, we have at $p \in M$,

$$\left(\tilde{\nabla} d\Psi\right)^{\mu}_{\alpha\beta} = \left[\frac{\partial^2 \Psi^\mu}{\partial x^\beta \partial x^\alpha} - \frac{\partial \Psi^\mu}{\partial x^\gamma} {}^{M}\Gamma^{\gamma}_{\beta\alpha}\right] = \frac{\partial^2 \Psi^\mu}{\partial x^\beta \partial x^\alpha} \tag{4.17}$$

$$= \left[\frac{\partial^2 \Psi^\mu}{\partial z^\beta \partial z^\alpha} + \sum_{r=m+1}^{k} \frac{\partial \Psi^\mu}{\partial z^r} \Pi^r_{\beta\alpha}\right]. \tag{4.18}$$

For a hypersurface M ($k = m+1$), it is $\Pi^r_{\beta\alpha} = \delta_{\beta\alpha}\kappa_\alpha$ if the coordinates x^α are aligned with the principal directions and κ_α are the principal curvatures of M at p.

Note that (4.17) is not an approximation, but the true second derivative of Ψ at p on M. This is due to the following argument: If we allowed for higher order terms in $f^{m+1}, .., f^k$, we could fit M exactly in a local neighbourhood around p, such that x would be coordinates of M and not its second order approximation. However, since the computation of Christoffel symbols at p and of (4.17) requires only second derivatives of $f^{m+1}, .., f^k$ at p, we would obtain identical results.

A straightforward consequence from Proposition 4.10 is Corrolary 4.11 below, which gives simple extrinsic forms for the Eells and biharmonic energy for the case of manifold-valued input. These expressions are derived by replacing the second partial derivatives in (4.15) and (4.16) with the slightly more complicated expression (4.17). We only show the energy densities here, not the integrals, since the z coordinates are different for each point $p \in M$.

Corollary 4.11. *For general input manifolds M and a second order approximation as in Proposition 4.9, we obtain for the energy densities of the Eells and the biharmonic energy of Ψ at p as*

$$\textit{biharmonic:} \qquad \sum_{\mu=1}^{l} \sum_{\alpha=1}^{m} \left[\left(\frac{\partial^2 \Psi^\mu}{\partial z^\alpha \partial z^\alpha} + \sum_{r=m+1}^{k} \frac{\partial \Psi^\mu}{\partial z^r} \Pi^r_{\alpha\alpha}\right)^{\top}\right]^2, \tag{4.19}$$

$$\textit{Eells:} \qquad \sum_{\mu=1}^{l} \sum_{\alpha,\beta=1}^{m} \left[\left(\frac{\partial^2 \Psi^\mu}{\partial z^\beta \partial z^\alpha} + \sum_{r=m+1}^{k} \frac{\partial \Psi^\mu}{\partial z^r} \Pi^r_{\beta\alpha}\right)^{\top}\right]^2. \tag{4.20}$$

The principal curvatures can be computed directly for manifolds given in analytic form. For point cloud data one can estimate them using a local fit with a quadratic function.

4.5.3 Comparison of Intrinsic and Extrinsic Energies

The expression of the intrinsic second derivative in terms of extrinsic quantities allows us to discuss the differences of our approach, which penalises only intrinsic variations of the mapping, to the one recently proposed in [Hofer and Pottmann, 2004; Wallner et al., 2007], where extrinsic variations are penalised. Suppose the output manifold N is isometrically embedded in $\mathbb{R}^l$. One way to learn mappings $\Psi : M \to N \subset \mathbb{R}^l$ is to penalise the extrinsic derivatives in $\mathbb{R}^l$ in the regularisation functional, and to constrain $\Psi(x)$ to lie on N for all $x \in M$. In this section, we will briefly argue why this extrinsic energy has worse properties than our proposed intrinsically defined one. We demonstrate the difference for curves $\gamma : M \to N$, $M \subseteq \mathbb{R}$.

The extrinsic second-order regularisation functional $S_{\text{ex}}(\gamma)$ is given as

$$S_{\text{ex}}(\gamma) = \int_M \|\ddot{\gamma}\|^2 \, dt,$$

where $\ddot{\gamma}$ is the second derivative in $\mathbb{R}^l$. In contrast, the Eells energy $S_{\text{in}}(\gamma)$ reduces for curves to

$$S_{\text{in}}(\gamma) = \int_M \|\nabla_{\dot{\gamma}}\dot{\gamma}\|^2 \, dt.$$

In both cases one has the constraint $\gamma(x) \in N$ for all $x \in M$. The extrinsic and intrinsic derivative are related via $\ddot{\gamma} = \nabla_{\dot{\gamma}}\dot{\gamma} + \mathrm{II}(\dot{\gamma}, \dot{\gamma})$, where $\mathrm{II} : TN \times TN \to NN$ is the second fundamental form of N and NN denotes the normal bundle of N (since N is a submanifold of $\mathbb{R}^l$), see also Figure 4.2. That means that the extrinsic energy penalises the intrinsic tangential acceleration *and* the normal component. We have $\|\ddot{\gamma}\|^2 = \|\nabla_{\dot{\gamma}}\dot{\gamma}\|^2 + \|\mathrm{II}(\dot{\gamma}, \dot{\gamma})\|^2$, and therefore

$$S_{\text{ex}}(\gamma) = S_{\text{in}}(\gamma) + \int_M \|\mathrm{II}(\dot{\gamma}, \dot{\gamma})\|^2 \, dt.$$

Now if N has constant extrinsic curvature as for example the sphere, then $\|\mathrm{II}(\dot{\gamma}, \dot{\gamma})\|^2 = C \|\dot{\gamma}\|^2$ so that the extrinsic energy functional is just a combination of harmonic and Eells energy. For simplicity suppose that we are given only two data points. Using the intrinsic second order energy, we will find a connecting geodesic as the solution of the learning problem in (4.1), since geodesics have zero energy S_{in}. For the extrinsic energy, the harmonic part of the energy aims to contract the curve, thus the minimum of (4.1) will be a geodesic segment that ends short of the training points, depending on the regularisation parameter λ.

While in the above special situation the solutions are at least similar, the extrinsic energy leads to less intuitive solutions in the general case of non-constant extrinsic curvature. The following simple example shows that geodesic segments are no longer minimisers of the extrinsic energy S_{ex}, if the second fundamental form is non-constant. Yet, they would be global minimisers of the intrinsic energy S_{in}.

Assume now that the output manifold N is the graph of a smooth function $f : (0, \infty) \to \mathbb{R}$, that is, $N = \{x \in \mathbb{R}^2 | x_1 > 0, f(x_1) = x_2\}$ with $f(x) = \sqrt{\cosh(x)^2 - 1} / \tanh(x) - 1$. A unit-speed curve in N is given as $\gamma(t) = (\sinh^{-1}(t), f(\sinh^{-1}(t)))^T$, since the length of curves $g(s) = (s, f(s))^T$ for $0 \leq s \leq x$ is given as $\int_0^x (1 + (\frac{df}{ds})^2)^{1/2} ds = \sinh(x)$. Minimisers of the extrinsic energy subject to $\gamma(x) \in N$, $x > 0$, must have vanishing gradient along N, that is $\left\langle W, \gamma^{(4)}(t) \right\rangle = 0$ for all $W \in TN$ where $\gamma^{(4)}(t) = \frac{\partial^4}{\partial t^4}\gamma(t)$. We compute the tangential gradient as $\left\langle \dot{\gamma}(t), \gamma^{(4)}(t) \right\rangle$ and show the results in Figure 4.4.

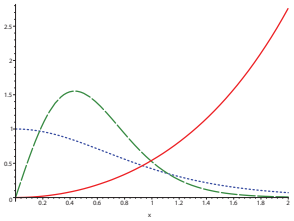


Figure 4.4: *Example showing that geodesics are in general not minimisers of the extrinsic second-order energy. Solid: the manifold N is given as the graph of a function $f : (0, \infty) \to \mathbb{R}$. Dotted: the curvature of N, that is the scalar second fundamental form, at $(x, f(x))^T \in N$ as a function of x. Dashed: gradient of the extrinsic energy $S_{\text{ex}}(\gamma)$ along TN for unit-speed curve $\gamma(t) \in N$. The gradient at $\gamma(t) = (x, f(x))^T$ is plotted as a function of x. While γ is a geodesic in N, the tangential gradient of $S_{\text{ex}}(\gamma)$ does not vanish.*

The tangential gradient does not vanish in areas where the graph of f has non-vanishing extrinsic curvature. This implies that even so γ is a geodesic it is not a minimiser of the extrinsic energy S_{ex}.

4.6 Implementation

A classic route to solve our variational learning problem (4.1) would be to derive the Euler/Lagrange variational equations and to solve these. We have computed these equations in Additional Material 4.10.4, but that leads to a system of coupled fourth-order (partial) differential equations which is numerically very difficult to solve. Similar to finite element methods, we instead tackle the problem by directly minimising the optimisation problem 4.1. This way, only second derivatives are needed, and furthermore no boundary conditions have to be specified explicitly.

In the following we will explain how objective (4.1) can be expressed in terms of a finite number of parameters, and how these can then be optimised efficiently with a pseudo-Newton method to yield the optimal map ϕ. All information about the manifolds that are used in a specific application is made available to the optimisation routine through a number of interface functions. An implementation of these interface functions for the manifolds that are used in the experiments in Section 4.7, namely spheres, combinations thereof, and point clouds, is described afterwards.

Since we aim at using the tools from the previous section, we will throughout this section assume that M and N are isometrically embedded in $\mathbb{R}^k$, $\mathbb{R}^l$ respectively, and the targeted function is thus represented as $\Psi : M \subseteq \mathbb{R}^k \to \mathbb{R}^l$.

4.6.1 The Optimisation

Concerning the representation of Ψ consider the following arguments. If the output space was Euclidean, then the Euler-Lagrange equations of the different energies derived in Theorem 4.29 would be linear differential equations that could elegantly be solved using Green's functions. A certain form of the representer theorem would then guarantee that the minimiser of the objective function of (4.1) is a finite linear combination of these Green's functions [Wahba, 1990], which would allow reducing the function optimisation problem (4.1) to an optimisation problem in the parameters only. However, this result is critically dependent on the linear structure of the output space N, and no simple parametric form exists for

the minimiser of (4.1) if the output is a general Riemannian manifold simply because the set of all mappings from M to N is *not* even a vector space.

Since no simple representation of the function to optimise exists in the general manifold case, we have to resort to some form of discretisation. A straightforward approach would be gridding combined with finite difference approximations for the derivative operators. While we experimented with this at first [Steinke et al., 2008], we now propose to use a collocation-like approach by choosing a flexible smooth parametric function set, the local polynomials. In the future, we also plan to examine finite element methods. Compared to the gridding approach, the local polynomials allow for an analytical computation of the required derivatives, and empirically a good solution in this parametrisation often needed relatively few parameters. Note that the Bayes optimal solution will almost surely not lie in the selected function set, but we can approximate it more and more closely, if we increase the flexibility of the function class through the addition of additional polynomial centres.

Let M be an open subset or submanifold of $\mathbb{R}^k$, then we parametrise the μ-th component of the mapping $\Psi : \mathbb{R}^k \to \mathbb{R}^l$ as a local polynomial of low order, that is,

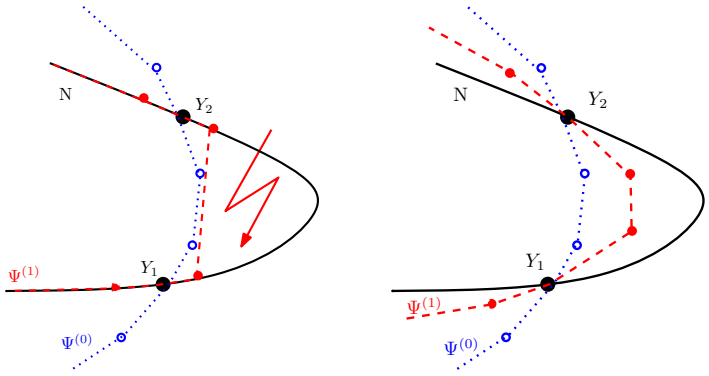
$$\Psi^\mu(x) = \frac{\sum_{i=1}^S k_{\sigma_i}(\|\Delta x_i\|) g(\Delta x_i, w_i^\mu)}{\sum_{j=1}^S k_{\sigma_j}(\|\Delta x_j\|)}.$$

Here, $g(\Delta x_i, w_i^\mu)$ is a first or second order polynomial in Δx_i with parameters w_i^μ, $\Delta x_i = (x - c_i)$ is the difference of x to the local polynomial centres c_i, and $k_{\sigma_i}(x) = k(r \equiv \frac{x}{\sigma_i}) = \frac{1}{6}(1-r)_+^6(6 + 36r + 82r^2 + 72r^3 + 30r^4 + 5r^5)$ is a compactly supported smoothing kernel with bandwidth σ_i [Schaback, 1995]. We choose the local polynomial centres c_i approximately uniformly distributed over M, thereby adapting the function class to the shape of the input manifold M. If we stack all parameters w_i^μ into a single vector w, then Ψ and its partial derivatives are just linear functions of w, which allows computing these values in parallel for many points using simple matrix multiplication.

We compute the energy integral (4.5) as a function of w, by summing up the energy density over an (approximately) uniform discretisation of M. The projection onto the tangent space, used in (4.19) and (4.20), and the second order approximation for computing intrinsic second derivatives, used in (4.19) and (4.20), are manifold specific and are explained below.

If N is non-Euclidean, which is the case we are mostly interested in, then we need to satisfy the constraints $\Psi(x) \in N$ for $x \in M$ throughout the optimisation process. We soften this condition and add it to the objective function as $\gamma \int_M d(\Psi(x), N)^2 dx$, where $d(y, N)$ denotes the Euclidean distance in $\mathbb{R}^l$ of a point $y \in \mathbb{R}^l$ to the manifold N. We increase the weight γ during the iterative optimisation process until all points are within a given prespecified distance of N. As initial solution, we compute the free solution, i.e. where N is assumed to be $\mathbb{R}^l$, in which case the problem becomes convex quadratic, since there are no constraints and no location dependent projections. The iteratively increasing penalisation of the distance to the manifold leads to a slow settling of the initial solution towards the target manifold. In contrast to a simple projection of the initial solution onto N, as done in [Steinke et al., 2008], this procedure is much more robust. The projection of Ψ can lead to large distortions which, in turn, can cause the optimisation to become numerically unstable or to stop in local minima. This problem is visualised with an example in Figure 4.5.

However, if we allow for $\Psi(x) \notin N$ during the optimisation, then we have to declare how the projection of the second derivative of Ψ onto the tangent space is meant and



(a) direct projections (b) soft constraint

Figure 4.5: *(a) Projecting the initial, unconstrained solution $\Psi^{(0)}$ directly onto the target manifold $N \subset \mathbb{R}^l$ can lead to large deformations in high curvature regions. Large deformations can cause the computation of the second derivative of $\Psi^{(1)}$ to become numerically unstable. (b) A slow settling of the solution towards the manifold increases numerical stability.*

how we deal with the loss term in this case. We propose to determine the projection using the iso-distance manifolds $N_{\Psi(x)} = \{y \in \mathbb{R}^l | d(y, N) = d(\Psi(x), N)\}$ of N. For the loss we use the geodesic distance between the projection of $\Psi(X_i)$ onto N and Y_i, that is, $d_N(\text{argmin}_{y \in N} \|\Psi(X_i) - y\|, Y_i)$. These two constructions are sensible, since as the weight γ of the constraint $\gamma \int_M d(\Psi(x), N)^2 dx$ increases, Ψ will approach the manifold N, and both terms converge to the corresponding operations directly executed on the manifold N. The computation of $d(\Psi(x), N)$, the projection onto tangent spaces of iso-distance manifolds, and the computations of geodesic distances on N are again manifold specific and can be found below.

Having expressed all parts of the optimisation problem (4.1) in terms of the parameters w, we obtain an unconstrained non-linear optimisation problem $\min_w f(w)$ which we solve using a pseudo-Newton method as follows. For each update we compute the true gradient $\nabla f(w)$, but only an approximation of the Hessian $\tilde{\nabla}^2 f(w)$, that is, the Hessian of $f(w)$ but without the projection onto the tangent space of N in the Eells energy. We then perform a line search in the direction $-(\tilde{\nabla}^2 f(w))^{-1} \nabla f(w)$, and update w accordingly. Computing only an approximation of the exact Hessian is advantageous for two reasons. First of all it is computationally much simpler since no second derivative of the projection operator is required. Secondly, it adds to the robustness of the algorithm due to the following argument. The Eells energy does not penalise oscillations in normal direction of the manifold. While these cannot occur if $\Psi(w) \in N$ is strictly enforced, it can occur during the optimisation process where we have relaxed that constraint. Using the approximate Hessian we discourage such distorting oscillations, however we are still guaranteed to minimise the true Eells energy. This can be seen as follows. The approximate Hessian of the Eells energy is positive semi-definite. If we assume that the markers fix an optimal linear transformation, then the combined approximate Hessian of the whole objective (4.1) is positive definite, and the multiplication of the gradient with the inverse of this matrix just corresponds to a change of the used inner product of the Euclidean embedding space. We thereby do *not* change the optimisation objective, and this pseudo-Newton type approach thus has at least the convergence guarantees of simple gradient descent. Finally, note that computation of

the descent direction $-(\tilde{\nabla}^2 f(w))^{-1}\nabla f(w)$ can be performed efficiently with sparse methods, since the compact support of the smoothing kernel k implies sparsity of approximate Hessian $\tilde{\nabla}^2 f(w)$.

4.6.2 Manifold Operations

It remains to describe how we perform the required manifold specific operations. Firstly, we need to be able to project onto the tangent space of the output manifold N and its iso-distance manifolds. Secondly, we need to be able to project from the embedding space $\mathbb{R}^l$ of the output manifold onto N, and thirdly, we require geodesic distances on N. Furthermore, for curved input manifolds M we need the principle curvatures to compute the intrinsic second derivatives, see Proposition 4.10.

In this section we focus on the manifolds that we used in our experiments, that is, spheres $\mathcal{S}^{n-1} \subseteq \mathbb{R}^n$ in different dimensions, combinations thereof, and two dimensional surfaces in $\mathbb{R}^3$ which are given as point clouds with surface normals. Note that the projection $P^\top$ onto the tangent space of N and its iso-distance manifolds can conveniently be performed for any embedded manifold, if we have access to a signed distance function η of the manifold N. The projection $P^\top$ at $x \in \mathbb{R}^l$ is then given as $P^\top(x) = 1 - \frac{1}{\|\nabla\eta(x)\|^2}\nabla\eta(x)\nabla\eta(x)^T$.

For the unit spheres $\mathcal{S}^{n-1} \subset \mathbb{R}^n$, for example the circle $\mathcal{S}^1$ or the 3D sphere $\mathcal{S}^2$, the signed distance function is simply given as $\eta(x) = 1 - \|x\|$. The projection from the embedding space onto the sphere is trivial and the geodesic distance is $d(x,y) = \arccos\left(\frac{<x,y>}{\|x\|\|y\|}\right)$ for $x, y \in \mathcal{S}^{n-1}$. Furthermore, the principle curvatures of $\mathcal{S}^2$ both have the value -1 for all $x \in \mathcal{S}^2$.

Now consider combinations of spheres with the direct sum metric, for example, $\mathcal{S}^{1,2} = \mathcal{S}^1 \times \mathcal{S}^1$ with metric $g^{\mathcal{S}^{1,2}} = g^{\mathcal{S}^1} \oplus g^{\mathcal{S}^1}$. Here, all the manifold operations can be performed component-wise. The geodesic distance is also just the sum of the corresponding two geodesic distances on $\mathcal{S}^1$. This is because the curve γ that minimises the distance, $\int \sqrt{g(\dot\gamma,\dot\gamma)}dt$, between two points on $\mathcal{S}^{1,2}$ also minimises the squared distance, the harmonic energy $\int g(\dot\gamma,\dot\gamma)dt$ [Lee, 1997; Eells and Sampson, 1964]. The harmonic energy, however, decomposes trivially. Note furthermore that, if the quadratic loss is used, then the complete learning objective (4.1) can be decomposed into two independent problems, which can be solved separately. In contrast, if $\mathcal{S}^{1,2}$ is given the metric of a torus embedded in $\mathbb{R}^3$, the components are coupled non-trivially and no decomposition is possible.

For point cloud surfaces in 3D, there exist many known methods to construct signed distance functions, e.g. [Ohtake et al., 2003; Steinke et al., 2005]. Here, we choose a particularly simple approach to compute the signed distance value $\eta(p)$ for some test point $p \in \mathbb{R}^l$: we first search for the closest point to p in the point cloud, then compute a local second order approximation there based on the 10 nearest neighbours using least squares, and finally use the distance to this second order approximation as the desired signed distance function η. The computation of the distance to the local second order approximation $(x^1, x^2, f(x^1, x^2))$ involves solving third order equations. However, since we assume that our manifolds are densely sampled, we will always obtain local coordinates (p^1, p^2, p^3) for p with small values for p^1, p^2. Thus, a good approximation to the true distance is to use $\eta(p) = p^3 - f(p^1, p^2)$. The so-constructed signed distance function readily allows to compute the required projections onto the tangent spaces. Furthermore, the same procedure also allows to determine the closest point on N for a given query point, just using

$(p^1, p^2, f(p^1, p^2))$. If the point cloud serves as an input manifold M, the same local second order approximations give trivial access to the required principal curvatures.

What remains is the geodesic distance for point clouds. One can either use approaches like [Kimmel and Sethian, 1998], or alternatively geodesic distances can be computed using the length of a curve which minimises the harmonic energy and whose endpoints are fixed at the two points of interest [Steinke et al., 2008]. However, since in our surface registration problem we used rather large weights for the loss, $\Psi(X_i)$ and Y_i were always very close on the surface. In this case the geodesic distance can be well approximated by the Euclidean one, so that for performance reasons we directly used the Euclidean distance.

4.7 Experiments

We now show some illustrative examples for regression between Riemannian manifolds. The examples show an increasing amount of theoretical and algorithmic complexity.

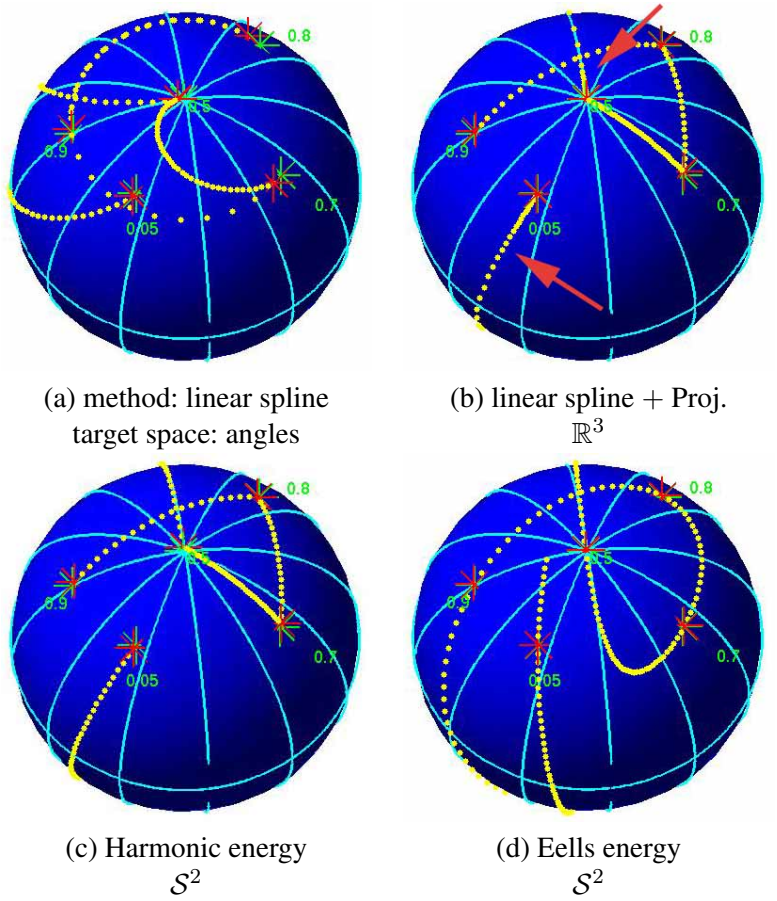
4.7.1 Curves on Spheres

To understand the basic problems of manifold-valued regression and to get a qualitative idea of the features of our approach, it is helpful to discuss Figure 4.6 in detail. The aim is here to fit a curve on the sphere $\mathcal{S}^2 \subseteq \mathbb{R}^3$ through 6 given data points. Thus, we have a regression problem $\phi : [0, 1] \to \mathcal{S}^2$.

A naive first idea to solve this problem could be to parametrise the surface of the sphere using spherical coordinates, and to interpolate the coordinates of the given data points using linear splines (For visualisation purposes we use linear splines corresponding to first order differential energies here). This is computationally attractive since the coordinates form a linear space, such that the splines can be computed using simple basis function expansions. However, as shown in Figure 4.6(a), no path can go through the parametrisation boundary at $-\pi$ and π, and moreover, the geometry is heavily distorted by the non-linear parametrisation mapping from $\mathcal{S}^2$ to $(-\pi, \pi) \times (0, \pi)$. Another naive idea, shown in Figure 4.6(b), is to first compute a linear spline in $\mathbb{R}^3$ and then project it radially onto the sphere. While the trajectory can now surround the sphere, the metric is still distorted through the projection. This can be seen in that the yellow points which are equally spaced in the input, are not equally spaced in the output, see the locations indicated by the red arrows in Figure 4.6(b).

Manifold adapted approaches are much better suited for this regression problem. In Figure 4.6(c), the harmonic energy (4.3) is used in the learning objective (4.1). Note that the yellow points are now equally spaced between any two data points, up to small distortions resulting from the 2D visualisation. However, since the minimisers of the harmonic energy are piecewise geodesic [Machado et al., 2006], the curve is not differentiable at the data points. It also does not extend outside of the first/last marker. Using the Eells energy both these problems are avoided, see Figure 4.6(d). The curves are smooth and they extrapolate linearly, or more precisely geodesically.

Turning to quantitative analysis, we should expect that a manifold adapted approach is much better at approximating some unknown curve from which just a few noisy observations are available. We tested this claim with a ground-truth curve given in spherical coordinates as $\theta(t) = (40t^2, 1.3\pi t + \pi \sin(\pi t))$. The K training inputs were sampled uniformly from



(a) method: linear spline
target space: angles

(b) linear spline + Proj.
$\mathbb{R}^3$

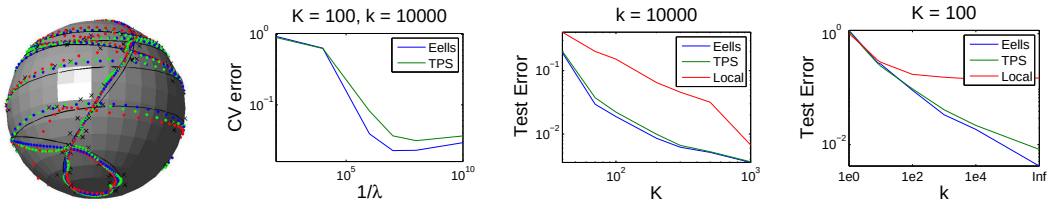
(c) Harmonic energy
$\mathcal{S}^2$

(d) Eells energy
$\mathcal{S}^2$

Figure 4.6: *The interval* $[0,1]$ *is mapped onto the unit sphere* $\mathcal{S}^2$ *in 3D. Green markers show the given data points* $Y_i \in \mathcal{S}^2$*, respective training times* $X_i \in [0,1]$ *are given as numbers close-by. Red markers indicate* $\Psi(X_i)$ *for the approximating spline* $\Psi : [0,1] \to \mathcal{S}^2$*. Yellow dots mark the* Ψ*-images of equally spaced points in* $[0,1]$.

$[0,1]$, the outputs were perturbed by "additive" noise from the von Mises distribution with concentration parameter k. The von Mises distribution is the maximum entropy distribution on the sphere for fixed mean and variance [Mardia and Jupp, 2000], and thus is the analogue to the Gaussian distribution for spheres. In the experiments the optimal regularisation parameter λ was determined by performing 10-fold cross-validation and the experiment was repeated 10 times for each size of the training sample K and noise parameter k to obtain statistical significance.

We compare our framework for non-parametric regression between manifolds with standard cubic smoothing splines in $\mathbb{R}^3$ – the equivalent of thin-plate splines (TPS) for one input dimension – projected radially on the sphere, and also with the local manifold-valued Nadaraya-Watson estimator of [Davis et al., 2007]. As can be seen in Figure 4.7, our globally regularised approach performs significantly better than [Davis et al., 2007] for this task. One can observe in Figure 4.7(a) that even in places where the estimated curve of [Davis et al., 2007] follows the ground truth relatively closely, the spacing between points varies greatly. These sampling dependent speed changes, that are not seen in the ground truth curve, cannot be avoided without a global smoothness prior such as for example the Eells energy. The Eells approach also outperforms the projected TPS method, in particular for



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

Figure 4.7: *Regression from* $[0, 1]$ *to the sphere* $\mathcal{S}^2$. *(a) Noisy data samples (black crosses) of the black ground-truth curve. The blue dots show the estimated curve for our Eells-regularised approach, the green dots depict thin-plate splines (TPS) in* $\mathbb{R}^3$ *radially projected onto the sphere, and the red dots show results for the local approach of [Davis et al., 2007]. (b) Cross-validation errors for given sample size* K *and noise concentration* k. *Von-Mises distributed noise in this case corresponds roughly to Gaussian noise with standard deviation* 0.01. *(c) Test errors for different* K, *but fixed* k. *In all experiments the regularisation parameter* λ *is found using cross-validation. (d) Test errors for different* k, *but fixed* K.

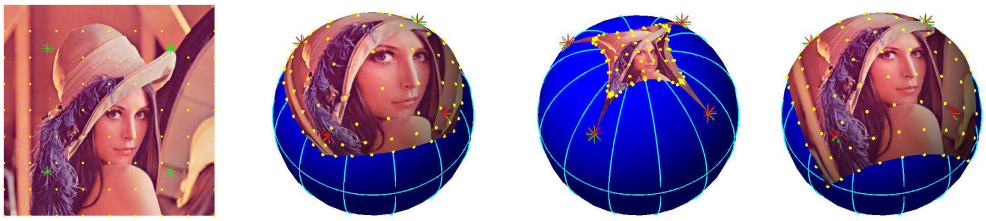
small sample sizes and reasonable noise levels. For a fixed noise level of $k = 10000$ we showed using a paired t-test that our reduction in test error is statistically significant at level $\alpha = 5\%$ for the sample sizes $K = 70, 200, 300, 500$. Clearly, as the curve is very densely sampled for high K, both approaches perform similar, since the problem then is essentially local and the manifold is locally linear. However, for small sample sizes, i.e. for situations where the a priori information is more important, the TPS method is outperformed by the proposed Eells-regularised approach, showing that this is a much more natural prior for this situation.

4.7.2 Mapping Two-Dimensional Patches

Similarly to the last section, we demonstrate qualitative differences between projected TPS, the harmonic energy and the Eells energy solution, here. However, we now consider the two-dimensional input manifold $M = [0, 1]^2 \subset \mathbb{R}^2$, that is, the task is to map a two-dimensional patch onto 3D surfaces.

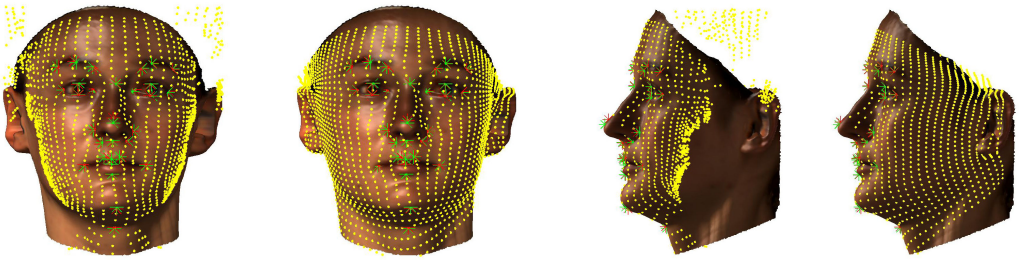
This setup is useful for many geometric modelling tasks such as surface parametrisation, re-meshing, or texture mapping. For example, one could use a regular grid mapped onto the surface of an object to reorganise the mesh according to a rectangular 2D coordinate system. This often improves the compressibility of a mesh, makes it easier to control and deform the mesh, and increases the numerical stability of many algorithms that are run on the mesh afterwards [Kalberer et al., 2007]. For this parametrisation task, one often computes mappings from the surface to $\mathbb{R}^2$, see [Sheffer et al., 2006] for an overview. However, there are also many applications where the inverse mapping is required. In this case, one could try to invert the forward mapping, but this may be costly and the estimated forward mapping need not even be invertible. Alternatively, one could directly estimate the inverse mapping from the $\mathbb{R}^2$ domain onto the manifold using our proposed approach.

In Figure 4.8 we compare different approaches targeting the sphere $\mathcal{S}^2 \in \mathbb{R}^3$. In (b), we first compute the thin-plate spline solution in $\mathbb{R}^3$, which in this case yields a plane cutting through the 4 given markers. We then project the plane radially onto the sphere. Observe the extreme fish-eye distortion resulting from projection. In (c), we show results for our varia-



(a) original in $\mathbb{R}^2$ (b) TPS to $\mathbb{R}^3$ + proj. (c) harmonic $\mathcal{S}^2$ (d) Eells $\mathcal{S}^2$

Figure 4.8: *The Lena image (a) is used to visualise a mapping from the unit square in $\mathbb{R}^2$ to the unit sphere $\mathcal{S}^2$ in $\mathbb{R}^3$. Green markers show the given data point pairs, red stars on $\mathcal{S}^2$ denote positions of the input markers in $\mathbb{R}^2$ mapped to the sphere by the approximating spline. TPS means thin-plate spline mapping from $\mathbb{R}^2$ to $\mathbb{R}^3$ and then projected onto $\mathcal{S}^2$.*



TPS to $\mathbb{R}^3$ + Proj. Eells to surface manifold TPS to $\mathbb{R}^3$ + Proj. Eells to surface

Figure 4.9: *Mapping a regular grid in $\mathbb{R}^2$ (yellow points) onto a face manifold in $\mathbb{R}^3$. Green and red markers as in Figure 4.8.*

tional setting, but using the harmonic energy. This approach is commonly used in geometric modelling, e.g., [Zayer et al., 2005], although mostly targeting linear spaces. The mapped image does not fill the convex hull of the training points, and we observe an unnatural contraction of the image. This is why the harmonic energy is traditionally only used for input domains without boundary, or when the output boundary can be fixed a priori. While there exist some methods to alleviate this problem [Zayer et al., 2005], a theoretically clean way would be to use the proposed Eells energy as a regulariser, see (d). Since the Eells energy does not try to minimise the distances between the points, but the variation of distances, it is much less prone to contraction of the image. It allows to extrapolate nicely out of the convex hull of the marker points. Furthermore, the distortion minimising property of the Eells energy can be observed here nicely. While it is not possible to exactly map all geodesics in the input to geodesics in the output, the Eells regularised approach works performs well in this respect compared to the projection approach in (b).

Similar effects for a less symmetric 3D object are observed in Figure 4.10, which shows two types of regressions from $[0, 1]^2$ to a face manifold guided by 30 markers. The markers were placed on feature points of the face such as eyes and mouth, their input position in $\mathbb{R}^2$ was determined by projecting the 3D points onto the surface of a vertical cylinder through the head.

4.7.3 Surface / Head Correspondence

Computing correspondence between the surfaces of different, but similar objects, such as for example human heads, is a central problem in shape processing. A dense correspondence map, that is, an assignment of all points of one head to the anatomically equivalent points on the other head, allows one to perform morphing [Schölkopf et al., 2005], or to build linear object models [Blanz and Vetter, 1999], also known as active appearance models [Cootes et al., 2001], which are flexible tools for computer graphics as well as computer vision. While the problem is well-studied, it remains a difficult problem which is still actively investigated. Most approaches minimise a functional that consists of a local similarity measure and a smoothness functional or regulariser for the overall mapping. Motivated by the fact that the Eells energy favours simple "linear" mappings, we propose to use it as regulariser for correspondence maps between surface manifolds. For testing and highlighting the role of this "prior" independently of the choice of local similarity measure, we formulate the dense correspondence problem as a non-parametric regression problem between manifolds where 55 point correspondences on characteristic local texture or shape features are given (Only on the forehead we fix some less well-defined markers, to determine a relevant length-scale).

It is in general difficult to evaluate correspondences numerically, since for different heads anatomical equivalence is not easily specified. Here, we have used a subset of the head database of [Blanz and Vetter, 1999] and considered their correspondence as ground-truth. These correspondences are known to be perceptually highly plausible. We took the average head of one part of the database and registered it to the other 10 faces, using the mean distance to the correspondence of [Blanz and Vetter, 1999] as error score. Apart from the average deviation over the whole head, we also show results for an interior region, see Figure 4.10(d), for which the correspondence given by [Blanz and Vetter, 1999] is known to be more exact compared to other regions as, for example, around the ear or below the chin.

We compared our approach against [Schölkopf et al., 2005] and a thin-plate spline (TPS) like approach. The TPS method represents the initial solution of our approach, that is, a mapping into $\mathbb{R}^3$ minimising the TPS energy (4.9), which is then projected onto the target manifold. [Schölkopf et al., 2005] use a volume-deformation based approach that directly finds smooth mappings from surface to surface, without the need of projection, but their regulariser does not take into account the true distances along the surface. We did not compare against [Davis et al., 2007], since their approach requires computing a large number of geodesics in each iteration, which is computationally prohibitive on point clouds. In order to obtain a sufficiently flexible, yet not too high-dimensional function set for our implementation, we place polynomial centres c_i on all markers points and also use a coarse, approximately uniform sampling of the other parts of the manifold. Free parameters, that is, the regularisation parameter λ and the density of additional polynomial centres, were chosen by 10-fold cross-validation for our and the TPS method, by manual inspection for the approach of [Schölkopf et al., 2005].

One computed correspondence example is shown in Figure 4.10, the average over all 10 test heads is summarised in the table below.

	TPS	Eells	[Schölkopf et al., 2005]
Mean error for the full head in mm	2.90	2.16	2.15
Mean error for the interior in mm	1.49	1.17	1.36

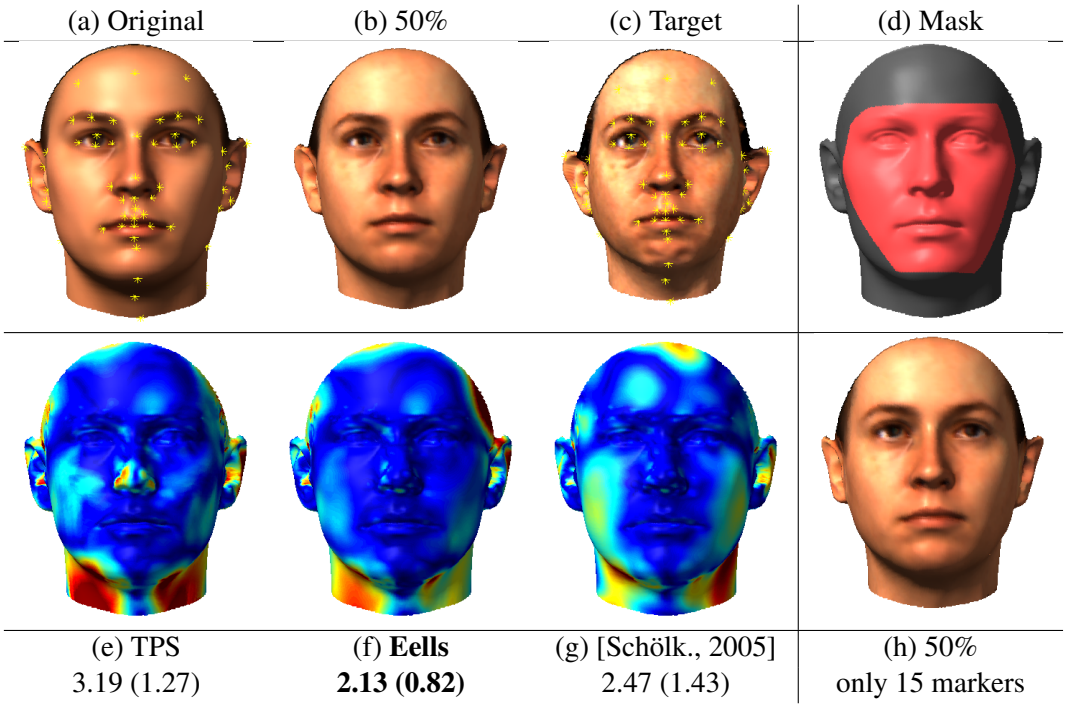


Figure 4.10: *Correspondence computation from the original head in (a) to the target head in (c) with* 55 *markers (yellow crosses). A resulting* 50% *morph using our method is shown in (c). Distance of the computed correspondence to the correspondence of [Blanz and Vetter, 1999] is colour-coded in (e) - (g) for different methods. The numbers below give the average distance in mm over the whole head, in brackets the average over an interior region (red area in (d). Using our method with only 15 markers, see (h), still yields visually plausible morphing results.*

The proposed manifold-adapted Eells approach performs much better than the TPS method, especially in regions of high curvature such as around the nose as the error heatmaps in Figure 4.10 show. Compared to [Schölkopf et al., 2005], our method finds a smoother, more plausible solution, also on large texture-less areas such as the forehead or the cheeks.

We also tried using many less markers with our Eells energy-based method. While the alignment of small texture details then becomes troublesome which negatively affects a numeric evaluation against [Blanz and Vetter, 1999], the overall visual impression is still fairly good, see Figure 4.10(h). This shows once more, that the Eells energy is a suitable prior for mappings between 3D object surfaces.

4.7.4 Learning of Task-Space Tracking

Now, consider a skeleton based model in animation or robotics. As a running example we use a model of a robot arm, see Figure 4.11(a). Most movement tasks are not defined through the model's joint angles $\boldsymbol{q} \in \mathcal{S}^{1,n} = \mathcal{S}^1 \times \cdots \times \mathcal{S}^1$ but rather by the motion of an end-effector $\boldsymbol{x} \in \mathbb{R}^m$, the fingertip. Thus, task-space planning and control requires the inverse kinematic mapping of the task onto the joint space.

Most interesting models are redundant $n > m$, i.e. there is a whole set of joint angles which all put the finger tip at the same location. Some of these will look natural, others won't. A

controller that just focuses on keeping the end effector on the desired trajectory may thus lead to rather undesirable postures. In practice it may be quite hard to specify all (soft) constraints to avoid such postures for a high-dimensional system explicitly, and it may be much easier to specify a number of example postures. We therefore propose to generate joint-space trajectories that stay close to previously observed postures. The necessary generalisation of the examples to a complete map from the task space to the preferred postures in joint space can be learnt well with our proposed approach for manifold-valued regression.

Typically, redundancy resolution is achieved by pulling the robot towards a single rest posture as implemented for example in the 3DSMax HI controller. In this case no generalisation is necessary. Alternatively, learning of postures has been proposed by [Grochow et al., 2004] who use Gaussian process regression. However, since some joints can rotate by 360° our manifold-adapted regression is much better suited for such a situation.

Formally, we assume that we are given a desired path $\boldsymbol{x}_d(t) \in \mathbb{R}^m$ of the finger tip. At time t, we aim at determining $\delta\boldsymbol{q}$ in the model's joint angles $\boldsymbol{q} \in \mathcal{S}^{1,n}$ such that the new posture $\boldsymbol{q} + \delta\boldsymbol{q}$ with tip position $\boldsymbol{x}(\boldsymbol{q} + \delta\boldsymbol{q})$ is close to the desired position $\boldsymbol{x}_d(t)$ and at the same time is similar to training postures in this region of task space. For generalising locally preferred postures $\boldsymbol{q}_1, .., \boldsymbol{q}_k$ at positions $\boldsymbol{x}_1, .., \boldsymbol{x}_k$ to all reachable positions in task space, we use our manifold-valued regression approach to learn a mapping $\boldsymbol{q}_{\text{pred}} : \mathbb{R}^m \to \mathcal{S}^{1,n}$. We then choose $\delta\boldsymbol{q}$ such that it solves the optimisation problem

$$\begin{aligned} \min{}_{\delta\boldsymbol{q}} \quad & \left\| [\boldsymbol{x}(\boldsymbol{q} + \delta\boldsymbol{q}) - \boldsymbol{x}] - \delta\boldsymbol{x}_d - \kappa[\boldsymbol{x}_d(t) - \boldsymbol{x}] \right\|^2 \\ & + \lambda_1 \left\| \delta\boldsymbol{q} \right\|^2 + \lambda_2 d^2_{\mathcal{S}^3_1}(\boldsymbol{q} + \delta\boldsymbol{q}, \boldsymbol{q}_{\text{pred}}(x)). \end{aligned} \tag{4.21}$$

Firstly, this cost tries to keep the finger tip on the desired trajectory with a feedback term with gain κ. Secondly, we prefer small steps $\delta\boldsymbol{q}$, and lastly try to minimise the distance between $\boldsymbol{q} + \delta\boldsymbol{q}$ and suitably generalised training examples $\boldsymbol{q}_{\text{pred}}$. The trade-off between the different objectives is controlled by the weighting coefficients λ_1 and λ_2. The presented control law, has local, data-derived preferred postures instead of a single global rest posture which helps to avoid unnatural postures.

Taking the derivative of (4.21) with respect to $\delta\boldsymbol{q}$ and equating to zero we arrive at the following control law,

$$\delta\boldsymbol{q} = (\boldsymbol{J}\boldsymbol{J}^T)^{-1}\boldsymbol{J}^T \left[\lambda_1 \big(\delta\boldsymbol{x}_d - \kappa[\boldsymbol{x}_d(t) - \boldsymbol{x}] \big) + \lambda_2 \nabla d^2_{\mathcal{S}^3_1}(\boldsymbol{q} + \delta\boldsymbol{q}, \boldsymbol{q}_{\text{pred}}(x)) \right]$$

where $\boldsymbol{J}$ is the forward kinematic Jacobian $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}}(\boldsymbol{q})$.

The presented method is evaluated on the three link ($n = 3$) arm model, see Figure 4.11(a). For better visualisation we chose a planar configuration ($m = 2$). Many postures $\boldsymbol{q}$ yield the same end effector location $\boldsymbol{x}$, see Figure 4.11(b). Training postures in Figure 4.11(c) are bent to the right for points $\boldsymbol{x}$ right of the base, to the left otherwise. From 15 examples (black crosses in Figure 4.11(d)) we learn the function $\boldsymbol{q}_{\text{pred}}(x)$; its first component is colour coded in Figure 4.11(d). Note the direct transition from $-\pi$ to π would be impossible with normal thin-plate splines, since they are not aware of the fact that π and $-\pi$ actually encode the same angle. While the standard resolved motion rate controller [Nakanishi et al., 2005; Spong et al., 2006] ($\lambda_2 = 0$) results in intuitively quite unnatural poses (red boxes in Figure 4.11(f,g)) despite a null-space term, ours stays close to the more natural training set. Also, when plotting the middle and outer angles — for which the training data imply a kind of soft constraints, see gray areas in Figure 4.11(h) — our controller consistently stays

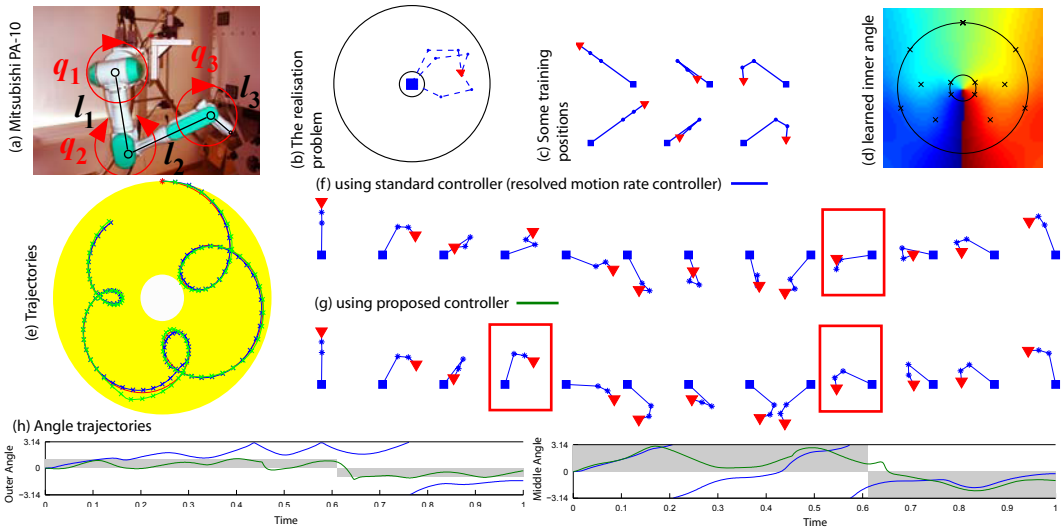


Figure 4.11: *(a) Example system: Mitsubishi PA-10 with three planar degrees of freedom where two have no joint limits (the others are locked). (b) Many postures of a three link arm in two dimensions yield the same tip position. (c) Some training postures. (d) The inner most angle of the arm generalised to the unit square in task space, $\mathbb{R}^2$. Angle $-\pi$ corresponds to dark blue, π to dark red, training points are marked as black crosses. (e) The desired task space trajectory (red) is followed by both the resolved motion rate controller [Spong et al., 2006] (blue) and our controller (green). The reachable space is yellow. (f,g) Postures during the trajectory. (h) Inner and outer angle plotted over time. The gray areas show the region of the training values for the current x position (right hand side positive angles, left hand side negative ones).*

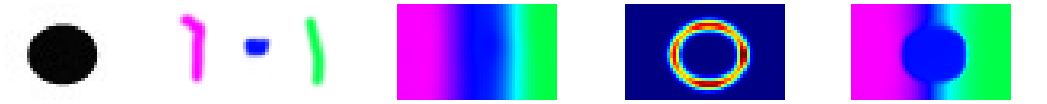
closer while full-filling the task to follow $\boldsymbol{x}_d(t)$ equally well as the default approach, see Figure 4.11(e).

4.7.5 Colour Interpolation

Another potential field of application for manifold-valued splines is colour processing, since perceptually colours have a circular structure [Shepard, 1980]. This property is used in the HSV colour space, where H, the hue value, is a circular variable. Potential applications of our regression framework include colourisation as in [Levin et al., 2004] or image compression which will be discussed here.

For smoothing colour values over a gray-scale image, that is, regression of type $\phi : \Omega \subset \mathbb{R}^2 \to \mathcal{S}^1$ where Ω is the image domain, it makes sense to take into account the presence of edges in the intensity. Edges can be included via a non-uniform metric in the input space. A one pixel distance could be termed large, if it crosses an edge, and small otherwise. This way our smoothing spline which varies slowly in units measured by the metric could express sharp changes over edges, whereas it would vary slowly within objects.

We define metric $g_{ij}(x) = a(x)\delta_{ij}$ on M with $a : \Omega \to \mathbb{R}_+$, $a(x) = \|\nabla I(x)\|^2$, where $\nabla I(x)$ is the gradient of the gray-scale image. While it is not obvious how to embed the thus defined manifold M isometrically into a Euclidean space, we can compute the derivative $\nabla_a d\Psi_b^r$ much more easily here. For $\Psi : \Omega \to \mathbb{R}^2$, the Christoffel symbols ${}^M\Gamma_{\alpha\beta}^{\gamma}$ necessary



Original (a) | Marker data (b) | TPS uniform metric (c) | Metric (d) | TPS adapted metric (e)

Figure 4.12: *Image (a) is coloured by interpolating the colours in (b) in HSV colour space, the H channel is modelled as $\mathcal{S}^1$. (c) shows results for the Eells energy with a uniform metric. However, we can extract edges from the original image (a) and use them as a scalar metric (d). The Eells interpolation then does not interpolate across edges (e), as the metric implies a large distance between the inner and the outer area of the circle.*

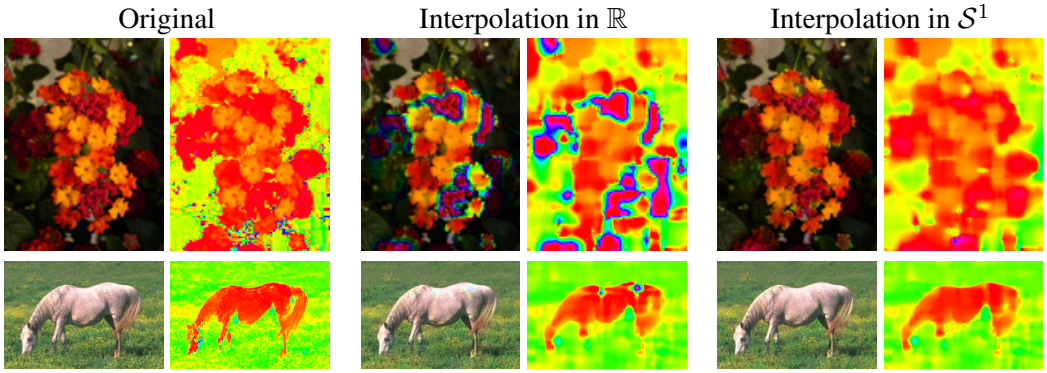


Figure 4.13: *The original images (left) are compressed via a HSV space method. During compression we randomly discard 98% of the H channel of the original images (left column right), but we keep the full S and V information. At decompression time, we interpolate the H values either using normal splines from the image pixels to $[0,1] \in \mathbb{R}$ (middle column), or the Eells energy for splines targeting the circle $\mathcal{S}^1$ (right column). We obtain the H images shown in the right columns. When combining the interpolated H channels with the additionally stored S and V channels we obtain the images shown to the left of the H images.*

for (4.6) follow from ${}^M\Gamma^\kappa_{\beta\alpha} = \frac{1}{2}g^{\kappa\mu}\left(\partial_\beta g_{\alpha\mu} + \partial_\alpha g_{\beta\mu} - \partial_\mu g_{\alpha\beta}\right)$ [Lee, 1997] by simple calculation. It is

$$\nabla'_b d\Psi^r_c = \left[\frac{\partial^2\Psi^\mu}{\partial x^\beta \partial x^\alpha} - \frac{\partial_\beta a}{2a}\frac{\partial\Psi^\mu}{\partial x^\alpha} - \frac{\partial_\alpha a}{2a}\frac{\partial\Psi^\mu}{\partial x^\beta} + \sum_\gamma \frac{\partial_\gamma a}{2a}\frac{\partial\Psi^\mu}{\partial x^\gamma}\right] dx^\beta_b \otimes dx^\alpha_c \otimes \frac{\partial^r}{\partial y^\mu}.$$

This expression is linear in Ψ. It can easily be included into the optimisation framework described in Section 4.6.

The effects of a non-uniform metric for smoothing over images are demonstrated in Figure 4.12, where we aim at colouring a black and white image of a circle (a). We interpolate given H colour values (b) over the image, fixing the S and V channel values to 1. A uniform metric (c) misses to take into account the shape of the circle. In (d), we then compute the norm of the (a)-image gradients to be used as the multiplier $a(x)$ of the metric δ_{ij}. We then arrive at an interpolation that is much better suited to the image structure (e).

The same technique is used for image compression in Figure 4.13. The compression consists of the following steps: first, we transform the RGB image into HSV colour space. We sample randomly 500 pixels of H values, corresponding to $2-3\%$ of all values. We store

these values and also the S and V components for the whole image. During decompression we interpolate the H channel of the image using out proposed Eells regularised approach. The mapping $\Psi : \mathbb{R}^2 \to \mathcal{S}^1$ is learned using an edge-adapted metric as above, where the edges are extracted from the stored S and V channel. The HSV colour image is finally transformed back to normal RGB values. Some experimental results are summarised below. RGB values range from 0 to 1, the error is the RGB root mean squared error over the whole image.

	Horse	Flower
Image size	135 x 200	133 x 100
Error interpol. in $\mathbb{R}$	0.029	0.144
Error interpol. in $\mathcal{S}^1$	0.028	0.042

While the overall compression rate and quality is certainly not state-of-the-art in well-developed image compression, the example may nevertheless show that manifold-valued regression is able to capture important regularities in natural datasets such as colour images. It might be possible to include such knowledge into a more sophisticated state-of-the-art compression scheme in the future.

4.7.6 Run-Times

The run-times of our implementation for different problems varied considerably. The lines in Figure 4.6 took between 1 and 2 seconds, the correspondence computations in Figure 4.10 around 2 minutes.

The critical variable for determining the run-time was the number of polynomial centres, the kernel width, and the number of discretisation points of the energy integral. These factors determine the size and the sparsity of the matrices for computing the Ψ-function and its derivatives at the discretisation points x_i from the parameter vector w. Building these matrices and multiplying with them during the calculation of the gradient and the (pseudo-) Hessian of the objective function (4.1) were the most time-consuming parts of the optimisation. Solving the linear system for determining the descend direction, given reasonable sparseness was not so critical in comparison.

4.8 Further Topics in Manifold-Valued Learning

After having seen an implementation and some practical experiments for the proposed Eells energy-based regression approach, let us step back and consider some more mathematical and statistical issues of non-parametric regression between Riemannian manifolds. It will turn out that there are some very interesting and sometimes surprising differences of regression between two Riemannian manifolds to multivariate regression. The results derived here are rather preliminary and the purpose is more to point out interesting problems than providing already a fully developed solution.

4.8.1 Function Spaces

In the regularised risk minimisation problem (4.1) the objective is minimised over all smooth mappings $C^\infty(M, N)$. It is a classical problem in variational analysis that this space

is not sufficient to guarantee the existence of a minimiser since it is not complete. For Euclidean output, one therefore introduces the Sobolev-space $W^{s,2p}(M,\mathbb{R}^l)$, so far $p=1$, as the completion of $C^s(M,\mathbb{R}^l)$ with respect to the norm

$$\|\phi\|_s^{2p} = \sum_{\mu=1}^{l}\sum_{r=0}^{s}\int_M \|\nabla_1 \dots \nabla_r \phi^\mu\|^{2p}\, dV.$$

The functions in $W^{s,2p}(M,\mathbb{R}^l)$ need not be in $C^s(M,\mathbb{R}^l)$, but at least it is known that a (weak) minimiser of (4.1) exists. For example for linear splines, the minimisers using the harmonic energy in $W^{1,2}(\mathbb{R},\mathbb{R})$ are piecewise linear, but not differentiable at the data points $\phi(X_i)$. Under strong assumptions a similar result for linear splines in manifolds has been derived without extending the theory of Sobolev-spaces to the manifold-output situation [Machado et al., 2006]. However, a general approach which uses less assumptions and which is also valid for higher dimensional input requires rather complicated generalisations.

One problem that occurs even for Euclidean output spaces is that if the input dimension $m=\dim(M)$ is greater or equal to $2p$-times the order s of the regulariser, that is, $m \geq 2ps$, then the functions in $W^{s,2p}(M,\mathbb{R}^l)$ need not be continuous and the values of such functions at a point can be changed arbitrarily without changing the function in a $W^{s,2p}$ sense [Evans, 1998]. Since our learning scheme in (4.1) corresponds to minimizing the weighted sum of (parts of) the $W^{s,2p}$-norm, $p=1$, and a point-wise defined loss over all functions in $W^{s,2p}$ ($s=1$ for the harmonic energy and $s=2$ for the biharmonic and Eells energy), the minimizing function for $m \geq 2ps$ could always be chosen as the zero function with delta peaks interpolating the training values. This solution would obviously not be able to generalise, rendering the proposed learning setup invalid in this case.

A classic route to circumvent this problem for Euclidean outputs is to resort to higher order regularisation keeping $p=1$ [Wahba, 1990; Wendland, 2005]. The optimal solution in this case is given in terms of Green's functions which can be computed analytically for any order of regularisation. In the manifold setting, however, such an analytical solution does not exist and we have to discretise $\phi: M \to N$. Higher order regularisation then leads to ever more complicated expressions for the derivatives of ϕ, which renders an implementation increasingly problematic. Instead, we could thus try to increase p for regression between manifolds, that is, changing the regulariser to use the $2p$-norm of the energy density instead of the 2-norm. An experimental evaluation of this idea, however, is subject to future work.

The second problem concerning the analysis of learning between manifolds in Sobolev spaces is manifold specific. If the output manifold is non-Euclidean, then any space of functions targeting that manifold *cannot be a vector space*. This is problematic in that the vector space concept is typically one of the first abstractions that is introduced in any derivation of Sobolev spaces. Avoiding this property thus requires one to make fundamental changes right from the start. Instead of a vector space structure, the space of admissible functions should be rather thought of as an infinite dimensional manifold where the tangent spaces have Hilbert space structure. Some results in this direction can be found in [Hélein and Wood, 2008; Wang, 2004] who examine harmonic maps between Riemannian manifolds. However, they do not examine the learning problem (4.1) or higher order regularisation, and an in-depth analysis of these settings remains an open issue.

In all the experiments in this work the condition $m < 2ps$ was satisfied (except in Figure 4.8 (c), which gives another explanation of the bad behaviour of harmonic energy regularisation in this case). We thus assumed that the minimisers of (4.1) existed in $W^{s,2p}$

also for manifold-valued output. If so, they can be well-approximated by smooth functions [Evans, 1998]. Furthermore, for any discretisation, we have argued that the resulting finite-dimensional non-linear optimisation problem is minimised by the proposed minimisation algorithm, at least locally.

4.8.2 Homotopy and Consistency

In the following we will explore the non-trivial topological structure of manifold-valued mappings.

Definition 4.12. *Two continuous mappings ϕ_1, ϕ_2 from M to N are said to be homotopic if there exists a continuous mapping $\Psi : M \times [0,1] \to N$ with $\Psi(x,0) = \phi_1(x)$ and $\Psi(x,1) = \phi_2(x)$.*

Homotopy defines an equivalence relation on $C(M,N)$. We denote the set of the resulting equivalence classes, the so called homotopy classes, by $[M,N]$. One says that $[M,N]$ is trivial, if it consists just of the homotopy class of the constant map. It is easy to see that $[M,\mathbb{R}^l]$, the homotopy class of mappings considered in manifold learning, is trivial. However, for the manifold-valued regression problem this is generally not the case which has interesting theoretical as well as practical implications.

Typically, the regularised empirical risk minimisation problem is solved using a descent-type algorithm which continuously deforms the current mapping ϕ. This implies that the homotopy class is preserved during optimisation and thus the homotopy class of the final solution is determined by the initial solution. Theoretically, one could just search in all components of $C(M,N)$, which is however practically not possible, e.g. $[\mathcal{S}^1,\mathcal{S}^1]$ is isomorphic to the set of integers - the number of cycles around the circle).

The following theorem provides a first step towards a consistent training procedure for manifold-valued mappings, where $[M,N]$ is non-trivial. It is shown for mappings $\gamma : \mathcal{S}^1 \to \mathcal{S}^1$ that for large enough sample size the initial solution $\hat{\gamma}$ constructed by piecewise geodesic interpolation of the training points has the same homotopy class as the Bayes optimal solution γ^*,

$$\gamma^* = \underset{\gamma \text{ measurable}}{\arg\min}\ \mathbb{E}_{Y,X}\, d^2(\gamma(X), Y),$$

provided that $\gamma^* \in C^1(\mathcal{S}^1,\mathcal{S}^1)$ and the problem is deterministic, that is $\mathrm{P}(\gamma^*(X) \neq Y) = 0$.

Theorem 4.13. *Given K training points $(X_i, Y_i) \in \mathcal{S}^1 \times \mathcal{S}^1$, let h be the maximal geodesic nearest neighbour distance of $\{X_i\}_{i=1}^K$. If the Bayes optimal solution γ^* is deterministic, smooth and $\left\|\dot{\gamma^*}\right\| \leq L$ and $h < \frac{\pi}{L}$, then the piecewise geodesic interpolant of the training data is in the same homotopy class as γ^*.*

Proof. Let X_i and X_j be nearest neighbours in $\mathcal{S}^1$. We have $\int_{X_i}^{X_j} \left\|\dot{\gamma^*}\right\| dt \leq L\, d_{S_1}(X_i, X_j) \leq L\, h$. With $L\, h < \pi$ we know that γ can have made no cycle around $\mathcal{S}^1$ between X_i and X_j. Moreover, the length of the shortest path between Y_i and Y_j is also bounded by $L\, h < \pi$. Thus the geodesic $\hat{\gamma}$ interpolating (X_i, Y_i) and (X_j, Y_j) is homotopic to the segment of $\gamma^*|_{X_i}^{X_j}$. Since this holds for any neighbouring points of the training data, the whole curves γ^* and $\hat{\gamma}$ are homotopic. □

The theorem can be easily extended to non-deterministic problems where $\mathrm{P}(Y|X)$ is sufficiently concentrated and to the setting where $(X_i, Y_i)_{i=1}^K$ is a random sample from P on $\mathcal{S}^1 \times \mathcal{S}^1$. The generalisation of this result to more general domains is non-trivial, and is an interesting problem of future research.

4.8.3 Capacity of Totally Geodesic Maps

In Section 4.4.1 we have shown that totally geodesic maps are a suitable generalisation of the linear maps in Euclidean space to Riemannian manifolds. While linear maps are considered as simple mappings of very limited capacity, this does not necessarily apply to totally geodesic maps as the following example shows.

We consider again mappings from $M = \mathcal{S}^1$ to $N = \mathcal{S}^1$. In standard angular coordinates, all totally geodesic maps in this setting are of the form $\phi_a(x) = a\,x + b$ for $a \in \mathbb{N}$ and $b \in [0, 2\pi)$. The following theorem which is a classical result in number theory shows that this set of mappings can fit any given set of training points arbitrarily well and thus has infinite capacity.

Theorem 4.14. *[Apostol, 1990, p.154] Let $(X_i, Y_i) \in \mathcal{S}^1 \times \mathcal{S}^1$, $i = 1, \ldots, K$, be the training data. Then there exists for any set of training data and any $\varepsilon > 0$ a $a \in \mathbb{N}$ such that*

$$\max_{i=1,\ldots,K} d(\phi_a(X_i), Y_i) \leq \varepsilon,$$

where $\phi_a : \mathcal{S}^1 \to \mathcal{S}^1$, $\phi_a(x) = \mathrm{mod}(ax + b, 2\pi)$.

Since totally geodesic mappings are not penalised by the Eells energy, the solution of regularised empirical risk minimisation in (4.1) is always given by the geodesic ϕ_a, that obviously overfits the training data. However, note that the integer a which corresponds to the number of cycles around the circle of ϕ_a (empirically) grows exponentially with the number of data points. This is the reason why we did not encounter this phenomenon in the implementation of [Steinke et al., 2008]. The above phenomenon still holds if the input space is the real line or a closed interval. At least for regression into $\mathcal{S}^1$ this example thus suggests that the null-space of both the Eells and the biharmonic energy of manifold-valued mappings is already too large to be useful. Since for the harmonic energy one has $S_{\mathrm{harm}}(\phi_a) = 2\pi a$, one should, at least in theory, use either the harmonic energy or a combination of the harmonic and a second-order energy in this case.

4.9 Conclusion

This chapter has presented a universal, theoretically sound framework for regression between two Riemannian manifolds based on regularised empirical risk minimisation. The discussed energies are only dependent on the geometry of the input and output manifold, but not on their respective parametric representation. We have derived an intuitively desirable property of the proposed Eells energy, namely that it favours the so-called totally geodesic maps, a suitable generalisation of linear maps. Our implementation and our experimental results have further supported the benefits of using a truly manifold-adapted approach and especially the Eells energy.

Throughout the chapter we tried to convey that the problem of manifold-valued regression is far from being a trivial generalisation of the Euclidean case, and there remain many challenging and interesting open questions in the mathematical and statistical analysis of this problem. On the practical side, an interesting question is whether there exists a compact but flexible representation for general mappings between Riemannian manifolds. Since our implementation is based on discretisation, it is so far limited to low dimensional input spaces, however, for many statistical problems higher-dimensional input would be desirable, requiring a more compact function representation. In Euclidean space this is typically done with sparse basis function expansions. However, since manifold-valued output does not allow for the addition of functions, this route cannot be undertaken here. The construction of compact, yet flexible representations for mappings between general Riemannian manifolds thus remains an important open project.

4.10 Additional Material

4.10.1 The Pull-Back Connection, its Curvature, and Green's Theorem

This section is a review of basic ingredients of connections and curvature of vector bundles. With the exception of the extension of the Green's theorem to the tensor product connection the material can be found in [Eells and Lemaire, 1983].

Let M be a smooth, connected, orientable Riemannian manifold. Let V be a smooth vector bundle over M of finite rank with base projection $\pi : V \to M$. We denote by $C(V)$ the vector space of smooth sections of V, i.e. of smooth maps $\sigma : M \to V$ such that $\pi \circ \sigma = \mathbf{1}_M$. Let V and W be two vector bundles over M, then we denote by

- V^* is the dual bundle of V,
- $V \oplus W$ is the direct sum of V and W,
- $V \otimes W$ is the tensor product of V and W,
- $\otimes^p V$ the p-th tensor power of V,
- $\wedge^p V$ the p-th exterior power of V (completely antisymmetric),
- $\odot^p V$ the p-th tensor power of V (completely symmetric).

A very important concept for manifold-valued mappings is the pull-back bundle $\phi^{-1}W$.

Definition 4.15. *If $\phi : M \to N$ and W is a vector bundle over N, we denote by $\phi^{-1}W$ the **pull-back bundle**, whose fibre over $x \in M$ is $W_{\phi(x)}$, the fibre of W over $\phi(x)$.*

Next we define the Riemannian metric and the connection on vector bundles.

Definition 4.16. *A **Riemannian metric** on a vector bundle V is a section a in $C(V^* \odot V^*)$, which induces on each fibre a positive definite inner product. Let $\sigma, \rho \in C(V)$, then we use $\langle \sigma, \rho \rangle := a(\sigma, \rho)$.*

Similar to the case of the tangent bundle one can introduce the musical isomorphisms to define maps $V \to V^*$ and $V^* \to V$. One can also define a Riemannian metric on the pull-back bundle. Let $\phi : M \to N$ and W be a vector bundle over N with metric b. We can identify $\sigma, \rho \in (\phi^{-1}W)_x$ with $\sigma, \rho \in W_{\phi(x)}$ and thereby define $\langle \sigma, \rho \rangle_b$.

Definition 4.17. *A **linear connection** on a vector bundle V over M is a bilinear map ∇ on spaces of sections,*

$$\nabla : C(TM) \times C(V) \to C(V),$$

written $\nabla : (X, \sigma) \mapsto \nabla_X \sigma,\ X \in C(TM),\ \sigma \in C(V)$, such that for $f \in C(M)$ we have

- $\nabla_{fX}\sigma = f \nabla_X \sigma$,
- $\nabla_X(f\sigma) = X(f)\ \sigma + f\ \nabla_X \sigma$.

Since ∇ is linear in its first argument we also write in abstract index notation $X^a \nabla_a \sigma^{t_1,\ldots,t_s}_{b_1,\ldots,b_r}$ for a (s, r) vector bundle V.

Definition 4.18. *Let ${}^V\nabla$ and ${}^W\nabla$ be connections on V and W.*

1. *The **dual connection** on V^* is defined by*

$$\theta \in C(V^*),\ \sigma \in C(V); \qquad (\nabla_X \theta)(\sigma) = X(\theta(\sigma)) - \theta(\nabla_X \sigma). \tag{4.22}$$

2. *The **direct sum connection** on $V \oplus W$ is defined as,*

$$\sigma \in C(V), \lambda \in C(W); \qquad \nabla_X(\sigma \oplus \lambda) = {}^V\nabla_X \sigma \oplus {}^W\nabla_X \lambda. \tag{4.23}$$

3. *The **tensor product connection** on $V \otimes W$ is defined as,*

$$\sigma \in C(V), \lambda \in C(W); \qquad \nabla_X(\sigma \otimes \lambda) = {}^V\nabla_X \sigma \otimes \lambda + \sigma \otimes {}^W\nabla_X \lambda. \tag{4.24}$$

The following definition of the pull-back connection is the central key to the definition of energy functionals for manifold-valued mappings.

Definition 4.19. *For a smooth map $\phi : M \to N$ and a vector bundle W over N with connection ${}^W\nabla$, we define the **pull-back or induced connection** on $\phi^{-1}W$ as the connection ∇' on $\phi^{-1}W$ such that for each $x \in M$, $X \in T_xM$ and $\lambda \in C(W)$, we have*

$$\nabla'_X (\phi^* \lambda) = \phi^* \left({}^W\nabla_{d\phi(X)} \lambda \right),$$

where $d\phi : T_xM \to T_{\phi(x)}N$ is the push-forward or differential of ϕ and $\phi^*\lambda = \lambda \circ \phi \in C(\phi^{-1}W)$. In abstract index notation

$$\nabla'_a \lambda(\phi(x)) = d\phi^r_a \, {}^W\nabla_r \lambda \Big|_{\phi(x)}.$$

This definition which formally only applies to elements $\phi^*\lambda \in \phi^{-1}W$ derived from $\lambda \in C(W)$ can be uniquely extended to all elements of $\phi^{-1}W$ using the defining properties of a connection [Eells and Lemaire, 1983].

Definition 4.20. *A **Riemannian structure on a bundle** V is a pair (∇, a), where a is a Riemannian metric, ∇ is a connection and $\nabla a = 0$, where ∇a is defined using the tensor product connection in Eq. (4.24).*

The condition $\nabla a = 0$ means that for all $X \in C(TM),\ \sigma, \omega \in C(V)$ we have

$$X \langle \sigma, \omega \rangle = \langle \nabla_X \sigma, \omega \rangle + \langle \sigma, \nabla_X \omega \rangle,$$

i.e. the connection is compatible with the inner product. It is straightforward to check that if $({}^V\nabla, a)$ and $({}^W\nabla, b)$ are Riemannian structures on V and W respectively, then the direct sum, the tensor product and the pull-back -connection are again Riemannian structures.

Definition 4.21. *The **curvature tensor of a connection** is the map $R : C(TM) \wedge C(TM) \otimes C(V) \to C(V)$ defined by*

$$R(X,Y)\sigma = \nabla_X \nabla_Y \sigma - \nabla_Y \nabla_X \sigma - \nabla_{[X,Y]}\sigma = -R(Y,X)\sigma. \tag{4.25}$$

Lemma 4.22. *Let R^V and R^W be the curvature tensors of V and W. Then it holds,*

- *for V^*, $(R(X,Y)\theta)(\sigma) = -\theta(R(X,Y)\sigma)$ for all $X, Y \in C(TM)$ and $\theta \in C(V^*)$ and $\sigma \in C(V)$,*
- *for $V \oplus W$, $R(X,Y)(\sigma \oplus \lambda) = R^V(X,Y)\sigma \oplus R^W(X,Y)\lambda$ where $\lambda \in C(W)$,*
- *for $V \otimes W$, $R(X,Y)(\sigma \otimes \lambda) = R^V(X,Y)\sigma \otimes \lambda + \sigma \otimes R^W(X,Y)\lambda$,*
- *for $\phi^{-1}W$, $R_x(X,Y)\rho(x) = R^W_{\phi(x)}(d\phi(X), d\phi(Y))\rho(x)$ where $\rho \in C(\phi^{-1}W)$.*

From here on, we only consider connections derived from the Levi-Civita connections on tangent bundles on M and N. In particular, for the smooth map $\phi : M \to N$ we repeatedly consider on $\phi^{-1}TN$ the pull-back connection ∇' of the Levi-Civita connection on N. Let the metric on M be g, the metric on N be h. Furthermore, let ${}^M\nabla$ and ${}^N\nabla$ be the Levi-Civita connections for the tangent bundles of M and N. For a mixed tensor $T^r_a \in T^*M \otimes \phi^{-1}TN$ we apply the tensor product connection by using ${}^M\nabla$ for T^*M and ∇' for $\phi^{-1}TN$. By some abuse of notation we use the same symbol ∇' for all tensor product connections on $\otimes^k TM \otimes^l T^*M \otimes \phi^{-1}TN$, and also refer to it as the *pull-back connection* for all these bundles. The following recipe for a covariant derivative of the mixed tensor T can be generalised in a straightforward manner.

$$\begin{aligned}\nabla'_b T^r_a &= \nabla'_b(T^\mu_\alpha\, dx^\alpha_a \otimes \partial^r_\mu)\\ &:= \left({}^M\nabla_b T^\mu_\alpha\right) dx^\alpha_a \otimes \partial^r_\mu + T^\mu_\alpha \left({}^M\nabla_b dx^\alpha_a\right) \otimes \partial^r_\mu + T^\mu_\alpha\, dx^\alpha_a \otimes \left(\nabla'_b \partial^r_\mu\right).\end{aligned}$$

As an example consider the differential $d\phi^r_a : T_xM \to T_{\phi(x)}N$,

$$d\phi^r_a(x) = \frac{\partial \phi^\mu}{\partial x^\alpha} dx^\alpha_a\Big|_x \otimes \frac{\partial^r}{\partial y^\mu}\Big|_{\phi(x)} = {}^M\nabla_a \phi^\mu\Big|_x \otimes \frac{\partial^r}{\partial y^\mu}\Big|_{\phi(x)}.$$

With the Christoffel symbols ${}^M\Gamma^\gamma_{\beta\alpha}$ and ${}^N\Gamma^\mu_{\nu\rho}$ for the connections on M and N the coordinate expression of $\nabla'_b d\phi^r_a$ is

$$\begin{aligned}\nabla'_b d\phi^r_a &= {}^M\nabla_b {}^M\nabla_a \phi^\mu \otimes \frac{\partial^r}{\partial y^\mu} + {}^M\nabla_a \phi^\mu \otimes \nabla'_b \frac{\partial^r}{\partial y^\mu}\\ &= \Big[\frac{\partial^2 \phi^\mu}{\partial x^\beta \partial x^\alpha} + \frac{\partial \phi^\mu}{\partial x^\gamma} {}^M\Gamma^\gamma_{\beta\alpha} + \frac{\partial \phi^\rho}{\partial x^\alpha}\frac{\partial \phi^\nu}{\partial x^\beta} {}^N\Gamma^\mu_{\nu\rho}\Big] dx^\beta_b \otimes dx^\alpha_a \otimes \frac{\partial^r}{\partial y^\mu}.\end{aligned}$$

One can read off that $\nabla'_b d\phi^r_a = \nabla'_a d\phi^r_b$, because the Levi-Civita connections on M and N are symmetric implying that ${}^M\Gamma^\gamma_{\beta\alpha} = {}^M\Gamma^\gamma_{\alpha\beta}$ and ${}^N\Gamma^\mu_{\nu\rho} = {}^N\Gamma^\mu_{\rho\nu}$. With this in mind, we can show a small lemma which will be useful later on.

Lemma 4.23. *Let $\phi : M \to N$ and $X, Y \in C(TM)$, then we have*

$$\nabla'_X(d\phi(Y)) - \nabla'_Y(d\phi(X)) = d\phi([X,Y]),$$

where $[X,Y]$ is the Lie-bracket.

Proof. It is

$$\begin{aligned}&X^b\nabla'_b(d\phi^r_a Y^a) - Y^b\nabla'_b(d\phi^r_a X^a)\\ &\quad = d\phi^r_a(X^b\, {}^M\nabla_b Y^a - Y^b\, {}^M\nabla_b X^a) + X^b Y^a[\nabla'_b d\phi^r_a - \nabla'_a d\phi^r_b] = d\phi^r_a[X,Y]^a,\end{aligned}$$

where we have used in the first equality the definition of the pull-back connection for tensor product spaces and in the second equality the definition of the Lie bracket together with $\nabla'_b d\phi^r_a = \nabla'_a d\phi^r_b$. □

The generalisation of Green's theorem to the case of the pull-back connection is as follows.

Lemma 4.24. *Let $T \in C(\otimes^{p+1} T^*M \otimes \phi^{-1}TN)$ and $S \in C(\otimes^p T^*M \otimes \phi^{-1}TN)$, then with ∇' being the pull-back connection, we have*

$$\int_M \langle T, \nabla' S \rangle = \int_{\partial M} \langle T, N \otimes S \rangle - \int_M \langle \mathrm{trace}_g \nabla' T, S \rangle ,$$

where N is the covector associated to the normal vector at ∂M and the trace is taken with respect to the first two indices. In abstract index notation the expression can be written as,

$$\begin{aligned}
&\int_M g^{ac_0} g^{b_1 c_1} \dots g^{b_p c_p}\, h_{rs}\, T^r_{c_0 \dots c_p}\, \nabla'_a S^s_{b_1 \dots b_p} \\
&= \int_{\partial M} g^{ac_0} g^{b_1 c_1} \dots g^{b_p c_p}\, h_{rs}\, T^r_{c_0 \dots c_p}\, N_a S^s_{b_1 \dots b_p} \\
&- \int_M g^{ac_0} g^{b_1 c_1} \dots g^{b_p c_p}\, h_{rs}\, \nabla'_a T^r_{c_0 \dots c_p}\, S^s_{b_1 \dots b_p}.
\end{aligned}$$

Proof. We show the result for $T \in C(T^*M \otimes \phi^{-1}TN)$ and $S \in C(\phi^{-1}TN)$ using explicit coordinates. The extension to higher tensor powers in T^*M is then a straightforward calculation. With $\nabla'_a S^s = \nabla'_a (S^\nu \frac{\partial^s}{\partial y^\nu}) = ({}^M\nabla_a S^\nu) \frac{\partial^s}{\partial y^\nu} + S^\nu \nabla'_a \frac{\partial^s}{\partial y^\nu}$ we can write the part of the covariant derivative associated to the pull-back connection explicitly,

$$\int_M g^{ab}\, h_{rs}\, T^r_b\, \nabla'_a S^s = \int_M g^{ab}\, h_{\mu\nu}\, T^\mu_b [{}^M\nabla_a S^\nu + S^\rho\, {}^N\Gamma^\nu_{\rho\omega} d\phi^\omega_a], \tag{4.26}$$

where ${}^N\Gamma^\nu_{\rho\omega}$ are the Christoffel-symbols of N. Furthermore, we have

$$\begin{aligned}
&\int_M g^{ab} h_{\mu\nu} T^\mu_b\, {}^M\nabla_a S^\nu \\
&= \int_M g^{ab}\, {}^M\nabla_a (h_{\mu\nu} T^\mu_b S^\nu) - \int_M g^{ab}\, ({}^M\nabla_a h_{\mu\nu}) T^\mu_b S^\nu - \int_M g^{ab} h_{\mu\nu} ({}^M\nabla_a T^\mu_b) S^\nu \\
&= \int_{\partial M} N^b h_{rs} T^r_b S^s - \int_M g^{ab} \frac{\partial h_{\mu\nu}}{\partial y^\rho}\, d\phi^\rho_a T^\mu_b S^\nu - \int_M g^{ab}\, h_{\mu\nu}\, S^\nu\, {}^M\nabla_a T^\mu_b ,
\end{aligned}$$

where we use the normal Green's theorem from differential geometry [Lee, 1997] in the second equation. With $\frac{\partial h_{\mu\nu}}{\partial y^\rho} = h_{\nu\omega}\, {}^N\Gamma^\omega_{\rho\mu} + h_{\mu\omega}\, {}^N\Gamma^\omega_{\rho\nu}$ we obtain

$$\int_M g^{ab} \frac{\partial h_{\mu\nu}}{\partial y^\rho}\, d\phi^\rho_a\, T^\mu_b\, S^\nu = \int_M g^{ab} [h_{\nu\omega}\, {}^N\Gamma^\omega_{\rho\mu} + h_{\mu\omega}\, {}^N\Gamma^\omega_{\rho\nu}]\, d\phi^\rho_a\, T^\mu_b\, S^\nu .$$

Plugging the expression for $\int_M g^{ab} h_{\mu\nu} T^\mu_b\, {}^M\nabla_a S^\nu$ into Equation (4.26) we obtain

$$\begin{aligned}
\int_M g^{ab}\, h_{rs}\, T^r_b\, \nabla'_a S^s &= \int_{\partial M} N^b\, h_{rs}\, T^r_b\, S^s - \int_M g^{ab}\, h_{\mu\nu}\, S^\nu\, [{}^M\nabla_a T^\mu_b + T^\omega_b\, {}^N\Gamma^\mu_{\omega\alpha}\, d\phi^\alpha_a] \\
&= \int_{\partial M} N^b\, h_{rs}\, T^r_b\, S^s - \int_M g^{ab}\, h_{rs}\, S^s\, \nabla'_a T^r_b .
\end{aligned}$$

□

4.10.2 Proofs of Section 4.4

Proposition 4.5. We have for $X^a, Y^b \in TM$, $X^a \nabla'_a (Y^b d\phi^r_b) = X^a Y^b \nabla'_a d\phi^r_b + X^a d\phi^r_b \nabla_a Y^b$. This yields

$$X^a Y^b \nabla'_a d\phi^r_b = X^a \nabla'_a (Y^b d\phi^r_b) - X^a d\phi^r_b \nabla_a Y^b.$$

The last equation can be rewritten in a more transparent way using the definition of the pull-back connection as

$$X^a Y^b \nabla'_a d\phi^r_b = (X^a d\phi^s_a)^N \nabla_s (Y^b d\phi^r_b) - X^a d\phi^r_b \nabla_a Y^b,$$

where the right hand side is just a different notation of ${}^N\nabla_{d\phi(X)} d\phi(Y) - d\phi({}^M\nabla_X Y)$. The above equation thus shows that ϕ is connection preserving if and only if $\nabla'_a d\phi^r_b = 0$. Moreover, $\nabla'_a d\phi^r_b = 0$ implies that geodesics are mapped onto geodesics. Suppose $\gamma : (-\varepsilon, \varepsilon) \to M$ is a geodesic on M. Then given $\nabla'_a d\phi^r_b = 0$ we obtain,

$$0 = {}^N\nabla_{d\phi(\dot\gamma)} d\phi(\dot\gamma) - d\phi({}^M\nabla_{\dot\gamma}\dot\gamma) = {}^N\nabla_{d\phi(\dot\gamma)} d\phi(\dot\gamma) = 0,$$

where we have used that ${}^M\nabla_{\dot\gamma}\dot\gamma = 0$ since γ is a geodesic. Therefore the mapped curve $\gamma' : (-\varepsilon, \varepsilon) \to N$ defined as $\gamma' = \phi \circ \gamma$ is also a geodesic. Conversely, ${}^N\nabla_{d\phi(\dot\gamma)} d\phi(\dot\gamma) - d\phi({}^M\nabla_{\dot\gamma}\dot\gamma) = 0$ for all geodesics implies $\nabla'_a d\phi^r_b = 0$. □

Theorem 4.6. One can write the difference between the biharmonic and Eells energy as a divergence of a vector field on M plus some curvature terms. We define,

$$F_b = h_{rs}\, g^{cd} \left(d\phi^r_b\, \nabla'_c d\phi^s_d - d\phi^r_c\, \nabla'_b d\phi^s_d \right).$$

We have

$$\begin{aligned} g^{ab}\, \nabla'_a F_b = h_{rs}\, g^{ab}\, g^{cd} \Big(& \nabla'_a d\phi^r_b\, \nabla'_c d\phi^s_d \\ & + d\phi^r_b\, \nabla'_a \nabla'_c d\phi^s_d - \nabla'_a d\phi^r_c \nabla'_b d\phi^s_d - d\phi^r_c \nabla'_a \nabla'_b d\phi^s_d \Big). \end{aligned} \tag{4.27}$$

Thus the divergence contains the energy densities of the Eells and biharmonic energy plus two other terms. The last term in (4.27) can be rewritten using $\nabla'_b d\phi^s_d = \nabla'_d \phi^s_b$ and

$$\nabla'_a \nabla'_d d\phi^s_b = \nabla'_d \nabla'_a d\phi^s_b - R^M{}_{adb}{}^e\, d\phi^s_e + R^N{}_{tuv}{}^s\, d\phi^t_a\, d\phi^u_d\, d\phi^v_b,$$

where we have used Appendix 4.10.1 and specifically Lemma 4.22 for elements in $T^*M \otimes \phi^{-1}TN$ like $d\phi^s_b$. The first term of this new expansion and the second term in (4.27) cancel. Applying the extended Green's theorem, Lemma 4.24, we obtain the desired result. □

4.10.3 Extrinsic Representation of the Pull-Back Connection and Proofs of Section 4.5

Here, we compute a representation of the pull-back connection for manifolds N which are isometrically embedded in Euclidean space.

Lemma 4.25. *Let $i : N \to \mathbb{R}^l$ be an isometric embedding and denote by h the metric of N and by y^μ coordinates in N. Then the following quantities can be computed using the embedding i,*

$$h_{\mu\nu} = \sum_{\alpha=1}^{l} \frac{\partial i^\alpha}{\partial y^\mu}\frac{\partial i^\alpha}{\partial y^\nu}, \qquad h_{\mu\nu}\,{}^N\Gamma^\nu_{\omega\rho} = \sum_{\alpha=1}^{l} \frac{\partial^2 i^\alpha}{\partial y^\omega \partial y^\rho}\frac{\partial i^\alpha}{\partial y^\mu}.$$

The projection $P : T_z\mathbb{R}^l \to T_z N$, $V \mapsto PV$ can be computed as

$$(PV)^r = h^{rs}\delta_{su}V^u = h^{\mu\nu}\sum_{\alpha=1}^{l}\frac{\partial i^\alpha}{\partial y^\nu}V^\alpha\frac{\partial^r}{\partial y^\mu}.$$

Proof. We have $h = i^*\delta$, where δ is the metric in $\mathbb{R}^l$. Thus, we obtain

$$h_{rs} = \delta_{\alpha\beta}(i^*dz^\alpha)_r \otimes (i^*dz^\beta)_s = \delta_{\alpha\beta}\frac{\partial i^\alpha}{\partial y^\mu}\frac{\partial i^\beta}{\partial y^\nu}\, dy^\mu_e \otimes dy^\nu_f.$$

It holds $h_{\mu\nu}\,{}^N\Gamma^\nu_{\omega\rho} = \frac{1}{2}\Big(\partial_\omega h_{\rho\mu} + \partial_\rho h_{\omega\mu} - \partial_\mu h_{\omega\rho}\Big)$. With

$$\partial_\omega h_{\rho\mu} = \sum_{\alpha=1}^{l}\left[\frac{\partial^2 i^\alpha}{\partial y^\omega\partial y^\rho}\frac{\partial i^\alpha}{\partial y^\mu} + \frac{\partial i^\alpha}{\partial y^\rho}\frac{\partial^2 i^\alpha}{\partial y^\omega \partial y^\mu}\right],$$

we arrive after a short calculation at the desired result. The projection $P : T_z\mathbb{R}^l \to T_z N$ can be written as $P = \sum_{i=1}^n e_i \langle e_i, \cdot\rangle$, where $\{e_i^r\}_{i=1}^n$ is an orthonormal basis in $T_z N$. Then $h^{rs} = \sum_{i=1}^n e_i^r e_i^s$ and thus $P_b^r = h^{rs}\delta_{sb}$. We have $\delta_{sb} = \sum_{\alpha=1}^l dz_s^\alpha dz_b^\alpha$, where z^α are Cartesian coordinates in $\mathbb{R}^l$. The tangential projection of dz^α is given as $(i^*dz^\alpha)_b = \frac{\partial i^\alpha}{\partial y^\nu}dy^\nu_b$. Thus $P_b^r = h^{\mu\nu}\sum_{\alpha=1}^l \frac{\partial i^\alpha}{\partial y^\nu}\, dz_b^\alpha \otimes \frac{\partial^r}{\partial y^\mu}$. □

Definition 4.26. *Let ∇' be the connection pull-back by ϕ and $\tilde\nabla$ the connection pull-back by $\Psi = i \circ \phi$. The pull-back second fundamental from $\Pi' : TM \otimes \phi^{-1}TN \to (\phi^{-1}TN)^\perp$ is defined via*

$$X^a\tilde\nabla_a S^r = X^a\nabla'_a S^r + X^a\Pi'^r_{as}S^s.$$

Lemma 4.27. *Let $i : N \to \mathbb{R}^l$ be an isometric embedding of N. The second fundamental form of N, ${}^N\Pi^e_{gf} : TN \otimes TN \to (TN)^\perp$ can be expressed in terms of the embedding i as,*

$$\begin{aligned}{}^N\Pi^u_{rs} =& \Big[\frac{\partial^2 i^\alpha}{\partial y^\mu \partial y^\nu}\frac{\partial^u}{\partial z^\alpha} - P^\rho_\beta\frac{\partial^2 i^\beta}{\partial y^\mu\partial y^\nu}\frac{\partial^u}{\partial y^\rho}\Big]\, dy^\mu_r\, dy^\nu_s \\ =& \left(\frac{\partial^2 i^\alpha}{\partial y^\mu\partial y^\nu}\frac{\partial^u}{\partial z^\alpha}\right)^\perp \otimes dy^\mu_r \otimes dy^\nu_s.\end{aligned}$$

The pull-back second fundamental form $\Pi'^e_{ab} : TM \otimes \phi^{-1}TN \to \phi^{-1}(TN)^\perp$ *can be computed as*

$$\Pi'^r_{as} = d\phi^u_a\,{}^N\Pi^r_{us}.$$

Proof. For $S^r \in \phi^{-1}TN$ one obtains with $d\Psi^f_a = d\phi^f_a$ from Theorem 4.7,

$$\tilde\nabla_a S^r = d\Psi^s_a\,{}^{\mathbb{R}^l}\nabla_s S^r = d\phi^s_a\left[{}^N\nabla_s S^r + {}^N\Pi^r_{su}\,S^u\right] = \nabla'_a S^r + d\phi^s_a\,{}^N\Pi^r_{su}S^u.$$

One can check that the result generalises to covariant derivatives of $\otimes^m T^*M \otimes \phi^{-1}TN$. □

Now the proofs of Section 4.5 can be derived as follows.

Theorem 4.7. We have $\Psi = i \circ \phi$. With $\frac{\partial^r}{\partial y^\mu} = \frac{\partial i^\alpha}{\partial y^\mu}\frac{\partial^r}{\partial z^\alpha}$ we get

$$d\Psi^r_a = \frac{\partial \Psi^\alpha}{\partial x^\beta}\, dx^\beta_a \otimes \frac{\partial^r}{\partial z^\alpha} = \frac{\partial i^\alpha}{\partial y^\mu}\,\frac{\partial \phi^\mu}{\partial x^\beta}\, dx^\beta_a \otimes \frac{\partial^r}{\partial z^\alpha} = \frac{\partial \phi^\mu}{\partial x^\beta}\, dx^\beta_a \otimes \frac{\partial^r}{\partial y^\mu} = d\phi^r_a.$$

We have ${}^{\mathbb{R}^l}\nabla_s V^r = {}^N\nabla_s V^r + \Pi^r_{su} V^u$. Therefore we can decompose the pull-back connection $\tilde{\nabla}$ related to Ψ and ∇' related to ϕ as follows for $T^s_{a_1\ldots,a_m} \in \otimes^m TM \otimes \phi^{-1}TN$

$$\tilde{\nabla}_b T^s_{a_1\ldots,a_m} = \nabla'_b T^s_{a_1\ldots,a_m} + \Pi'^s_{br}\; T^r_{a_1\ldots a_m},$$

where we have used the pull-back second fundamental form $\Pi'^s_{br} \in \phi^{-1}(TN)^\perp \otimes TM \otimes \phi^{-1}TN$, $\Pi'^s_{br} = d\phi^u_b\, {}^N\Pi^s_{ur}$. □

Theorem 4.8. A direct application of Theorem 4.7 together with the fact that $g^{ab} = \delta^{ab}$ for Cartesian coordinates and $\tilde{\nabla}_b d\Psi^r_a = \frac{\partial^2 \Psi^\mu}{\partial x^\alpha \partial x^\beta} dx^\alpha_b \otimes dx^\beta_a \otimes \frac{\partial^r}{\partial z^\mu}$ yields the results. □

Proposition 4.9. Let $\gamma(t)$ be a geodesic on M with $\gamma(0) = p$. A Taylor expansion of γ around p with respect to the ambient space $\mathbb{R}^k$ yields

$$\gamma(t) = \gamma(0) + \gamma'(0)t + \frac{1}{2}\gamma''(0)t^2 + O(t^3).$$

It is $\gamma'' = {}^M\nabla_{\gamma'}\gamma' + \Pi(\gamma', \gamma')$, where $\Pi : T_pM \times T_pM \to N_pM$ is the second fundamental form or extrinsic curvature of M, N_pM is the normal space of M (the subspace orthogonal to the tangent space T_pM in $\mathbb{R}^k$) [Lee, 1997, p. 140]. Since γ is a geodesic, ${}^M\nabla_{\gamma'}\gamma' = 0$ and thus $\gamma'' = \Pi(\gamma', \gamma')$. Plugging this into the Taylor expansion, we obtain

$$\gamma(t) = \gamma(0) + \gamma'(0)t + \frac{t^2}{2}\Pi(\gamma', \gamma') + O(t^3),$$

where $\gamma'(0) \in T_pM$ and $\Pi(\gamma', \gamma') \in N_pM$. We deduce that, if we introduce orthonormal coordinates x^α for the subspace $p + T_pM$ with origin at $p \in M$ and extend this to a full Cartesian coordinate system of $\mathbb{R}^k$, the first part of the theorem follows. For a hypersurface M the normal space N_pM is one-dimensional, $\Pi(X, Y) = h(X, Y)N$, where N is the normal vector at p and $h : T_pM \times T_pM \to \mathbb{R}$. Thus, in coordinates, h is just a $m \times m$-symmetric matrix with eigenvalues κ_α, $\alpha = 1, \ldots, m$ and in the basis formed by the eigenvectors it is $h(X, Y) = \sum_{\alpha=1}^m \kappa_\alpha X^\alpha Y^\alpha$. □

Proposition 4.10. The function $i : \mathbb{R}^m \to \mathbb{R}^k$ defined as $(x^1, \ldots, x^m) \mapsto i(x) = (x^1, \ldots, x^m, f^{m+1}(x), \ldots, f^k(x))$, can be seen as the embedding of the second order approximation of M into $\mathbb{R}^k$. The induced metric is given as

$$g_{\alpha\beta} = \sum_{r=1}^k \frac{\partial i^r}{\partial x^\alpha}\frac{\partial i^r}{\partial x^\beta} = \begin{cases} 1 + \sum_{r=m+1}^k \left(\frac{\partial f^r}{\partial x^\alpha}\right)^2, & \text{if } \alpha = \beta, \\ \sum_{r=m+1}^k \frac{\partial f^r}{\partial x^\alpha}\frac{\partial f^r}{\partial x^\beta}, & \text{if } \alpha \neq \beta. \end{cases}$$

Since the functions f^r are all quadratic in the coordinates x^α, we immediately see that $g_{\alpha\beta}(0) = \delta_{\alpha\beta}$. Moreover, we have

$$\frac{\partial g_{\alpha\beta}}{\partial x^\gamma} = \begin{cases} 2\sum_{r=m+1}^s \frac{\partial^2 f^r}{\partial x^\gamma \partial x^\alpha}\frac{\partial f^r}{\partial x^\alpha}, & \text{if } \alpha = \beta, \\ \sum_{r=m+1}^s \left(\frac{\partial^2 f^r}{\partial x^\gamma \partial x^\alpha}\frac{\partial f^r}{\partial x^\beta} + \frac{\partial f^r}{\partial x^\alpha}\frac{\partial^2 f^r}{\partial x^\gamma \partial x^\beta}\right), & \text{if } \alpha \neq \beta. \end{cases}$$

Again, since f^r are quadratic functions in x^α we have $\frac{\partial g_{\alpha\beta}}{\partial x^\gamma} = 0$ at the origin. Now, the Christoffel symbols in local coordinates x^α are given as [Lee, 1997, p. 70] $\Gamma^\gamma_{\alpha\beta} = \frac{1}{2}g^{\gamma\rho}(\partial_\alpha g_{\beta\rho} + \partial_\beta g_{\alpha\rho} - \partial_\rho g_{\alpha\beta})$, and with the previous result, we also obtain $\Gamma^\gamma_{\alpha\beta} = 0$ at the origin. Finally, we have

$$\frac{\partial^2 \Psi^\mu}{\partial x^\beta \partial x^\alpha} = \frac{\partial^2 \Psi^\mu}{\partial z^r \partial z^u}\frac{\partial z^r}{\partial x^\alpha}\frac{\partial z^u}{\partial x^\beta} + \frac{\partial \Psi^\mu}{\partial z^r}\frac{\partial^2 z^r}{\partial x^\alpha \partial x^\beta},$$

and

$$\frac{\partial z^r}{\partial x^\alpha} = \begin{cases} 1, & \text{if } r = \alpha, \\ 0, & \text{if } r \leq m \text{ and } r \neq \alpha, \\ \frac{\partial f^r}{\partial x^\alpha}, & \text{if } r > m, \end{cases} \qquad \frac{\partial^2 z^r}{\partial x^\beta \partial x^\alpha} = \begin{cases} 0, & \text{if } r \leq m, \\ \Pi^r_{\alpha\beta}, & \text{if } r > m, \end{cases}$$

from the result in (4.17) follows. □

4.10.4 Variation of the Harmonic, Biharmonic and Eells Energy

In this section, we derive necessary conditions for the minimiser of the energy functionals, that is, the Euler-Lagrange equations. The variation of the energy functionals is based on the extended Green's theorem, Lemma 4.24, and the commutator formula from Lemma 4.28, for the exchange of derivatives of the induced connection.

Let $I = (-\varepsilon, \varepsilon)$, then we denote by $\phi(t, x)$, $t \in I$, a variation of the mapping ϕ such that $\phi(0, x) = \phi(x)$ and by $T(M \times I)$ the tangent space of the product manifold $M \times I$. Note that $T(M \times I)$ is isomorphic to $TM \oplus TI$. The product metric is given as $g = g_{TM} \oplus g_{TI}$ and is block-diagonal in any local coordinate system. This implies that also all other structures on the product manifold like Christoffel-symbols or curvature tensor have this block-diagonal structure.

Lemma 4.28. *Let ∇' be the pull-back connection on $T^*(M \times I) \otimes \phi^{-1}TN$, then*

$$\frac{\partial^a}{\partial t}\nabla'_a d\phi^r_b = \nabla'_b \frac{\partial \phi^r}{\partial t} = \nabla'_b\Big(d\phi^r_a \frac{\partial^a}{\partial_t}\Big), \tag{4.28}$$

$$\frac{\partial^c}{\partial t}\nabla'_c\nabla'_a d\phi^r_b = \nabla'_a\nabla'_b \frac{\partial \phi^r}{\partial t} + R^N_{suv}{}^r \frac{\partial \phi^s}{\partial t}\, d\phi^u_a\, d\phi^v_b. \tag{4.29}$$

Proof. Since $\frac{\partial}{\partial_t}$ and $\frac{\partial}{\partial_{x_i}}$ are coordinate vectors, we have $[\frac{\partial}{\partial_t}, \frac{\partial}{\partial_{x_i}}] = 0$. Moreover, the tensor product of the pull-back connection of $\phi^{-1}TN$ and $T^*(M \times I)$ is compatible with the Riemannian structure on $T^*(M \times I) \otimes \phi_t^{-1}TN$ (note that $T^*(M \times I) \simeq T^*M \oplus TI$ so that the metric is block-diagonal). We use the result of Lemma 4.23 with $Y^a = \frac{\partial^a}{\partial t} \in T(M \times I)$,

$$X^b\nabla'_b\Big(d\phi^r_a \frac{\partial^a}{\partial t}\Big) - \frac{\partial^a}{\partial t}\nabla'_a\Big(d\phi^r_b X^b\Big) = 0.$$

With $\frac{\partial^b}{\partial t}\nabla'_b\Big(d\phi^r_a X^a\Big) = X^a\frac{\partial^b}{\partial t}\nabla'_b d\phi^r_c + d\phi^r_c\frac{\partial^b}{\partial t}\nabla'_b X^a$ and $\frac{\partial^a}{\partial t}\nabla'_a X^b = 0$ (X^b is a vector field on M and does not change with t) we obtain

$$\nabla'_b\Big(d\phi^r_a \frac{\partial^a}{\partial_t}\Big) = \nabla'_b \frac{\partial \phi^r}{\partial t} = \frac{\partial^a}{\partial t}\nabla'_b d\phi^r_a = \frac{\partial^a}{\partial t}\nabla'_a d\phi^r_b, \tag{4.30}$$

where the last equality follows by the symmetry of $\nabla'_d d\phi^r_c$. Taking the derivative of Equation 4.30 we get

$$\nabla'_a \nabla'_b \frac{\partial \phi^r}{\partial t} = \left(\nabla'_a \frac{\partial^c}{\partial t}\right) \nabla'_c d\phi^r_b + \frac{\partial^c}{\partial t} \nabla'_a \nabla'_c d\phi^r_b = \frac{\partial^c}{\partial t} \nabla'_a \nabla'_c d\phi^r_b,$$

where we have used that $\left(\nabla'_a \frac{\partial^c}{\partial t}\right)\Big|_{TM} = 0$. We will now exchange the order of the derivatives in front of $d\phi^r_b$ using the definition of the curvature tensor for objects of type $T^*(M \times I) \otimes \phi^{-1} TN$,

$$\nabla'_c \nabla'_a d\phi^r_b = \nabla'_a \nabla'_c d\phi^r_b - R^{M\times I\ d}_{cab}\ d\phi^r_d + R^N_{suv}{}^r\ d\phi^s_c d\phi^u_a d\phi^b_u,$$

where we have used that the curvature tensor of $M \times I$ is the direct sum of the curvature of M and the curvature of I which is zero. Moreover, we have due to the block-diagonal structure of the curvature tensor $\frac{\partial^c}{\partial t} R^{M\times I\ d}_{cab} = 0$. Using the previous result we get,

$$\frac{\partial^c}{\partial t} \nabla'_c \nabla'_a d\phi^r_b = \nabla'_a \nabla'_b \frac{\partial \phi^r}{\partial t} + R^N_{suv}{}^r\ \frac{\partial \phi^s}{\partial t} d\phi^u_a d\phi^v_u$$

□

The previous theorem basically tells us that the time derivative commutes with the pull-back connection. But the "Hessian" does not commute with the time derivative and one gets an additional curvature term.

Theorem 4.29. *Let $I = (-\varepsilon, \varepsilon)$ and $\phi(t, x) : I \times M \to N$ be a variation of the mapping $\phi = \phi(0, x)$ and $W^b = \frac{\partial}{\partial t}\phi^b_t\big|_{t=0}$ the variational vector field at $t = 0$.*
*The variation of the **harmonic energy** is given as,*

$$\frac{1}{2}\frac{d}{dt} S_{\text{harmonic}}(\phi_t)\Big|_{t=0} = -\int_M g^{ac} h_{rs} W^r \nabla'_c d\phi^s_a\, dV + \int_{\partial M} h_{rs} N^c W^r d\phi^s_c d\tilde{V},$$

*The variation of the **Eells energy** is given as,*

$$\begin{aligned}\frac{1}{2}\frac{d}{dt} S_{\text{Eells}}(\phi_t)\Big|_{t=0} = &\int_M g^{ab} g^{cd} h_{rs} W^r \left[\nabla'_c \nabla'_a \nabla'_b d\phi^s_d + R^N_{twv}{}^s\, d\phi^v_a\, d\phi^w_c\, \nabla'_b d\phi^t_d\right] dV \\ &+ \int_{\partial M} h_{rs}\ g^{ab}\ N^c \left[\nabla'_a W^r \nabla'_c d\phi^s_b - W^r\ \nabla'_a \nabla'_b d\phi^s_c\right] d\tilde{V},\end{aligned}$$

*The variation of the **biharmonic energy** is given as,*

$$\begin{aligned}\frac{1}{2}\frac{d}{dt} S_{\text{biharmonic}}(\phi_t)\Big|_{t=0} = &\int_M g^{ac} g^{bd} h_{rs} W^r \left[\nabla'_c \nabla'_a \nabla'_b d\phi^s_d + R^N_{twv}{}^s\, d\phi^v_a\, d\phi^w_c\, \nabla'_b d\phi^t_d\right] dV \\ &+ \int_{\partial M} h_{rs}\ g^{ab}\ N^c \left[\nabla'_c W^r \nabla'_b d\phi^s_a - W^r\ \nabla'_c \nabla'_b d\phi^s_a\right] d\tilde{V}\end{aligned}$$

where $d\tilde{V}$ is the volume element of the boundary ∂M, $R^N_{uvw}{}^s$ is the curvature tensor of N and N^a is the normal vector field at ∂M.

Proof. For the harmonic energy we get with Lemma 4.28 and the extended Green's theorem 4.24,

$$\begin{aligned}\frac{1}{2}\frac{d}{dt} S_{\text{harmonic}}(\phi_t)\Big|_{t=0} &= \int_M g^{ab}\, h_{rs}\, \nabla'_a W^r\, d\phi^s_b\, dV(x) \\ &= \int_{\partial M} W^r\, h_{rs}\, N^b\, d\phi^s_b - \int_M W^r\, h_{rs}\, g^{ab}\, \nabla'_a d\phi^s_b.\end{aligned}$$

For the Eells energy we use the commutator of Theorem 4.28 and obtain,

$$\frac{1}{2}\frac{d}{dt}S_{\text{Eells}}(\phi_t) = \int_M g^{ab}\, g^{cd}\, h_{rs}\, \nabla'_a \nabla'_c \frac{\partial \phi^r}{\partial t}\, \nabla'_b (d\phi_t)^s_d dV \\ + \int_M g^{ab}\, g^{cd}\, h_{rs}\, R^N_{uvw}{}^r\, \frac{\partial \phi^u_t}{\partial t}\, (d\phi_t)^v_a\, (d\phi_t)^w_c\, \nabla'_b (d\phi_t)^s_d\, dV.$$

One has $\nabla'_b (d\phi_t)^s_d\Big|_{t=0} = \nabla'_b d\phi^s_d$. Applying twice the extended Green's theorem we obtain

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}S_{\text{Eells}}(\phi_t)\Big|_{t=0} &= \int_M g^{ab}\, g^{cd}\, h_{rs}\, \nabla'_a \nabla'_c W^r\, \nabla'_b d\phi^s_d\, dV \\ &\quad + \int_M g^{ab}\, g^{cd}\, h_{rs}\, R^N_{uvw}{}^r\, W^u\, d\phi^v_a\, d\phi^w_c\, \nabla'_b d\phi^s_d\, dV \\ &= \int_{\partial M} N^b\, g^{cd}\, h_{rs}\, \nabla'_c W^r\, \nabla'_b d\phi^s_d\, d\tilde{V} \\ &\quad - \int_{\partial M} g^{ab}\, N^d\, h_{rs}\, W^r\, \nabla'_a \nabla'_b d\phi^s_d\, d\tilde{V} \\ &\quad + \int_M g^{ab}\, g^{cd}\, h_{rs}\, W^r\, \nabla'_c \nabla'_a \nabla'_b d\phi^s_d\, dV \\ &\quad + \int_M g^{ab}\, g^{cd}\, h_{rs}\, R^N_{uvw}{}^r\, W^u\, d\phi^v_a\, d\phi^w_c\, \nabla'_b d\phi^s_d\, dV.\end{aligned}$$

The result follows noting that $R_{uvws} = R_{wsuv}$. The variation of the biharmonic energy can be derived analogously. □

A necessary condition for a minimiser of the energy $S(\phi)$ is that $\frac{d}{dt}S(\phi_t)\Big|_{t=0} = 0$ for all vector fields $W = \frac{\partial \phi}{\partial t}$.

Corollary 4.30. *For all points in the interior of $M \backslash \{X_1, .., X_K\}$ the minimiser $\phi : M \to N$ of the learning objective (4.1) satisfies for the*

$$\begin{aligned}&\textit{harm. energy:} && g^{ac}\nabla'_c d\phi^r_a = 0, \\ &\textit{biharm. energy:} && g^{ac}g^{bd}\Big[\nabla'_c\nabla'_a\nabla'_b d\phi^r_d + R^N_{twv}{}^r\, d\phi^v_a\, d\phi^w_c\, \nabla'_b d\phi^t_d\Big] = 0, \\ &\textit{Eells energy:} && g^{ab}g^{cd}\Big[\nabla'_c\nabla'_a\nabla'_b d\phi^r_d + R^N_{twv}{}^r\, d\phi^v_a\, d\phi^w_c\, \nabla'_b d\phi^t_d\Big] = 0.\end{aligned}$$

The following are natural boundary conditions at ∂M for the

$$\begin{aligned}&\textit{harm. energy:} && N^c d\phi^r_c = 0, \\ &\textit{biharm. energy:} && g^{ab}\,\nabla'_b d\phi^r_a = 0,\ \ N^c g^{ab}\nabla'_c\nabla'_b d\phi^r_a = 0, \\ &\textit{Eells energy:} && N^c\nabla'_c d\phi^r_b = 0,\ \ N^c g^{ab}\nabla'_a\nabla'_b d\phi^r_c = 0.\end{aligned}$$

The boundary conditions for the biharmonic and Eells energy are sufficient but not necessary for a minimiser. That means they guarantee that the sum of the two boundary terms in the variation vanishes, however, they are not the weakest possible conditions on ϕ. The given boundary conditions are nevertheless "natural" in the sense, that both ϕ and its derivative can be arbitrarily chosen on the boundary.

4.10.5 Table of Symbols

Symbol	Description
M,N	input, output manifold
m, n	dimension of M, N
x,y	coordinates on M, N
p	point on M, or in $\mathbb{R}^l$
a,b,c,d	abstract indices on M
r,s,t	abstract indices on N
α,β,γ	summation indices on M
μ,ν,ρ	summation indices on N
g_{ab},h_{ab}	Riemannian metric on M, N
${}^M\nabla$,${}^N\nabla$	Levi-Civita connections on M, N
${}^M\Gamma^{\alpha}_{\beta\gamma}$, ${}^N\Gamma^{\rho}_{\nu\mu}$	Christoffel symbols of the Levi-Civita connection on M, N
d_M,d_N	Riemannian metric on M, N
T_xM,T_yM	tangent space of M, N at x, y
ϕ	mapping from M to N
Ψ	mapping from M to $\mathbb{R}^l$
∇'	pull-back connection on M via ϕ
$\tilde{\nabla}$	pull-back connection on M via Ψ
$\mathbb{R}^k$,$\mathbb{R}^l$	embedding space of M, N
z	coordinates of the embedding spaces
X_i,X_j	training data inputs
Y_i,Y_j	training data outputs
K	number of training data points

Bibliography

Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97.

Apostol, T. (1990). *Modular Functions and Dirichlet Series in Number Theory*. Springer, New York.

Archambeau, C., Cornford, D., Opper, M., and Shawe-Taylor, J. (2007). Gaussian Process Approximations of Stochastic Differential Equations. *Journal of Machine Learning Research, Workshop and Conference Proceedings*, 1:1–16.

Aubert, G. and Kornprobst, P. (2006). *Mathematical Problems in Image Processing*. Springer, New York, second edition.

Belkin, M. and Niyogi, P. (2004). Semi-supervised learning on manifolds. *Machine Learning*, 56:209–239.

Bhattacharya, R. and Patrangenaru, V. (2003). Large sample theory of intrinsic and extrinsic sample means on manifolds I. *Annals of Statistics*, 31(1):1–29.

Blackwell, D. and Maitra, M. (1984). Factorization of probability measures and absolutely measurable sets. *Proceedings of the American Mathematical Society*, 92(2):251–254.

Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *SIGGRAPH'99 Conference Proceedings*, pages 187–194, Los Angeles. ACM Press.

Bochner, S. (1933). Monotone Funktionen, Stieltjessche Integrate und harmonische Analyse. *Mathematische Annalen*, 108:378–410.

Bogachev, V. I. (1998). *Gaussian Measures*. American Mathematical Society, Providence, RI.

Bousquet, O., Boucheron, S., and Lugosi, G. (2004). Introduction to statistical learning theory. In *Advanced Lectures in Machine Learning*, pages 169–207. Springer.

Buss, S. R. and Fillmore, J. P. (2001). Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics*, 20(2):95–126.

Camarinha, M., Silvia Leite, F., and Crouch, P. (1995). Splines of class C^k on non-euclidean spaces. *IMA Journal of Mathematical Control and Information*, 12(4):399–410.

Canu, S. and Smola, A. (2006). Kernel methods and the exponential family. *Neurocomputing*, 69(7-9):714–720.

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, 10:273–304.

Cokus, S. J., Rose, S., Haynor, D., Gronbech-Jensen, N., and Pellegrini, M. (2006). Modelling the network of cell cycle transcription factors in the yeast saccharomyces cerevisiae. *BMC Bioinformatics*, 7(38).

Cooke, T., Steinke, F., Wallraven, C., and Bülthoff, H. (2005). A similarity-based approach to perceptual feature validation. In *Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization*, pages 59 – 66, New York, NY, USA. ACM Press.

Cootes, T., Edwards, G., and Taylor, C. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

Curtain, R. and Zwart, H. (1995). *An Introduction to Infinite Dimensional Linear Systems Theory*. Springer.

Davis, B. C., Fletcher, P. T., Bullitt, E., and Joshi, S. (2007). Population shape regression from random design data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–7.

Daw, N., O'Doherty, J., Dayan, P., Seymour, B., and Dolan, R. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095):876–879.

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer.

DREAM (2006). The DREAM project, NYAS eBriefing. http://www.nyas.org/ebrief.

Duchamp, T. and Stuetzle, W. (2003). Spline smoothing on surfaces. *Journal of Computational and Graphical Statistics*, 12(2):354–381.

Eells, J. and Lemaire, L. (1983). *Selected topics in harmonic maps*. American Mathematical Society, Providence, RI.

Eells, J. and Sampson, J. H. (1964). Harmonic mappings of Riemannian manifolds. *American Journal of Mathematics*, 86(1):109–160.

Evans, L. (1998). *Partial differential equations*. American Mathematical Society, Providence, RI.

Fire, A., Xu, S., Montgomery, M. K., Kostas, S. A., Driver, S. E., and Mello, C. C. (1998). Potent and specific genetic interference by double-stranded RNA in caenorhabditis elegans. *Nature*, 391(6669):806–811.

Fisher, N. I., Lewis, T., and Embleton, B. J. J. (1993). *Statistical Analysis of Spherical Data*. Cambridge University Press, Cambridge, UK.

Floater, M. and Hormann, K. (2005). Surface parameterization: a tutorial and survey. In *Advances In Multiresolution For Geometric Modelling*. Springer.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3/4):601–620.

Gabriel, S. and Kajiya, K. (1985). Spline interpolation in curved space. In *SIGGRAPH'85 Course Notes on State of the Art Image Synthesis*.

Gardner, T. S., Cantor, C. R., and Collins, J. J. (2000). Construction of a genetic toggle switch in escherichia coli. *Nature*, 403(6767):339–342.

Girosi, F., Jones, M., and Poggio, T. (1993). Priors, stabilizers and basis functions: From regularization to radial, tensor and additive splines. A.I. Memo No. 1430, MIT.

Girosi, F., Jones, M., and Poggio, T. (1995). Regularization theory and neural network architectures. *Neural Computation*, 7:219–267.

Graepel, T. (2003). Solving Noisy Linear Operator Equations by Gaussian Processes: Application to Ordinary and Partial Differential Equations. In *Proceedings of the 20th International Conference on Machine Learning*, volume 20, pages 234–241.

Grochow, K., Martin, S., Hertzmann, A., and Popović, Z. (2004). Style-based inverse kinematics. *ACM Transactions on Graphics*, 23(3):522–531.

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., and Young, R. A. (2002). Bayesian methods for elucidating genetic regulatory networks. *IEEE Intelligent Systems*, 17(2):37–43.

Heckman, N. E. and Ramsay, J. O. (2000). Penalized regression with model-based penalties. *Canadian Journal of Statistics*, 28:241–258.

Hein, M., Audibert, J.-Y., and von Luxburg, U. (2007). Graph Laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8:1325–1370.

Hein, M. and Bousquet, O. (2004). Kernels, associated structures and generalizations. Technical Report 127, Max Planck Institute for Biological Cybernetics, Tübingen, Germany.

Hélein, F. and Wood, J. C. (2008). Harmonic maps. In *Handbook on global analysis*, pages 417–491. Elsevier.

Heuser, H. (1991). *Lehrbuch der Analysis, Teil 2*. B. G. Teubner, Stuttgart, Germany.

Hofer, M. and Pottmann, H. (2004). Energy-minimizing splines in manifolds. *ACM Transactions on Graphics*, 23:284–293.

Hofmann, M., Steinke, F., Scheel, V., Charpiat, G., Farquhar, J., Aschoff, P., Brady, M., Schölkopf, B., and Pichler, B. J. (2008). MRI-Based Attenuation Correction for PET/MRI: A Novel Approach Combining Pattern Recognition and Atlas Registration. *Journal of Nuclear Medicine*, 49(11):1875–1883.

Huang, Y. and McColl, W. (1997). Analytical inversion of general tridiagonal matrices. *Journal of Physics A: Mathematical and General*, 30:7919–7933.

Hume, D. (1748). *An Enquiry Concerning Human Understanding*.

Ideker, T., Thorsson, V., and Karp, R. (2000). Discovery of regulatory interactions through perturbation: inference and experimental design. In *Pacific Symposium on Biocomputing*, pages 305–316.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Julier, S. and Uhlmann, J. (1997). A new extension of the Kalman filter to nonlinear systems. In Kadar, I., editor, *Proceedings of the Conference on Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 182–193.

Kalberer, F., Nieser, M., and Polthier, K. (2007). QuadCover -surface parameterization using branched coverings. In *Computer Graphics Forum*, volume 26, pages 375–384. Blackwell Synergy.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Karcher, H. (1977). Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30:509–541.

Kendall, W. (1990). Probability, convexity, and harmonic maps with small image. I. Uniqueness and fine existence. *Proceedings of the London Mathematical Society*, 61(2):371–406.

Kholodenko, B. N., Kiyatkin, A., Bruggeman, F. J., Sontag, E., Westerhoff, H. V., and Hoek, J. B. (2002). Untangling the wires: A strategy to trace functional interactions in signaling and gene networks. *Proceedings of the National Academy of Sciences*, 99(20):12841–12846.

Kilian, M., Mitra, N., and Pottmann, H. (2007). Geometric modeling in shape space. *ACM Transactions on Graphics*, 26(3).

Kimeldorf, G. and Wahba, G. (1970). A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines. *The Annals of Mathematical Statistics*, 41(2):495–502.

Kimmel, R. and Sethian, J. (1998). Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences*, 95(15):8431–8435.

Kushner, H. and Budhiraja, A. (2000). A nonlinear filtering algorithm based on an approximation of the conditional distribution. *IEEE Transactions on Automatic Control*, pages 580–585.

Laplace, P.-S. (1814). *Essai philosophique sur les probabilités*.

Lawrence, N. D. and Quiñonero-Candela, J. (2006). Local distance preservation in the GP-LVM through back constraints. In *Proceedings of the International Conference in Machine Learning*, pages 513–520.

Lee, J. M. (1997). *Riemannian Manifolds - An introduction to curvature*. Springer, New York.

Levin, A., Lischinski, D., and Weiss, Y. (2004). Colorization using optimization. *ACM Transactions on Graphics*, 23(3):689–694.

Ljung, L. (1999). *System Identification – Theory for the user, 2nd edition*. Prentice Hall, Upper Saddle River, New Jersey.

Machado, L., Leite, F. S., and Hüper, K. (2006). Riemannian means as solutions of variational problems. *LMS Journal of Computation and Mathematics*, 9:86–103.

Madych, W. and Nelson, S. (1990). Multivariate Interpolation and Conditionally Positive Definite Functions. II. *Mathematics of Computation*, 54(189):211–230.

Mardia, K. and Jupp, P. (2000). *Directional statistics*. Wiley, New York.

Marsden, J. and Ratiu, T. (1999). *Introduction to Mechanics and Symmetry*. Springer.

Massonnet, D., Rossi, M., Carmona, C., Adragna, F., Peltzer, G., Feigl, K., and Rabaute, T. (1993). The displacement field of the Landers earthquake mapped by radar interferometry. *Nature*, 364(6433):138–142.

Maurer, A. (1984). Ockham's razor and Chatton's anti-razor. *Mediaeval Studies*, 46:463–475.

Micchelli, C. and Pontil, M. (2005). On learning vector-valued functions. *Neural Computation*, 17:177–204.

Minka, T. (2001). Expectation propagation for approximate Bayesian inference. In *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 362–369, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Minka, T. (2004). Power EP. Technical Report MSR-TR-2004-149, Mirosoft Research, Cambridge.

Mémoli, F., Sapiro, G., and Osher, S. (2004). Solving variational problems and partial differential equations mapping into general target manifolds. *Journal of Computational Physics*, 195(1):263–292.

Montaldo, S. and Oniciuc, C. (2005). A short survey on biharmonic maps between Riemannian manifolds. *ArXiv Mathematics e-prints*, page math/0510636.

Nakanishi, J., Cory, R., Mistry, M., Peters, J., and Schaal, S. (2005). Comparative experiments on task space control with redundancy resolution. In *Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems*.

Nash, J. (1956). The imbedding problem for Riemannian manifolds. *Annals of Mathematics*, 63(1):20–63.

Nishikawa, S. (2002). *Variational Problems in Geometry*. American Mathematical Society, Providence, RI.

Noakes, L., Heinzinger, G., and Paden, B. (1989). Cubic Splines on Curved Spaces. *IMA Journal of Mathematical Control and Information*, 6:465–473.

Noakes, L. and Popiel, T. (2007). Geometry for robot path planning. *Robotica*, 25:691–701.

O'Hagan, A. (1994). *Bayesian Inference*, volume 2B of *Kendall's Advanced Theory of Statistics*. Arnold, London.

Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., and Seidel, H.-P. (2003). Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22:463–470.

Oksendal, B. (2002). *Stochastic differential equations: an introduction with applications*. Springer, 6th edition.

Opper, M. and Winther, O. (2000a). Gaussian processes for classification: Mean field algorithms. *Neural Computation*, 12(11):2655–2684.

Opper, M. and Winther, O. (2000b). Gaussian Processes for Classification: Mean-Field Algorithms. *Neural Computation*, 12(11):2655–2684.

Peeters, R. and Westra, R. (2004). On the identification of sparse gene regulatory networks. In *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*.

Popper, K. (1934). *Logik der Forschung*.

Rahman, I. U., Drori, I., Stodden, V. C., Donoho, D. L., and Schroder, P. (2005). Multiscale representations for manifold-valued data. *Multiscale Modeling and Simulation*, 4(4):1201–1232.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer, second edition.

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

Rogers, S. and Girolami, M. (2005). A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics*, 21(14):3131–3137.

Schaback, R. (1995). Creating surfaces from scattered data using radial basis functions. In Daehlen, M., Lyche, T., and Schumaker, L., editors, *Mathematical Methods for Curves and Surfaces*, pages 477–496. Vanderbilt University Press, Nashville.

Schmidt, H., Cho, K.-H., and Jacobsen, E. (2005). Identification of small scale biochemical networks based on general type system perturbations. *FEBS Journal*, 272:2141–2151.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA.

Schölkopf, B., Steinke, F., and Blanz, V. (2005). Object correspondence as a machine learning problem. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 05)*.

Seeger, M. (2005). Expectation propagation for exponential families. Technical report, University of California at Berkeley. See www.kyb.tuebingen.mpg.de/bs/people/seeger.

Seeger, M. (2008). Bayesian inference and optimal design in the sparse linear model. *Journal of Machine Learning Research*, 9:759–813.

Seeger, M., Steinke, F., and Tsuda, K. (2006). Bayesian inference and optimal design in the sparse linear model. Technical report, Max Planck Institute for Biologic Cybernetics, Tübingen, Germany. See www.kyb.tuebingen.mpg.de/bs/people/seeger.

Seeger, M., Steinke, F., and Tsuda, K. (2007). Bayesian inference and optimal design in the sparse linear model. In *AISTATS07: Proceedings of the 11th International Workshop on AI and Statistics*.

Sheffer, A., Praun, E., and Rose, K. (2006). Mesh Parameterization Methods and Their Applications. *Foundations and Trends in Computer Graphics and Vision*, 2(2):105–171.

Shepard, R. (1980). Multidimensional Scaling, Tree-Fitting, and Clustering. *Science*, 210(4468):390–398.

Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (2002). Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274.

Smola, A. and Kondor, R. (2003). Kernels and regularization on graphs. In *Proceedings of the Conference on Learning Theory*. Springer, Berlin.

Smola, A., Schölkopf, B., and Müller, K. (1998). The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649.

Smolen, P., Baxter, D., and Byrne, J. (2000). Mathematical Modeling of Gene Networks. *Neuron*, 26:567–580.

Sontag, E., Kiyatkin, A., and Kholodenko, B. N. (2004). Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data. *Bioinformatics*, 20(12):1877–1886.

Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot Modeling and Control*. Wiley.

Srivastava, A. (2000). A Bayesian approach to geometric subspace estimation. *IEEE Transactions on Signal Processing*, 48(5):1390–1400.

Steinke, F. and Hein, M. (2009). Non-parametric regression between Riemannian manifolds. In *Advances in Neural Information Processing Systems*, volume 21.

Steinke, F., Hein, M., Peters, J., and Schölkopf, B. (2008). Manifold-valued Thin-Plate Splines with Applications in Computer Graphics. *Computer Graphics Forum*, 27(2):437–448.

Steinke, F., Hein, M., and Schölkpof, B. (2009). Non-parametric regression between general Riemannian manifolds. *SIAM Journal on Imaging Science*. (submitted).

Steinke, F. and Schölkopf, B. (2006). Machine learning methods for estimating operator equations. In *Proceedings of the 14th IFAC Symposium on System Identification (SYSID 2006)*. Elsevier.

Steinke, F. and Schölkopf, B. (2008). Kernels, regularization and differential equations. *Pattern Recognition*, 41(11):3271–3286.

Steinke, F., Schölkopf, B., and Blanz, V. (2007a). Learning dense 3D correspondence. In Schölkopf, B. and J. Platt, T. H., editors, *Advances in Neural Information Processing Systems*, volume 19, pages 1313–1320, Cambridge, MA, USA. MIT Press.

Steinke, F., Schölkopf, B., and Blanz, V. (2005). Support vector machines for 3D shape processing. *Computer Graphics Forum*, 24:285–294.

Steinke, F., Seeger, M., and Tsuda, K. (2007b). Experimental design for efficient identification of gene regulatory networks using sparse bayesian models. *BMC Systems Biology*, 1(51):1–15.

Tegnér, J., Yeung, M. K. S., Hasty, J., and Collins, J. J. (2003). Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proceedings of the National Academy of Sciences*, 100(10):5944–5949.

Tenenbaum, J., Silva, V., and Langford, J. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society B*, 58:267–288.

Tikhonov, A. (1943). On the stability of inverse problems. In *CR (Dokl.) Acad. Sci. URSS, n. Ser.*, volume 39, pages 176–179.

Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244.

Tipping, M. and Bishop, C. (2003). Bayesian Image Super-resolution. In *Advances in Neural Information Processing Systems*, volume 15, pages 1279 – 1286.

Urakawa, H. (1993). *Calculus of Variations and Harmonic Maps*. American Mathematical Society, Providence, RI.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.

Vert, J.-P., Foveau, N., Lajaunie, C., and Vandenbrouck, Y. (2006). An accurate and interpretable model for siRNA efficacy prediction. *BMC Bioinformatics*, 7(1):520.

von Dassow, G., Meir, E., Munro, E. M., and Odell, G. M. (2000). The segment polarity network is a robust developmental module. *Nature*, 406:188–192.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wahba, G. (1990). *Spline models for observational data*. Society for Industrial and Applied Mathematics.

Wald, R. (1984). *General Relativity*. The University of Chicago Press, Chicago.

Walder, C., Schölkopf, B., and Chapelle, O. (2006). Implicit surface modelling with a globally regularised basis of compact support. *Computer Graphics Forum*, 25(3):635–644.

Wallner, J., Pottmann, H., and Hofer, M. (2007). Fair webs. *The Visual Computer*, 23(1):83–94.

Wang, C. (2004). Stationary biharmonic maps from $\mathbb{R}^m$ into a Riemannian manifold. *Communications on Pure and Applied Mathematics*, 57:419–444.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440.

Wendland, H. (2005). *Scattered Data Approximation*. Cambridge University Press, Cambridge, UK.

Wolpert, D. (1996). The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation*, 8(7):1341–1390.

Yeung, M. K. S., Tegnér, J., and Collins, J. J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences*, 99:6163–6168.

Yoo, C. and Cooper, G. (2003). A Computer-Based Microarray Experiment Design-System for Gene-Regulation Pathway Discovery. *AMIA Annual Symposium Proceedings*, 2003:733–737.

Zayer, R., Rössl, C., and Seidel, H. (2005). Setting the boundary free: A composite approach to surface parameterization. *Symposium on Geometry Processing*, pages 91–100.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, volume 20.

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Herrenberg, 6.2.2009

Florian Steinke