Universität Münster
Institut für Theoretische Physik I

Bayesian Field Theory

Nonparametric Approaches to Density Estimation, Regression, Classification, and Inverse Quantum Problems

Jörg C. Lemm

This publication can be retrieved from the URL:
http://pauli.uni-muenster.de/~lemm/

Email: lemm@uni-muenster.de

Bayesian Field Theory

Nonparametric Approaches to Density Estimation, Regression, Classification, and Inverse Quantum Problems

Jörg C. Lemm*

Institut für Theoretische Physik I
Universität Münster
Wilhelm–Klemm–Str.9
D–48149 Münster, Germany

Abstract

Bayesian field theory denotes a nonparametric Bayesian approach for learning functions from observational data. Based on the principles of Bayesian statistics, a particular Bayesian field theory is defined by combining two models: a likelihood model, providing a probabilistic description of the measurement process, and a prior model, providing the information necessary to generalize from training to non–training data. The particular likelihood models discussed in the paper are those of general density estimation, Gaussian regression, clustering, classification, and models specific for inverse quantum problems. Besides problem typical hard constraints, like normalization and non–negativity for probabilities, prior models have to implement all the specific, and often vague, *a priori* knowledge available for a specific task. Nonparametric prior models discussed in the paper are Gaussian processes, mixtures of Gaussian processes, and non–quadratic potentials. Prior models are made flexible by including hyperparameters.

*Email: lemm@uni-muenster.de, WWW: http://pauli.uni-muenster.de/~lemm/

In particular, the adaption of mean functions and covariance operators of Gaussian process components is discussed in detail. Even if constructed using Gaussian process building blocks, Bayesian field theories are typically non–Gaussian and have thus to be solved numerically. According to increasing computational resources the class of non–Gaussian Bayesian field theories of practical interest which are numerically feasible is steadily growing. Models which turn out to be computationally too demanding can serve as starting point to construct easier to solve parametric approaches, using for example variational techniques.

Contents

1 Introduction

The last decade has seen a rapidly growing interest in learning from observational data. Increasing computational resources enabled successful applications of empirical learning algorithms in various areas including, for example, time series prediction, image reconstruction, speech recognition, computer tomography, and inverse scattering and inverse spectral problems for quantum mechanical systems. Empirical learning, i.e., the problem of finding underlying general laws from observations, represents a typical inverse problem and is usually ill–posed in the sense of Hadamard [220, 221, 224, 146, 115, 226]. It is well known that a successful solution of such problems requires additional *a priori* information. It is *a priori* information which controls the generalization ability of a learning system by providing the link between available empirical "training" data and unknown outcome in future "test" situations.

We will focus mainly on nonparametric approaches, formulated directly in terms of the function values of interest. Parametric methods, on the other hand, impose typically implicit restrictions which are often extremely difficult to relate to available *a priori* knowledge. Combined with a Bayesian framework [11, 16, 33, 145, 201, 174, 18, 69, 212, 35, 235, 42, 105, 104], a nonparametric approach allows a very flexible and interpretable implementation of *a priori* information in form of stochastic processes. Nonparametric Bayesian methods can easily be adapted to different learning situations and have therefore been applied to a variety of empirical learning problems, including regression, classification, density estimation and inverse quantum problems [170, 237, 143, 142, 138, 222]. Technically, they are related to kernel and regularization methods which often appear in the form of a roughness penalty approach [221, 224, 192, 211, 153, 228, 90, 83, 116, 226]. Computationally, working with stochastic processes, or discretized versions thereof, is more demanding than, for example, fitting a small number of parameters. This holds especially for such applications where one cannot take full advantage of the convenient analytical features of Gaussian processes. Nevertheless, it seems to be the right time to study nonparametric Bayesian approaches also for non–Gaussian problems as they become computationally feasible now at least for low dimensional systems and, even if not directly solvable, they provide a well defined basis for further approximations.

In this paper we will in particular study general density estimation problems. Those include, as special cases, regression, classification, and certain types of clustering. In density estimation the functions of interest are the probability densities $p(y|x,h)$, of producing output ("data") y under condition x and unknown state of Nature h. Considered as function of h, for fixed y, x, the function $p(y|x,h)$ is also known as *likelihood* function and a Bayesian approach to density estimation is based on a probabilistic model for likelihoods $p(y|x,h)$. We will concentrate on situations where y and x are real variables, possibly multi–dimensional. In a nonparametric approach, the variable h represents the whole likelihood function $p(y|x,h)$. That means, h may be seen as the collection of the numbers $0 \le p(y|x,h) \le 1$ for all x and all y. The dimension of h is thus infinite, if the number of values which the variables x and/or y can take is infinite. This is the case for real x and/or y.

A learning problem with discrete y variable is also called a *classification problem*. Restricting to Gaussian probabilities $p(y|x,h)$ with fixed variance leads to (Gaussian) *regression problems*. For regression problems the aim is to find an optimal regression function $h(x)$. Similarly, adapting a mixture of Gaussians allows *soft clustering* of data points. Furthermore, extracting relevant features from the predictive density $p(y|x,\text{data})$ is the Bayesian analogue of *unsupervised learning*. Other special density estimation problems are, for example, *inverse problems in quantum mechanics* where h represents a unknown potential to be determined from observational data [143, 142, 138, 222]. Special emphasis will be put on the explicit and flexible implementation of *a priori* information using, for example, mixtures of Gaussian prior processes with adaptive, non–zero mean functions for the mixture components.

Let us now shortly explain what is meant by the term "Bayesian Field Theory": From a physicists point of view functions, like $h(x,y) = p(y|x,h)$, depending on continuous variables x and/or y, are often called a 'field'.[1] Most times in this paper we will, as common in field theories in physics, not parameterize these fields but formulate the relevant probability densities or stochastic processes, like the prior $p(h)$ or the posterior $p(h|f)$, directly in terms of the field values $h(x,y)$, e.g., $p(h|f) = p(h(x,y), x \in X, y \in Y|f)$. (In the parametric case, discussed in Chapter 4, we obtain a probability density $p(h|f) = p(\xi|f)$ for fields $h(x,y,\xi)$ parameterized by ξ.)

The possibility to solve Gaussian integrals analytically makes Gaussian processes, or (generalized) free fields in the language of physicists, very at-

[1] We may also remark that for example statistical field theories, which encompass quantum mechanics and quantum field theory in their Euclidean formulation, are technically similar to a nonparametric Bayesian approach [249, 103, 127].

tractive for nonparametric learning. Unfortunately, only the case of Gaussian regression is completely Gaussian. For general density estimation problems the likelihood terms are non–Gaussian, and even for Gaussian priors additional non–Gaussian restrictions have to be included to ensure non–negativity and normalization of densities. Hence, in the general case, density estimation corresponds to a non–Gaussian, i.e., interacting field theory.

As it is well known from physics, a continuum limit for non-Gaussian theories, based on the definition of a renormalization procedure, can be highly nontrivial to construct. (See [20, 5] for an renormalization approach to density estimation.) We will in the following not discuss such renormalization procedures but focus more on practical, numerical learning algorithms, obtained by discretizing the problem (typically, but not necessarily in coordinate space). This is similar to what is done in lattice field theories.

Gaussian problems live effectively in a space with dimension not larger than the number of training data. This is not the case for non–Gaussian problems. Hence, numerical implementations of learning algorithms for non–Gaussian problems require to discretize the functions of interest. This can be computationally challenging.

For low dimensional problems, however, many non–Gaussian models are nowadays solvable on a standard PC. Examples include predictions of one–dimensional time series or the reconstruction of two–dimensional images. Higher dimensional problems require additional approximations, like projections into lower dimensional subspaces or other variational approaches. Indeed, it seems that a most solvable high dimensional problems live effectively in some low dimensional subspace.

There are special situations in classification where non–negativity and normalization constraints are fulfilled automatically. In that case, the calculations can still be performed in a space of dimension not larger than the number of training data. Contrasting Gaussian models, however the equations to be solved are then typically nonlinear.

Summarizing, we will call a nonparametric Bayesian model to learn a function one or more continuous variables a *Bayesian field theory*, having especially in mind non–Gaussian models. A large variety of Bayesian field theories can be constructed by combining a specific likelihood models with specific functional priors (see Tab. 1). The resulting flexibility of nonparametric Bayesian approaches is probably their main advantage.

likelihood model	prior model
describes	
measurement process (Chap. 2)	generalization behavior (Chap. 2)
is determined by	
parameters (Chap. 3, 4)	hyperparameters (Chap. 5)
Examples include	
density estimation(Sects. 3.1–3.6, 6.2)	hard constraints (Chap. 2)
regression (Sects. 3.7, 6.3)	Gaussian prior factors (Chap. 3)
classification (Sect. 3.8)	mixtures of Gauss. (Sects. 6.1–6.4)
inverse quantum theory (Sect. 3.9)	non–quadratic potentials(Sect. 6.5)
Learning algorithms are treated in Chapter 7.	

Table 1: A Bayesian approach is based on the combination of two models, a likelihood model, describing the measurement process used to obtain the training data, and a prior model, enabling generalization to non–training data. Parameters of the prior model are commonly called hyperparameters. In "nonparametric" approaches the collection of all values of the likelihood function itself are considered as the parameters. A nonparametric Bayesian approach for likelihoods depending on one or more real variables is in this paper called a Bayesian field theory.

The paper is organized as follows: Chapter 2 summarizes the Bayesian framework as needed for the subsequent chapters. Basic notations are defined, an introduction to Bayesian decision theory is given, and the role of *a priori* information is discussed together with the basics of a Maximum A Posteriori Approximation (MAP), and the specific constraints for density estimation problems. Gaussian prior processes, being the most commonly used prior processes in nonparametric statistics, are treated in Chapter 3. In combination with Gaussian prior models, this section also introduces the likelihood models of density estimation, (Sections 3.1, 3.2, 3.3) Gaussian regression and clustering (Section 3.7), classification (Section 3.8), and inverse quantum problems (Section 3.9). Notice, however, that all these likelihood models can also be combined with the more elaborated prior models discussed in the following sections of the paper. Parametric approaches, useful if a numerical solution of a full nonparametric approach is not feasible, are the topic of Chapter 4. Hyperparameters, parameterizing prior processes and making them more flexible, are considered in Section 5. Two possibilities to go beyond Gaussian processes, mixture models and non–quadratic potentials, are presented in Section 6. Chapter 7 focuses on learning algorithms, i.e., on methods to solve the stationarity equations resulting from a Maximum A Posteriori Approximation. In this section one can also find numerical solutions of Bayesian field theoretical models for general density estimation.

2 Bayesian framework

2.1 Basic model and notations

2.1.1 Independent, dependent, and hidden variables

Constructing theories means introducing concepts which are not directly observable. They should, however, explain empirical findings and thus have to be related to observations. Hence, it is useful and common to distinguish observable (visible) from non–observable (hidden) variables. Furthermore, it is often convenient to separate visible variables into dependent variables, representing results of measurements the theory is aiming to explain, and independent variables which are under explicit control and specify the kind of measurements performed.

Hence, we will consider the following three groups of variables

1. observable (visible) independent variables x,
2. observable (visible) dependent variables y,
3. not directly observable (hidden, latent) variables h.

This characterization of variables translates to the following factorization property, defining the model we will study,

$$p(x, y, h) = p(y|x, h)\, p(x)\, p(h). \tag{1}$$

In particular, we will be interested in scenarios where $x = (x_1, x_2, \cdots)$ and analogously $y = (y_1, y_2, \cdots)$ are decomposed into independent components, meaning that $p(y|x, h) = \prod_i p(y_i|x_i, h)$ and $p(x) = \prod_i p(x_i)$ factorize. Then,

$$p(x, y, h) = p(h) \prod_i p(y_i|x_i, h)\, p(x_i). \tag{2}$$

Fig.1 shows a graphical representation of the factorization model (2) as a directed acyclic graph [185, 125, 107, 200]. The x_i and/or y_i itself can also be vectors.

The interpretation will be as follows: Variables $h \in H$ represent possible *states of (the model of) Nature*, being the invisible conditions for dependent variables y. The set H defines the space of all possible states of Nature for the model under study. We assume that states h are not directly observable and all information about $p(h)$ comes from observed variables (data) y, x. A given set of observed data results in a *state of knowledge* f numerically represented by the *posterior density* $p(h|f)$ over states of Nature.

Independent variables $x \in X$ describe the visible conditions (measurement situation, measurement device) under which dependent variables (measurement results) y have been observed (measured). According to Eq. (1) they are independent of h, i.e., $p(x|h) = p(x)$. The conditional density $p(y|x, h)$ of the dependent variables y is also known as *likelihood* of h (under y given x). Vector–valued y can be treated as a collection of one–dimensional y with the vector index being part of the x variable, i.e., $y_\alpha(x) = y(x, \alpha) = y(\tilde{x})$ with $\tilde{x} = (x, \alpha)$.

In the setting of *empirical learning* available knowledge is usually separated into a finite number of *training data* $D = \{(x_i, y_i)|1 \le i \le n\}$ $=\{(x_D, y_D)$ and, to make the problem well defined, additional *a priori* information D_0. For data $D \cup D_0$ we write $p(h|f) = p(h|D, D_0)$. Hypotheses h

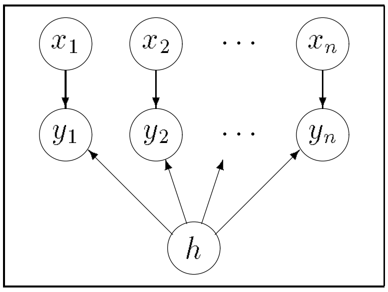

Figure 1: Directed acyclic graph for the factorization model (2).

represent in this setting functions $h(x, y) = p(y|x, h)$ of two (possibly multi-dimensional) variables y, x. In density estimation y is a continuous variable (the variable x may be constant and thus be skipped), while in classification problems y takes only discrete values. In regression problems on assumes $p(y|x, h)$ to be Gaussian with fixed variance, so the function of interest becomes the regression function $h(x) = \int dy\, y p(y|x, h)$.

2.1.2 Energies, free energies, and errors

Often it is more convenient to work with log–probabilities $L = \ln p$ than with probabilities. Firstly, this ensures non–negativity of probabilities $p = e^L \geq 0$ for arbitrary L. (For $p = 0$ the log–probability becomes $L = -\infty$.) Thus, when working with log–probabilities one can skip the non–negativity constraint which would be necessary when working with probabilities. Secondly, the multiplication of probabilities for independent events, yielding their joint probability, becomes a sum when written in terms of L. Indeed, from $p(A, B)$ $= p(A \text{ AND } B) = p(A)p(B)$ it follows for $L(A, B) = \ln P(A, B)$ that $L(A, B)$ $=L(A \text{ AND } B) = L(A)L(B)$. Especially in the limit where an infinite number of events is combined by AND, this would result in an infinite product for p but yields an integral for L, which is typically easier to treat.

Besides the requirement of being non–negative, probabilities have to be normalized, e.g., $\int dx\, p(x) = 1$. When dealing with a large set of elementary events normalization is numerically a nontrivial task. It is then convenient to work as far as possible with unnormalized probabilities $Z(x)$ from which normalized probabilities are obtained as $p(x) = Z(x)/Z$ with partition sum Z $= \int dx\, Z(x)$. Like for probabilities, it is also often advantageous to work with the logarithm of unnormalized probabilities, or to get positive numbers (for $p(x) < 1$) with the negative logarithm $E(x) = -(1/\beta) \ln Z(x)$, in physics also known as *energy*. (For the role of β see below.) Similarly, $F = -(1/\beta) \ln Z$ is known as *free energy*.

Defining the energy we have introduced a parameter β. Varying the parameter β generates an exponential family of densities which is frequently used in practice by (simulated or deterministic) annealing techniques for minimizing free energies [114, 156, 199, 43, 1, 203, 243, 68, 244, 245]. In physics β is known as *inverse temperature* and plays the role of a Lagrange multiplier in the maximum entropy approach to statistical physics. Inverse temperature β can also be seen as an external field coupling to the energy. Indeed, the free energy F is a generating function for the cumulants of the energy, meaning that cumulants of E can be obtained by taking derivatives of F with respect to β [62, 9, 13, 163]. For a detailled discussion of the relations between probability, log–probability, energy, free energy, partition sums, generating functions, and also bit numbers and information see [133].

The posterior $p(h|f)$, for example, can so be written as

$$\begin{aligned} p(h|f) &= e^{L(h|f)} = \frac{Z(h|f)}{Z(H|f)} = \frac{e^{-\beta E(h|f)}}{Z(H|f)} \\ &= e^{-\beta(E(h|f)-F(H|f))} = e^{-\beta E(h|f)+c(H|f)}, \end{aligned} \tag{3}$$

with (posterior) *log–probability*

$$L(h|f) = \ln p(h|f), \tag{4}$$

unnormalized (posterior) probabilities or *partition sums*

$$Z(h|f), \qquad Z(H|f) = \int dh\, Z(h|f), \tag{5}$$

(posterior) *energy*

$$E(h|f) = -\frac{1}{\beta} \ln Z(h|f) \tag{6}$$

and (posterior) *free energy*

$$F(H|f) = -\frac{1}{\beta} \ln Z(H|f) \tag{7}$$

$$= -\frac{1}{\beta} \ln \int dh\, e^{-\beta E(h|f)}, \tag{8}$$

yielding

$$Z(h|f) = e^{-\beta E(h|f)}, \qquad (9)$$
$$Z(H|f) = \int dh\, e^{-\beta E(h|f)}, \qquad (10)$$

where $\int dh$ represent a (functional) integral, for example over variables (functions) $h(x,y) = p(y|x,h)$, and

$$c(H|f) = -\ln Z(H|f) = \beta F(H|f). \qquad (11)$$

Note that we did not include the β–dependency of the functions Z, F, c in the notation.

For the sake of clarity, we have chosen to use the common notation for conditional probabilities also for energies and the other quantities derived from them. The same conventions will also be used for other probabilities, so we will write for example for likelihoods

$$p(y|x,h) = e^{-\beta'(E(y|x,h)-F(Y|x,h))}, \qquad (12)$$

for $y \in Y$. Inverse temperatures may be different for prior and likelihood. Thus, we may choose $\beta' \neq \beta$ in Eq. (12) and Eq. (3).

In Section 2.3 we will discuss the maximum a posteriori approximation where an optimal h is found by maximizing the posterior $p(h|f)$. Since maximizing the posterior means minimizing the posterior energy $E(h|f)$ the latter plays the role of an *error functional* for h to be minimized. This is technically similar to the minimization of an regularized error functional as it appears in regularization theory or empirical risk minimization, and which is discussed in Section 2.5.

Let us have a closer look to the integral over model states h. The variables h represent the parameters describing the data generating probabilities or likelihoods $p(y|x,h)$. In this paper we will mainly be interested in "nonparametric" approaches where the (x,y,h)–dependent numbers $p(y|x,h)$ itself are considered to be the primary degrees of freedom which "parameterize" the model states h. Then, the integral over h is an integral over a set of real variables indexed by x, y, under additional non–negativity and normalization condition.

$$\int dh \rightarrow \int \left(\prod_{x,y} dp(y|x,h) \right). \qquad (13)$$

Mathematical difficulties arise for the case of continuous x, y where $p(h|f)$ represents a stochastic process. and the integral over h becomes a functional integral over (non–negative and normalized) functions $p(y|x,h)$. For Gaussian processes such a continuum limit can be defined [51, 77, 228, 144, 150] while the construction of continuum limits for non–Gaussian processes is highly non–trivial (See for instance [48, 37, 103, 249, 189, 233, 234, 34, 206] for perturbative approaches or [77] for a non–perturbative ϕ^4–theory.) In this paper we will take the numerical point of view where all functions are considered to be finally discretized, so the h–integral is well–defined ("lattice regularization" [41, 204, 163]).

2.1.3 Posterior and likelihood

Bayesian approaches require the calculation of posterior densities. Model states h are commonly specified by giving the data generating probabilities or likelihoods $p(y|x,h)$. Posteriors are linked to likelihoods by Bayes' theorem

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}, \tag{14}$$

which follows at once from the definition of conditional probabilities, i.e., $p(A,B) = p(A|B)p(B) = p(B|A)p(A)$. Thus, one finds

$$p(h|f) = p(h|D,D_0) = \frac{p(D|h)\,p(h|D_0)}{p(D|D_0)} = \frac{p(y_D|x_D,h)\,p(h|D_0)}{p(y_D|x_D,D_0)} \tag{15}$$

$$= \frac{p(h|D_0)\prod_i p(x_i,y_i|h)}{\int dh\, p(h|D_0)\prod_i p(x_i,y_i|h)} = \frac{p(h|D_0)\prod_i p(y_i|x_i,h)}{\int dh\, p(h|D_0)\prod_i p(y_i|x_i,h)}, \tag{16}$$

using $p(y_D|x_D,D_0,h) = p(y_D|x_D,h)$ for the training data likelihood of h and $p(h|D_0,x_i) = p(h|D_0)$. The terms of Eq. (15) are in a Bayesian context often referred to as

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}. \tag{17}$$

Eqs.(16) show that the posterior can be expressed equivalently by the joint likelihoods $p(y_i,x_i|h)$ or conditional likelihoods $p(y_i|x_i,h)$. When working with joint likelihoods, a distinction between y and x variables is not necessary. In that case x can be included in y and skipped from the notation. If, however, $p(x)$ is already known or is not of interest working with conditional likelihoods is preferable. Eqs.(15,16) can be interpreted as updating (or learning) formula used to obtain a new posterior from a given prior probability if new data D arrive.

In terms of energies Eq. (16) reads,

$$p(h|f) = \frac{e^{-\beta\sum_i E(y_i|x_i,h)-\beta E(h|D_0)}}{Z(Y_D|x_D,h)\,Z(H|D_0)}\left(\int dh'\,\frac{e^{-\beta\sum_i E(y_i|x_i,h')-\beta E(h'|D_0)}}{Z(Y_D|x_D,h')\,Z(H|D_0)}\right)^{-1}, \quad (18)$$

where the same temperature $1/\beta$ has been chosen for both energies and the normalization constants are

$$Z(Y_D|x_D,h) \;=\; \prod_i \int dy_i\, e^{-\beta E(y_i|x_i,h)}, \quad (19)$$

$$Z(H|D_0) \;=\; \int dh\, e^{-\beta E(h|D_0)}. \quad (20)$$

The predictive density we are interested in can be written as the ratio of two correlation functions under $p_0(h)$,

$$p(y|x,f) \;=\; < p(y|x,h) >_{H|f} \quad (21)$$

$$= \frac{< p(y|x,h)\prod_i p(y_i|x_i,h) >_{H|D_0}}{< \prod_i p(y_i|x_i,h) >_{H|D_0}}, \quad (22)$$

$$= \frac{\int dh\, p(y|x,h)\, e^{-\beta E_{\rm comb}}}{\int dh\, e^{-\beta E_{\rm comb}}} \quad (23)$$

where $< \cdots >_{H|D_0}$ denotes the expectation under the prior density $p_0(h)$ $= p(h|D_0)$ and the *combined likelihood and prior energy* $E_{\rm comb}$ collects the h–dependent energy and free energy terms

$$E_{\rm comb} = \sum_i E(y_i|x_i,h) + E(h|D_0) - F(Y_D|x_D,h), \quad (24)$$

with

$$F(Y_D|x_D,h) = -\frac{1}{\beta}\ln Z(Y_D|x_D,h). \quad (25)$$

Going from Eq. (22) to Eq. (23) the normalization factor $Z(H|D_0)$ appearing in numerator and denominator has been canceled.

We remark that for continuous x and/or y the likelihood energy term $E(y_i|x_i,h)$ describes an ideal, non–realistic measurement because realistic measurements cannot be arbitrarily sharp. Considering the function $p(\cdot|\cdot,h)$ as element of a Hilbert space its values may be written as scalar product

$p(y|x,h) = (v_{xy},\, p(\cdot|\cdot,h)\,)$ with a function v_{xy} being also an element in that Hilbert space. For continuous x and/or y this notation is only formal as v_{xy} becomes unnormalizable. In practice a measurement of $p(\cdot|\cdot,h)$ corresponds to a normalizable $v_{\tilde{x}\tilde{y}} = \int dy \int dx\, \vartheta(x,y) v_{xy}$ where the kernel $\vartheta(x,y)$ has to ensure normalizability. (Choosing normalizable $v_{\tilde{x}\tilde{y}}$ as coordinates the Hilbert space of $p(\cdot|\cdot,h)$ is also called a reproducing kernel Hilbert space [183, 112, 113, 228, 144].) The data terms then become

$$p(\tilde{y}_i|\tilde{x}_i,h) = \frac{\int dy\, dx\, \vartheta(\tilde{x},\tilde{y};x,y) p(y,x|h)}{\int d\tilde{y}\, dy\, dx \vartheta(\tilde{y},\tilde{y};x,y) p(y,x|h)}. \tag{26}$$

The notation $p(y_i|x_i,h)$ is understood as limit $\vartheta(x,y) \to \delta(x-x_i)\delta(y-y_i)$ and means in practice that $\vartheta(x,y)$ is very sharply centered. We will assume that the discretization, finally necessary to do numerical calculations, will implement such an averaging.

2.1.4 Predictive density

Within a Bayesian approach predictions about (e.g., future) events are based on the *predictive probability density*, being the expectation of probability for y for given (test) situation x, training data D and prior data D_0

$$p(y|x,f) = p(y|x,D,D_0) = \int dh\, p(h|f)\, p(y|x,h) = <p(y|x,h)>_{H|f}\,. \tag{27}$$

Here $<\cdots>_{H|f}$ denotes the expectation under the posterior $p(h|f) = p(h|D,D_0)$, the state of knowledge f depending on prior and training data. Successful applications of Bayesian approaches rely strongly on an adequate choice of the model space H and model likelihoods $p(y|x,h)$.

Note that $p(y|x,f)$ is in the convex cone spanned by the possible states of Nature $h \in H$, and typically not equal to one of these $p(y|x,h)$. The situation is illustrated in Fig. 2. During learning the predictive density $p(y|x,f)$ tends to approach the true $p(y|x,h)$. Because the training data are random variables, this approach is stochastic. (There exists an extensive literature analyzing the stochastic properties of learning and generalization from a statistical mechanics perspective [63, 64, 65, 231, 239, 178]).

2.1.5 Mutual information and learning

The aim of learning is to generalize the information obtained from training data to non–training situations. For such a *generalization* to be possible,

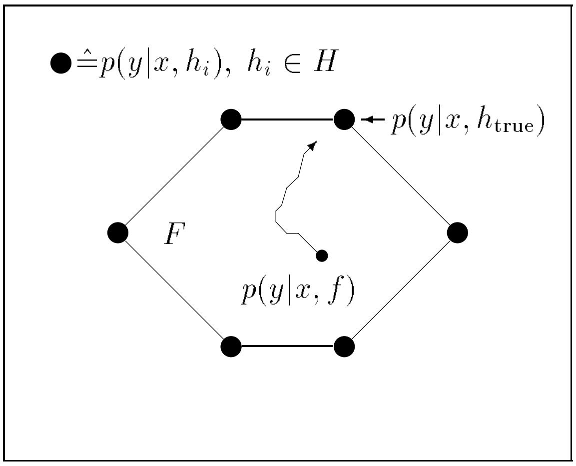

Figure 2: The predictive density $p(y|x, f)$ for a state of knowledge $f = f(D, D_0)$ is in the convex hull spanned by the possible states of Nature h_i characterized by the likelihoods $p(y|x, h_i)$. During learning the actual predictive density $p(y|x, f)$ tends to move stochastically towards the extremal point $p(y|x, h_{\text{true}})$ representing the "true" state of Nature.

there must exist a, at least partially known, relation between the likelihoods $p(y_i|x_i, h)$ for training and for non–training data. This relation is typically provided by *a priori* knowledge.

One possibility to quantify the relation between two random variables y_1 and y_2, representing for example training and non–training data, is to calculate their *mutual information*, defined as

$$M(Y_1, Y_2) = \sum_{y_1 \in Y_1, y_2 \in Y_2} p(y_1, y_2) \ln \frac{p(y_1, y_2)}{p(y_1)p(y_2)}. \tag{28}$$

It is also instructive to express the mutual information in terms of (average) information content or entropy, which, for a probability function $p(y)$, is defined as

$$H(Y) = -\sum_{y \in Y} p(y) \ln p(y). \tag{29}$$

We find

$$M(Y_1, Y_2) = H(Y_1) + H(Y_2) - H(Y_1, Y_2), \tag{30}$$

meaning that the mutual information is the sum of the two individual entropies diminished by the entropy common to both variables.

To have a compact notation for a family of predictive densities $p(y_i|x_i, f)$ we choose a vector $x = (x_1, x_2, \cdots)$ consisting of all possible values x_i and corresponding vector $y = (y_1, y_2, \cdots)$, so we can write

$$p(y|x, f) = p(y_1, y_2, \cdots |x_1, x_2, \cdots, f). \tag{31}$$

We now would like to characterize a state of knowledge f corresponding to predictive density $p(y|x, f)$ by its mutual information. Thus, we generalize the definition (28) from two random variables to a random vector y with components y_i, given vector x with components x_i and obtain the conditional mutual information

$$M(Y|x, f) = \int \left(\prod_i dy_i\right) p(y|x, f) \ln \frac{p(y|x, f)}{\prod_j p(y_j|x_j, f)}, \tag{32}$$

or

$$M(Y|x, f) = \left(\int dy_i \, H(Y_i|x, f) - H(Y|x, f)\right), \tag{33}$$

in terms of conditional entropies

$$H(Y|x, f) = -\int dy \, p(y|x, f) \ln p(y|x, f). \tag{34}$$

In case not a fixed vector x is given, like for example $x = (x_1, x_2, \cdots)$, but a density $p(x)$, it is useful to average the conditional mutual information and conditional entropy by including the integral $\int dx \, p(x)$ in the above formulae.

It is clear from Eq. (32) that predictive densities which factorize

$$p(y|x, f) = \prod_i p(y_i|x_i, f), \tag{35}$$

have a mutual information of zero. Hence, such *factorial states* do not allow any generalization from training to non–training data. A special example are the possible states of Nature or pure states h, which factorize according to the definition of our model

$$p(y|x, h) = \prod_i p(y_i|x_i, h). \tag{36}$$

Thus, pure states do not allow any further generalization. This is consistent with the fact that pure states represent the natural endpoints of any learning process.

It is interesting to see, however, that there are also other states for which the predictive density factorizes. Indeed, from Eq. (36) it follows that any (prior or posterior) probability $p(h)$ which factorizes leads to a factorial state,

$$p(h) = \prod_i p(h(x_i)) \Rightarrow p(y|x,f) = \prod_i p(y_i|x_i,f). \tag{37}$$

This means generalization, i.e., (non–local) learning, is impossible when starting from a *factorial prior.*

A factorial prior provides a very clear reference for analyzing the role of a–priori information in learning. In particular, with respect to a prior factorial in local variables x_i, learning may be decomposed into two steps, one increasing, the other lowering mutual information:

1. Starting from a factorial prior, new *non–local data* D_0 (typically called *a priori* information) produce a new non–factorial state with non–zero mutual information.

2. *Local data* D (typically called training data) stochastically reduce the mutual information. Hence, learning with local data corresponds to a *stochastic decay of mutual information.*

Pure states, i.e., the extremal points in the space of possible predictive densities, do not have to be deterministic. Improving measurement devices, stochastic pure states may be further decomposed into finer components g, so that

$$p(y_i|x_i,h) = \int dg\, p(g)\, p(y_i|x_i,g). \tag{38}$$

Imposing a non–factorial prior $p(g)$ on the new, finer hypotheses g enables again non–local learning with local data, leading asymptotically to one of the new pure states $p(y_i|x_i,g)$.

Let us exemplify the stochastic decay of mutual information by a simple numerical example. Because the mutual information requires the integration over all y_i variables we choose a problem with only two of them, y_a and y_b corresponding to two x values x_a and x_b. We consider a model with four states of Nature h_l, $1 \le l \le 4$, with Gaussian likelihood $p(y|x,h) = (\sqrt{2\pi}\sigma)^{-1} \exp\left(-(y-h_i(x))^2/(2\sigma^2)\right)$ and local means $h_l(x_j) = \pm 1$.

Selecting a "true" state of Nature h, we sample 50 data points $D_i = (x_i, y_i)$ from the corresponding Gaussian likelihood using $p(x_a) = p(x_b) =$

0.5. Then, starting from a given, factorial or non–factorial, prior $p(h|D_0)$ we sequentially update the predictive density,

$$p(y|x, f(D_{i+1}, \cdots, D_0)) = \sum_{l=1}^{4} p(y|x, h_l)\, p(h_l|D_{i+1}, \cdots, D_0), \tag{39}$$

by calculating the posterior

$$p(h_l|D_{i+1}, \cdots, D_0) = \frac{p(y_{i+1}|x_{i+1}, h_l)\, p(h_j|D_i \cdots, D_0)}{p(y_{i+1}|x_{i+1}, D_i \cdots, D_0)}. \tag{40}$$

It is easily seen from Eq. (40) that factorial states remain factorial under local data.

Fig. 3 shows that indeed the mutual information decays rapidly. Depending on the training data, still the wrong hypothesis h_l may survive the decay of mutual information. Having arrived at a factorial state, further learning has to be local. That means, data points for x_i can then only influence the predictive density for the corresponding y_i and do not allow generalization to the other y_j with $j \neq i$.

For a factorial prior $p(h_l) = p(h_l(x_a))p(h_l(x_b))$ learning is thus local from the very beginning. Only very small numerical random fluctuations of the mutual information occur, quickly eliminated by learning. Thus, the predictive density moves through a sequence of factorial states.

2.2 Bayesian decision theory

2.2.1 Loss and risk

In *Bayesian decision theory* a set A of possible actions a is considered, together with a function $l(x, y, a)$ describing the *loss* l suffered in situation x if y appears and action a is selected [16, 128, 185, 201]. The loss averaged over *test data* x, y, and possible states of Nature h is known as *expected risk*,

$$\begin{aligned} r(a, f) &= \int dx\, dy\, p(x)\, p(y|x, f)\, l(x, y, a). & (41)\\ &= < l(x, y, a) >_{X,Y|f} & (42)\\ &= < r(a, h) >_{H|f} & (43) \end{aligned}$$

where $< \cdots >_{X,Y|f}$ denotes the expectation under the joint predictive density $p(x, y|f) = p(x)p(y|x, f)$ and

$$r(a, h) = \int dx\, dy\, p(x)\, p(y|x, h)\, l(x, y, a). \tag{44}$$

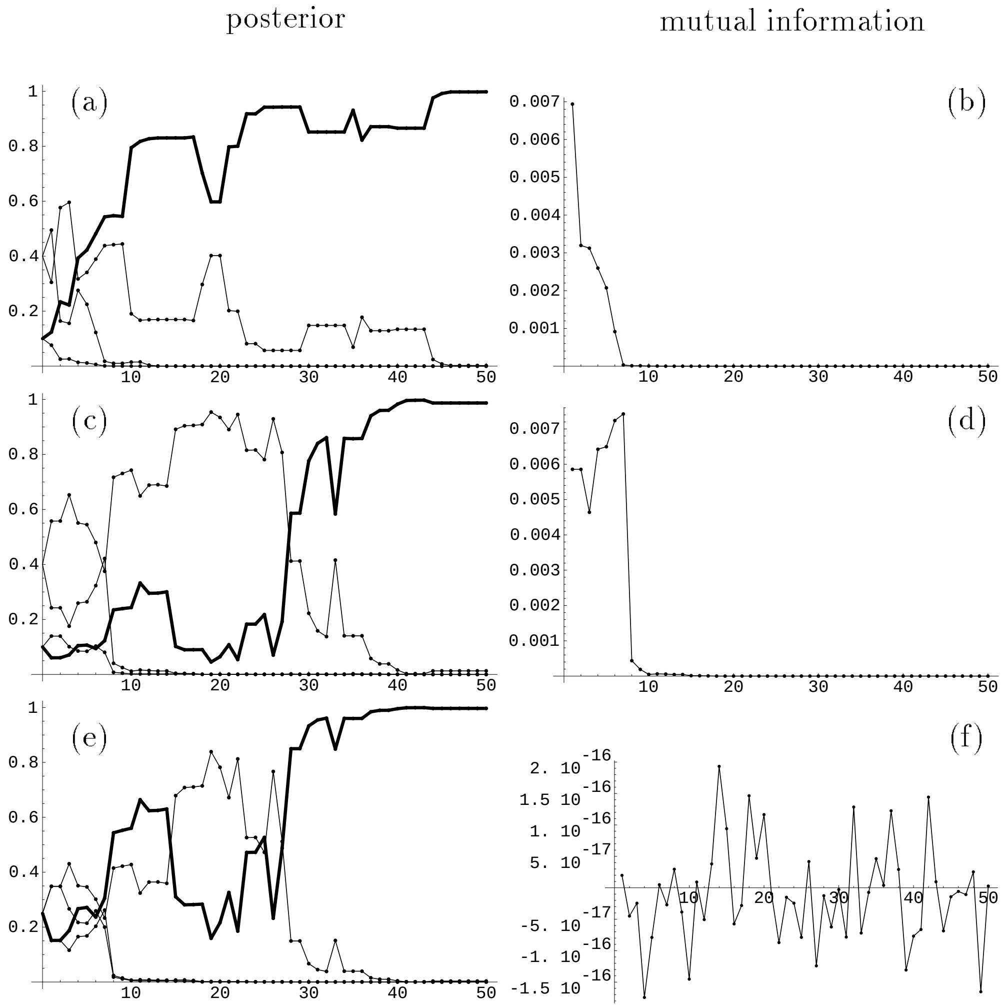

Figure 3: The decay of mutual information during learning: Model with 4 possible states h_l representing Gaussian likelihoods $p(y_i|x_i, h_l)$ with means ± 1 for two different x_i values. Shown are posterior probabilities $p(h_l|f)$ (a, c, e, on the left hand side, the posterior of the true h_l is shown by a thick line) and mutual information $M(y)$ (b, d, f, on the right hand side) during learning 50 training data. (a, b): The mutual information decays during learning and becomes quickly practically zero. (c, d): For "unlucky" training data the wrong hypothesis h_i can dominate at the beginning. Nevertheless, the mutual information decays and the correct hypothesis has finally to be found through "local" learning. (e, f): Starting with a factorial prior the mutual information is and remains zero, up to artificial numerical fluctuations. For (e, f) the same random data have been used as for (c, d).

The aim is to find an optimal action a^*

$$a^* = \text{argmin}_{a\in A} r(a,f). \tag{45}$$

2.2.2 Loss functions for approximation

Log–loss: A typical loss function for *density estimation problems* is the *log–loss*

$$l(x,y,a) = -b_1(x)\ln p(y|x,a) + b_2(x,y) \tag{46}$$

with some a–independent $b_1(x) > 0$, $b_2(x,y)$ and actions a describing probability densities

$$\int dy\, p(y|x,a) = 1,\ \forall x \in X, \forall a \in A. \tag{47}$$

Choosing $b_2(x,y) = \ln p(y|x,f)$ and $b_1(x) = 1$ gives

$$r(a,f) = \int dx\, dy\, p(x)p(y|x,f)\ln\frac{p(y|x,f)}{p(y|x,a)} \tag{48}$$

$$= <\ln\frac{p(y|x,f)}{p(y|x,a)}>_{X,Y|f} \tag{49}$$

$$= <\text{KL}(\, p(y|x,f),\, p(y|x,a)\,)>_X, \tag{50}$$

which shows that minimizing log–loss is equivalent to minimizing the (x–averaged) *Kullback–Leibler entropy* $\text{KL}(\, p(y|x,f),\, p(y|x,a)\,)$[122, 123, 13, 46, 53].

While the paper will concentrate on log–loss we will also give a short summary of loss functions for *regression problems*. (See for example [16, 201] for details.) Regression problems are special density estimation problems where the considered possible actions are restricted to y–independent functions $a(x)$.

Squared–error loss: The most common loss function for regression problems (see Sections 3.7, 3.7.2) is the squared–error loss. It reads for one–dimensional y

$$l(x,y,a) = b_1(x)\,(y - a(x))^2 + b_2(x,y), \tag{51}$$

with arbitrary $b_1(x) > 0$ and $b_2(x,y)$. In that case the optimal function $a(x)$ is the *regression function* of the posterior which is the mean of the predictive density

$$a^*(x) = \int dy\, y\, p(y|x,f) = <y>_{Y|x,f}\,. \tag{52}$$

This can be easily seen by writing

$$
\begin{aligned}
(y-a(x))^2 &= \Big(y-<y>_{Y|x,f}+<y>_{Y|x,f}-a(x)\Big)^2 &(53)\\
&= \Big(y-<y>_{Y|x,f}\Big)^2+\Big(a(x)-<y>_{Y|x,f}\Big)^2\\
&\quad -2\Big(y-<y>_{Y|x,f}\Big)\Big(a(x)-<y>_{Y|x,f}\Big), &(54)
\end{aligned}
$$

where the first term in (54) is independent of a and the last term vanishes after integration over y according to the definition of $<y>_{Y|x,f}$. Hence,

$$
r(a,f)=\int dx\, b_1(x)p(x)\Big(a(x)-<y>_{Y|x,f}\Big)^2+\text{const.} \tag{55}
$$

This is minimized by $a(x)=<y>_{Y|x,f}$. Notice that for Gaussian $p(y|x,a)$ with fixed variance log–loss and squared-error loss are equivalent. For multi–dimensional y one–dimensional loss functions like Eq. (51) can be used when the component index of y is considered part of the x–variables. Alternatively, loss functions depending explicitly on multidimensional y can be defined. For instance, a general quadratic loss function would be

$$
l(x,y,a)=\sum_{k,k'}(y_k-a_k)\mathbf{K}(k,k')(y_{k'}-a_{k'}(x)). \tag{56}
$$

with symmetric, positive definite kernel $\mathbf{K}(k,k')$.

Absolute loss: For absolute loss

$$
l(x,y,a)=b_1(x)|y-a(x)|+b_2(x,y), \tag{57}
$$

with arbitrary $b_1(x)>0$ and $b_2(x,y)$. The risk becomes

$$
\begin{aligned}
r(a,f) &= \int dx\, b_1(x)p(x)\int_{-\infty}^{a(x)}dy\,(a(x)-y)\,p(y|x,f)\\
&\quad +\int dx\, b_1(x)p(x)\int_{a(x)}^{\infty}dy\,(y-a(x))\,p(y|x,f)+\text{const.} &(58)\\
&= 2\int dx\, b_1(x)p(x)\int_{m(x)}^{a(x)}dy\,(a(x)-y)\,p(y|x,f)+\text{const.}', &(59)
\end{aligned}
$$

where the integrals have been rewritten as $\int_{-\infty}^{a(x)}=\int_{-\infty}^{m(x)}+\int_{m(x)}^{a(x)}$ and $\int_{a(x)}^{\infty}=\int_{a(x)}^{m(x)}+\int_{m(x)}^{\infty}$ introducing a median function $m(x)$ which satisfies

$$
\int_{-\infty}^{m(x)}dy\,p(y|x,f)=\frac{1}{2},\ \forall x\in X, \tag{60}
$$

so that

$$a(x)\left(\int_{-\infty}^{m(x)} dy\, p(y|x,f) - \int_{m(x)}^{\infty} dy\, p(y|x,f)\right) = 0,\ \forall x \in X. \tag{61}$$

Thus the risk is minimized by any *median function* $m(x)$.

δ–loss and 0–1 loss : Another possible loss function, typical for classification tasks (see Section 3.8), like for example image segmentation [153], is the δ–loss for continuous y or 0–1–loss for discrete y

$$l(x,y,a) = -b_1(x)\delta\left(y - a(x)\right) + b_2(x,y), \tag{62}$$

with arbitrary $b_1(x) > 0$ and $b_2(x,y)$. Here δ denotes the Dirac δ–functional for continuous y and the Kronecker δ for discrete y. Then,

$$r(a,f) = -\int dx\, b_1(x)p(x)\, p(\,y\!=\!a(x)\,|x,f) + \text{const.}, \tag{63}$$

so the optimal a corresponds to any *mode function* of the predictive density. For Gaussians mode and median are unique, and coincide with the mean.

2.2.3 General loss functions and unsupervised learning

Choosing actions a in specific situations often requires the use of specific loss functions. Such loss functions may for example contain additional terms measuring *costs* of choosing action a not related to approximation of the predictive density. Such costs can quantify aspects like the simplicity, implementability, production costs, sparsity, or understandability of action a.

Furthermore, instead of approximating a whole density it often suffices to extract some of its features, like identifying clusters of similar y–values, finding independent components for multidimensional y, or mapping to an approximating density with lower dimensional x. This kind of *exploratory data analysis* is the Bayesian analogue to *unsupervised learning methods*. Such methods are on one hand often utilized as a preprocessing step but are, on the other hand, also important to choose actions for situations where specific loss functions can be defined.

From a Bayesian point of view general loss functions require in general an explicit two–step procedure [132]: 1. Calculate (an approximation of) the predictive density, and 2. Minimize the expectation of the loss function under that (approximated) predictive density. (Empirical risk minimization, on the other hand, minimizes the empirical average of the (possibly regularized) loss function, see Section 2.5.) (For a related example see for instance [139].)

For a Bayesian version of cluster analysis, for example, partitioning a predictive density obtained from empirical data into several clusters, a possible loss function is

$$l(x, y, a) = (y - a(x, y))^2, \tag{64}$$

with action $a(x, y)$ being a mapping of y for given x to a finite number of cluster centers (prototypes). Another example of a clustering method based on the predictive density is Fukunaga's valley seeking procedure [61].

For multidimensional x a space of actions $a(\mathbf{P}_x x, y)$ can be chosen depending only on a (possibly adaptable) lower dimensional projection of x.

For multidimensional y with components y_i it is often useful to identify independent components. One may look, say, for a linear mapping $\tilde{y} = \mathbf{M}y$ minimizing the correlations between different components of the 'source' variables $\tilde{y}$ by minimizing the loss function

$$l(y, y', \mathbf{M}) = \sum_{i \neq j} \tilde{y}_i \, \tilde{y}'_j, \tag{65}$$

with respect to $\mathbf{M}$ under the joint predictive density for y and y' given x, x', D, D_0. This includes a Bayesian version of blind source separation (e.g. applied to the so called cocktail party problem [14, 7]), analogous to the treatment of Molgedey and Schuster [162]. Interesting projections of multidimensional y can for example be found by projection pursuit techniques [59, 102, 108, 211].

2.3 Maximum A Posteriori Approximation

In most applications the (usually very high or even formally infinite dimensional) h–integral over model states in Eq. (23) cannot be performed exactly. The two most common methods used to calculate the h integral approximately are *Monte Carlo integration* [155, 91, 95, 198, 16, 70, 199, 21, 219, 238, 69, 170, 180, 202, 171] and *saddle point approximation* [16, 45, 30, 172, 17, 249, 201, 69, 76, 132]. The latter approach will be studied in the following.

For that purpose, we expand $E_{\rm comb}$ of Eq. (24) with respect to h around some h^*

$$\begin{aligned} E_{\rm comb}(h) &= e^{(\Delta h, \nabla)} E(h)\Big|_{h=h^*} \qquad (66) \\ &= E_{\rm comb}(h^*) + (\Delta h, \nabla(h^*)) + \frac{1}{2}(\Delta h, \mathbf{H}(h^*)\Delta h) + \cdots \end{aligned}$$

with $\Delta h = (h - h^*)$, gradient ∇ (not acting on Δh), Hessian $\mathbf{H}$, and round brackets $(\cdots, \cdots)$ denoting scalar products. In case $p(y|x,h)$ is parameterized independently for every x, y the states h represent a parameter set indexed by x and y, hence

$$\nabla(x, y; h^*) = \left.\frac{\delta E_{\rm comb}(h)}{\delta h(x,y)}\right|_{h=h^*} = \left.\frac{\delta E_{\rm comb}(p(y'|x',h))}{\delta p(y|x,h)}\right|_{h=h^*}, \tag{67}$$

$$\mathbf{H}(x, y, x', y'; h^*) = \left.\frac{\delta^2 E_{\rm comb}(h)}{\delta h(x,y)\delta h(x',y')}\right|_{h=h^*} = \left.\frac{\delta^2 E_{\rm comb}(p(y''|x'',h))}{\delta p(y|x,h)\delta p(y'|x',h)}\right|_{h=h^*}, \tag{68}$$

are functional derivatives [97, 106, 29, 36] (or partial derivatives for discrete x, y) and for example

$$(\Delta h, \nabla h^*) = \int dx\, dy\ (h(x,y) - h^*(x,y))\, \nabla(h^*)(x,y). \tag{69}$$

Choosing h^* to be the location of a local minimum of $E_{\rm comp}(h)$ the linear term in (66) vanishes. The second order term includes the Hessian and corresponds to a Gaussian integral over h which could be solved analytically

$$\int dh\, e^{-\frac{1}{2}(\Delta h, \mathbf{H}\Delta h)} = (2\pi)^{\frac{d}{2}} (\det \mathbf{H})^{-\frac{1}{2}}, \tag{70}$$

for a d–dimensional h–integral. However, using the same approximation for the h–integrals in numerator and denominator of Eq. (23), expanding then also $p(y|x,h)$ around h^*, and restricting to the first (h–independent) term $p(y|x,h^*)$ of that expansion, the factor (70) cancels, even for infinite d. (The result is the zero order term of an expansion of the predictive density in powers of $1/\beta$. Higher order contributions can be calculated by using Wick's theorem [45, 30, 172, 249, 109, 163, 132].) The final approximative result for the predictive density (27) is very simple and intuitive

$$p(y|x,f) \approx p(y|x,h^*), \tag{71}$$

with

$$h^* = \text{argmin}_{h\in H} E_{comb} = \text{argmax}_{h\in H}\, p(h|f) = \text{argmax}_{h\in H}\, p(y_D|x_D,h) p(h|D_0). \tag{72}$$

The saddle point (or Laplace) approximation is therefore also called *Maximum A Posteriori Approximation* (MAP). Notice that the same h^* also maximizes the integrand of the evidence of the data y_D

$$p(y_D|x_D, D_0) = \int dh\, p(y_D|x_D, h)p(h|D_0). \tag{73}$$

This is due to the assumption that $p(y|x, h)$ is slowly varying at the stationary point and has not to be included in the saddle point approximation for the predictive density. For (functional) differentiable E_{comb} Eq. (72) yields the stationarity equation,

$$\frac{\delta E_{\rm comb}(h)}{\delta h(x, y)} = 0. \tag{74}$$

The functional $E_{\rm comb}$ including training and prior data (regularization, stabilizer) terms is also known as (regularized) *error functional* for h.

In practice a saddle point approximation may be expected useful if the posterior is peaked enough around a single maximum, or more general, if the posterior is well approximated by a Gaussian centered at the maximum. For asymptotical results one would have to require

$$-\frac{1}{\beta}\sum_i L(y_i|x_i, h), \tag{75}$$

to become β–independent for $\beta \rightarrow \infty$ with some β being the same for the prior and data term. (See [40, 242]). If for example $\frac{1}{n}\sum_i L(y_i|x_i, h)$ converges for large number n of training data the low temperature limit $1/\beta \rightarrow 0$ can be interpreted as large data limit $n \rightarrow \infty$,

$$n E_{\rm comb} = n\left(-\frac{1}{n}\sum_i L(y_i|x_i, h) + \frac{1}{n}E(h|D_0)\right). \tag{76}$$

Notice, however, the factor $1/n$ in front of the prior energy. For Gaussian $p(y|x, h)$ temperature $1/\beta$ corresponds to variance σ^2

$$\frac{1}{\sigma^2}E_{\rm comb} = \frac{1}{\sigma^2}\left(\frac{1}{2}\sum_i (y_i - h(x_i))^2 + \sigma^2 E(h|D_0)\right). \tag{77}$$

For Gaussian prior this would require simultaneous scaling of data and prior variance.

We should also remark that for continuous x,y the stationary solution h^* needs not to be a typical representative of the process $p(h|f)$. A common example is a Gaussian stochastic process $p(h|f)$ with prior energy $E(h|D_0)$ related to some smoothness measure of h expressed by derivatives of $p(y|x,h)$. Then, even if the stationary h^* is smooth, this needs not to be the case for a typical h sampled according to $p(h|f)$. For Brownian motion, for instance, a typical sample path is not even differentiable (but continuous) while the stationary path is smooth. Thus, for continuous variables only expressions like $\int dh\, e^{-\beta E(h)}$ can be given an exact meaning as a Gaussian measure, defined by a given covariance with existing normalization factor, but not the expressions dh and $E(h)$ alone [51, 62, 228, 110, 83, 144].

Interestingly, the stationary h^* yielding maximal posterior $p(h|f)$ is not only useful to obtain an approximation for the predictive density $p(y|x,f)$ but is also the optimal solution a^* for a Bayesian decision problem with log–loss and $a \in A = H$.

Indeed, for a Bayesian decision problem with log–loss (46)

$$\text{argmin}_{a\in H} r(a,h) = h, \tag{78}$$

and analogously,

$$\text{argmin}_{a\in F} r(a,f) = f. \tag{79}$$

This is proved as follows: Jensen's inequality states that

$$\int dy\, p(y) g(q(y)) \ge g(\int dy\, p(y) q(y)), \tag{80}$$

for any convex function g and probability $p(y) \ge 0$ with $\int dy\, p(y) = 1$. Thus, because the logarithm is concave

$$-\int dy\, p(y|x,h) \ln \frac{p(y|x,a)}{p(y|x,h)} \ge -\ln \int dy\, p(y|x,h) \frac{p(y|x,a)}{p(y|x,h)} = 0 \tag{81}$$

$$\Rightarrow -\int dy\, p(y|x,h) \ln p(y|x,a) \ge -\int dy\, p(y|x,h) \ln p(y|x,h), \tag{82}$$

with equality for $a = h$. Hence

$$r(a,h) \;=\; -\int dx \int dy\, p(x) p(y|x,h) \left(b_1(x) \ln p(y|x,a) - b_2(x,y)\right) \tag{83}$$

$$
\begin{aligned}
&= -\int dx\, p(x) b_1(x) \int dy\, p(y|x,h) \ln p(y|x,a) + \text{const.} && (84)\\
&\geq -\int dx\, p(x) b_1(x) \int dy\, p(y|x,h) \ln p(y|x,h) + \text{const.} && (85)\\
&= r(h,h), && (86)
\end{aligned}
$$

with equality for $a = h$. For $a \in F$ replace $h \in H$ by $f \in F$. This proves Eqs. (78) and (79).

2.4 Normalization, non–negativity, and specific priors

Density estimation problems are characterized by their normalization and non–negativity condition for $p(y|x,h)$. Thus, the prior density $p(h|D_0)$ can only be non–zero for such h for which $p(y|x,h)$ is positive and normalized over y for all x. (Similarly, when solving for a distribution function, i.e., the integral of a density, the non–negativity constraint is replaced by monotonicity and the normalization constraint by requiring the distribution function to be 1 on the right boundary.) While the non–negativity constraint is local with respect to x and y, the normalization constraint is nonlocal with respect to y. Thus, implementing a normalization constraint leads to nonlocal and in general non–Gaussian priors.

For classification problems, having discrete y values (classes), the normalization constraint requires simply to sum over the different classes and a Gaussian prior structure with respect to the x–dependency is not altered [236]. For general density estimation problems, however, i.e., for continuous y, the loss of the Gaussian structure with respect to y is more severe, because non–Gaussian functional integrals can in general not be performed analytically. On the other hand, solving the learning problem numerically by discretizing the y and x variables, the normalization term is typically not a severe complication.

To be specific, consider a Maximum A Posteriori Approximation, minimizing

$$
\beta E_{\text{comb}} = -\sum_i L(y_i|x_i,h) + \beta E(h|D_0), \tag{87}
$$

where the likelihood free energy $F(Y_D|x_D,h)$ is included, but not the prior free energy $F(H|D_0)$ which, being h–independent, is irrelevant for minimization with respect to h. The prior energy $\beta E(h|D_0)$ has to implement the non–negativity and normalization conditions

$$Z_X(x,h) = \int dy_i \, p(y_i|x_i,h) = 1, \qquad \forall x_i \in X_i, \forall h \in H \tag{88}$$

$$p(y_i|x_i,h) \ge 0, \qquad \forall y_i \in Y_i, \forall x_i \in X_i, \forall h \in H. \tag{89}$$

It is useful to isolate the normalization condition and non–negativity constraint defining the class of density estimation problems from the rest of the problem specific priors. Introducing the *specific prior information* $\tilde{D}_0$ so that $D_0 = \{\tilde{D}_0, \text{normalized}, \text{positive}\}$, we have

$$p(h|\tilde{D}_0, \text{norm.}, \text{pos.}) = \frac{p(\text{norm.}, \text{pos.}|h) p(h|\tilde{D}_0)}{p(\text{norm.}, \text{pos.}|\tilde{D}_0)}, \tag{90}$$

with deterministic, $\tilde{D}_0$–independent

$$p(\text{norm.}, \text{pos.}|h) = p(\text{norm.}, \text{pos.}|h, \tilde{D}_0) \tag{91}$$

$$= p(\text{norm.}|h) p(\text{pos.}|h) = \delta(Z_X - 1) \prod_{xy} \Theta\Big(p(y|x,h)\Big), \tag{92}$$

and step function Θ. (The density $p(\text{norm.}|h)$ is normalized over all possible normalizations of $p(y|x,h)$, i.e., over all possible values of Z_X, and $p(\text{pos.}|h)$ over all possible sign combinations.) The h–independent denominator $p(\text{norm.}, \text{pos.}|\tilde{D}_0)$ can be skipped for error minimization with respect to h. We define the *specific prior* as

$$p(h|\tilde{D}_0) \propto e^{-E(h|\tilde{D}_0)}. \tag{93}$$

In Eq. (93) the specific prior appears as posterior of a h–generating process determined by the parameters $\tilde{D}_0$. We will call therefore Eq. (93) the *posterior form* of the specific prior. Alternatively, a specific prior can also be in *likelihood form*

$$p(\tilde{D}_0, h|\text{norm.}, \text{pos.}) = p(\tilde{D}_0|h) \, p(h|\text{norm.}, \text{pos.}). \tag{94}$$

As the likelihood $p(\tilde{D}_0|h)$ is conditioned on h this means that the normalization $Z = \int d\tilde{D}_0 \, e^{-E(\tilde{D}_0|h)}$ remains in general h–dependent and must be included when minimizing with respect to h. However, Gaussian specific priors with h–independent covariances have the special property that according to

Eq. (70) likelihood and posterior interpretation coincide. Indeed, representing Gaussian specific prior data $\tilde{D}_0$ by a mean function $t_{\tilde{D}_0}$ and covariance $\mathbf{K}^{-1}$ (analogous to standard training data in the case of Gaussian regression, see also Section 3.5) one finds due to the fact that the normalization of a Gaussian is independent of the mean (for uniform (meta) prior $p(h)$)

$$
\begin{aligned}
p(h|\tilde{D}_0) &= \frac{e^{-\frac{1}{2}(h-t_{\tilde{D}_0},\mathbf{K}(h-t_{\tilde{D}_0}))}}{\int dh\, e^{-\frac{1}{2}(h-t_{\tilde{D}_0},\mathbf{K}(h-t_{\tilde{D}_0}))}} &(95)\\
= p(t_{\tilde{D}_0}|h,\mathbf{K}) &= \frac{e^{-\frac{1}{2}(h-t_{\tilde{D}_0},\mathbf{K}(h-t_{\tilde{D}_0}))}}{\int dt\, e^{-\frac{1}{2}(h-t,\mathbf{K}(h-t))}}. &(96)
\end{aligned}
$$

Thus, for Gaussian $p(t_{\tilde{D}_0}|h,\mathbf{K})$ with h–independent normalization the specific prior energy in likelihood form becomes analogous to Eq. (93)

$$
p(t_{\tilde{D}_0}|h,\mathbf{K}) \propto e^{-E(t_{\tilde{D}_0}|h,\mathbf{K})}, \tag{97}
$$

and specific prior energies can be interpreted both ways.

Similarly, the complete likelihood factorizes

$$
p(\tilde{D}_0, \text{norm.}, \text{pos.}|h) = p(\text{norm.}, \text{pos.}|h)\, p(\tilde{D}_0|h). \tag{98}
$$

According to Eq. (92) non–negativity and normalization conditions are implemented by step and δ–functions. The non–negativity constraint is only active when there are locations with $p(y|x,h) = 0$. In all other cases the gradient has no component pointing into forbidden regions. Due to the combined effect of data, where $p(y|x,h)$ has to be larger than zero by definition, and smoothness terms the non–negativity condition for $p(y|x,h)$ is usually (but not always) fulfilled. Hence, if strict positivity is checked for the final solution, then it is not necessary to include extra non–negativity terms in the error (see Section 3.2.1). For the sake of simplicity we will therefore not include non–negativity terms explicitly in the following. In case a non–negativity constraint has to be included this can be done using Lagrange multipliers, or alternatively, by writing the step functions in $p(\text{pos.}|h) \propto \prod_{x,y} \Theta(p(y|x,h))$

$$
\Theta(x-a) = \int_a^\infty d\xi \int_{-\infty}^\infty d\eta e^{i\eta(\xi-x)}, \tag{99}
$$

and solving the ξ–integral in saddle point approximation (See for example [63, 64, 65]).

Including the normalization condition in the prior $p_0(h|D_0)$ in form of a δ–functional results in a posterior probability

$$p(h|f) \propto e^{\sum_i L_i(y_i|x_i,h) - E(h|\tilde{D}_0) + \tilde{c}(H|\tilde{D}_0)} \prod_{x \in X} \delta \left(\int dy \, e^{L(y|x,h)} - 1 \right) \tag{100}$$

with constant $\tilde{c}(H|\tilde{D}_0) = -\ln \tilde{Z}(h|\tilde{D}_0)$ related to the normalization of the specific prior $e^{-E(h|\tilde{D}_0)}$. Writing the δ–functional in its Fourier representation

$$\delta(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dk \, e^{ikx} = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} dk \, e^{-kx}, \tag{101}$$

i.e.,

$$\delta(\int dy \, e^{L(y|x,h)} - 1) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} d\Lambda_X(x) \, e^{\Lambda_X(x)\left(1 - \int dy \, e^{L(y|x,h)}\right)}, \tag{102}$$

and performing a saddle point approximation with respect to $\Lambda_X(x)$ (which is exact in this case) yields

$$p(h|f) \propto e^{\sum_i L_i(y_i|x_i,h) - E(h|\tilde{D}_0) + \tilde{c}(H|\tilde{D}_0) + \int dx \, \Lambda_X(x)\left(1 - \int dy e^{L(y|x,h)}\right)}. \tag{103}$$

This is equivalent to the Lagrange multiplier approach. Here the stationary $\Lambda_X(x)$ is the Lagrange multiplier vector (or function) to be determined by the normalization conditions for $p(y|x,h) = e^{L(y|x,h)}$. Besides the Lagrange multiplier terms it is numerically sometimes useful to add additional terms to the log–posterior which vanish for normalized $p(y|x,h)$.

2.5 Empirical risk minimization

In the previous sections the error functionals we will try to minimize in the following have been given a Bayesian interpretation in terms of the log–posterior density. There is, however, an alternative justification of error functionals using the Frequentist approach of *empirical risk minimization* [224, 225, 226].

Common to both approaches is the aim to minimize the *expected risk* for action a

$$r(a,f) = \int dx \, dy \, p(x,y|f(D,D_0)) \, l(x,y,a). \tag{104}$$

The expected risk, however, cannot be calculated without knowledge of the true $p(x,y|f)$. In contrast to the Bayesian approach of modeling $p(x,y|f)$

the Frequentist approach approximates the expected risk by the *empirical risk*

$$E(a) = \hat{r}(a, f) = \sum_i l(x_i, y_i, a), \tag{105}$$

i.e., by replacing the unknown true probability by an observable empirical probability. Here it is essential for obtaining asymptotic convergence results to assume that training data are sampled according to the true $p(x, y|f)$ [224, 52, 194, 128, 226]. Notice that in contrast in a Bayesian approach the density $p(x_i)$ for training data D does according to Eq. (16) not enter the formalism because D enters as conditional variable. For a detailed discussion of the relation between quadratic error functionals and Gaussian processes see for example [181, 183, 184, 112, 113, 153, 228, 144].

From that Frequentist point of view one is not restricted to logarithmic data terms as they arise from the posterior–related Bayesian interpretation. However, like in the Bayesian approach, training data terms are not enough to make the minimization problem well defined. Indeed this is a typical inverse problem [224, 115, 226] which can, according to the classical regularization approach [220, 221, 165], be treated by including additional *regularization (stabilizer) terms* in the loss function l. Those regularization terms, which correspond to the prior terms in a Bayesian approach, are thus from the point of view of empirical risk minimization a technical tool to make the minimization problem well defined.

The *empirical generalization error* for a test or validation data set independent from the training data D, on the other hand, is measured using only the data terms of the error functional without regularization terms. In empirical risk minimization this empirical generalization error is used, for example, to determine adaptive (hyper–)parameters of regularization terms. A typical example is a factor multiplying the regularization terms controlling the trade–off between data and regularization terms. Common techniques using the empirical generalization error to determine such parameters are *cross–validation* or *bootstrap* like techniques [166, 6, 230, 216, 217, 81, 39, 228, 54]. From a strict Bayesian point of view those parameters would have to be integrated out after defining an appropriate prior [16, 147, 149, 24].

2.6 Interpretations of Occam's razor

The principle to prefer simple models over complex models and to find an optimal trade–off between fitting data and model complexity is often referred to as Occam's razor (William of Occam, 1285–1349). Regularization terms, penalizing for example non–smooth ("complex") functions, can be seen as an implementation of Occam's razor.

The related phenomena appearing in practical learning is called over–fitting [224, 96, 24]. Indeed, when studying the generalization behavior of trained models on a test set different from the training set, it is often found that there is an optimal model complexity. Complex models can due to their higher flexibility achieve better performance on the training data than simpler models. On a test set independent from the training set, however, they can perform poorer than simpler models.

Notice, however, that the Bayesian interpretation of regularization terms as (*a priori*) information about Nature and the Frequentist interpretation as additional cost terms in the loss function are *not* equivalent. Complexity priors reflects the case where Nature is known to be simple while complexity costs express the wish for simple models without the assumption of a simple Nature. Thus, while the practical procedure of minimizing an error functional with regularization terms appears to be identical for empirical risk minimization and a Bayesian Maximum A Posteriori Approximation, the underlying interpretation for this procedure is different. In particular, because the Theorem in Section 2.3 holds only for log–loss, the case of loss functions differing from log–loss requires from a Bayesian point of view to distinguish explicitly between model states h and actions a. Even in saddle point approximation, this would result in a two step procedure, where in a first step the hypothesis h^*, with maximal posterior probability is determined, while the second step minimizes the risk for action $a \in A$ under that hypothesis h^* [132].

2.7 *A priori* information and *a posteriori* control

Learning is based on data, which includes training data as well as *a priori* data. It is prior knowledge which, besides specifying the space of local hypothesis, enables generalization by providing the necessary link between measured training data and not yet measured or non–training data. The strength of this connection may be quantified by the mutual information of training and non–training data, as we did in Section 2.1.5.

Often, the role of *a priori* information seems to be underestimated. There are theorems, for example, proving that asymptotically learning results become independent of *a priori* information if the number of training data goes to infinity. This, however, is correct only if the space of hypotheses h is already sufficiently restricted and if *a priori* information means knowledge in addition to that restriction.

In particular, let us assume that the number of potential test situations x, is larger than the number of training data one is able to collect. As the number of actual training data has to be finite, this is always the case if x can take an infinite number of values, for example if x is a continuous variable. The following arguments, however, are not restricted to situations were one considers an infinite number of test situations, we just assume that their number is too large to be completely included in the training data.

If there are x values for which no training data are available, then learning for such x must refer to the mutual information of such test data and the available training data. Otherwise, training would be useless for these test situations. This also means, that the generalization to non–training situations can be arbitrarily modified by varying *a priori* information.

To make this point very clear, consider the rather trivial situation of learning a deterministic function $h(x)$ for a x variable which can take only two values x_1 and x_2, from which only one can be measured. Thus, having measured for example $h(x_1) = 5$, then "learning" $h(x_2)$ is not possible without linking it to $h(x_1)$. Such prior knowledge may have the form of a "smoothness" constraint, say $|h(x_1) - h(x_2)| \leq 2$ which would allow a learning algorithm to "generalize" from the training data and obtain $3 \leq h(x_2) \leq 7$. Obviously, arbitrary results can be obtained for $h(x_2)$ by changing the prior knowledge. This exemplifies that generalization can be considered as a mere reformulation of available information, i.e., of training data and prior knowledge. Except for such a rearrangement of knowledge, a learning algorithm does not add any new information to the problem. (For a discussion of the related "no–free-lunch" theorems see [240, 241].)

Being extremely simple, this example nevertheless shows a severe problem. If the result of learning can be arbitrary modified by *a priori* information, then it is critical which prior knowledge is implemented in the learning algorithm. This means, that prior knowledge needs an empirical foundation, just like standard training data have to be measured empirically. Otherwise, the result of learning cannot expected to be of any use.

Indeed, the problem of appropriate *a priori* information is just the old induction problem, i.e., the problem of learning general laws from a finite number of observations, as already been discussed by the ancient Greek philosophers. Clearly, this is not a purely academic problem, but is extremely important for every system which depends on a successful control of its environment. Modern applications of learning algorithms, like speech recognition or image understanding, rely essentially on correct *a priori* information. This holds especially for situations where only few training data are available, for example, because sampling is very costly.

Empirical measurement of *a priori* information, however, seems to be impossible. The reason is that we must link every possible test situation to the training data. We are not able to do this in practice if, as we assumed, the number of potential test situations is larger than the number of measurements one is able to perform.

Take as example again a deterministic learning problem like the one discussed above. Then measuring *a priori* information might for example be done by measuring (e.g., bounds on) all differences $h(x_1) - h(x_i)$. Thus, even if we take the deterministic structure of the problem for granted, the number of such differences is equal to the number of potential non–training situations x_i we included in our model. Thus, measuring *a priori* information does not require fewer measurements than measuring directly all potential non–training data. We are interested in situations where this is impossible.

Going to a probabilistic setting the problem remains the same. For example, even if we assume Gaussian hypotheses with fixed variance, measuring a complete mean function $h(x)$, say for continuous x, is clearly impossible in practice. The same holds thus for a Gaussian process prior on h. Even this very specific prior requires the determination of a covariance and a mean function (see Chapter 3).

As in general empirical measurement of *a priori* information seems to be impossible, one might thus just try to guess some prior. One may think, for example, of some "natural" priors. Indeed, the term "*a priori*" goes back to Kant [111] who assumed certain knowledge to be necessarily be given "*a priori*" without reference to empirical verification. This means that we are either only able to produce correct prior assumptions, for example because incorrect prior assumptions are "unthinkable", or that one must typically be lucky to implement the right *a priori* information. But looking at the huge number of different prior assumptions which are usually possible (or "thinkable"), there seems no reason why one should be lucky. The question thus remains, how can prior assumptions get empirically verified.

Also, one can ask whether there are "natural" priors in practical learning tasks. In Gaussian regression one might maybe consider a "natural" prior to be a Gaussian process with constant mean function and smoothness–

related covariance. This may leave a single regularization parameter to be determined for example by cross–validation. Formally, one can always even use a zero mean function for the prior process by subtracting a base line or reference function. Thus does, however, not solve the problem of finding a correct prior, as now that reference function has to be known to relate the results of learning to empirical measurements. In principle *any* function could be chosen as reference function. Such a reference function would for example enter a smoothness prior. Hence, there is no "natural" constant function and from an abstract point of view no prior is more "natural" than any other.

Formulating a general law refers implicitly (and sometimes explicitly) to a "ceteris paribus" condition, i.e., the constraint that all relevant variables, not explicitly mentioned in the law, are held constant. But again, verifying a "ceteris paribus" condition is part of an empirical measurement of *a priori* information and by no means trivial.

Trying to be cautious and use only weak or "uninformative" priors does also not solve the principal problem. One may hope that such priors (which may be for example an improper constant prior for a one–dimensional real variable) do not introduce a completely wrong bias, so that the result of learning is essentially determined by the training data. But, besides the problem to define what exactly an uninformative prior has to be, such priors are in practice only useful if the set of possible hypothesis is already sufficiently restricted, so "the data can speak for themselves" [69]. Hence, the problem remains to find that priors which impose the necessary restrictions, so that uninformative priors can be used.

Hence, as measuring *a priori* information seems impossible and finding correct *a priori* information by pure luck seems very unlikely, it looks like also successful learning is impossible. It is a simple fact, however, that learning can be successful. That means there must be a way to control *a priori* information empirically.

Indeed, the problem of measuring *a priori* information may be artificial, arising from the introduction of a large number of *potential* test situations and correspondingly a large number of hidden variables h (representing what we call "Nature") which are not all observable.

In practice, the number of *actual* test situations is also always finite, just like the number of training data has to be. This means, that not *all* potential test data but only the actual test data must be linked to the training data. Thus, in practice it is only a finite number of relations which must be under control to allow successful generalization. (See also Vapnik's distinction between induction and transduction problems. [226]: In induction problems one tries to infer a whole function, in transduction problems one is only interested in predictions for a few specific test situations.)

This, however, opens a possibility to control *a priori* information empirically. Because we do not know which test situation will occur, such an empirical control cannot take place at the time of training. This means *a priori* information has to be implemented at the time of measuring the test data. In other words, *a priori* information has to be implemented by the measurement process [132, 135].

Again, a simple example may clarify this point. Consider the prior information, that a function h is bounded, i.e., $a \le h(x) \le b$, $\forall x$. A direct measurement of this prior assumption is practically not possible, as it would require to check every value $h(x)$. An implementation within the measurement process is however trivial. One just has to use a measurement device which is only able to to produce output in the range between a and b. This is a very realistic assumption and valid for all real measurement devices. Values smaller than a and larger than b have to be filtered out or actively projected into that range. In case we nevertheless find a value out of that range we either have to adjust the bounds or we exchange the "malfunctioning" measurement device with a proper one. Note, that this range filter is only needed at the finite number of actual measurements. That means, *a priori* information can be implemented by *a posteriori* control at the time of testing.

A realistic measurement device does not only produce bounded output but shows also always *input noise* or *input averaging*. A device with input noise has noise in the x variable. That means if one intends to measure at x_i the device measures instead at $x_i + \Delta$ with Δ being a random variable. A typical example is translational noise, with Δ being a, possibly multidimensional, Gaussian random variable with mean zero. Similarly, a device with input averaging returns a weighted average of results for different x values instead of a sharp result. Bounded devices with translational input noise, for example, will always measure smooth functions [129, 23, 132]. (See Fig. 4.) This may be an explanation for the success of smoothness priors.

The last example shows, that to obtain adequate *a priori* information it can be helpful in practice to analyze the measurement process for which learning is intended. The term "measurement process" does here not only refer to a specific device, e.g., a box on the table, but to the collection of all

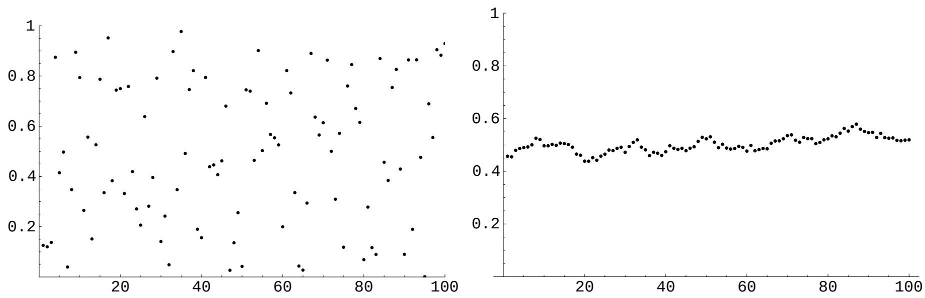

Figure 4: The l.h.s. shows a bounded random function which does not allow generalization from training to non–training data. Using a measurement device with input averaging (r.h.s.) or input noise the function becomes learnable.

processes which lead to a measurement result.

We may remark that measuring a measurement process is as difficult or impossible as a direct measurement of *a priori* information. What has to be ensured is the validity of the necessary restrictions during a finite number of actual measurements. This is nothing else than the implementation of a probabilistic rule producing y given the test situation and the training data. In other words, what has to be implemented is the predictive density $p(y|x, D)$. This predictive density indeed only depends on the actual test situation and the finite number of training data. (Still, the probability density for a real y cannot strictly be empirically verified or controlled. We may take it here, for example, as an approximate statement about frequencies.) This shows the tautological character of learning, where measuring *a priori* information means controlling directly the corresponding predictive density.

The *a posteriori* interpretation of *a priori* information can be related to a constructivistic point of view. The main idea of constructivism can be characterized by a sentence of Vico (1710): *Verum ipsum factum* — the truth is the same as the made [227]. (For an introduction to constructivism see [232] and references therein, for constructive mathematics see [25].)

3 Gaussian prior factors

3.1 Gaussian prior factor for log–probabilities

3.1.1 Lagrange multipliers: Error functional E_L

In this chapter we look at density estimation problems with Gaussian prior factors. We begin with a discussion of functional priors which are Gaussian in probabilities or in log–probabilities, and continue with general Gaussian prior factors. Two sections are devoted to the discussion of covariances and means of Gaussian prior factors, as their adequate choice is essential for practical applications. After exploring some relations of Bayesian field theory and empirical risk minimization, the last three sections introduce the specific likelihood models of regression, classification, inverse quantum theory.

We begin with a discussion of Gaussian prior factors in L. As Gaussian prior factors correspond to quadratic error (or energy) terms, consider an error functional with a quadratic regularizer in L

$$(L, \mathbf{K}L) = ||L||^2_{\mathbf{K}} = \frac{1}{2}\int dx\, dy\, dx'dy' L(x,y)\mathbf{K}(x,y;x',y')L(x',y'), \tag{106}$$

writing for the sake of simplicity from now on $L(x,y)$ for the log–probability $L(y|x,h) = \ln p(y|x,h)$. The operator $\mathbf{K}$ is assumed symmetric and positive semi–definite and positive definite on some subspace. (We will understand positive semi–definite to include symmetry in the following.) For positive (semi–)definite $\mathbf{K}$ the scalar product defines a (semi–)norm by

$$||L||_{\mathbf{K}} = \sqrt{(L, \mathbf{K}L)}, \tag{107}$$

and a corresponding distance by $||L - L'||_{\mathbf{K}}$. The quadratic error term (106) corresponds to a Gaussian factor of the prior density which have been called the specific prior $p(h|\tilde{D}_0) = p(L|\tilde{D}_0)$ for L. In particular, we will consider here the posterior density

$$p(h|f) \propto e^{\sum_i L_i(x_i,y_i) - \frac{1}{2}\int dx dy dx' dy' L(x,y)\mathbf{K}(x,y;x',y')L(x',y') + \int dx\, \Lambda_X(x)\left(1 - \int dy\, e^{L(x,y)}\right) + \tilde{c}}, \tag{108}$$

where prefactors like β are understood to be included in $\mathbf{K}$. The constant $\tilde{c}$ referring to the specific prior is determined by the determinant of $\mathbf{K}$ according to Eq. (70). Notice however that not only the likelihood $\sum_i L_i$ but also the complete prior is usually *not* Gaussian due to the presence of the normalization conditions. (An exception is Gaussian regression, see Section 3.7.) The posterior (108) corresponds to an error functional

$$E_L = \beta E_{\rm comb} = -(L, N) + \frac{1}{2}(L, \mathbf{K}L) + (e^L - \delta(y), \Lambda_X), \tag{109}$$

with *likelihood vector (or function)*

$$L(x, y) = L(y|x, h), \tag{110}$$

data vector (function)

$$N(x, y) = \sum_i^n \delta(x - x_i)\delta(y - y_i), \tag{111}$$

Lagrange multiplier vector (function)

$$\Lambda_X(x, y) = \Lambda_X(x), \tag{112}$$

probability vector (function)

$$e^L(x, y) = e^{L(x,y)} = P(x, y) = p(y|x, h), \tag{113}$$

and

$$\delta(y)(x, y) = \delta(y). \tag{114}$$

According to Eq. (111) $N/n = P_{\rm emp}$ is an *empirical density function* for the joint probability $p(x, y|h)$.

We end this subsection by defining some notations. Functions of vectors (functions) and matrices (operators), different from multiplication, will be understood element-wise like for example $(e^L)(x, y) = e^{L(x,y)}$. Only multiplication of matrices (operators) will be interpreted as matrix product. Element-wise multiplication has then to be written with the help of diagonal matrices. For that purpose we introduce diagonal matrices made from vectors (functions) and denoted by the corresponding bold letters. For instance,

$$\begin{aligned}
\mathbf{I}(x, y; x', y') &= \delta(x - x')\delta(y - y'), && (115)\\
\mathbf{L}(x, y; x', y') &= \delta(x - x')\delta(y - y')L(x, y), && (116)\\
\mathbf{P}(x, y; x', y') &= \mathbf{e^L}(x, y; x', y') && (117)\\
&= \delta(x - x')\delta(y - y')P(x, y), && (118)\\
\mathbf{N}(x, y; x', y') &= \delta(x - x')\delta(y - y')\,N(x, y), && (119)\\
\mathbf{\Lambda}_X(x, y; x', y') &= \delta(x - x')\delta(y - y')\Lambda_X(x), && (120)
\end{aligned}$$

or

$$L = \mathbf{L}I, \quad P = \mathbf{P}I, \quad e^L = \mathbf{e^L}I, \quad N = \mathbf{N}I, \quad \Lambda_X = \mathbf{\Lambda}_X I, \tag{121}$$

where

$$I(x,y) = 1. \tag{122}$$

Being diagonal all these matrices commute with each other. *Element-wise multiplication* can now be expressed as

$$\begin{aligned}(\mathbf{KL})(x',y',x,y) &= \int dx''dy''\mathbf{K}(x',y',x'',y'')\mathbf{L}(x'',y'',x,y)\\ &= \int dx''dy''\mathbf{K}(x',y',x'',y'')L(x,y)\delta(x-x'')\delta(y-y'')\\ &= \mathbf{K}(x',y',x,y)L(x,y).\end{aligned} \tag{123}$$

In general this is not equal to $L(x',y')\mathbf{K}(x',y',x,y)$. In contrast, the *matrix product* $\mathbf{K}L$ with vector L

$$(\mathbf{K}L)(x',y') = \int dx\,dy\,\mathbf{K}(x',y',x,y)L(x,y), \tag{124}$$

does not depend on x, y anymore, while the *tensor product* or outer product,

$$(\mathbf{K}\otimes L)(x'',y'',x,y,x',y') = \mathbf{K}(x'',y'',x',y')L(x,y), \tag{125}$$

depends on additional x'', y''.

Taking the variational derivative of (108) with respect to $L(x,y)$ using

$$\frac{\delta L(x',y')}{\delta L(x,y)} = \delta(x-x')\delta(y-y') \tag{126}$$

and setting the gradient equal to zero yields the stationarity equation

$$0 = N - \mathbf{K}L - \mathbf{e^L}\Lambda_X. \tag{127}$$

Alternatively, we can write $\mathbf{e^L}\Lambda_X = \mathbf{\Lambda}_X e^L = \mathbf{P}\Lambda_X$.

The Lagrange multiplier function Λ_X is determined by the normalization condition

$$Z_X(x) = \int dy\, e^{L(x,y)} = 1, \quad \forall x \in X, \tag{128}$$

which can also be written

$$Z_X = \mathbf{I}_X P = \mathbf{I}_X e^L = I \quad \text{or} \quad \mathbf{Z}_X = \mathbf{I}, \tag{129}$$

in terms of normalization vector,

$$Z_X(x, y) = Z_X(x), \tag{130}$$

normalization matrix,

$$\mathbf{Z_X}(x, y; x', y') = \delta(x - x')\delta(y - y')\, Z_X(x), \tag{131}$$

and identity on X,

$$\mathbf{I}_X(x, y; x', y') = \delta(x - x'). \tag{132}$$

Multiplication of a vector with $\mathbf{I}_X$ corresponds to y–integration. Being a non–diagonal matrix $\mathbf{I}_X$ does in general not commute with diagonal matrices like $\mathbf{L}$ or $\mathbf{P}$. Note also that despite $\mathbf{I}_X e^L = \mathbf{I}_X \mathbf{e^L} I = \mathbf{I} I = I$ in general $\mathbf{I}_X \mathbf{P} = \mathbf{I}_X \mathbf{e^L} \ne \mathbf{I} = \mathbf{Z}_X$. According to the fact that $\mathbf{I}_X$ and $\mathbf{\Lambda}_X$ commute, i.e.,

$$\mathbf{I}_X \mathbf{\Lambda}_X = \mathbf{\Lambda}_X \mathbf{I}_X \Leftrightarrow [\mathbf{\Lambda}_X, \mathbf{I}_X] = \mathbf{\Lambda}_X \mathbf{I}_X - \mathbf{I}_X \mathbf{\Lambda}_X = 0, \tag{133}$$

(introducing the commutator $[A, B] = AB - BA$), and that the same holds for the diagonal matrices

$$[\mathbf{\Lambda}_X, \mathbf{e^L}] = [\mathbf{\Lambda}_X, \mathbf{P}] = 0, \tag{134}$$

it follows from the normalization condition $\mathbf{I}_X P = I$ that

$$\mathbf{I}_X \mathbf{P} \Lambda_X = \mathbf{I}_X \mathbf{\Lambda_X} P = \mathbf{\Lambda_X} \mathbf{I}_X P = \mathbf{\Lambda_X} I = \Lambda_X, \tag{135}$$

i.e.,

$$0 = (\mathbf{I} - \mathbf{I}_X \mathbf{e^L})\Lambda_X = (\mathbf{I} - \mathbf{I}_X \mathbf{P})\Lambda_X. \tag{136}$$

For $\Lambda_X(x) \ne 0$ Eqs.(135,136) are equivalent to the normalization (128). If there exist directions at the stationary point L^* in which the normalization of P changes, i.e., the normalization constraint is active, a $\Lambda_X(x) \ne 0$ restricts the gradient to the normalized subspace (Kuhn–Tucker conditions [57, 19, 99, 193]). This will clearly be the case for the unrestricted variations of $p(y, x)$ which we are considering here. Combining $\Lambda_X = \mathbf{I}_X \mathbf{P} \Lambda_X$ for $\Lambda_X(x) \ne 0$ with the stationarity equation (127) the Lagrange multiplier function is obtained

$$\Lambda_X = \mathbf{I}_X\, (N - \mathbf{K} L) = N_X - (\mathbf{I}_X \mathbf{K} L). \tag{137}$$

Here we introduced the vector

$$N_X = \mathbf{I}_X N, \tag{138}$$

with components

$$N_X(x,y) = N_X(x) = \sum_i \delta(x - x_i) = n_x, \tag{139}$$

giving the number of data available for x. Thus, Eq. (137) reads in components

$$\Lambda_X(x) = \sum_i \delta(x - x_i) - \int dy''\, dx' dy'\, \mathbf{K}(x, y''; x', y') L(x', y'). \tag{140}$$

We remark, that for a Laplacian $\mathbf{K}$ and appropriate boundary conditions the integral term in equation (140) vanishes.

Inserting now the equation (140) for Λ_X into the stationarity equation (127) yields

$$0 = N - \mathbf{K}L - \mathbf{e^L}(N_X - \mathbf{I}_X \mathbf{K} L) = \left(\mathbf{I} - \mathbf{e^L} \mathbf{I}_X\right) (N - \mathbf{K}L)\,. \tag{141}$$

Eq. (141) possesses, besides normalized solutions we are looking for, also possibly unnormalized solutions fulfilling $N = \mathbf{K}L$ for which Eq. (137) yields $\Lambda_X = 0$. That happens because we used Eq. (135) which is also fulfilled for $\Lambda_X(x) = 0$. Such a $\Lambda_X(x) = 0$ does not play the role of a Lagrange multiplier. For parameterizations of L where the normalization constraint is not necessarily active at a stationary point $\Lambda_X(x) = 0$ can be possible for a normalized solution L^*. In that case normalization has to be checked.

It is instructive to define

$$T_L = N - \Lambda_X e^L, \tag{142}$$

so the stationarity equation (127) acquires the form

$$\mathbf{K}L = T_L, \tag{143}$$

which reads in components

$$\int dx' dy'\, \mathbf{K}(x, y; x', y') L(x', y') = \sum_i \delta(x - x_i)\delta(y - y_i) - \mathbf{\Lambda}_X(x)\, e^{L(x,y)}, \tag{144}$$

which is in general a non–linear equation because T_L depends on L. For existing (and not too ill–conditioned) $\mathbf{K}^{-1}$ the form (143) suggest however an iterative solution of the stationarity equation according to

$$L^{i+1} = \mathbf{K}^{-1} T_L(L^i), \tag{145}$$

for discretized L, starting from an initial guess L^0. Here the Lagrange multiplier Λ_X has to be adapted so it fulfills condition (137) at the end of iteration. Iteration procedures will be discussed in detail in Section 7.

3.1.2 Normalization by parameterization: Error functional E_g

Referring to the discussion in Section 2.3 we show that Eq. (141) can alternatively be obtained by ensuring normalization, instead of using Lagrange multipliers, explicitly by the parameterization

$$L(x,y) = g(x,y) - \ln \int dy'\, e^{g(x,y')}, \quad L = g - \ln Z_X, \tag{146}$$

and considering the functional

$$E_g = -\Big(N,\, g - \ln Z_X\,\Big) + \frac{1}{2}\Big(\,g - \ln Z_X\,,\, \mathbf{K}\,(g - \ln Z_X)\,\Big). \tag{147}$$

The stationary equation for $g(x,y)$ obtained by setting the functional derivative $\delta E_g/\delta g$ to zero yields again Eq. (141). We check this, using

$$\frac{\delta \ln Z_X(x')}{\delta g(x,y)} = \delta(x-x')e^{L(x,y)}, \quad \frac{\delta \ln Z_X}{\delta g} = \mathbf{I}_X \mathbf{e^L} = \Big(\mathbf{e^L}\mathbf{I}_X\Big)^T, \tag{148}$$

and

$$\frac{\delta L(x',y')}{\delta g(x,y)} = \delta(x-x')\delta(y-y') - \delta(x-x')e^{L(x,y)}, \quad \frac{\delta L}{\delta g} = \mathbf{I} - \mathbf{I}_X\mathbf{e^L}, \tag{149}$$

where $\frac{\delta L}{\delta g}$ denotes a matrix, and the superscript ^{T} the transpose of a matrix. We also note that despite $\mathbf{I}_X = \mathbf{I}_X^T$

$$\mathbf{I}_X\mathbf{e^L} \ne \mathbf{e^L}\mathbf{I}_X = (\mathbf{I}_X\mathbf{e^L})^T, \tag{150}$$

is not symmetric because $\mathbf{e^L}$ depends on y and does not commute with the non–diagonal $\mathbf{I}_X$. Hence, we obtain the stationarity equation of functional E_g written in terms of $L(g)$ again Eq. (141)

$$0 = -\left(\frac{\delta L}{\delta g}\right)^T \frac{\delta E_g}{\delta L} = G_L - \mathbf{e^L}\Lambda_X = \Big(\mathbf{I} - \mathbf{e^L}\mathbf{I}_X\Big)\,(N - \mathbf{K}L)\,. \tag{151}$$

Here $G_L = N - \mathbf{K}L = -\delta E_g/\delta L$ is the L–gradient of $-E_g$. Referring to the discussion following Eq. (141) we note, however, that solving for g instead for L no unnormalized solutions fulfilling $N = \mathbf{K}L$ are possible.

In case $\ln Z_X$ is in the zero space of $\mathbf{K}$ the functional E_g corresponds to a Gaussian prior in g alone. Alternatively, we may also directly consider a Gaussian prior in g

$$\tilde{E}_g = -\Big(N,\, g - \ln Z_X\Big) + \frac{1}{2}\Big(\,g\,,\, \mathbf{K}\, g\,\Big), \tag{152}$$

with stationarity equation

$$0 = N - \mathbf{K}g - \mathbf{e}^{\mathbf{L}} N_X. \tag{153}$$

Notice, that expressing the density estimation problem in terms of g, nonlocal normalization terms have not disappeared but are part of the likelihood term. As it is typical for density estimation problems, the solution g can be calculated in X–data space, i.e., in the space defined by the x_i of the training data. This still allows to use a Gaussian prior structure with respect to the x–dependency which is especially useful for classification problems [236].

3.1.3 The Hessians $\mathbf{H}_L$, $\mathbf{H}_g$

The Hessian $\mathbf{H}_L$ of $-E_L$ is defined as the matrix or operator of second derivatives

$$\mathbf{H}_L(L)(x,y;x'y') = \left.\frac{\delta^2(-E_L)}{\delta L(x,y)\delta L(x',y')}\right|_L. \tag{154}$$

For functional (109) and fixed Λ_X we find the Hessian by taking the derivative of the gradient in (127) with respect to L again. This gives

$$\mathbf{H}_L(L)(x,y;x'y') = -\mathbf{K}(x,y;x'y') - \delta(x-x')\delta(y-y')\Lambda_X(x)e^{L(x,y)} \tag{155}$$

or

$$\mathbf{H}_L = -\mathbf{K} - \mathbf{\Lambda}_X \mathbf{e}^{\mathbf{L}}. \tag{156}$$

The addition of the diagonal matrix $\mathbf{\Lambda}_X \mathbf{e}^{\mathbf{L}} = \mathbf{e}^{\mathbf{L}}\mathbf{\Lambda}_X$ can result in a negative definite $\mathbf{H}$ even if $\mathbf{K}$ has zero modes, like for a differential operator $\mathbf{K}$ with periodic boundary conditions. Note, however, that $\mathbf{\Lambda}_X \mathbf{e}^{\mathbf{L}}$ is diagonal and therefore symmetric, but not necessarily positive definite, because $\Lambda_X(x)$ can be negative for some x. Depending on the sign of $\Lambda_X(x)$ the normalization condition $Z_X(x) = 1$ for that x can be replaced by the inequality $Z_X(x) \le 1$ or $Z_X(x) \ge 1$. Including the L–dependence of Λ_X and with

$$\frac{\delta e^{L(x',y')}}{\delta g(x,y)} = \delta(x-x')\delta(y-y')e^{L(x,y)} - \delta(x-x')e^{L(x,y)}e^{L(x',y')}, \tag{157}$$

i.e.,

$$\frac{\delta e^L}{\delta g} = \left(\mathbf{I} - \mathbf{e^L}\,\mathbf{I}_X\right)\mathbf{e^L} = \mathbf{e^L} - \mathbf{e^L}\,\mathbf{I}_X\mathbf{e^L}, \tag{158}$$

or

$$\frac{\delta P}{\delta g} = \mathbf{P} - \mathbf{P}\,\mathbf{I}_X\mathbf{P}, \tag{159}$$

we find, written in terms of L,

$$\mathbf{H}_g(L)(x,y;x',y') = \left.\frac{\delta^2(-E_g)}{\delta g(x,y)\delta g(x',y')}\right|_L$$

$$= \int dx''dy'' \left.\left(\frac{\delta^2(-E_g)}{\delta L(x,y)\delta L(x'',y'')}\frac{\delta L(x'',y'')}{\delta g(x',y')} + \frac{\delta(-E_g)}{\delta L(x'',y'')}\frac{\delta^2 L(x'',y'')}{\delta g(x,y)\delta g(x',y')}\right)\right|_L$$

$$\begin{aligned}
= \quad & -\mathbf{K}(x,y;x',y') - e^{L(x',y')}e^{L(x,y)}\int dy''dy'''\mathbf{K}(x',y'';x,y''') \\
& + e^{L(x',y')}\int dy''\mathbf{K}(x',y'';x,y) + e^{L(x,y)}\int dy''\mathbf{K}(x',y';x,y'') \\
& - \delta(x-x')\delta(y-y')e^{L(x,y)}\left(N_X(x) - \int dy''(\mathbf{K}L)(x,y'')\right) \\
& + \delta(x-x')e^{L(x,y)}e^{L(x',y')}\left(N_X(x) - \int dy''(\mathbf{K}L)(x,y'')\right).
\end{aligned} \tag{160}$$

The last term, diagonal in X, has dyadic structure in Y, and therefore for fixed x at most one non–zero eigenvalue. In matrix notation the Hessian becomes

$$\begin{aligned}
\mathbf{H}_g &= -\left(\mathbf{I} - \mathbf{e^L}\mathbf{I}_X\right)\mathbf{K}\left(\mathbf{I} - \mathbf{I}_X\mathbf{e^L}\right) - \left(\mathbf{I} - \mathbf{e^L}\mathbf{I}_X\right)\mathbf{\Lambda}_X\mathbf{e^L} \\
&= -\left(\mathbf{I} - \mathbf{P}\mathbf{I}_X\right)\left[\mathbf{K}\left(\mathbf{I} - \mathbf{I}_X\mathbf{P}\right) + \mathbf{\Lambda}_X\mathbf{P}\right],
\end{aligned} \tag{161}$$

the second line written in terms of the probability matrix. The expression is symmetric under $x \leftrightarrow x'$,$y \leftrightarrow y'$, as it must be for a Hessian and as can be verified using the symmetry of $\mathbf{K} = \mathbf{K}^T$ and the fact that $\mathbf{\Lambda}_X$ and $\mathbf{I}_X$ commute, i.e., $[\mathbf{\Lambda}_X, \mathbf{I}_X] = 0$. Because functional E_g is invariant under a shift transformation, $g(x,y) \rightarrow g'(x,y) + c(x)$, the Hessian has a space of zero modes with the dimension of X. Indeed, any y–independent function (which can have finite L^1–norm only in finite Y–spaces) is a left eigenvector of $\left(\mathbf{I} - \mathbf{e^L}\mathbf{I}_X\right)$ with eigenvalue zero. Thus, where necessary, the pseudo inverse of $\mathbf{H}$ have to be used instead of the inverse. Alternatively, additional constraints on g can be added which remove zero modes, like for example boundary conditions.

3.2 Gaussian prior factor for probabilities

3.2.1 Lagrange multipliers: Error functional E_P

We write $P(x,y) = p(y|x,h)$ for the probability of y conditioned on x and h. We consider now a regularizing term which is quadratic in P instead of L. This corresponds to a factor within the posterior probability (the specific prior) which is Gaussian with respect to P.

$$p(h|f) \propto e^{\sum_i \ln P_i(x_i,y_i) - \frac{1}{2}\int dx dy dx' dy' P(x,y) \mathbf{K}(x,y;x',y') P(x',y') + \int dx\, \Lambda_X(x)\left(1-\int dy\, P(x,y)\right) + \tilde{c}}, \tag{162}$$

or written in terms of $L = \ln P$ for comparison,

$$p(h|f) \propto e^{\sum_i L_i(x_i,y_i) - \frac{1}{2}\int dx dy dx' dy' e^{L(x,y)} \mathbf{K}(x,y;x',y') e^{L(x',y')} + \int dx\, \Lambda_X(x)\left(1-\int dy\, e^{L(x,y)}\right) + \tilde{c}}. \tag{163}$$

Hence, the error functional is

$$E_P = \beta E_{\rm comb} = -(\ln P, N) + \frac{1}{2}(P, \mathbf{K}\, P) + (\, P - \delta(y)\, , \Lambda_X). \tag{164}$$

In particular, the choice $\mathbf{K} = \frac{\lambda}{2}\mathbf{I}$, i.e.,

$$\frac{\lambda}{2}(P,\ P) = \frac{\lambda}{2}||P||^2, \tag{165}$$

can be interpreted as a smoothness prior with respect to the distribution function of P (see Section 3.3).

In functional (164) we have only implemented the normalization condition for P by a Lagrange multiplier and not the non–negativity constraint. This is sufficient if $P(x,y) > 0$ (i.e., $P(x,y)$ not equal zero) at the stationary point because then $P(x,y) > 0$ holds also in some neighborhood and there are no components of the gradient pointing into regions with negative probabilities. In that case the non–negativity constraint is not active at the stationarity point. As probabilities have to be positive at data points, smoothness constraints result for example typically in positive probabilities everywhere where not set to zero explicitly by boundary conditions. If, however, the stationary point has locations with $P(x,y) = 0$ at non–boundary points, then the component of the gradient pointing in the region with negative probabilities has to be projected out by introducing Lagrange parameters for each $P(x,y)$. This may happen, for example, if the regularizer rewards oscillatory behavior.

The stationarity equation for E_P is

$$0 = \mathbf{P}^{-1} N - \mathbf{K} P - \Lambda_X, \tag{166}$$

with the diagonal matrix $\mathbf{P}(x', y'; x, y) = \delta(x - x')\delta(y - y')P(x, y)$, or multiplied by $\mathbf{P}$

$$0 = N - \mathbf{P}\mathbf{K} P - \mathbf{P}\Lambda_X. \tag{167}$$

Probabilities $P(x, y)$ are unequal zero at observed data points (x_i, y_i) so $\mathbf{P}^{-1} N$ is well defined.

Combining the normalization condition Eq. (135) for $\Lambda_X(x) \neq 0$ with Eq. (166) or (167) the Lagrange multiplier function Λ_X is found as

$$\Lambda_X = \mathbf{I}_X \left(N - \mathbf{P}\mathbf{K} P \right) = N_X - \mathbf{I}_X \mathbf{P}\mathbf{K} P, \tag{168}$$

where

$$\mathbf{I}_X \mathbf{P}\mathbf{K} P(x, y) = \int dy' dx'' dy''\, P(x, y')\mathbf{K}(x, y'; x'', y'')P(x'', y'').$$

Eliminating Λ_X in Eq. (166) by using Eq. (168) gives finally

$$0 = (\mathbf{I} - \mathbf{I}_X \mathbf{P})(\mathbf{P}^{-1} N - \mathbf{K} P), \tag{169}$$

or for Eq. (167)

$$0 = (\mathbf{I} - \mathbf{P}\mathbf{I}_X)(N - \mathbf{P}\mathbf{K} P). \tag{170}$$

For similar reasons as has been discussed for Eq. (141) unnormalized solutions fulfilling $N - \mathbf{P}\mathbf{K} P$ are possible. Defining

$$T_P = \mathbf{P}^{-1} N - \Lambda_X = \mathbf{P}^{-1} N - N_X - \mathbf{I}_X \mathbf{P}\mathbf{K} P, \tag{171}$$

the stationarity equation can be written analogously to Eq. (143) as

$$\mathbf{K} P = T_P, \tag{172}$$

with $T_P = T_P(P)$, suggesting for existing $\mathbf{K}^{-1}$ an iteration

$$P^{i+1} = \mathbf{K}^{-1} T_P(P^i), \tag{173}$$

starting from some initial guess P^0.

3.2.2 Normalization by parameterization: Error functional E_z

Again, normalization can also be ensured by parameterization of P and solving for unnormalized probabilities z, i.e.,

$$P(x,y) = \frac{z(x,y)}{\int dy\, z(x,y)}, \quad P = \frac{z}{Z_X}. \tag{174}$$

The corresponding functional reads

$$E_z = -\left(N, \ln \frac{z}{Z_X}\right) + \frac{1}{2}\left(\frac{z}{Z_X}, \mathbf{K}\,\frac{z}{Z_X}\right). \tag{175}$$

We have

$$\frac{\delta z}{\delta z} = \mathbf{I}, \quad \frac{\delta Z_X}{\delta z} = \mathbf{I}_X, \quad \frac{\delta \ln z}{\delta z} = \mathbf{z}^{-1} = (\mathbf{Z}_X\mathbf{P})^{-1}, \quad \frac{\delta \ln Z_X}{\delta z} = \mathbf{Z}_X^{-1}\,\mathbf{I}_X, \tag{176}$$

with diagonal matrix $\mathbf{z}$ built analogous to $\mathbf{P}$ and $\mathbf{Z}_X$, and

$$\frac{\delta P}{\delta z} = \frac{\delta(z/Z_X)}{\delta z} = \mathbf{Z}_X^{-1}\left(\mathbf{I} - \mathbf{P}\mathbf{I}_X\right), \quad \frac{\delta \ln P}{\delta z} = \mathbf{Z}_X^{-1}\left(\mathbf{P}^{-1} - \mathbf{I}_X\right), \tag{177}$$

$$\frac{\delta Z_X^{-1}}{\delta z} = -\mathbf{Z}_X^{-2}\mathbf{I}_X, \quad \frac{\delta P^{-1}}{\delta z} = -\mathbf{P}^{-2}\,\mathbf{Z}_X^{-1}\left(\mathbf{I} - \mathbf{P}\mathbf{I}_X\right). \tag{178}$$

The diagonal matrices $[\mathbf{Z}_X, \mathbf{P}] = 0$ commute, as well as $[\mathbf{Z}_X, \mathbf{I}_X] = 0$, but $[\mathbf{P}, \mathbf{I}_X] \neq 0$. Setting the gradient to zero and using

$$\left(\mathbf{I} - \mathbf{P}\mathbf{I}_X\right)^T = \left(\mathbf{I} - \mathbf{I}_X\mathbf{P}\right), \tag{179}$$

we find

$$\begin{aligned} 0 &= -\left(\frac{\delta P}{\delta z}\right)^T \frac{\delta E_z}{\delta P} \\ &= \mathbf{Z}_X^{-1}\left[\left(\mathbf{P}^{-1} - \mathbf{I}_X\right) N - \left(\mathbf{I} - \mathbf{I}_X\mathbf{P}\right)\mathbf{K}P\right] \\ &= \mathbf{Z}_X^{-1}\left(\mathbf{I} - \mathbf{I}_X\mathbf{P}\right)\left(\mathbf{P}^{-1}N - \mathbf{K}P\right) \\ &= \mathbf{Z}_X^{-1}\left(\mathbf{I} - \mathbf{I}_X\mathbf{P}\right) G_P = \mathbf{Z}_X^{-1}\left(G_P - \Lambda_X\right) = \left(\mathbf{G}_P - \mathbf{\Lambda}_X\right) Z_X^{-1}, \end{aligned} \tag{180}$$

with P–gradient $G_P = \mathbf{P}^{-1}N - \mathbf{K}P = -\delta E_z/\delta P$ of $-E_z$ and $\mathbf{G}_P$ the corresponding diagonal matrix. Multiplied by $\mathbf{Z}_X$ this gives the stationarity equation (172).

3.2.3 The Hessians $\mathbf{H}_P$, $\mathbf{H}_z$

We now calculate the Hessian of the functional $-E_P$. For fixed Λ_X one finds the Hessian by differentiating again the gradient (166) of $-E_P$

$$\mathbf{H}_P(P)(x,y;x'y') = -\mathbf{K}(x'y';x,y) - \delta(x-x')\delta(y-y')\sum_i \frac{\delta(x-x_i)\delta(y-y_i)}{P^2(x,y)}, \tag{181}$$

i.e.,

$$\mathbf{H}_P = -\mathbf{K} - \mathbf{P}^{-2}\mathbf{N}. \tag{182}$$

Here the diagonal matrix $\mathbf{P}^{-2}\mathbf{N}$ is non–zero only at data points.

Including the dependence of Λ_X on P one obtains for the Hessian of $-E_z$ in (175) by calculating the derivative of the gradient in (180)

$$\begin{aligned}
\mathbf{H}_z(x,y;x',y\prime) = -\frac{1}{Z_X(x)}\Big[\mathbf{K}(x,y;x',y\prime)\\
-\int dy''\Big(p(x,y'')\mathbf{K}(x,y'';x',y') + \mathbf{K}(x,y;x',y'')p(x',y'')\Big)\\
+\int dy''dy'''p(x,y'')\mathbf{K}(x,y'';x',y''')p(x',y''')\\
+\delta(x-x')\delta(y-y')\sum_i \frac{\delta(x-x_i)\delta(y-y_i)}{p^2(x,y)} - \delta(x-x')\sum_i \delta(x-x_i)\\
-\delta(x-x')\int dx''dy''\Big(\mathbf{K}(x,y;x'',y'')p(x'',y'') + p(x'',y'')\mathbf{K}(x'',y'';x',y')\Big)\\
+2\,\delta(x-x')\int dy''dx'''dy'''p(x,y'')\mathbf{K}(x,y'';x''',y''')p(x''',y''')\Big]\frac{1}{Z_X(x')},
\end{aligned} \tag{183}$$

i.e.,

$$\begin{aligned}
\mathbf{H}_z &= \mathbf{Z}_X^{-1}\left(\mathbf{I}-\mathbf{I}_X\mathbf{P}\right)\Big(-\mathbf{K}-\mathbf{P}^{-2}\mathbf{N}\Big)\left(\mathbf{I}-\mathbf{P}\mathbf{I}_X\right)\mathbf{Z}_X^{-1}\\
&\quad -\mathbf{Z}_X^{-1}\left(\mathbf{I}_X\left(\mathbf{G}_P-\mathbf{\Lambda}_X\right)+\left(\mathbf{G}_P-\mathbf{\Lambda}_X\right)\mathbf{I}_X\right)\mathbf{Z}_X^{-1}, &(184)\\
&= -\mathbf{Z}_X^{-1}\Big[\left(\mathbf{I}-\mathbf{I}_X\mathbf{P}\right)\mathbf{K}\left(\mathbf{I}-\mathbf{P}\mathbf{I}_X\right)+\mathbf{P}^{-2}\mathbf{N}\\
&\quad -\mathbf{I}_X\mathbf{P}^{-1}\mathbf{N}-\mathbf{N}\mathbf{P}^{-1}\mathbf{I}_X+\mathbf{I}_X\mathbf{N}\mathbf{I}_X\\
&\quad +\mathbf{I}_X\mathbf{G}_P+\mathbf{G}_P\mathbf{I}_X-2\,\mathbf{I}_X\mathbf{\Lambda}_X\Big]\mathbf{Z}_X^{-1}. &(185)
\end{aligned}$$

Here we used $[\mathbf{\Lambda}_X,\mathbf{I}_X]=0$. It follows from the normalization $\int dy\,P(x,y) = 1$ that any y–independent function is right eigenvector of $(\mathbf{I}-\mathbf{I}_X\mathbf{P})$ with zero eigenvalue. Because $\Lambda_X = \mathbf{I}_X \mathbf{P} G_P$ this factor or its transpose is also contained in the second line of Eq. (184), which means that $\mathbf{H}_z$ has a zero mode. Indeed, functional E_z is invariant under multiplication of z with a y–independent factor. The zero modes can be projected out or removed by including additional conditions, e.g. by fixing one value of z for every x.

3.3 General Gaussian prior factors

3.3.1 The general case

In the previous sections we studied priors consisting of a factor (the specific prior) which was Gaussian with respect to P or $L = \ln P$ and additional normalization (and non–negativity) conditions. In this section we consider the situation where the probability $p(y|x,h)$ is expressed in terms of a function $\phi(x,y)$. That means, we assume a, possibly non–linear, operator $P = P(\phi)$ which maps the function ϕ to a probability. We can then formulate a learning problem in terms of the function ϕ, meaning that ϕ now represents the hidden variables or unknown state of Nature h.[2] Consider the case of a specific prior which is Gaussian in ϕ, i.e., which has a specific prior energy quadratic in ϕ

$$\frac{1}{2}(\,\phi\,,\,\mathbf{K}\,\phi\,). \tag{186}$$

This means we are lead to error functionals of the form

$$E_\phi = -(\,\ln P(\phi)\,,\,N\,) + \frac{1}{2}(\,\phi\,,\,\mathbf{K}\,\phi\,) + (\,P(\phi)\,,\,\Lambda_X\,), \tag{187}$$

where we have skipped the ϕ–independent part of the Λ_X–terms. In general cases also the non–negativity constraint has to be implemented.

To express the functional derivative of functional (187) with respect to ϕ we define besides the diagonal matrix $\mathbf{P} = \mathbf{P}(\phi)$ the Jacobian, i.e., the matrix of derivatives $\mathbf{P}' = \mathbf{P}'(\phi)$ with matrix elements

$$\mathbf{P}'(x,y;x',y';\phi) = \frac{\delta P(x',y';\phi)}{\delta\phi(x,y)}. \tag{188}$$

The matrix $\mathbf{P}'$ is diagonal for point–wise transformations, i.e., for $P(x,y;\phi) = P(\,\phi(x,y)\,)$. In such cases we use P' to denote the vector of diagonal elements

[2] Besides ϕ also the hyperparameters discussed in Chapter 5 belong to the hidden variables h.

of $\mathbf{P}'$. An example is the previously discussed transformation $L = \ln P$ for which $\mathbf{P}' = \mathbf{P}$. The stationarity equation for functional (187) becomes

$$0 = \mathbf{P}'(\phi)\mathbf{P}^{-1}(\phi)N - \mathbf{K}\phi - \mathbf{P}'(\phi)\Lambda_X, \tag{189}$$

and for existing $\mathbf{P}\mathbf{P}'^{-1}$ $=(\mathbf{P}'\mathbf{P}^{-1})^{-1}$ (for nonexisting inverse see Section 4.1),

$$0 = N - \mathbf{P}\mathbf{P}'^{-1}\mathbf{K}\,\phi - \mathbf{P}\Lambda_X. \tag{190}$$

From Eq. (190) the Lagrange multiplier function can be found by integration, using the normalization condition $\mathbf{I}_X P = I$, in the form $\mathbf{I}_X\mathbf{P}\Lambda_X = \Lambda_X$ for $\Lambda_X(x) \neq 0$. Thus, multiplying Eq. (190) by $\mathbf{I}_X$ yields

$$\Lambda_X = \mathbf{I}_X\left(N - \mathbf{P}\mathbf{P}'^{-1}\mathbf{K}\,\phi\right) = N_X - \mathbf{I}_X\mathbf{P}\mathbf{P}'^{-1}\mathbf{K}\,\phi. \tag{191}$$

Λ_X is now eliminated by inserting Eq. (191) into Eq. (190)

$$0 = (\mathbf{I} - \mathbf{P}\mathbf{I}_X)\left(N - \mathbf{P}\mathbf{P}'^{-1}\mathbf{K}\,\phi\right). \tag{192}$$

A simple iteration procedure, provided $\mathbf{K}^{-1}$ exists, is suggested by writing Eq. (189) in the form

$$\mathbf{K}\phi = T_\phi, \quad \phi^{i+1} = \mathbf{K}^{-1}T_\phi(\phi^i), \tag{193}$$

with

$$T_\phi(\phi) = \mathbf{P}'\mathbf{P}^{-1}N - \mathbf{P}'\Lambda_X. \tag{194}$$

Table 2 lists constraints to be implemented explicitly for some choices of ϕ.

3.3.2 Example: Square root of P

We already discussed the cases $\phi = \ln P$ with $P' = P = e^L$, $P/P' = 1$ and $\phi = P$ with $P' = 1$, $P/P' = P$. The choice $\phi = \sqrt{P}$ yields the common L_2–normalization condition over y

$$1 = \int dy\, \phi^2(x,y), \quad \forall x \in X, \tag{195}$$

which is quadratic in ϕ, and $P = \phi^2$, $P' = 2\phi$, $P/P' = \phi/2$. For real ϕ the non–negativity condition $P \geq 0$ is automatically satisfied [82, 211].

ϕ	$P(\phi)$	constraints	
$P(x,y)$	$P = P$	norm	non–negativity
$z(x,y)$	$P = z/\int z\,dy$	—	non–negativity
$L(x,y) = \ln P$	$P = e^L$	norm	—
$g(x,y)$	$P = e^g/\int e^g\,dy$	—	—
$\Phi = \int^y dy'\,P$	$P = d\Phi/dy$	boundary	monotony

Table 2: Constraints for specific choices of ϕ

For $\phi = \sqrt{P}$ and a negative Laplacian inverse covariance $\mathbf{K} = -\Delta$, one can relate the corresponding Gaussian prior to the *Fisher information* [38, 211, 207]. Consider, for example, a problem with fixed x (so x can be skipped from the notation and one can write $P(y)$) and a d_y–dimensional y. Then one has, assuming the necessary differentiability conditions and vanishing boundary terms,

$$(\,\phi,\,\mathbf{K}\,\phi\,) = -(\,\phi,\,\Delta\,\phi\,) = \int dy\,\sum_k^{d_y}\left|\frac{\partial\phi}{\partial y_k}\right|^2 \tag{196}$$

$$= \sum_k^{d_y}\int\frac{dy}{4P(y)}\left(\frac{\partial P(y)}{\partial y_k}\right)^2 = \frac{1}{4}\sum_k^{d_y} I_k^F(0), \tag{197}$$

where $I_k^F(0)$ is the Fisher information, defined as

$$I_k^F(y_0) = \int dy\frac{\left|\frac{\partial P(y-y^0)}{\partial y^0}\right|^2}{P(y-y^0)} = \int dy\left|\frac{\partial\ln P(y-y^0)}{\partial y_k^0}\right|^2 P(y-y_k^0), \tag{198}$$

for the family $P(\cdot - y^0)$ with location parameter vector y^0.

A connection to quantum mechanics can be found considering the case without training data

$$E_\phi = \frac{1}{2}(\,\phi,\ \mathbf{K}\,\phi\,) + (\Lambda_X,\,\phi), \tag{199}$$

having the homogeneous stationarity equation

$$\mathbf{K}\,\phi = -2\Phi\Lambda_X. \tag{200}$$

For x–independent Λ_X this is an eigenvalue equation. Examples include the quantum mechanical Schrödinger equation where $\mathbf{K}$ corresponds to the system Hamiltonian and

$$-2\Lambda_X = \frac{(\phi,\,\mathbf{K}\,\phi)}{(\phi,\,\phi)}, \tag{201}$$

to its ground state energy. In quantum mechanics Eq. (201) is the basis for variational methods (see Section 4) to obtain approximate solutions for ground state energies [55, 197, 27].

Similarly, one can take $\phi = \sqrt{-(L-L_{\max})}$ for L bounded from above by $L_{\max}$ with the normalization

$$1 = \int dy\, e^{-\phi^2(x,y)+L_{\max}}, \quad \forall x \in X, \tag{202}$$

and $P = e^{-\phi^2+L_{\max}}$, $P' = -2\phi P$, $P/P' = -1/(2\phi)$.

3.3.3 Example: Distribution functions

Instead in terms of the probability density function, one can formulate the prior in terms of its integral, the distribution function. The density P is then recovered from the distribution function ϕ by differentiation,

$$P(\phi) = \prod_k^{d_y} \frac{\partial \phi}{\partial y_k} = \prod_k^{d_y} \nabla_{y_k}\phi = \bigotimes_k^{d_y} \mathbf{R}_k^{-1}\phi. = \mathbf{R}^{-1}\phi, \tag{203}$$

resulting in a non–diagonal $\mathbf{P}'$. The inverse of the derivative operator $\mathbf{R}^{-1}$ is the integration operator $\mathbf{R} = \bigotimes_k^{d_y}\mathbf{R}_k P$ with matrix elements

$$\mathbf{R}(x,y;x',y') = \delta(x-x')\theta(y-y'), \tag{204}$$

i.e.,

$$\mathbf{R}_k(x,y;x',y') = \delta(x-x')\theta(y_k-y_k')\prod_{l\neq k}\delta(y_l-y_l'). \tag{205}$$

Thus, (203) corresponds to the transformation of (x–conditioned) density functions P in (x–conditioned) distribution functions $\phi = \mathbf{R}P$, i.e., $\phi(x,y) =$

$\int_{-\infty}^{y} P(x,y')dy'$. Because $\mathbf{R^T K R}$ is positive (semi–)definite if $\mathbf{K}$ is, a specific prior which is Gaussian in the distribution function ϕ is also Gaussian in the density P. $\mathbf{P}'$ becomes

$$\mathbf{P}'(x,y;x',y') = \frac{\delta\left(\prod_k^{d_y} \nabla_{y_{k'}} \phi(x',y')\right)}{\delta\phi(x,y)} = \delta(x-x')\prod_k^{d_y}\delta'(y_k-y_k'). \tag{206}$$

Here the derivative of the δ–function is defined by formal partial integration

$$\int_{-\infty}^{\infty} dy'\, f(y')\delta'(y-y') = f(y')\delta(y'-y)|_{-\infty}^{\infty} - f'(y). \tag{207}$$

Fixing $\phi(x,-\infty)=0$ the variational derivative $\delta/(\delta\phi(x,-\infty))$ is not needed. The normalization condition for P becomes for the distribution function $\phi = \mathbf{R}P$ the boundary condition $\phi(x,\infty)=1$, $\forall x \in X$. The non–negativity condition for P corresponds to the monotonicity condition $\phi(x,y)\ge\phi(x,y')$, $\forall y \ge y'$, $\forall x \in X$ and to $\phi(x,-\infty)\ge 0$, $\forall x\in X$.

3.4 Covariances and invariances

3.4.1 Approximate invariance

Prior terms can often be related to the assumption of approximate invariances or approximate symmetries. A Laplacian smoothness functional, for example, measures the deviation from translational symmetry under infinitesimal translations.

Consider for example a linear mapping

$$\phi \to \tilde\phi = \mathbf{S}\phi, \tag{208}$$

given by the operator $\mathbf{S}$. To compare ϕ with $\tilde\phi$ we define a (semi–)distance defined by choosing a positive (semi–)definite $\mathbf{K}_S$, and use as error measure

$$\frac{1}{2}\Big((\phi-\mathbf{S}\phi),\,\mathbf{K}_S(\phi-\mathbf{S}\phi)\Big) = \frac{1}{2}\Big(\phi,\,\mathbf{K}\phi\Big). \tag{209}$$

Here

$$\mathbf{K} = (\mathbf{I}-\mathbf{S})^T\mathbf{K}_S(\mathbf{I}-\mathbf{S}) \tag{210}$$

is positive semi–definite if $\mathbf{K}_S$ is. Conversely, every positive semi–definite $\mathbf{K}$ can be written $\mathbf{K}=\mathbf{W}^T\mathbf{W}$ and is thus of form (210) with $\mathbf{S}=\mathbf{I}-\mathbf{W}$ and

$\mathbf{K}_S = \mathbf{I}$. Including terms of the form of (210) in the error functional forces ϕ to be similar to $\tilde{\phi}$.

A special case are mappings leaving the norm invariant

$$(\phi,\, \phi) = (\mathbf{S}\phi, \mathbf{S}\phi) = (\phi,\, \mathbf{S}^T\mathbf{S}\phi). \tag{211}$$

For real ϕ and $\tilde{\phi}$ i.e., $(\mathbf{S}\phi) = (\mathbf{S}\phi)^*$, this requires $\mathbf{S}^T = \mathbf{S}^{-1}$ and $\mathbf{S}^* = \mathbf{S}$. Thus, in that case $\mathbf{S}$ has to be an orthogonal matrix $\in O(N)$ and can be written

$$\mathbf{S}(\theta) = e^{\mathbf{A}} = e^{\sum_i \theta_i \mathbf{A}_i} = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\sum_i \theta_i \mathbf{A}_i \right)^k, \tag{212}$$

with antisymmetric $\mathbf{A} = -\mathbf{A}^T$ and real parameters θ_i. Selecting a set of (generators) $\mathbf{A}_i$ the matrices obtained be varying the parameters θ_i form a Lie group. Up to first order the expansion of the exponential function reads $\mathbf{S} \approx 1 + \sum_i \nabla_i \mathbf{A}_i$. Thus, we can define an error measure with respect to an infinitesimal transformation by

$$\frac{1}{2}\sum_i \left(\frac{\phi - (1+\theta_i\mathbf{A}_i)\phi}{\theta_i},\, \mathbf{K}_S \frac{\phi - (1+\theta_i\mathbf{A}_i)\phi}{\theta_i} \right) = \frac{1}{2}(\phi,\, \sum_i \mathbf{A}_i^T \mathbf{K}_S \mathbf{A}_i \phi). \tag{213}$$

3.4.2 Approximate symmetries

Next we come to the special case of symmetries, i.e., invariance under under coordinate transformations. Symmetry transformations $\mathbf{S}$ change the arguments of a function ϕ. For example for the translation of a function $\phi(x) \to \tilde{\phi}(x) = \mathbf{S}\phi(x) = \phi(x - c)$. Therefore it is useful to see how $\mathbf{S}$ acts on the arguments of a function. Denoting the (possibly improper) eigenvectors of the coordinate operator $\mathbf{x}$ with eigenvalue x by $(\cdot,\, x) = |x)$, i.e., $\mathbf{x}|x) = x|x)$, function values can be expressed as scalar products, e.g. $\phi(x) = (x,\, \phi)$ for a function in x, or, in two variables, $\phi(x, y) = (x \otimes y,\, \phi)$. (Note that in this 'eigenvalue' notation, frequently used by physicists, for example $2|x) \neq |2x)$.) Thus, we see that the action of $\mathbf{S}$ on some function $h(x)$ is equivalent to the action of $\mathbf{S}^T$ ($= \mathbf{S}^{-1}$ if orthogonal) on $|x)$

$$\mathbf{S}\phi(x) = (x, \mathbf{S}\phi) = (\mathbf{S}^T x, \phi), \tag{214}$$

or for $\phi(x, y)$

$$\mathbf{S}\phi(x, y) = \left(\mathbf{S}^T(x \otimes y),\, \phi\right). \tag{215}$$

Assuming $\mathbf{S} = \mathbf{S}_x \mathbf{S}_y$ we may also split the action of $\mathbf{S}$,

$$\mathbf{S}\phi(x, y) = \left((\mathbf{S}_x^T x) \otimes y,\, \mathbf{S}_y \phi \right). \tag{216}$$

This is convenient for example for vector fields in physics where x and $\phi(\cdot, y)$ form three dimensional vectors with y representing a linear combination of component labels of ϕ.

Notice that, for a general operator $\mathbf{S}$, the transformed argument $\mathbf{S}|x)$ does not have to be an eigenvector of the coordinate operator $\mathbf{x}$ again. In the general case $\mathbf{S}$ can map a specific $|x)$ to arbitrary vectors being linear combinations of all $|x')$, i.e., $\mathbf{S}|x) = \int dx'\, S(x, x')|x')$. A general orthogonal $\mathbf{S}$ maps an orthonormal basis to another orthonormal basis. Coordinate transformations, however, are represented by operators $\mathbf{S}$, which map coordinate eigenvectors $|x)$ to other coordinate eigenvectors $|\sigma(x))$. Hence, such coordinate transformations $\mathbf{S}$ just changes the argument x of a function ϕ into $\sigma(x)$, i.e.,

$$\mathbf{S}\phi(x) = \phi(\sigma(x)), \tag{217}$$

with $\sigma(x)$ a permutation or a one–to–one coordinate transformation. Thus, even for an arbitrary nonlinear coordinate transformation σ the corresponding operator $\mathbf{S}$ in the space of ϕ is linear. (This is one of the reasons why linear functional analysis is so useful.)

A special case are linear coordinate transformations for which we can write $\phi(x) \rightarrow \tilde{\phi}(x) = \mathbf{S}\phi(x) = \phi(Sx)$, with S (in contrast to $\mathbf{S}$) acting in the space of x. An example of such S are coordinate rotations which preserve the norm in x–space, analogously to Eq. (211) for ϕ, and form a Lie group $S(\theta) = e^{\sum_i \theta_i A_i}$ acting on coordinates, analogously to Eq. (212).

3.4.3 Example: Infinitesimal translations

A Laplacian smoothness prior, for example, can be related to an approximate symmetry under infinitesimal translations. Considering the group of d–dimensional translations which is generated by the gradient operator ∇, this can be verified by recalling the multidimensional Taylor formula for expansion of ϕ at x

$$\mathbf{S}(\theta)\phi(x) = e^{\sum_i \theta_i \nabla_i} \phi(x) = \sum_{k=0}^{\infty} \frac{\left(\sum_i \theta_i \nabla_i\right)^k}{k!} \phi(x) = \phi(x + \theta). \tag{218}$$

Up to first order $\mathbf{S} \approx 1 + \sum_i \theta_i \Delta_i$. Hence, for infinitesimal translations, the error measure of Eq. (213) becomes

$$\frac{1}{2}\sum_i \left(\frac{\phi - (1+\theta_i \nabla_i)\phi}{\theta_i},\, \frac{\phi - (1+\theta_i \nabla_i)\phi}{\theta_i} \right) = \frac{1}{2}(\phi,\, \sum_i \nabla_i^T \nabla_i \phi) = -\frac{1}{2}(\phi,\, \Delta\phi), \tag{219}$$

assuming vanishing boundary terms and choosing $\mathbf{K}_S = \mathbf{I}$. This is the classical Laplacian smoothness term.

3.4.4 Example: Approximate periodicity

As another example, lets us discuss the implementation of approximate periodicity. To measure the deviation from exact periodicity let us define the difference operators

$$\nabla_\theta^R \phi(x) = \phi(x+\theta) - \phi(x), \tag{220}$$

$$\nabla_\theta^L \phi(x) = \phi(x) - \phi(x-\theta). \tag{221}$$

For periodic boundary conditions $(\nabla_\theta^L)^T = -\nabla_\theta^R$, where $(\nabla_\theta^L)^T$ denotes the transpose of ∇_θ^L. Hence, the operator,

$$\Delta_\theta = \nabla_\theta^L \nabla_\theta^R = -(\nabla_\theta^R)^T \nabla_\theta^R, \tag{222}$$

defined similarly to the Laplacian, is negative (semi) definite, and a possible error term, enforcing approximate periodicity with period θ, is

$$\frac{1}{2}(\nabla_R(\theta)\phi,\, \nabla_R(\theta)\phi) = -\frac{1}{2}(\phi,\, \Delta_\theta \phi) = \frac{1}{2}\int dx\, |\phi(x) - \phi(x+\theta)|^2. \tag{223}$$

As every periodic function with $\phi(x) = \phi(x+\theta)$ is in the null space of Δ_θ typically another error term has to be added to get a unique solution of the stationarity equation. Choosing, for example, a Laplacian smoothness term, yields

$$-\frac{1}{2}(\phi,\, (\Delta + \lambda\Delta_\theta)\, \phi). \tag{224}$$

In case θ is not known, it can be treated as hyperparameter as discussed in Section 5.

Alternatively to an implementation by choosing a semi–positive definite operator $\mathbf{K}$ with symmetric functions in its null space, approximate symmetries can be implemented by giving explicitly a symmetric reference function $t(x)$. For example, $\frac{1}{2}(\phi - t,\, \mathbf{K}(\phi - t)\,)$ with $t(x) = t(x+\theta)$. This possibility will be discussed in the next section.

3.5 Non–zero means

A prior energy term $(1/2)(\phi,\, \mathbf{K}\,\phi)$ measures the squared $\mathbf{K}$–distance of ϕ to the zero function $t \equiv 0$. Choosing a zero mean function for the prior process is calculationally convenient for Gaussian priors, but by no means mandatory. In particular, a function ϕ is in practice often measured relative to some non–trivial base line. Without further *a priori* information that base line can in principle be an arbitrary function. Choosing a zero mean function that base line does not enter the formulae and remains hidden in the realization of the measurement process. On the the other hand, including explicitly a non–zero mean function t, playing the role of a function *template* (or reference, target, prototype, base line) and being technically relatively straightforward, can be a very powerful tool. It allows, for example, to parameterize $t(\theta)$ by introducing hyperparameters (see Section 5) and to specify explicitly different maxima of multimodal functional priors (see Section 6. [132, 133, 134, 135, 136]). All this cannot be done by referring to a single baseline.

Hence, in this section we consider error terms of the form

$$\frac{1}{2}\Big(\phi - t,\, \mathbf{K}\,(\phi - t)\Big). \tag{225}$$

Mean or template functions t allow an easy and straightforward implementation of prior information in form of examples for ϕ. They are the continuous analogue of standard training data. The fact that template functions t are most times chosen equal to zero, and thus do not appear explicitly in the error functional, should not obscure the fact that they are of key importance for any generalization. There are many situations where it can be very valuable to include non–zero prior means explicitly. Template functions for ϕ can for example result from learning done in the past for the same or for similar tasks. In particular, consider for example $\tilde{\phi}(x)$ to be the output of an empirical learning system (neural net, decision tree, nearest neighbor methods, ...) being the result of learning the same or a similar task. Such a $\tilde{\phi}(x)$ would be a natural candidate for a template function $t(x)$. Thus, we see that template functions could be used for example to allow *transfer* of knowledge between similar tasks or to *include the results of earlier learning* on the same task in case the original data are lost but the output of another learning system is still available.

Including non–zero template functions generalizes functional E_ϕ of Eq.

(187) to

$$E_\phi = -(\ln P(\phi),\, N) + \frac{1}{2}\Big(\phi - t,\, \mathbf{K}\,(\phi - t)\Big) + (P(\phi),\, \Lambda_X) \tag{226}$$

$$= -(\ln P(\phi),\, N) + \frac{1}{2}(\phi,\, \mathbf{K}\,\phi) - (J,\, \phi) + (P(\phi),\, \Lambda_X) + \text{const.} \tag{227}$$

In the language of physics $J = \mathbf{K}t$ represents an *external field* coupling to $\phi(x, y)$, similar, for example, to a magnetic field. A non–zero field leads to a non–zero expectation of ϕ in the no–data case. The ϕ–independent constant stands for the term $\frac{1}{2}(t,\, \mathbf{K}\,t)$, or $\frac{1}{2}(J,\, \mathbf{K}^{-1}\,J)$ for invertible $\mathbf{K}$, and can be skipped from the error/energy functional E_ϕ.

The stationarity equation for an E_ϕ with non–zero template t contains an inhomogeneous term $\mathbf{K}t = J$

$$0 = \mathbf{P}'(\phi)\mathbf{P}^{-1}(\phi)N - \mathbf{P}'(\phi)\Lambda_X - \mathbf{K}\,(\phi - t)\,, \tag{228}$$

with, for invertible $\mathbf{P}\mathbf{P}'^{-1}$ and $\Lambda_X \neq 0$,

$$\Lambda_X = \mathbf{I}_X\left(N - \mathbf{P}\mathbf{P}'^{-1}\mathbf{K}\,(\phi - t)\right). \tag{229}$$

Notice that functional (226) can be rewritten as a functional with zero template $t \equiv 0$ in terms of $\widetilde{\phi} = \phi - t$. That is the reason why we have not included non–zero templates in the previous sections. For general non–additive combinations of squared distances of the form (225) non–zero templates cannot be removed from the functional as we will see in Section 6. Additive combinations of squared error terms, on the other hand, can again be written as one squared error term, using a generalized 'bias–variance'–decomposition

$$\frac{1}{2}\sum_{j=1}^{N}\Big(\phi - t_j,\, \mathbf{K}_j\,(\phi - t_j)\Big) = \frac{1}{2}\Big(\phi - t,\, \mathbf{K}\,(\phi - t)\Big) + E_{\min} \tag{230}$$

with *template average*

$$t = \mathbf{K}^{-1}\sum_{j=1}^{N}\mathbf{K}_j t_j, \tag{231}$$

assuming the existence of the inverse of the operator

$$\mathbf{K} = \sum_{j=1}^{N}\mathbf{K}_j. \tag{232}$$

and *minimal energy/error*

$$E_{\min} = \frac{N}{2} V(t_1, \cdots t_N) = \frac{1}{2} \sum_{j=1}^{N} (t_j,\, \mathbf{K}_j\, t_j) - (t,\, \mathbf{K}\, t), \tag{233}$$

which up to a factor $N/2$ represents a generalized template variance V. We end with the remark that adding error terms corresponds in its probabilistic Bayesian interpretation to ANDing independent events. For example, if we wish to implement that ϕ is likely to be smooth AND mirror symmetric, we may add two squared error terms, one related to smoothness and another to mirror symmetry. According to (230) the result will be a single squared error term of form (225).

Summarizing, we have seen that there are many potentially useful applications of non–zero template functions. Technically, however, non–zero template functions can be removed from the formalism by a simple substitution $\phi' = \phi - t$ if the error functional consists of an additive combination of quadratic prior terms. As most regularized error functionals used in practice have additive prior terms this is probably the reason that they are formulated for $t \equiv 0$, meaning that non–zero templates functions (base lines) have to be treated by including a preprocessing step switching from ϕ to ϕ'. We will see in Section 6 that for general error functionals templates cannot be removed by a simple substitution and do enter the error functionals explicitly.

3.6 Quadratic density estimation and empirical risk minimization

Interpreting an energy or error functional E probabilistically, i.e., assuming $-\beta E + c$ to be the logarithm of a posterior probability under study, the form of the training data term has to be $-\sum_i \ln P_i$. Technically, however, it would be easier to replace that data term by one which is quadratic in the function ϕ of interest.

Indeed, we have mentioned in Section 2.5 that such functionals can be justified within the framework of empirical risk minimization. From that Frequentist point of view an error functional $E(P)$, is not derived from a log–posterior, but represents an empirical risk $\hat{r}(P, f) = \sum_i l(x_i, y_i, P)$, approximating an expected risk $r(P, f)$ for action $a = P$. This is possible under the assumption that training data are sampled according to the true $p(x, y|f)$. In that interpretation one is therefore not restricted to a log–loss for training data but may as well choose for training data a quadratic loss like

$$\frac{1}{2}\Big(P - P_{\rm emp},\, \mathbf{K}_D\,(P - P_{\rm emp})\Big), \tag{234}$$

choosing a reference density Pemp and a real symmetric positive (semi) definite $\mathbf{K}_D$.

Approximating a joint probability $p(x,y|h)$ the reference density $P_{\rm emp}$ would have to be the joint empirical density

$$P^{\rm joint}_{\rm emp}(x,y) = \frac{1}{n}\sum_i^n \delta(x-x_i)\delta(y-y_i), \tag{235}$$

i.e., $P^{\rm joint}_{\rm emp} = N/n$, as obtained from the training data. Approximating conditional probabilities $p(y|x,h)$ the reference $P_{\rm emp}$ has to be chosen as conditional empirical density,

$$P_{\rm emp}(x,y) = \frac{\sum_i \delta(x-x_i)\delta(y-y_i)}{\sum_i \delta(x-x_i)} = \frac{N(x,y)}{n_x}, \tag{236}$$

or, defining the diagonal matrix $\mathbf{N}_X(x,x',y,y') = \delta(x-x')\delta(y-y')N_X(x) = \delta(x-x')\delta(y-y')\sum_i \delta(x-x_i)$

$$P_{\rm emp} = \mathbf{N}_X^{-1} N. \tag{237}$$

This, however, is only a valid expression if $N_X(x) \ne 0$, meaning that for all x at least one measured value has to be available. For x variables with a large number of possible values, this cannot be assumed. For continuous x variables it is even impossible.

Hence, approximating conditional empirical densities either non–data x–values must be excluded from the integration in (234) by using an operator $\mathbf{K}_D$ containing the projector $\sum_{x'\in x_D}\delta(x-x')$, or $P_{\rm emp}$ must be defined also for such non–data x–values. For existing $V_X = \mathbf{I}_X 1 = \int dy\, 1$, a possible extension $\tilde P_{\rm emp}$ of $P_{\rm emp}$ would be to assume a uniform density for non–data x values, yielding

$$\tilde P_{\rm emp}(x,y) = \left\{ \begin{array}{ll} \frac{\sum_i \delta(x-x_i)\delta(y-y_i)}{\sum_i \delta(x-x_i)} & \text{for} \quad \sum_i \delta(x-x_i) \ne 0, \\ \frac{1}{\int dy\, 1} & \text{for} \quad \sum_i \delta(x-x_i) = 0. \end{array} \right. \tag{238}$$

This introduces a bias towards uniform probabilities, but has the advantage to give a empirical density for all x and to fulfill the conditional normalization requirements.

Instead of a quadratic term in P, one might consider a quadratic term in the log–probability L. The log–probability, however, is minus infinity at all non–data points $(x, y) \notin D$. To work with a finite expression, one can choose small $\epsilon(y)$ and approximate $P_{\rm emp}$ by

$$P^\epsilon_{\rm emp}(x,y) = \frac{\epsilon(y) + \sum_i \delta(x-x_i)\delta(y-y_i)}{\int dy\, \epsilon(y) + \sum_i \delta(x-x_i)}, \tag{239}$$

provided $\int dy\, \epsilon(y)$ exists. For $\epsilon(y) \ne 0$ also $P^\epsilon_{\rm emp}(x,y) \ne 0$, $\forall x$ and $L^\epsilon_{\rm emp} = \ln P^\epsilon_{\rm emp} > -\infty$ exists.

A quadratic data term in P results in an error functional

$$\tilde E_P = \frac{1}{2}\Big(P - P_{\rm emp},\, \mathbf{K}_D\,(P - P_{\rm emp})\Big) + \frac{1}{2}(P,\, \mathbf{K}\,P) + (P,\, \Lambda_X), \tag{240}$$

skipping the constant part of the Λ_X–terms. In (240) the empirical density $P_{\rm emp}$ may be replaced by $\tilde P_{\rm emp}$ of (238).

Positive (semi–)definite operators $\mathbf{K}_D$ have a square root and can be written in the form $\mathbf{R}^T\mathbf{R}$. One possibility, skipping for the sake of simplicity x in the following, is to choose as square root $\mathbf{R}$ the integration operator, i.e., $\mathbf{R} = \bigotimes_k \mathbf{R}_k$ and $\mathbf{R}(y, y') = \theta(y - y')$. Thus, $\phi = \mathbf{R}P$ transforms the density function P in the distribution function ϕ, and we have $P = P(\phi) = \mathbf{R}^{-1}\phi$. Here the inverse $\mathbf{R}^{-1}$ is the differentiation operator $\prod_k \nabla_{y_k}$ (with appropriate boundary conditions) and $\left(\mathbf{R}^T\right)^{-1}\mathbf{R}^{-1} = -\prod_k \Delta_k$ is the product of one–dimensional Laplacians $\Delta_k = \partial^2/\partial y_k^2$. Adding for example a regularizing term $\frac{\lambda}{2}(P,\, P)$ as in (165) gives

$$\tilde E_P = \frac{1}{2}\Big(\, P - P_{\rm emp}\,,\, \mathbf{R}^T\mathbf{R}\,(P - P_{\rm emp})\Big) + \frac{\lambda}{2}(P,\, P) \tag{241}$$

$$= \frac{1}{2}\left(\Big(\phi - \phi_{\rm emp},\, \phi - \phi_{\rm emp}\Big) - \lambda\Big(\phi,\, \prod_k \Delta_k\, \phi\Big)\right) \tag{242}$$

$$= \frac{1}{2m^2}\Big(\phi,\, (-\prod_k \Delta_k + m^2\mathbf{I})\phi\Big) - (\phi\,, \phi_{\rm emp}) + \frac{1}{2}(\phi_{\rm emp},\, \phi_{\rm emp}). \tag{243}$$

with $m^2 = \lambda^{-1}$. Here the empirical distribution function $\phi_{\rm emp} = \mathbf{R}P_{\rm emp}$ is given by $\phi_{\rm emp}(y) = \frac{1}{n}\sum_i \theta(y - y_i)$ (or, including the x variable, $\phi_{\rm emp}(x,y) = \frac{1}{N_X(x)}\sum_{x'\in x_D} \delta(x - x')\theta(y - y_i)$ for $N_X(x) \ne 0$ which could be extended to a linear $\tilde\phi = \mathbf{R}\tilde P_{\rm emp}$ for $N_X(x) = 0$). The stationarity equation yields

$$\phi = m^2\left(-\prod_k \Delta_k + m^2\mathbf{I}\right)^{-1}\phi_{\rm emp}. \tag{244}$$

For $d_y = 1$ (or $\phi = \prod_k \phi$) the operator becomes $(-\Delta + m^2\mathbf{I})^{-1}$ which has the structure of a free massive propagator for a scalar field with mass m^2 and is calculated in section 7.2.3. As already mentioned the normalization and non–negativity condition for P appear for ϕ as boundary and monotonicity conditions. For non–constant P the monotonicity condition has not to be implemented by Lagrange multipliers as the gradient at the stationary point has no components pointing into the forbidden area. (But the conditions nevertheless have to be checked.) Kernel methods of density estimation, like the use of Parzen windows, can be founded on such quadratic regularization functionals [224]. Indeed, the one–dimensional Eq. (244) is equivalent to the use of Parzens kernel in density estimation [182, 169].

3.7 Regression

3.7.1 Gaussian regression

An important special case of density estimation leading to quadratic data terms is regression for independent training data with Gaussian likelihoods

$$p(y_i|x_i,h) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(y_i-h(x_i))^2}{2\sigma^2}}, \tag{245}$$

with fixed, but possibly x_i–dependent, variance σ^2. In that case $P(x,y) = p(y_i|x_i,h)$ is specified by $\phi = h$ and the logarithmic term $\sum_i \ln P_i$ becomes quadratic in the regression function $h(x_i)$, i.e., of the form (225). In an interpretation as empirical risk minimization quadratic error terms corresponds to the choice of a squared error loss function $l(x,y,a) = (y - a(x))^2$ for action $a(x)$. Similarly, the technical analogon of Bayesian priors are additional (regularizing) cost terms.

We have remarked in Section 2.3 that for continuous x measurement of $h(x)$ has to be understood as measurement of a $h(\tilde{x}) = \int dx\, \vartheta(x)h(x)$ for sharply peaked $\vartheta(x)$. We assume here that the discretization of h used in numerical calculations takes care of that averaging. Divergent quantities like δ–functionals, used here for convenience, will then not be present.

We now combine Gaussian data terms and a Gaussian (specific) prior with prior operator $\mathbf{K}_0(x,x')$ and define for training data x_i, y_i the operator

$$\mathbf{K}_i(x,x') = \delta(x - x_i)\delta(x - x'), \tag{246}$$

and training data templates $t = y_i$. We also allow a general prior template t_0 but remark that it is often chosen identically zero. According to (230) the resulting functional can be written in the following forms, useful for different purposes,

$$
\begin{aligned}
E_h &= \frac{1}{2}\sum_{i=1}^{n}(h(x_i)-y_i)^2 + \frac{1}{2}(\,h-t_0,\,\mathbf{K}_0\,(h-t_0)\,)_X && (247)\\
&= \frac{1}{2}\sum_{i=1}^{n}(\,h-t_i,\,\mathbf{K}_i(h-t_i)\,)_X + \frac{1}{2}(\,h-t_0,\,\mathbf{K}_0\,(h-t_0)\,)_X && (248)\\
&= \frac{1}{2}\sum_{i=0}^{n}(\,h-t_i,\,\mathbf{K}_i(h-t_i)\,)_X && (249)\\
&= \frac{1}{2}(h-t_D,\,\mathbf{K}_D(h-t_D))_X + \frac{1}{2}(h-t_0,\,\mathbf{K}_0(h-t_0))_X + E^D_{\min} && (250)\\
&= \frac{1}{2}(\,h,\,\mathbf{K}\,h\,)_X - (\,h,\,\mathbf{K}\,t\,)_X + \frac{1}{2}\sum_{i=0}^{n}(\,t_i,\,\mathbf{K}_i t_i\,)_X && (251)\\
&= \frac{1}{2}(\,h-t,\,\mathbf{K}\,(h-t)\,)_X + E_{\min}, && (252)
\end{aligned}
$$

with

$$
\mathbf{K}_D = \sum_{i=1}^{n}\mathbf{K}_i, \quad t_D = \mathbf{K}_D^{-1}\sum_{i=1}^{n}\mathbf{K}_i t_i, \tag{253}
$$

$$
\mathbf{K} = \sum_{i=0}^{n}\mathbf{K}_i, \quad t = \mathbf{K}^{-1}\sum_{i=0}^{n}\mathbf{K}_i t_i, \tag{254}
$$

and h–independent minimal errors,

$$
\begin{aligned}
E^D_{\min} &= \frac{1}{2}\left(\sum_{i=1}^{n}(t_i,\ \mathbf{K}_i t_i)_X - (t_D,\ \mathbf{K}_D t_D)_X\right), && (255)\\
E_{\min} &= \frac{1}{2}\left(\sum_{i=0}^{n}(t_i,\ \mathbf{K}_i t_i)_X - (t,\ \mathbf{K} t)_X\right), && (256)
\end{aligned}
$$

being proportional to the "generalized variances" $V_D = 2E^D_{\min}/n$ and $V = 2E_{\min}/(n+1)$. The scalar product $(\cdot,\cdot)_X$ stands for x–integration only, for the sake of simplicity however, we will skip the subscript X in the following. The data operator $\mathbf{K}_D$

$$
\mathbf{K}_D(x,x') = \sum_{i=1}^{n}\delta(x-x_i)\delta(x-x') = n_x\,\delta(x-x'), \tag{257}
$$

contains for discrete x on its diagonal the number of measurements at x,

$$
n_x = N_X(x) = \sum_{i=1}^{n}\delta(x-x_i), \tag{258}
$$

which is zero for x not in the training data. As already mentioned for continuous x a integration around a neighborhood of x_i is required. $\mathbf{K}_D^{-1}$ is a short hand notation for the inverse within the space of training data

$$\mathbf{K}_D^{-1} = (\mathbf{I}_D \mathbf{K}_D \mathbf{I}_D)^{-1} = \delta(x - x')/n_x, \tag{259}$$

$\mathbf{I}_D$ denoting the projector into the space of training data

$$\mathbf{I}_D = \delta(x - x') \sum_{i=1}^{\tilde{n}} \delta(x - x_i). \tag{260}$$

Notice that the sum is not over all n training points x_i but only over the $\tilde{n} \le n$ different x_i. (Again for continuous x an integration around x_i is required to ensure $\mathbf{I}_D^2 = \mathbf{I}_D$). Hence, the data template t_D becomes the mean of y–values measured at x

$$t_D(x) = \frac{1}{n_x} \sum_{\substack{j=1 \\ x_j = x}}^{n_x} y(x_j), \tag{261}$$

and $t_D(x) = 0$ for $n_x = 0$. Normalization of $P(x, y)$ is not influenced by a change in $h(x)$ so the Lagrange multiplier terms have been skipped.

The stationarity equation is most easily obtained from (252),

$$0 = \mathbf{K}(h - t). \tag{262}$$

It is linear and has on a space where $\mathbf{K}^{-1}$ exists the unique solution

$$h = t. \tag{263}$$

We remark that $\mathbf{K}$ can be invertible (and usually is so the learning problem is well defined) even if $\mathbf{K}_0$ is not invertible. The inverse $\mathbf{K}^{-1}$, necessary to calculate t, is training data dependent and represents the covariance operator/matrix of a Gaussian posterior process. In many practical cases, however, the prior covariance $\mathbf{K}_0^{-1}$ (or in case of a null space a pseudo inverse of $\mathbf{K}_0$) is directly given or can be calculated. Then an inversion of a finite dimensional matrix in data space is sufficient to find the minimum of the energy E_h [228, 76].

Invertible $\mathbf{K}_0$: Let us first study the case of an invertible $\mathbf{K}_0$ and consider the stationarity equation as obtained from (248) or (250)

$$\begin{aligned} 0 &= \sum_{i=1}^{n} \mathbf{K}_i (h - t_i) + \mathbf{K}_0 (h - t_0) && (264) \\ &= \mathbf{K}_D (h - t_D) + \mathbf{K}_0 (h - t_0). && (265) \end{aligned}$$

For existing $\mathbf{K}_0^{-1}$

$$h = t_0 + \mathbf{K}_0^{-1}\mathbf{K}_D(t_D - h), \tag{266}$$

one can introduce

$$a = \mathbf{K}_D(t_D - h), \tag{267}$$

to obtain

$$h = t_0 + \mathbf{K}_0^{-1}a. \tag{268}$$

Inserting Eq. (268) into Eq. (267) one finds an equation for a

$$\left(I + \mathbf{K}_D\mathbf{K}_0^{-1}\right) a = \mathbf{K}_D(t_D - t_0). \tag{269}$$

Multiplying Eq. (269) from the left by the projector $\mathbf{I}_D$ defined in (260) and using

$$\mathbf{K}_D\mathbf{I}_D = \mathbf{I}_D\mathbf{K}_D, \quad a = \mathbf{I}_D a, \quad t_D = \mathbf{I}_D t_D, \tag{270}$$

one obtains an equation in data space

$$\left(I_D + \mathbf{K}_D\mathbf{K}_{0,DD}^{-1}\right) a = \mathbf{K}_D(t_D - t_{0,D}), \tag{271}$$

where

$$\mathbf{K}_{0,DD}^{-1} = (\mathbf{K}_0^{-1})_{DD} = \mathbf{I}_D\mathbf{K}_0^{-1}\mathbf{I}_D \ne (\mathbf{K}_{0,DD})^{-1}, \quad t_{0,D} = \mathbf{I}_D t_0. \tag{272}$$

Thus,

$$a = \mathbf{C}_{DD}\, b, \tag{273}$$

where

$$\mathbf{C}_{DD} = \left(\mathbf{I}_D + \mathbf{K}_D\mathbf{K}_{0,DD}^{-1}\right)^{-1}, \tag{274}$$

and

$$b = \mathbf{K}_D(t_D - t_0). \tag{275}$$

In components Eq. (273) reads,

$$\sum_l \left(\delta_{kl} + n_{x_k}\mathbf{K}_0^{-1}(x_k, x_l)\right) a(x_l) = n_{x_k}\left(t_D(x_k) - t_0(x_k)\right). \tag{276}$$

Having calculated a the solution h is given by Eq. (268)

$$h = t_0 + \mathbf{K}_0^{-1}\mathbf{C}_{DD}b = t_0 + \mathbf{K}_0^{-1}\left(\mathbf{K}_D^{-1} + \mathbf{K}_{0,DD}^{-1}\right)^{-1}(t_D - t_0). \tag{277}$$

Eq. (277) can also be obtained directly from Eq. (263) and the definitions (254), without introducing the auxiliary variable a, using the decomposition $\mathbf{K}_0 t_0 = -\mathbf{K}_D t_0 + (\mathbf{K}_0 + \mathbf{K}_D) t_0$ and

$$\mathbf{K}^{-1}\mathbf{K}_D = \mathbf{K}_0^{-1}\left(\mathbf{I} + \mathbf{K}_D\mathbf{K}_0^{-1}\right)^{-1}\mathbf{K}_D = \mathbf{K}_0^{-1}\sum_{m=0}^{\infty}\left(-\mathbf{K}_D\mathbf{K}_0^{-1}\right)^m\mathbf{K}_D \tag{278}$$

$$= \mathbf{K}_0^{-1}\sum_{m=0}^{\infty}\left(-\mathbf{K}_D\mathbf{I}_D\mathbf{K}_0^{-1}\mathbf{I}_D\right)^m\mathbf{K}_D = \mathbf{K}_0^{-1}\left(\mathbf{I}_D + \mathbf{K}_D\mathbf{K}_{0,DD}^{-1}\right)^{-1}\mathbf{K}_D. \tag{279}$$

$\mathbf{K}_0^{-1}\mathbf{C}_{DD}$ is also known as equivalent kernel due to its relation to kernel smoothing techniques [210, 94, 90, 76].

Interestingly, Eq. (268) still holds for non–quadratic data terms of the form $g_D(h)$ with any differentiable function fulfilling $g(h) = g(h_D)$, where $h_D = \mathbf{I}_D h$ is the restriction of h to data space. Hence, also the function of functional derivatives with respect to $h(x)$ is restricted to data space, i.e., $g'(h_D) = g'_D(h_D)$ with $g'_D = \mathbf{I}_D g'$ and $g'(h,x) = \delta g(h)/\delta h(x)$. For example, $g(h) = \sum_{i=1}^n V(h(x_i) - y_i)$ with V a differentiable function. The finite dimensional vector a is then found by solving a nonlinear equation instead of a linear one [73, 75].

Furthermore, one can study vector fields, i.e., the case where, besides possibly x, also y, and thus $h(x)$, is a vector for given x. (Considering the variable indicating the vector components of y as part of the x–variable, this is a situation where a fixed number of one–dimensional y, corresponding to a subspace of X with fixed dimension, is always measured simultaneously.) In that case the diagonal $\mathbf{K}_i$ of Eq. (246) can be replaced by a version with non–zero off–diagonal elements $\mathbf{K}_{\alpha,\alpha'}$ between the vector components α of y. This corresponds to a multi–dimensional Gaussian data generating probability

$$p(y_i|x_i,h) = \frac{\det \mathbf{K}_i^{\frac{1}{2}}}{(2\pi)^{\frac{k}{2}}} e^{-\frac{1}{2}\sum_{\alpha,\alpha'}(y_{i,\alpha}-h_\alpha(x_i))\mathbf{K}_{i,\alpha,\alpha'}(x_i)(y_{i,\alpha'}-h_{\alpha'}(x_i))}, \tag{280}$$

for k–dimensional vector y_i with components $y_{i,\alpha}$.

Non-invertible $\mathbf{K}_0$: For non–invertible $\mathbf{K}_0$ one can solve for h using the Moore–Penrose inverse $\mathbf{K}_0^{\#}$. Let us first recall some basic facts [58, 164, 186, 187, 126, 15, 120]. A pseudo inverse $\mathbf{B}$ of (a possibly non–square) $\mathbf{A}$ is defined by the conditions

$$\mathbf{BAB} = \mathbf{B}, \quad \mathbf{ABA} = \mathbf{A}, \tag{281}$$

and becomes for real $\mathbf{A}$ the unique Moore–Penrose inverse $\mathbf{A}^{\#}$ if

$$(\mathbf{A}\mathbf{A}^{\#})^T = \mathbf{A}\mathbf{A}^{\#}, \quad (\mathbf{A}^{\#}\mathbf{A})^T = \mathbf{A}^{\#}\mathbf{A}. \tag{282}$$

A linear equation

$$\mathbf{A}x = b \tag{283}$$

is solvable if

$$\mathbf{A}\mathbf{A}^{\#}b = b. \tag{284}$$

In that case the solution is

$$x = \mathbf{A}^{\#}b + x^0 = \mathbf{A}^{\#}b + y - \mathbf{A}^{\#}\mathbf{A}y, \tag{285}$$

where $x^0 = y - \mathbf{A}^{\#}\mathbf{A}y$ is solution of the homogeneous equation $\mathbf{A}x^0 = 0$ and vector y is arbitrary. Hence, x^0 can be expanded in an orthonormalized basis ψ_l of the null space of $\mathbf{A}$

$$x^0 = \sum_l c_l \psi_l. \tag{286}$$

For a diagonal

$$\mathbf{D} = \begin{pmatrix} \widetilde{\mathbf{D}} & 0 \\ 0 & 0 \end{pmatrix}, \tag{287}$$

where $\widetilde{\mathbf{D}}$ is invertible, the Moore–Penrose inverse is

$$\mathbf{D}^{\#} = \begin{pmatrix} \widetilde{\mathbf{D}}^{-1} & 0 \\ 0 & 0 \end{pmatrix}. \tag{288}$$

For a square matrix $\mathbf{D}$ the product

$$\mathbf{D}\mathbf{D}^{\#} = \mathbf{D}^{\#}\mathbf{D} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \mathbf{I}_1^D \tag{289}$$

becomes the projector on the space where $\mathbf{D}$ is invertible. Similarly, $\mathbf{I}_0^D = \mathbf{I} - \mathbf{I}_1^D$ is the projector on the null space of $\mathbf{D}$.

For a general (possible non–square) $\mathbf{A}$ there always exist orthogonal $\mathbf{O_1}$ and $\mathbf{O_2}$ so it can be written in orthogonal normal form $\mathbf{A} = \mathbf{O_1}\mathbf{D}\mathbf{O_2}$, where $\mathbf{D}$ is of the form of Eq. (287) [120]. The corresponding Moore–Penrose inverse is $\mathbf{A} = \mathbf{O_2^{-1}}\mathbf{D}^{\#}\mathbf{O_1^{-1}}$.

Similarly, for an $\mathbf{A}$ which can be diagonalized, i.e., for a square matrix $\mathbf{A} = \mathbf{M}^{-1}\mathbf{D}\mathbf{M}$ with diagonal $\mathbf{D}$, the Moore–Penrose inverse is $\mathbf{A}^{\#} = \mathbf{M}^{-1}\mathbf{D}^{\#}\mathbf{M}$. From Eq. (289) it follows then that

$$\mathbf{A}\mathbf{A}^{\#} = \mathbf{A}^{\#}\mathbf{A} = \mathbf{I}_1 = \mathbf{I} - \mathbf{I}_0. \tag{290}$$

where $\mathbf{I}_1 = \mathbf{M}^{-1}\mathbf{D}\mathbf{D}^{\#}\mathbf{M} = \mathbf{M}^{-1}\mathbf{I}_1^D\mathbf{M}$ projects on the space where $\mathbf{A}$ is invertible and $\mathbf{I}_0 = \mathbf{I} - \mathbf{I}_1 = \sum_l \psi_l\psi_l^T$ is the projector into the null space of $\mathbf{A}$. Then from the definition (281) it follows

$$\mathbf{I}_1\mathbf{A}\mathbf{I}_1 = \mathbf{A}, \qquad \mathbf{I}_1\mathbf{A}^{\#}\mathbf{I}_1 = \mathbf{A}^{\#}. \tag{291}$$

Sometimes it is convenient to write in that case

$$\mathbf{A}^{\#} = (\mathbf{I}_1\mathbf{A}\mathbf{I}_1)^{-1} \tag{292}$$

where the inversion is understood to be performed in the subspace where $\mathbf{A}$ is invertible. Expressed by projectors the solvability condition Eq. (284) becomes

$$\mathbf{I}_1 b = b, \quad \text{i.e.,} \quad \mathbf{I}_0 b = 0, \tag{293}$$

or in terms of a basis ψ_l of the null space

$$(\psi_l,\, b) = 0,\ \forall l. \tag{294}$$

This means that the inhomogeneity b must have no components within the null space of $\mathbf{A}$. Furthermore we can then write

$$x^0 = \mathbf{I}_0\, y, \tag{295}$$

showing that an x^0 can be obtained by projecting an arbitrary y on the null space of $\mathbf{A}$. Collecting the results for an $\mathbf{A}$ which can be diagonalized, Eq. (285) can be expressed in terms of projection operators as follows:

$$x = \mathbf{I}_1\mathbf{A}^{\#}b + \mathbf{I}_0 y = (\mathbf{I}_1\mathbf{A}\mathbf{I}_1)^{-1}\mathbf{I}_1 b + \mathbf{I}_0\, y. \tag{296}$$

Thus a solution x is obtained by inverting $\mathbf{A}$ in the subspace where it is invertible and adding an arbitrary component of the null space.

Now we apply this to Eq. (265) where $\mathbf{K}_0$ is diagonalizable because being positive semi–definite. (In this case $\mathbf{M}$ is an orthogonal matrix and the entries of D are real and larger or equal to zero.) Hence, one obtains under the condition

$$\mathbf{I}_0 \left(\mathbf{K}_0 t_0 + \mathbf{K}_D (t_D - h) \right) = 0, \tag{297}$$

for Eq. (285)

$$h = \mathbf{K}_0^{\#} \left(\mathbf{K}_0 t_0 + \mathbf{K}_D (t_D - h) \right) + h^0, \tag{298}$$

where $\mathbf{K}_0 h^0 = 0$ so that $h^0 = \sum_l c_l \psi_l$ can be expanded in an orthonormalized basis ψ_l of the null space of $\mathbf{K}_0$, assumed here to be of finite dimension. To find an equation in data space define the vector

$$a = \mathbf{K}_D (t_D - h), \tag{299}$$

to get from Eqs.(297) and (298)

$$\begin{aligned} 0 &= (\, \psi_l,\, \mathbf{K}_0 t_0) + (\, \psi_l,\, a),\ \forall l & (300) \\ h &= \mathbf{K}_0^{\#} \left(\mathbf{K}_0 t_0 + a \right) + \sum_l c_l \psi_l. & (301) \end{aligned}$$

These equations have to be solved for a and the coefficients c_l. Inserting Eq. (301) into the definition (299) gives

$$(\mathbf{I} + \mathbf{K}_D \mathbf{K}_0^{\#}) a = \mathbf{K}_D t_D - \mathbf{K}_D \mathbf{I}_1 t_0 - \mathbf{K}_D \sum_l c_l \psi_l, \tag{302}$$

using $\mathbf{K}_0^{\#} \mathbf{K}_0 = \mathbf{I}_1$ according to Eq. (290). Using $a = \mathbf{I}_D\, a$ [where $\mathbf{I}_D$, the projector on the space of training data, is defined in (260)] the solvability condition (297) becomes

$$\sum_{i=1}^{\tilde{n}} \psi_l(x_i) a = -(\, \psi_l,\, \mathbf{K}_0 t_0\,), \forall l, \tag{303}$$

the sum going over different x_i only. Eq. (302) for a and c_l reads in data space, similar to Eq. (271),

$$a = \tilde{\mathbf{C}} \tilde{b}, \tag{304}$$

where $\tilde{\mathbf{C}}^{-1} = \mathbf{I} + \mathbf{K}_D \mathbf{K}_0^{\#}$ has been assumed invertible and $\tilde{b}$ is given by the right hand side of Eq. (302). Inserting into Eq. (301) the solution finally can be written

$$h = \mathbf{I}_1 t_0 + \mathbf{K}_0^{\#} \tilde{\mathbf{C}} \tilde{b} + \sum_l c_l \psi_l. \tag{305}$$

Again, general non–quadratic data terms $g(h_D)$ can be allowed. In that case $\delta g(h_D)/\delta h(x) = g'(h_D, x) = (\mathbf{I}_D g')(h_D, x)$ and Eq. (299) becomes the nonlinear equation

$$a = g'(h_D) = g'\Big(\mathbf{I}_D\left(\mathbf{K}_0^{\#}\left(\mathbf{K}_0 t_0 + \mathbf{K}_D(t_D - h)\right) + h^0\right)\Big). \tag{306}$$

The solution(s) a of that equation have then to be inserted in Eq. (301).

3.7.2 Exact predictive density

For Gaussian regression the predictive density under training data D and prior D_0 can be found analytically without resorting to a saddle point approximation. The predictive density is defined as the h-integral

$$\begin{aligned} p(y|x, D, D_0) &= \int dh\, p(y|x, h)p(h|D, D_0) \\ &= \frac{\int dh\, p(y|x, h)p(y_D|x_D, h)p(h|D_0)}{\int dh\, p(y_D|x_D, h)p(h|D_0)} \\ &= \frac{p(y, y_D|x, x_D, D_0)}{p(y_D|x_D, D_0)}. \end{aligned} \tag{307}$$

Denoting training data values y_i by t_i sampled with inverse covariance $\mathbf{K}_i$ concentrated on x_i and analogously test data values $y = y_{n+1}$ by t_{n+1} sampled with inverse (co–)variance $\mathbf{K}_{n+1}$, we have for $1 \le i \le n+1$

$$p(y_i|x_i, h) = \det(\mathbf{K}_i/2\pi)^{\frac{1}{2}} e^{-\frac{1}{2}\Big(h-t_i,\, \mathbf{K}_i(h-t_i)\Big)}, \tag{308}$$

and

$$p(h|D_0) = \det(\mathbf{K_0}/2\pi)^{\frac{1}{2}} e^{-\frac{1}{2}\Big(h-t_0,\, \mathbf{K}_0(h-t_0)\Big)}, \tag{309}$$

hence

$$p(y|x, D, D_0) = \frac{\int dh\, e^{-\frac{1}{2}\sum_{i=0}^{n+1}\Big(h-t_i,\, \mathbf{K}_i(h-t_i)\Big)+\frac{1}{2}\sum_{i=0}^{n+1}\ln\det_i(\mathbf{K}_i/2\pi)}}{\int dh\, e^{-\frac{1}{2}\sum_{i=0}^{n}\Big(h-t_i,\, \mathbf{K}_i(h-t_i)\Big)+\frac{1}{2}\sum_{i=0}^{n}\ln\det_i(\mathbf{K}_i/2\pi)}}. \tag{310}$$

Here we have this time written explicitly $\det_i(\mathbf{K}_i/2\pi)$ for a determinant calculated in that space where $\mathbf{K}_i$ is invertible. This is useful because for example in general $\det_i \mathbf{K}_i \det \mathbf{K}_0 \neq \det_i \mathbf{K}_i\mathbf{K}_0$. Using the generalized 'bias–variance'–decomposition (230) yields

$$p(y|x,D,D_0) = \frac{\int dh\, e^{-\frac{1}{2}\Big(h-t_+,\,\mathbf{K}_+(h-t_+)\Big)+\frac{n+2}{2}V_+ +\frac{1}{2}\sum_{i=0}^{n+1}\ln\det_i(\mathbf{K}_i/2\pi)}}{\int dh\, e^{-\frac{1}{2}\Big(h-t,\,\mathbf{K}(h-t)\Big)+\frac{n+1}{2}V+\frac{1}{2}\sum_{i=0}^{n}\ln\det_i(\mathbf{K}_i/2\pi)}}, \tag{311}$$

with

$$t = \mathbf{K}^{-1}\sum_{i=0}^{n}\mathbf{K}_i t_i, \qquad \mathbf{K} = \sum_{i=0}^{n}\mathbf{K}_i, \tag{312}$$

$$t_+ = \mathbf{K}_+^{-1}\sum_{i=0}^{n+1}\mathbf{K}_i t_i, \qquad \mathbf{K}_+ = \sum_{i=0}^{n+1}\mathbf{K}_i, \tag{313}$$

$$V = \frac{1}{n+1}\sum_{i=0}^{n}\Big(t_i,\,\mathbf{K}_i t_i\Big) - \Big(t,\,\frac{\mathbf{K}}{n+1}\,t\Big), \tag{314}$$

$$V_+ = \frac{1}{n+2}\sum_{i=0}^{n+1}\Big(t_i,\,\mathbf{K}_i t_i\Big) - \Big(t_+,\,\frac{\mathbf{K}_+}{n+2}\,t_+\Big). \tag{315}$$

Now the h–integration can be performed

$$p(y|x,D,D_0) = \frac{e^{-\frac{n+2}{2}V_+ +\frac{1}{2}\sum_{i=0}^{n+1}\ln\det_i(\mathbf{K}_i/2\pi)-\frac{1}{2}\ln\det(\mathbf{K}_+/2\pi)}}{e^{-\frac{n+1}{2}V+\frac{1}{2}\sum_{i=0}^{n}\ln\det_i(\mathbf{K}_i/2\pi)-\frac{1}{2}\ln\det(\mathbf{K}/2\pi)}} \tag{316}$$

Canceling common factors, writing again y for t_{n+1}, $\mathbf{K}_x$ for $\mathbf{K}_{n+1}$, $\det_x$ for $\det_{n+1}$, and using $\mathbf{K}_+t_+ = \mathbf{K}t+\mathbf{K}_x y$, this becomes

$$p(y|x,D,D_0) = e^{-\frac{1}{2}(y,\mathbf{K}_y\,y)+(y,\mathbf{K}_y\,t)+\frac{1}{2}\Big(t,(\mathbf{K}\mathbf{K}_+^{-1}\mathbf{K}-\mathbf{K})\,t\Big)+\frac{1}{2}\ln\det_x(\mathbf{K}_x\mathbf{K}_+^{-1}\mathbf{K}/2\pi)}. \tag{317}$$

Here we introduced $\mathbf{K}_y = \mathbf{K}_y^T = \mathbf{K}_x - \mathbf{K}_x\mathbf{K}_+^{-1}\mathbf{K}_x$ and used that

$$\det \mathbf{K}^{-1}\mathbf{K}_+ = \det(\mathbf{I}+\mathbf{K}^{-1}\mathbf{K}_x) = \det_x\mathbf{K}^{-1}\mathbf{K}_+ \tag{318}$$

can be calculated in the space of test data x. This follows from $\mathbf{K} = \mathbf{K}_+ - \mathbf{K}_x$ and the equality

$$\det\begin{pmatrix} 1+A & 0 \\ B & 1 \end{pmatrix} = \det(1+A) \tag{319}$$

with $A = \mathbf{I}_x\mathbf{K}^{-1}\mathbf{K}_x$, $B = (\mathbf{I}-\mathbf{I}_x)\mathbf{K}^{-1}\mathbf{K}_x$, and $\mathbf{I}_x$ denoting the projector into the space of test data x. Finally

$$\mathbf{K}_y = \mathbf{K}_x - \mathbf{K}_x\mathbf{K}_+^{-1}\mathbf{K}_x = \mathbf{K}_x\mathbf{K}_+^{-1}\mathbf{K} = (\mathbf{K}-\mathbf{K}\mathbf{K}_+^{-1}\mathbf{K}), \tag{320}$$

yields the correct normalization of the predictive density

$$p(y|x,D,D_0) = e^{-\frac{1}{2}\left(y-\bar{y},\,\mathbf{K}_y(y-\bar{y})\right)+\frac{1}{2}\ln\det_x(\mathbf{K}_y/2\pi)}, \tag{321}$$

with mean and covariance

$$\begin{aligned}
\bar{y} &= t = \mathbf{K}^{-1}\sum_{i=0}^{n}\mathbf{K}_i t_i, &(322)\\
\mathbf{K}_y^{-1} &= \left(\mathbf{K}_x - \mathbf{K}_x\mathbf{K}_+^{-1}\mathbf{K}_x\right)^{-1} = \mathbf{K}_x^{-1} + \mathbf{I}_x\mathbf{K}^{-1}\mathbf{I}_x. &(323)
\end{aligned}$$

It is useful to express the posterior covariance $\mathbf{K}^{-1}$ by the prior covariance $\mathbf{K}_0^{-1}$. According to

$$\begin{pmatrix} 1+A & B \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} (1+A)^{-1} & -(1+A)^{-1}B \\ 0 & 1 \end{pmatrix}, \tag{324}$$

with $A = \mathbf{K}_D\mathbf{K}_{0,DD}^{-1}$, $B = \mathbf{K}_D\mathbf{K}_{0,D\bar{D}}^{-1}$, and $\mathbf{K}_{0,DD}^{-1} = \mathbf{I}_D\mathbf{K}_0^{-1}\mathbf{I}_D$, $\mathbf{K}_{0,D\bar{D}}^{-1} = \mathbf{I}_D\mathbf{K}_0^{-1}\mathbf{I}_{\bar{D}}$, $\mathbf{I}_{\bar{D}} = \mathbf{I} - \mathbf{I}_D$ we find

$$\begin{aligned}
\mathbf{K}^{-1} &= \mathbf{K}_0^{-1}\left(\mathbf{I} + \mathbf{K}_D\mathbf{K}_0^{-1}\right)^{-1} &(325)\\
&= \mathbf{K}_0^{-1}\left(\left(\mathbf{I}_D + \mathbf{K}_D\mathbf{K}_{0,DD}^{-1}\right)^{-1} - \left(\mathbf{I}_D + \mathbf{K}_D\mathbf{K}_{0,DD}^{-1}\right)^{-1}\mathbf{K}_D\mathbf{K}_{0,D\bar{D}}^{-1} + \mathbf{I}_{\bar{D}}\right).
\end{aligned}$$

Notice that while $\mathbf{K}_D^{-1} = (\mathbf{I}_D\mathbf{K}_D\mathbf{I}_D)^{-1}$ in general $\mathbf{K}_{0,DD}^{-1} = \mathbf{I}_D\mathbf{K}_0^{-1}\mathbf{I}_D \neq (\mathbf{I}_D\mathbf{K}_0\mathbf{I}_D)^{-1}$. This means for example that $\mathbf{K}_0^{-1}$ has to be known to find $\mathbf{K}_{0,DD}^{-1}$ and it is not enough to invert $\mathbf{I}_D\mathbf{K}_0\mathbf{I}_D = \mathbf{K}_{0,DD} \neq (\mathbf{K}_{0,DD}^{-1})^{-1}$. In data space $\left(\mathbf{I}_D + \mathbf{K}_D\mathbf{K}_{0,DD}^{-1}\right)^{-1} = \left(\mathbf{K}_D^{-1} + \mathbf{K}_{0,DD}^{-1}\right)^{-1}\mathbf{K}_D^{-1}$, so Eq. (325) can be manipulated to give

$$\mathbf{K}^{-1} = \mathbf{K}_0^{-1}\left(\mathbf{I} - \mathbf{I}_D\left(\mathbf{K}_D^{-1} + \mathbf{K}_{0,DD}^{-1}\right)^{-1}\mathbf{I}_D\mathbf{K}_0^{-1}\right). \tag{326}$$

This allows now to express the predictive mean (322) and covariance (323) by the prior covariance

$$\begin{aligned}
\bar{y} &= t_0 + \mathbf{K}_0^{-1}\left(\mathbf{K}_D^{-1} + \mathbf{K}_{0,DD}^{-1}\right)^{-1}(t_D - t_0), &(327)\\
\mathbf{K}_y^{-1} &= \mathbf{K}_x^{-1} + \mathbf{K}_{0,xx}^{-1} - \mathbf{K}_{0,xD}^{-1}\left(\mathbf{K}_D^{-1} + \mathbf{K}_{0,DD}^{-1}\right)^{-1}\mathbf{K}_{0,Dx}^{-1}. &(328)
\end{aligned}$$

Thus, for given prior covariance $\mathbf{K}_0^{-1}$ both, $\bar{y}$ and $\mathbf{K}_y^{-1}$, can be calculated by inverting the $\tilde{n} \times \tilde{n}$ matrix $\widetilde{\mathbf{K}} = \left(\mathbf{K}_{0,DD}^{-1} + \mathbf{K}_D^{-1}\right)^{-1}$.

Comparison of Eqs.(327,328) with the maximum posterior solution h^* of Eq. (277) now shows that for Gaussian regression the exact predictive density $p(y|x, D, D_0)$ and its maximum posterior approximation $p(y|x, h^*)$ have the same mean

$$t = \int dy\, y\, p(y|x, D, D_0) = \int dy\, y\, p(y|x, h^*). \tag{329}$$

The variances, however, differ by the term $\mathbf{I}_x \mathbf{K}^{-1} \mathbf{I}_x$.

According to the results of Section 2.2.2 the mean of the predictive density is the optimal choice under squared–error loss (51). For Gaussian regression, therefore the optimal regression function $a^*(x)$ is the same for squared–error loss in exact and in maximum posterior treatment and thus also for log–loss (for Gaussian $p(y|x, a)$ with fixed variance)

$$a^*_{\text{MAP,log}} = a^*_{\text{exact,log}} = a^*_{\text{MAP,sq.}} = a^*_{\text{exact,sq.}} = h^* = t. \tag{330}$$

In case the space of possible $p(y|x, a)$ is not restricted to Gaussian densities with fixed variance, the variance of the optimal density under log–loss $p(y|x, a^*_{\text{exact,log}}) = p(y|x, D, D_0)$ differs by $\mathbf{I}_x \mathbf{K}^{-1} \mathbf{I}_x$ from its maximum posterior approximation $p(y|x, a^*_{\text{MAP,log}}) = p(y|x, h^*)$.

3.7.3 Gaussian mixture regression (cluster regression)

Generalizing Gaussian regression the likelihoods may be modeled by a mixture of m Gaussians

$$p(y|x, h) = \frac{\sum_k^m p(k)\, e^{-\frac{\beta}{2}(y - h_k(x))^2}}{\int dy\, \sum_k^m p(k)\, e^{-\frac{\beta}{2}(y - h_k(x))^2}}, \tag{331}$$

where the normalization factor is found as $\sum_k p(k) \left(\frac{\beta}{2\pi}\right)^{\frac{m}{2}}$. Hence, h is here specified by mixing coefficients $p(k)$ and a vector of regression functions $h_k(x)$ specifying the x–dependent location of the kth cluster centroid of the mixture model. A simple prior for $h_k(x)$ is a smoothness prior diagonal in the cluster components. As any density $p(y|x, h)$ can be approximated arbitrarily well by a mixture with large enough m such cluster regression models allows to interpolate between Gaussian regression and more flexible density estimation.

The posterior density becomes for independent data

$$p(h|D, D_0) = \frac{p(h|D_0)}{p(y_D|x_D, D_0)} \prod_i^n \frac{\sum_k^m p(k)\, e^{-\frac{\beta}{2}(y_i - h_k(x_i))^2}}{\sum_k^m p(k)\, \left(\frac{\beta}{2\pi}\right)^{\frac{m}{2}}}. \tag{332}$$

Maximizing that posterior is — for fixed x, uniform $p(k)$ and $p(h|D_0)$ — equivalent to the clustering approach of Rose, Gurewitz, and Fox for squared distance costs [203].

3.7.4 Support vector machines and regression

Expanding the regression function $h(x)$ in a basis of eigenfunctions Ψ_k of $\mathbf{K}_0$

$$K_0 = \sum_k \lambda_k \Psi_k \Psi_k^T, \quad h(x) = \sum_k n_k \Psi_k(x) \tag{333}$$

yields for functional (247)

$$E_h = \sum_i \left(\sum_k n_k \Psi_k(x_i) - y_i \right)^2 + \sum_k \lambda_k |n_k|^2. \tag{334}$$

Under the assumption of output noise for training data the data terms may for example be replaced by the logarithm of a mixture of Gaussians. Such mixture functions with varying mean can develop flat regions where the error is insensitive (robust) to changes of h. Analogously, Gaussians with varying mean can be added to obtain errors which are flat compared to Gaussians for large absolute errors. Similarly to such Gaussian mixtures the mean–square error data term $(y_i - h(x_i))^2$ may be replaced by an ϵ–insensitive error $|y_i - h(x_i)|_\epsilon$, which is zero for absolute errors smaller ϵ and linear for larger absolute errors (see Fig.5). This results in a quadratic programming problem and is equivalent to Vapnik's support vector machine [225, 74, 226, 214, 215, 49]. For a more detailed discussion of the relation between support vector machines and Gaussian processes see [229, 208].

3.8 Classification

In classification (or pattern recognition) tasks the independent visible variable y takes discrete values (group, cluster or pattern labels) [16, 61, 24, 47]. We write $y = k$ and $p(y|x,h) = P_k(x,h)$, i.e., $\sum_k P_k(x,h) = 1$. Having received classification data $D = \{(x_i, k_i) | 1 \le i \le n\}$ the density estimation

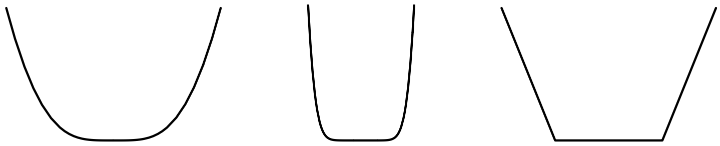

Figure 5: Three robust error functions which are insensitive to small errors. Left: Logarithm of mixture with two Gaussians with equal variance and different means. Middle: Logarithm of mixture with 11 Gaussians with equal variance and different means. Right: ϵ–insensitive error.

error functional for a prior on function ϕ (with components ϕ_k and $P = P(\phi)$) reads

$$E_{\rm cl.} = \sum_i^n \ln P_{k_i}(x_i;\phi) + \frac{1}{2}\Big(\phi - t,\, \mathbf{K}\,(\phi - t)\Big) + (P(\phi), \Lambda_X). \tag{335}$$

In classification the scalar product corresponds to an integral over x and a summation over k, e.g.,

$$\Big(\phi - t,\, \mathbf{K}\,(\phi - t)\Big) = \sum_{k,k'} \int dx\, dx' (\phi_k(x) - t_k(x)) \mathbf{K}_{k,k'}(x,x')(\phi_{k'}(x') - t_{k'}(x')), \tag{336}$$

and $(P, \Lambda_X) = \int dx\, \Lambda_X(x) \sum_k P_k(x)$.

For zero–one loss $l(x,k,a) = \delta_{k,a(x)}$ — a typical loss function for classification problems — the optimal decision (or *Bayes classifier*) is given by the mode of the predictive density (see Section 2.2.2), i.e.,

$$a(x) = \text{argmax}_k\, p(k|x,D,D_0). \tag{337}$$

In saddle point approximation $p(k|x,D,D_0) \approx p(k|x,\phi^*)$ where ϕ^* minimizing $E_{\rm cl.}(\phi)$ can be found by solving the stationarity equation (228).

For the choice $\phi_k = P_k$ non–negativity and normalization must be ensured. For $\phi = L$ with $P = e^L$ non–negativity is automatically fulfilled but the Lagrange multiplier must be included to ensure normalization.

Normalization is guaranteed by using unnormalized probabilities $\phi_k = z_k$, $P = z_k / \sum_l z_l$ (for which non–negativity has to be checked) or shifted log–likelihoods $\phi_k = g_k$ with $g_k = L_k + \ln \sum_l e^{L_l}$, i.e., $P_k = e^{g_k} / \sum_l e^{g_l}$. In that case the nonlocal normalization terms are part of the likelihood and no Lagrange multiplier has to be used [236]. The resulting equation can be solved in the

likelihood $p(y|x,h)$	problem type
of general form	density estimation
discrete y	classification
Gaussian with fixed variance	regression
mixture of Gaussians	clustering
quantum mechanical likelihood	inverse quantum mechanics

Table 3: Special cases of density estimation

space defined by the X–data (see Eq. (153)). The restriction of $\phi_k = g_k$ to linear functions $\phi_k(x) = w_k x + b_k$ yields log–linear models [154]. Recently a mean field theory for Gaussian Process classification has been developed [177, 179].

Table 3 lists some special cases of density estimation. The last line of the table, referring to inverse quantum mechanics, will be discussed in the next section.

3.9 Inverse quantum mechanics

Up to now we have formulated the learning problem in terms of a function ϕ having a simple, e.g., pointwise, relation to P. Nonlocalities in the relation between ϕ and P was only due to the normalization condition, or, working with the distribution function, due to an integration. *Inverse problems for quantum mechanical systems* provide examples of more complicated, nonlocal relations between likelihoods $p(y|x,h) = p(y|x,\phi)$ and the hidden variables ϕ the theory is formulated in. To show the flexibility of Bayesian Field Theory we will give in the following a short introduction to its application to inverse quantum mechanics. A more detailed discussion of inverse quantum problems including numerical applications can be found in [133, 143, 142, 138, 222].

The state of a quantum mechanical systems can be completely described by giving its density operator ρ. The density operator of a specific system depends on its preparation and its Hamiltonian, governing the time evolution of the system. The inverse problem of quantum mechanics consists in the reconstruction of ρ from observational data. Typically, one studies systems with identical preparation but differing Hamiltonians. Consider for example Hamiltonians of the form $\mathbf{H} = \mathbf{T} + \mathbf{V}$, consisting of a kinetic energy part $\mathbf{T}$ and a potential $\mathbf{V}$. Assuming the kinetic energy to be fixed, the inverse problem is that of reconstructing the potential $\mathbf{V}$ from measurements. A local potential $\mathbf{V}(y, y') = V(y)\delta(y-y')$ is specified by a function $V(y)$. Thus, for reconstructing a local potential it is the function $V(y)$ which determines the likelihood $p(y|x, h) = p(y|\mathbf{X}, \rho) = p(y|\mathbf{X}, V) = P(\phi)$ and it is natural to formulate the prior in terms of the function $\phi = V$. The possibilities of implementing prior information for V are similar to those we discuss in this paper for general density estimation problems. It is the likelihood model where inverse quantum mechanics differs from general density estimation.

Measuring quantum systems the variable x corresponds to a hermitian operator $\mathbf{X}$. The possible outcomes y of measurements are given by the eigenvalues of $\mathbf{X}$, i.e.,

$$\mathbf{X}|y>= y|y>, \tag{338}$$

where $|y>$, with dual $<y|$, denotes the eigenfunction with eigenvalue y. (For the sake of simplicity we assume nondegenerate eigenvalues, the generalization to the degenerate case being straightforward.) Defining the projector

$$\Pi_{\mathbf{X},y} = |y><y| \tag{339}$$

the likelihood model of quantum mechanics is given by

$$p(y|x, \rho) = \mathrm{Tr}(\Pi_{\mathbf{X},y}\rho). \tag{340}$$

In the simplest case, where the system is in a pure state, say the ground state φ_0 of $\mathbf{H}$ fulfilling

$$\mathbf{H}|\varphi_0>= E_0|\varphi_0>, \tag{341}$$

the density operator is

$$\rho = \rho^2 = |\varphi_0><\varphi_0|, \tag{342}$$

and the likelihood (340) becomes

$$p(y|x, h) = p(y|\mathbf{X}, \rho) = \mathrm{Tr}(|\varphi_0><\varphi_0|y><y|) = |\varphi_0(y)|^2. \tag{343}$$

Other common choices for ρ are shown in Table 4.

	ρ
general pure state	$\vert\psi><\psi\vert$
stationary pure state	$\vert\varphi_i(\mathbf{H})><\varphi_i(\mathbf{H})\vert$
ground state	$\vert\varphi_0(\mathbf{H})\vert><\varphi_0(\mathbf{H})\vert$
time–dependent pure state	$\vert\mathbf{U}(t,t_0)\psi(t_0)><\mathbf{U}(t,t_0)\psi(t_0)\vert$
scattering	$\lim_{\substack{t\to\infty \\ t_0\to-\infty}} \vert\mathbf{U}(t,t_0)\psi(t_0)><\mathbf{U}(t,t_0)\psi(t_0)\vert$
general mixture state	$\sum_k p(k)\ \vert\psi_k><\psi_k\vert$
stationary mixture state	$\sum_i p(i\vert\mathbf{H})\ \vert\varphi_i(\mathbf{H})><\varphi_i(\mathbf{H})\vert$
canonical ensemble	$(\mathrm{Tr}\, e^{-\beta\mathbf{H}})^{-1}e^{-\beta\mathbf{H}}$

Table 4: The most common examples of density operators for quantum systems. In this table ψ denotes an arbitrary pure state, φ_i represents an eigenstate of Hamiltonian $\mathbf{H}$. The unitary time evolution operator for a time–independent Hamiltonian $\mathbf{H}$ is given by $\mathbf{U} = e^{-i(t-t_0)\mathbf{H}}$. In scattering one imposes typically additional specific boundary conditions on the initial and final states.

In contrast to ideal measurements on classical systems, quantum measurements change the state of the system. Thus, in case one is interested in repeated measurements for the same ρ, that density operator has to be prepared before each measurement. For a stationary state at finite temperature, for example, this can be achieved by waiting until the system is again in thermal equilibrium.

For a Maximum A Posteriori Approximation the functional derivative of the likelihood is needed. Thus, for reconstructing a local potential we have to calculate

$$\delta_{V(y)} p(y_i|\mathbf{X}, V). \tag{344}$$

To be specific, let us assume we measure particle coordinates, meaning we have chosen $\mathbf{X}$ to be the coordinate operator. For a system prepared in the ground state of its Hamiltonian $\mathbf{H}$, we then have to find,

$$\delta_{V(y)} |\varphi_0(y_i)|^2. \tag{345}$$

For that purpose, we take the functional derivative of Eq. (341), which yields

$$(\mathbf{H} - E_0) |\delta_{V(y)} \varphi_0\!> = (\delta_{V(y)} \mathbf{H} - \delta_{V(y)} E_0) |\varphi_0\!> . \tag{346}$$

Projecting from the left by $<\!\varphi_0|$, using again Eq. (341) and the fact that for a local potential $\delta_{V(y)} \mathbf{H}(y', y'') = \delta(y - y')\delta(y' - y'')$, shows that

$$\delta_{V(y)} E_0 = <\!\varphi_0|\delta_{V(y)} \mathbf{H}|\varphi_0\!> = |\varphi_0(y)|^2. \tag{347}$$

Choosing $<\!\varphi_0|\delta_{V(y)}\varphi_0\!> = 0$ and inserting a complete basis of eigenfunctions $|\varphi_j\!>$ of $\mathbf{H}$, we end up with

$$\delta_{V(y)} \varphi_0(y_i) = \sum_{j \ne 0} \frac{1}{E_0 - E_j} \varphi_j(y_i) \varphi_j^*(y) \varphi_0(y). \tag{348}$$

From this the functional derivative of the quantum mechanical log–likelihood (345) corresponding to data point y_i can be obtained easily,

$$\delta_{V(y)} \ln p(y_i|\mathbf{X}, V) = 2\mathrm{Re} \left(\varphi_0(y_i)^{-1} \delta_{V(y)} \varphi_0(y_i) \right). \tag{349}$$

The MAP equations for inverse quantum mechanics are obtained by including the functional derivatives of the prior term for V. In particular, for a Gaussian prior with mean V_0 and inverse covariance $\mathbf{K}_V$, acting in the space of potential functions $V(y)$, its negative logarithm, i.e., its prior error functional, reads

$$\frac{1}{2} \Big(V - V_0,\ \mathbf{K}_V\, (V - V_0) \Big) + \ln Z_V, \tag{350}$$

with Z_V being the V–independent constant normalizing the prior over V. Collecting likelihood and prior terms, the stationarity equation finally becomes

$$0 = \sum_i \delta_{V(y)} \ln p(y_i|\mathbf{X}, V) - \mathbf{K}_V\, (V - V_0). \tag{351}$$

The Bayesian approach to inverse quantum problems is quite flexible and can be used for many different learning scenarios and quantum systems. By adapting Eq. (349), it can deal with measurements of different observables, for example, coordinates, momenta, energies, and with other density operators, describing, for example, time–dependent states or systems at finite temperature [143].

The treatment of bound state or scattering problems for quantum many–body systems requires additional approximations. Common are, for example, mean field methods, for bound state problems [55, 197, 27] as well as for scattering theory [78, 27, 140, 141, 130, 131, 223]. Referring to such mean field methods inverse quantum problems can also be treated for many–body systems [142].

4 Parameterizing likelihoods: Variational methods

4.1 General parameterizations

Approximate solutions of the error minimization problem are obtained by restricting the search (trial) space for $h(x,y) = \phi(x,y)$ (or $h(x)$ in regression). Functions ϕ which are in the considered search space are called *trial functions*. Solving a minimization problem in some restricted trial space is also called a *variational approach* [97, 106, 29, 36, 27]. Clearly, minimal values obtained by minimization within a trial space can only be larger or equal than the true minimal value, and from two variational approximations that with smaller error is the better one.

Alternatively, using parameterized functions ϕ can also be interpreted as implementing the *a priori* information that ϕ is known to have that specific parameterized form. (In cases where ϕ is only known to be approximately of a specific parameterized form, this should ideally be implemented using a prior with a parameterized template and the parameters be treated as hyperparameters as in Section 5.) The following discussion holds for both interpretations.

Any parameterization $\phi = \phi(\{\xi_l\})$ together with a range of allowed values for the parameter vector ξ defines a possible trial space. Hence we consider the error functional

$$E_{\phi(\xi)} = -(\,\ln P(\xi),\; N\,) + \frac{1}{2}(\,\phi(\xi),\; \mathbf{K}\,\phi(\xi)\,) + (\,P(\xi),\; \Lambda_X\,), \tag{352}$$

for ϕ depending on parameters ξ and $p(\xi) = p(\,\phi(\xi)\,)$. In the special case of

Gaussian regression this reads

$$E_{h(\xi)} = \frac{1}{2}(\, h(\xi) - t_D,\, \mathbf{K}_D\, h(\xi) - t_D\,) + \frac{1}{2}(\, h(\xi),\, \mathbf{K}\, h(\xi)\,). \tag{353}$$

Defining the matrix

$$\Phi'(l;x,y) = \frac{\partial \phi(x,y)}{\partial \xi_l} \tag{354}$$

the stationarity equation for the functional (352) becomes

$$0 = \Phi'\mathbf{P}'\mathbf{P}^{-1}N - \Phi'\mathbf{K}\phi - \Phi'\mathbf{P}'\Lambda_X. \tag{355}$$

Similarly, a parameterized functional E_ϕ with non–zero template t as in (226) would give

$$0 = \Phi'\mathbf{P}'\mathbf{P}^{-1}N - \Phi'\mathbf{K}\,(\phi - t) - \Phi'\mathbf{P}'\Lambda_X. \tag{356}$$

To have a convenient notation when solving for Λ_X we introduce

$$\mathbf{P}'_\xi = \Phi'(\xi)\mathbf{P}'(\phi), \tag{357}$$

i.e.,

$$\mathbf{P}'_\xi(l;x,y) = \frac{\partial P(x,y)}{\partial \xi_l} = \int dx'dy'\, \frac{\partial \phi(x',y')}{\partial \xi_l}\frac{\delta P(x,y)}{\delta \phi(x',y')}, \tag{358}$$

and

$$G_{\phi(\xi)} = \mathbf{P}'_\xi\mathbf{P}^{-1}N - \Phi'\mathbf{K}\phi, \tag{359}$$

to obtain for Eq. (355)

$$\mathbf{P}'_\xi\Lambda_X = G_{\phi(\xi)}. \tag{360}$$

For a parameterization ξ restricting the space of possible P the matrix $\mathbf{P}'_\xi$ is not square and cannot be inverted. Thus, let $(\mathbf{P}'_\xi)^\#$ be the Moore–Penrose inverse of $\mathbf{P}'_\xi$, i.e.,

$$(\mathbf{P}'_\xi)^\#\mathbf{P}'_\xi(\mathbf{P}'_\xi)^\# = \mathbf{P}'_\xi, \quad \mathbf{P}'_\xi(\mathbf{P}'_\xi)^\#\mathbf{P}'_\xi = (\mathbf{P}'_\xi)^\#, \tag{361}$$

and symmetric $(\mathbf{P}'_\xi)^\#\mathbf{P}'_\xi$ and $\mathbf{P}'_\xi(\mathbf{P}'_\xi)^\#$. A solution for Λ_X exists if

$$\mathbf{P}'_\xi(\mathbf{P}'_\xi)^\# G_{\phi(\xi)} = G_{\phi(\xi)}. \tag{362}$$

In that case the solution can be written

$$\Lambda_X = (\mathbf{P}'_\xi)^\# G_{\phi(\xi)} + V_\Lambda - (\mathbf{P}'_\xi)^\#\mathbf{P}'_\xi V_\Lambda, \tag{363}$$

with arbitrary vector V_Λ and

$$\Lambda_X^0 = V_\Lambda - (\mathbf{P}'_\xi)^\# \mathbf{P}'_\xi V_\Lambda \tag{364}$$

from the right null space of $\mathbf{P}'_\xi$, representing a solution of

$$\mathbf{P}'_\xi \Lambda_X^0 = 0. \tag{365}$$

Inserting for $\Lambda_X(x) \neq 0$ Eq. (363) into the normalization condition $\Lambda_X = \mathbf{I}_X \mathbf{P} \Lambda_X$ gives

$$\Lambda_X = \mathbf{I}_X \mathbf{P} \left((\mathbf{P}'_\xi)^\# G_{\phi(\xi)} + V_\Lambda - (\mathbf{P}'_\xi)^\# \mathbf{P}'_\xi V_\Lambda \right). \tag{366}$$

Substituting back in Eq. (355) Λ_X is eliminated yielding as stationarity equation

$$0 = \left(\mathbf{I} - \mathbf{P}'_\xi \mathbf{I}_X \mathbf{P} (\mathbf{P}'_\xi)^\# \right) G_{\phi(\xi)} - \mathbf{P}'_\xi \mathbf{I}_X \mathbf{P} \left(V_\Lambda - (\mathbf{P}'_\xi)^\# \mathbf{P}'_\xi V_\Lambda \right), \tag{367}$$

where $G_{\phi(\xi)}$ has to fulfill Eq. (362). Eq. (367) may be written in a form similar to Eq. (193)

$$\mathbf{K}_{\phi(\xi)}(\xi) = T_{\phi(\xi)} \tag{368}$$

with

$$T_{\phi(\xi)}(\xi) = \mathbf{P}'_\xi \mathbf{P}^{-1} N - \mathbf{P}'_\xi \Lambda_X, \tag{369}$$

but with

$$\mathbf{K}_{\phi(\xi)}(\xi) = \Phi' \mathbf{K} \Phi(\xi), \tag{370}$$

being in general a nonlinear operator.

4.2 Gaussian priors for parameters

Up to now we assumed the prior to be given for a function $\phi(\xi)(x, y)$ depending on x and y. Instead of a prior in a function $\phi(\xi)(x, y)$ also a prior in another not (x, y)–dependent function of the parameters $\psi(\xi)$ can be given. A Gaussian prior in $\psi(\xi) = W_\psi \xi$ being a linear function of ξ, results in a prior which is also Gaussian in the parameters ξ, giving a regularization term

$$\frac{1}{2}(\,\xi,\, W_\psi^T \mathbf{K}_\psi W_\psi\, \xi\,) = \frac{1}{2}(\,\xi,\, \mathbf{K}_\xi\, \xi\,), \tag{371}$$

where $\mathbf{K}_\xi = W_\psi^T \mathbf{K}_\psi W_\psi$ is not an operator in a space of functions $\phi(x,y)$ but a matrix in the space of parameters ξ. The results of Section 4.1 apply to this case provided the following replacement is made

$$\Phi' \mathbf{K} \phi \rightarrow \mathbf{K}_\xi \xi. \tag{372}$$

Similarly, a nonlinear ψ requires the replacement

$$\Phi' \mathbf{K} \phi \rightarrow \Psi' \mathbf{K}_\psi \psi, \tag{373}$$

where

$$\Psi'(k,l) = \frac{\partial \psi_l(\xi)}{\partial \xi_k}. \tag{374}$$

Thus, in the general case where a Gaussian (specific) prior in $\phi(\xi)$ and $\psi(\xi)$ is given,

$$\begin{aligned} E_{\phi(\xi),\psi(\xi)} &= -(\ln P(\xi),\, N\,) + (\,P(\xi),\, \Lambda_X\,) \\ &\quad +\frac{1}{2}(\,\phi(\xi),\, \mathbf{K}\,\phi(\xi)\,) + \frac{1}{2}(\,\psi(\xi),\, \mathbf{K}_\psi\,\psi(\xi)\,), \end{aligned} \tag{375}$$

or, including also non–zero template functions (means) t, t_ψ for ϕ and ψ as discussed in Section 3.5,

$$\begin{aligned} E_{\phi(\xi),\psi(\xi)} &= -(\ln P(\xi),\, N\,) + (\,P(\xi),\, \Lambda_X\,) \\ &\quad +\frac{1}{2}(\,\phi(\xi) - t,\, \mathbf{K}\,(\phi(\xi) - t)\,) \\ &\quad +\frac{1}{2}(\,\psi(\xi) - t_\psi,\, \mathbf{K}_\psi\,(\psi(\xi) - t_\psi)\,). \end{aligned} \tag{376}$$

The ϕ and ψ–terms of the energy can be interpreted as corresponding to a probability $p(\xi|t,\mathbf{K},t_\psi,\mathbf{K}_\psi)$, $(\neq p(\xi|t,\mathbf{K})\ p(\xi|t_\psi,\mathbf{K}_\psi))$, or, for example, to $p(t_\psi|\xi,\mathbf{K}_\psi)\ p(\xi|t,\mathbf{K})$ with one of the two terms term corresponding to a Gaussian likelihood with ξ–independent normalization.

The stationarity equation becomes

$$\begin{aligned} 0 &= \mathbf{P}_\xi' \mathbf{P}^{-1} N - \Phi' \mathbf{K}(\phi - t) - \Psi' \mathbf{K}_\psi(\psi - t_\psi) - \mathbf{P}_\xi' \Lambda_X && (377) \\ &= G_{\phi,\psi} - \mathbf{P}_\xi' \Lambda_X, && (378) \end{aligned}$$

which defines $G_{\phi,\psi}$, and for $\Lambda_X \neq 0$

$$\Lambda_X = \mathbf{I}_X \mathbf{P}\left((\mathbf{P}_\xi')^{\#} G_{\phi,\psi} + \Lambda_X^0 \right), \tag{379}$$

for $\mathbf{P}_\xi' \Lambda_X^0 = 0$.

Variable	Error	Stationarity equation	Λ_X
$L(x,y)$	E_L	$\mathbf{K}L = N - \mathbf{e^L}\Lambda_X$	$\mathbf{I}_X\,(N - \mathbf{K}L)$
$P(x,y)$	E_P	$\mathbf{K}P = \mathbf{P}^{-1}N - \Lambda_X$	$\mathbf{I}_X(N - \mathbf{PK}P)$
$\phi = \sqrt{P}$	$E_{\sqrt{P}}$	$\mathbf{K}\phi = 2\Phi^{-1}N - 2\Phi\Lambda_X$	$\mathbf{I}_X(N - \frac{1}{2}\Phi\mathbf{K}\phi)$
$\phi(x,y)$	E_ϕ	$\mathbf{K}\phi = \mathbf{P}'\mathbf{P}^{-1}N - \mathbf{P}'\Lambda_X$	$\mathbf{I}_X\left(N - \mathbf{PP}'^{-1}\mathbf{K}\,\phi\right)$
ξ	$E_{\phi(\xi)}$	$\Phi'\mathbf{K}\phi = \mathbf{P}'_\xi\mathbf{P}^{-1}N - \mathbf{P}'_\xi\Lambda_X$	$\mathbf{I}_X\mathbf{P}\left((\mathbf{P}'_\xi)^{\#}G_{\phi(\xi)} + \Lambda^0_X\right)$
ξ	$E_{\phi(\xi)\psi(\xi)}$	$\Phi'\mathbf{K}(\phi - t) + \Psi'\mathbf{K}_\psi(\psi - t_\psi)$ $= \mathbf{P}'_\xi\mathbf{P}^{-1}N - \mathbf{P}'_\xi\Lambda_X$	$\mathbf{I}_X\mathbf{P}\left((\mathbf{P}'_\xi)^{\#}G_{\phi,\psi} + \Lambda^0_X\right)$

Table 5: Summary of stationarity equations. For notations, conditions and comments see Sections 3.1.1, 3.2.1, 3.3.2, 3.3.1, 4.1 and 4.2.

4.3 Linear trial spaces

Solving a density estimation problem numerically, the function ϕ has to be discretized. This is done by expanding ϕ in a basis B_l (not necessarily orthonormal) and, choosing some $l_{\max}$, truncating the sum to terms with $l \le l_{\max}$,

$$\phi = \sum_{l=1}^{\infty} c_l B_l \rightarrow \phi = \sum_{l=1}^{l_{\max}} c_l B_l. \tag{380}$$

This, also called Ritz's method, corresponds to a finite linear trial space and is equivalent to solving a projected stationarity equation. Using a *discretization* (380) the functional (187) becomes

$$E_{\text{Ritz}} = -(\,\ln P(\phi),\; N\,) + \frac{1}{2}\sum_{kl} c_k c_l(\,B_k,\; \mathbf{K}\,B_l\,) + (\,P(\phi),\; \Lambda_X\,). \tag{381}$$

Solving for the coefficients c_l, $l \le l_{\max}$ to minimize the error results according to Eq.[355) and

$$\Phi'(l;x,y) = B_l(x,y), \tag{382}$$

in

$$0 = (\,B_l,\; \mathbf{P}'\mathbf{P}^{-1}\,N\,) - \sum_k c_k(\,B_l,\; \mathbf{K}\,B_k\,) - (\,B_l,\; \mathbf{P}'\,\Lambda_X\,), \forall l \le l_{\max}, \tag{383}$$

corresponding to the $l_{\max}$–dimensional equation

$$\mathbf{K}_B c = N_B(c) - \Lambda_B(c), \tag{384}$$

with

$$\begin{aligned}
c(l) &= c_l, & (385)\\
\mathbf{K}_B(l,k) &= (\, B_l,\, \mathbf{K}\, B_k\,), & (386)\\
N_B(c)(l) &= (\, B_l,\, \mathbf{P}'(\phi(c))\, \mathbf{P}^{-1}(\phi(c))\, N\,), & (387)\\
\Lambda_B(c)(l) &= (\, B_l,\, \mathbf{P}'(\phi(c))\, \Lambda_X\,). & (388)
\end{aligned}$$

Thus, for an orthonormal basis B_l Eq. (384) corresponds to Eq. (189) projected into the trial space by the projector $\sum_l B_l\, B_l^T$.

The so called *linear models* are obtained by the (very restrictive) choice

$$\phi(z) = \sum_{l=0}^{1} c_l B_l = c_0 + \sum_l c_l z_l \tag{389}$$

with $z = (x, y)$ and $B_0 = 1$ and $B_l = z_l$. Interactions, i.e., terms proportional to products of z–components like $c_{mn} z_m z_n$ can be included. Including all possible interaction would correspond to a multidimensional Taylor expansion of the function $\phi(z)$.

If the functions $B_l(z)$ are also parameterized this leads to mixture models for ϕ. (See Section 4.4.)

4.4 Mixture models

The function $\phi(z)$ can be approximated by a mixture model, i.e., by a linear combination of components functions

$$\phi(z) = \sum c_l B_l(\xi_l, z), \tag{390}$$

with parameter vectors ξ_l and constants c_l (which could also be included into the vector ξ_l) to be adapted. The functions $B_l(\xi_l, z)$ are often chosen to depend on one–dimensional combinations of the vectors ξ_l and z. For example they may depend on some distance $||\xi_l - z||$ ('local or distance approaches') or the projection of z in ξ_l–direction, i.e., $\sum_k \xi_{l,k} z_k$ ('projection approaches'). (For projection approaches see also Sections 4.5, 4.8 and 4.9).

A typical example are Radial Basis Functions (RBF) using Gaussian $B_l(\xi_l, z)$ for which centers (and possibly covariances and also number of components) can be adjusted. Other local methods include k–nearest neighbors methods (kNN) and learning vector quantizations (LVQ) and its variants. (For a comparison see [158].)

4.5 Additive models

Trial functions ϕ may be chosen as sum of simpler functions ϕ_l each depending only on part of the x and y variables. More precisely, we consider functions ϕ_l depending on projections $z_l = \mathbf{I}_l^{(z)} z$ of the vector $z = (x, y)$ of all x and y components. $\mathbf{I}_l^{(z)}$ denotes an projector in the vector space of z (and not in the space of functions $\Phi(x, y)$). Hence, ϕ becomes of the form

$$\phi(z) = \sum_l \phi_l(z_l), \tag{391}$$

so only one–dimensional functions ϕ_l have to be determined. Restricting the functions ϕ_l to a parameterized function space yields a "parameterized additive model"

$$\phi(z) = \sum_l \phi_l(\xi, z_l), \tag{392}$$

which has to be solved for the parameters ξ. The model can also be generalized to a model "additive in parameters ξ_l"

$$\phi(z) = \sum_l \phi_l(\xi_l, x, y), \tag{393}$$

where the functions $\phi_l(\xi_l, x, y)$ are not restricted to one–dimensional functions depending only on projections z_l on the coordinate axes. If the parameters ξ_l determine the component functions ϕ_l completely, this yields just the mixture models of Section 4.4. Another example is projection pursuit, discussed in Section 4.8), where a parameter vector ξ_l corresponds to a projections $\xi_l \cdot z$. In that case even for given ξ_l still a one–dimensional function $\phi_l(\xi_l \cdot z)$ has to be determined.

An ansatz like (391) is made more flexible by including also interactions

$$\phi(x, y) = \sum_l \phi_l(z_l) + \sum_{kl} \phi_{kl}(z_k, z_l) + \sum_{klm} \phi_{klm}(z_k, z_l, z_m) + \cdots. \tag{394}$$

The functions $\phi_{kl\cdots}(z_k, z_l, \cdots)$ can be chosen to depend on product terms like $z_{l,i}z_{k,j}$, or $z_{l,i}z_{k,j}z_{m,n}$, where $z_{l,i}$ denotes one–dimensional sub-variables of z_l.

In additive models in the narrower sense [218, 92, 93, 94] z_l is a subset of x, y components, i.e., $z_l \subseteq \{x_i | 1 \le i \le d_x\} \cup \{y_j | 1 \le j \le d_y\}$, d_x denoting the dimension of x, d_y the dimension of y. In regression, for example, one takes usually the one–element subsets $z_l = \{x_l\}$ for $1 \le l \le d_x$.

In more general schemes the projections of z do not have to be restricted to projections on the coordinates axes. In particular, the projections can be optimized too. For example, one–dimensional projections $\mathbf{I}_l^{(z)} z = w \cdot z$ with $z, w \in X \times Y$ (where $\cdot$ denotes a scalar product in the space of z variables) are used by *ridge approximation* schemes. They include for regression problems one–layer (and similarly multilayer) feedforward neural networks (see Section 4.9) projection pursuit regression (see Section 4.8), ridge regression [151, 152], and hinge functions [31]. For a detailed discussion of the regression case see [76].

The stationarity equation for E_ϕ becomes for the ansatz (391)

$$0 = \mathbf{P}_l' \mathbf{P}^{-1} N - \mathbf{K}\phi - \mathbf{P}_l' \Lambda_X, \tag{395}$$

with

$$\mathbf{P}_l'(z_l, z') = \frac{\delta P(z')}{\delta \phi_l(z_l)}. \tag{396}$$

Considering a density P being also decomposed into components P_l determined by the components ϕ_l

$$P(z) = \sum_l P_l(\phi_l(z_l)), \tag{397}$$

the derivative (396) becomes

$$\mathbf{P}_l'(z_l, z_l') = \frac{\delta P_l(z_l')}{\delta \phi_l(z_l)}, \tag{398}$$

so that specifying an additive prior

$$\frac{1}{2} \sum_{kl} (\, \phi_k - t_k,\, \mathbf{K}_{kl}\, (\phi_l - t_l)\,), \tag{399}$$

the stationary conditions are coupled equations for the component functions ϕ_l which, because $\mathbf{P}$ is diagonal, only contain integrations over z_l–variables

$$0 = \frac{\delta P_l}{\delta \phi_l} \mathbf{P}^{-1} N - \sum_k \mathbf{K}_{lk} (\phi_k - t_k) - \frac{\delta P_l}{\delta \phi_l} \Lambda_X. \tag{400}$$

For the parameterized approach (392) one finds

$$0 = \Phi_l' \mathbf{P}_l' \mathbf{P}^{-1} N - \Phi_l' \mathbf{K} \phi - \Phi_l' \mathbf{P}_l' \Lambda_X, \tag{401}$$

with

$$\Phi_l'(k, z_l) = \frac{\partial \phi_l(z_l)}{\partial \xi_k}. \tag{402}$$

For the ansatz (393) $\Phi_l'(k, z)$ would be restricted to a subset of ξ_k.

4.6 Product ansatz

A product ansatz has the form

$$\phi(z) = \prod_l \phi_l(z_l), \tag{403}$$

where $z_l = \mathbf{I}_l^{(z)} z$ represents projections of the vector z consisting of all x and y components. The ansatz can be made more flexible by using sum of products

$$\phi(z) = \sum_k \prod_l \phi_{k,l}(z_l). \tag{404}$$

The restriction of the trial space to product functions corresponds to the Hartree approximation in physics. (In a Hartree–Fock approximation the product functions are antisymmetrized under coordinate exchange.)

For additive $\mathbf{K} = \sum_l \mathbf{K}_l$ with $\mathbf{K}_l$ acting only on ϕ_l, i.e., $\mathbf{K}_l = \mathbf{K}_l \otimes \left(\bigotimes_{l' \neq l} \mathbf{I}_{l'} \right)$, with $\mathbf{I}_l$ the projector into the space of functions $\phi_l = \mathbf{I}_l \phi$, the quadratic regularization term becomes, assuming $\mathbf{I}_l \, \mathbf{I}_{l'} = \delta_{l,l'}$,

$$(\, \phi,\, \mathbf{K}\, \phi \,) = \sum_l (\, \phi_l,\, \mathbf{K}_l\, \phi_l \,) \prod_{l' \neq l} (\, \phi_{l'},\, \phi_{l'} \,). \tag{405}$$

For $\mathbf{K} = \bigotimes_l \mathbf{K}_l$ with a product structure with respect to ϕ_l

$$(\, \phi,\, \mathbf{K}\, \phi \,) = \prod_l (\, \phi_l,\, \mathbf{K}_l\, \phi_l \,). \tag{406}$$

In both cases the prior term factorizes into lower dimensional contributions.

4.7 Decision trees

Decision trees [32] implement functions which are piecewise constant on rectangular areas parallel to the coordinate axes z_l. Such an approach can be written in tree structure with nodes only performing comparisons of the form $x < a$ or $x > a$ which allows a very effective hardware implementation. Such a piecewise constant approach can be written in the form

$$\phi(z) = \sum_l c_l \prod_k \Theta(z_{\nu(l,k)} - a_{lk}) \tag{407}$$

with step function Θ and $z_{\nu(l,k)}$ indicating the component of z which is compared with the reference value a_{lk}. While there are effective constructive methods to build trees the use of gradient–based minimization or maximization methods would require, for example, to replace the step function by a sigmoid. In particular, decision trees correspond to neural networks at zero temperature, where sigmoids become step functions, and which are restricted to weight vectors in coordinate directions (see Section 4.9).

An overview over different variants of decision trees together with a comparison with rule–based systems, neural networks (see Section 4.9) techniques from applied statistics like linear discriminants, projection pursuit (see Section 4.8) and local methods like for example k-nearest neighbors methods (kNN), Radial Basis Functions (RBF), or learning vector quantization (LVQ) is given in [158].

4.8 Projection pursuit

Projection pursuit models [60, 102, 50] are a generalization of additive models (391) (and a special case of models (393) additive in parameters) where the projections of $z = (x, y)$ are also adapted

$$\phi(z) = \xi_0 + \sum_l \phi_l(\xi_{0,l} + \xi_l \cdot z). \tag{408}$$

For such a model one has to determine one–dimensional 'ridge' functions ϕ_l together with projections defined by vectors ξ_l and constants ξ_0, $\xi_{0,l}$. Adaptive projections may also be used for product approaches

$$\phi(z) = \prod_l \phi_l(\xi_{0,l} + \xi_l \cdot z). \tag{409}$$

Similarly, ϕ may be decomposed into functions depending on distances to adapted reference points (centers). That gives models of the form

$$\phi(z) = \prod_l \phi_l(||\xi_l - z||), \tag{410}$$

which require to adapt parameter vectors (centers) ξ_l and distance functions ϕ_l. For high dimensional spaces the number of centers necessary to cover a high dimensional space with fixed density grows exponentially. Furthermore, as the volume of a high dimensional sphere tends to be concentrated near its surface, the tails become more important in higher dimensions. Thus, typically, projection methods are better suited for high dimensional spaces than distance methods [211].

4.9 Neural networks

While in projection pursuit–like techniques the one–dimensional 'ridge' functions ϕ_l are adapted optimally, neural networks use ridge functions of a fixed sigmoidal form. The resulting lower flexibility following from fixing the ridge function is then compensated by iterating this parameterization. This leads to multilayer neural networks.

Multilayer neural networks have been become a popular tool for regression and classification problems [205, 124, 159, 96, 167, 231, 24, 200, 10]. One-layer neural networks, also known as perceptrons, correspond to the parameterization

$$\phi(z) = \sigma\left(\sum_l w_l z_l - b\right) = \sigma(v), \tag{411}$$

with a sigmoidal function σ, parameters $\xi = w$, projection $v = \sum_l w_l z_l - b$ and z_l single components of the variables x, y, i.e., $z_l = x_l$ for $1 \le l \le d_x$ and $z_l = y_l$ for $d_x + 1 \le l \le d_x + d_y$. (For neural networks with Lorentzians instead of sigmoids see [72].)

Typical choices for the sigmoid are $\sigma(v) = \tanh(\beta v)$ or $\sigma(v) = 1/(1 + e^{-2\beta v})$. The parameter β, often called inverse temperature, controls the sharpness of the step of the sigmoid. In particular, the sigmoid functions become a sharp step in the limit $\beta \to \infty$, i.e., at zero temperature. In principle the sigmoidal function σ may depend on further parameters which then — similar to projection pursuit discussed in Section 4.8 — would also have to be included in the optimization process. The threshold or bias b can be treated as weight if an additional input component is included clamped to the value 1.

A linear combination of perceptrons

$$\phi(x,y) = b + \sum_l W_l \sigma \left(\sum_k w_{lk} z_k - b_k \right), \tag{412}$$

has the form of a projection pursuit approach (408) but with fixed $\phi_l(v) = W_l \sigma(v)$.

In multi–layer networks the parameterization (411) is cascaded,

$$z_{k,i} = \sigma \left(\sum_{l=1}^{m_{i-1}} w_{kl,i} z_{l,i-1} - b_{k,i} \right) = \sigma(v_{k,i}), \tag{413}$$

with $z_{k,i}$ representing the output of the kth node (neuron) in layer i and

$$v_{k,i} = \sum_{l=1}^{m_{i-1}} w_{kl,i} z_{l,i-1} - b_{k,i}, \tag{414}$$

being the input for that node. This yields, skipping the bias terms for simplicity

$$\phi(z,w) = \sigma \left(\sum_{l_{n-1}}^{m_{n-1}} w_{l_{n-1},n} \sigma \left(\sum_{l_{n-2}}^{m_{n-2}} w_{l_{n-1} l_{n-2}, n-1} \cdots \sigma \left(\sum_{l_0}^{m_0} w_{l_1 l_0,1} z_{l_0,0} \right) \cdots \right) \right), \tag{415}$$

beginning with an input layer with $m_0 = d_x + d_y$ nodes (plus possibly nodes to implement the bias) $z_{l,0} = z_l$ and going over intermediate layers with m_i nodes $z_{l,i}$, $0 < i < n$, $1 \le l \le m_i$ to a single node output layer $z_n = \phi(x,y)$.

Commonly neural nets are used in regression and classification to parameterize a function $\phi(x,y) = h(x)$ in functionals

$$E = \sum_i (y_i - h(x_i,w))^2, \tag{416}$$

quadratic in h and without further regularization terms. In that case, regularization has to be assured by using either 1. a neural network architecture which is restrictive enough, 2. by using early stopping like training procedures so the full flexibility of the network structure cannot completely develop and destroy generalization, where in both cases the optimal architecture or algorithm can be determined for example by cross–validation or bootstrap techniques [166, 6, 230, 216, 217, 81, 39, 228, 54], or 3. by averaging over ensembles of networks [170]. In all these cases regularization is implicit in the parameterization of the network. Alternatively, explicit regularization or prior terms can be added to the functional. For regression or classification this is for example done in *learning by hints* [2, 3, 4] or *curvature–driven smoothing* with feedforward networks [22].

One may also remark that from a Frequentist point of view the quadratic functional is not interpreted as posterior but as squared–error loss $\sum_i (y_i - a(x_i, w))^2$ for actions $a(x) = a(x, w)$. According to Section 2.2.2 minimization of error functional (416) for data $\{(x_i, y_i)|1 \le i \le n\}$ sampled under the true density $p(x, y|f)$ yields therefore an empirical estimate for the regression function $\int dy\, y\, p(y|x, f)$.

We consider here neural nets as parameterizations for density estimation with prior (and normalization) terms explicitly included in the functional E_ϕ. In particular, the stationarity equation for functional (352) becomes

$$0 = \Phi'_w \mathbf{P}' \mathbf{P}^{-1} N - \Phi'_w \mathbf{K} \phi - \Phi'_w \mathbf{P}' \Lambda_X, \tag{417}$$

with matrix of derivatives

$$\begin{aligned} \Phi'_w(k, l, i; x, y) &= \frac{\partial \phi(x, y, w)}{\partial w_{kl,i}} \qquad (418) \\ &= \sigma'(v_n) \sum_{l_{n-1}} w_{l_{n-1},n} \sigma'(v_{l_{n-1},n-1}) \sum_{l_{n-2}} w_{l_{n-1} l_{n-2},n-1} \\ &\quad \cdots \sum_{l_{i+1}} w_{l_{i+2} l_{i+1},i+2} \sigma'(v_{l_{i+1},i+1}) w_{l_{i+1} k,i+1} \sigma'(v_{l_i,i}) z_{l,i-1}, \end{aligned}$$

and $\sigma'(v) = d\sigma(v)/dv$. While $\phi(x, y, w)$ is calculated by forward propagating $z = (x, y)$ through the net defined by weight vector w according to Eq. (415) the derivatives Φ' can efficiently be calculated by back–propagation according to Eq. (418). Notice that even for diagonal $\mathbf{P}'$ the derivatives are not needed only at data points but the prior and normalization term require derivatives at all x, y. Thus, in practice terms like $\Phi' \mathbf{K} \phi$ have to be calculated in a relatively poor discretization. Notice, however, that regularization is here not only due to the prior term but follows also from the restrictions implicit in a chosen neural network architecture. In many practical cases a relatively poor discretization of the prior term may thus be sufficient.

Table 6 summarizes the discussed approaches.

Ansatz	Functional form	to be optimized
linear ansatz	$\phi(z) = \sum_l \xi_l B_l(z)$	ξ_l
linear model with interaction	$\phi(z) = \xi_0 + \sum_l \xi_l z_l$ $+ \sum_{mn} \xi_{mn} z_m z_n + \cdots$	$\xi_0,\ \xi_l$ $\xi_{mn}, \cdots$
mixture model	$\phi(z) = \sum \xi_{0,l} B_l(\xi_l, z)$	$\xi_{0,l},\ \xi_l$
additive model with interaction	$\phi(z) = \sum_l \phi_l(z_l)$ $+ \sum_{mn} \phi_{mn}(z_m z_n) + \cdots$	$\phi_l(z_l)$ $\phi_{mn}(z_m z_n), \cdots$
product ansatz	$\phi(z) = \prod_l \phi_l(z_l)$	$\phi_l(z_l)$
decision trees	$\phi(z) = \sum_l \xi_l \prod_k \Theta(z_{\xi_{lk}} - \xi_{0,lk})$	$\xi_l,\ \xi_{0,lk},\ \xi_{lk}$
projection pursuit	$\phi(z) = \xi_0 + \sum_l \phi_l(\xi_{0,l} + \sum_l \xi_l z_l)$	$\phi_l,\ \xi_0,\ \xi_{0,l},\ \xi_l$
neural net (2 lay.)	$\phi(z) = \sigma(\sum_l \xi_l\, \sigma(\sum_k \xi_{lk} z_k))$	$\xi_l,\ \xi_{lk}$

Table 6: Some possible parameterizations.

5 Parameterizing priors: Hyperparameters

5.1 Prior normalization

In Chapter 4. parameterization of ϕ have been studied. This section now discusses parameterizations of the prior density $p(\phi|D_0)$. For Gaussian prior densities that means parameterization of mean and/or covariance. The parameters of the prior functional, which we will denote by θ, are in a Bayesian context also known as *hyperparameters*. Hyperparameters θ can be considered as part of the hidden variables.

In a full Bayesian approach the h–integral therefore has to be completed by an integral over the additional hidden variables θ. Analogously, the prior densities can be supplemented by priors for θ, also be called *hyperpriors*, with corresponding energies E_θ.

In saddle point approximation thus an additional stationarity equation will appear, resulting from the derivative with respect to θ. The saddle point approximation of the θ–integration (in the case of uniform hyperprior $p(\theta)$ and with the h–integral being calculated exactly or by approximation) is also known as ML–II prior [16] or evidence framework [85, 86, 213, 147, 148, 149, 24].

There are some cases where it is convenient to let the likelihood $p(y|x, h)$ depend, besides on a function ϕ, on a few additional parameters. In regression such a parameter can be the variance of the likelihood. Another example is the inverse temperature β introduced in Section 6.3, which, like ϕ also appears in the prior. Such parameters may formally be added to the "direct" hidden variables ϕ yielding an enlarged $\tilde{\phi}$. As those "additional likelihood parameters" are like other hyperparameters typically just real numbers, and not functions like ϕ, they can often be treated analogously to hyperparameters. For example, they may also be determined by cross–validation (see below) or by a low dimensional integration. In contrast to pure prior parameters, however, the functional derivatives with respect to such "additional likelihood parameters" contain terms arising from the derivative of the likelihood.

Within the Frequentist interpretation of error minimization as empirical risk minimization hyperparameters θ can be determined by minimizing the *empirical generalization error* on a new set of test or validation data D_T being independent from the training data D. Here the empirical generalization error is meant to be the pure data term $E_D(\theta) = E_D(\phi^*(\theta))$ of the error functional for ϕ^* being the optimal ϕ for the full regularized $E_\phi(\theta)$ at θ and for given training data D. Elaborated techniques include cross–validation and bootstrap methods which have been mentioned in Sections 2.5 and 4.9.

Within the Bayesian interpretation of error minimization as posterior maximization the introduction of hyperparameters leads to a new difficulty. The problem arises from the fact that it is usually desirable to interpret the error term E_θ as prior energy for θ, meaning that

$$p(\theta) = \frac{e^{-E_\theta}}{Z_\theta}, \tag{419}$$

with normalization

$$Z_\theta = \int d\theta \, e^{-E_\theta}, \tag{420}$$

represents the prior density for θ. Because the joint prior factor for ϕ and θ is given by the product

$$p(\phi, \theta) = p(\phi|\theta)p(\theta), \tag{421}$$

one finds

$$p(\phi|\theta) = \frac{e^{-E(\phi|\theta)}}{Z_\phi(\theta)}. \tag{422}$$

Hence, the ϕ–dependent part of the energy represents a *conditional prior energy* denoted here $E(\phi|\theta)$. As this conditional normalization

$$Z_\phi(\theta) = \int d\phi \, e^{-E(\phi|\theta)}, \tag{423}$$

is in general θ–dependent a normalization term

$$E_N(\theta) = \ln Z_\phi(\theta) \tag{424}$$

must therefore be included in the error functional when minimizing with respect to θ.

It is interesting to look what happens if $p(\phi, \theta)$ of Eq. (419) is expressed in terms of *joint energy* $E(\phi, \theta)$ as follows

$$p(\phi, \theta) = \frac{e^{-E(\phi,\theta)}}{Z_{\phi,\theta}}. \tag{425}$$

Then the joint normalization

$$Z_{\phi,\theta} = \int d\phi \, d\theta \, e^{-E(\phi,\theta)}, \tag{426}$$

is independent of ϕ and θ and could be skipped from the functional. However, in that case the term E_θ cannot easily be related to the prior $p(\theta)$.

Notice especially, that this discussion also applies to the case where E_θ is assumed to be uniform so it does not have to appear explicitly in the error functional. The two ways of expressing $p(\phi, \theta)$ by a joint or conditional energy, respectively, are equivalent if the joint density factorizes. In that case, however, θ and ϕ are independent, so θ cannot be used to parameterize the density of ϕ.

Numerically the need to calculate $Z_\phi(\theta)$ can be disastrous because normalization factors $Z_\phi(\theta)$ represent often an extremely high dimensional (functional) integral and are, in contrast to the normalization of P over y, very difficult to calculate.

There are, however, situations for which $Z_\phi(\theta)$ remains θ–independent. Let $p(\phi, \theta)$ stand for example for a Gaussian specific prior $p(\phi, \theta|\tilde{D}_0)$ (with the normalization condition factored out as in Eq. (90)). Then, because the normalization of a Gaussian is independent of its mean, parameterizing the mean $t = t(\theta)$ results in a θ–independent $Z_\phi(\theta)$.

Besides their mean, Gaussian processes are characterized by their covariance operators $\mathbf{K}^{-1}$. Because the normalization only depends on $\det \mathbf{K}$ a second possibility yielding θ–dependent $Z_\phi(\theta)$ are parameterized transformations of the form $\mathbf{K} \to \mathbf{OKO}^{-1}$ with orthogonal $\mathbf{O} = \mathbf{O}(\theta)$. Indeed, such transformations do not change the determinant $\det \mathbf{K}$. They are only non–trivial for multi–dimensional Gaussians.

For general parameterizations of density estimation problems, however, the normalization term $\ln Z_\phi(\theta)$ must be included. The only way to get rid of that normalization term would be to assume a *compensating hyperprior*

$$p(\theta) \propto Z_\phi(\theta), \tag{427}$$

resulting in an error term $E(\theta) = -\ln Z_\phi(\theta)$ compensating $E_N(\theta)$.

Thus, in the general case we have to consider the functional

$$E_{\theta,\phi} = -(\ln P(\phi),\, N) + (P(\phi),\, \Lambda_X) + E_\phi(\theta) + E_\theta + \ln Z_\phi(\theta). \tag{428}$$

writing $E(\phi|\theta) = E_\phi$ and $E(\theta) = E_\theta$. The stationarity conditions have the form

$$\begin{aligned}
\frac{\delta E_\phi}{\delta \phi} &= \mathbf{P}'(\phi)\mathbf{P}^{-1}(\phi)N - \mathbf{P}'(\phi)\Lambda_X, && (429)\\
\frac{\partial E_\phi}{\partial \theta} &= -\mathbf{Z}' Z_\phi^{-1}(\theta) - E_\theta', && (430)
\end{aligned}$$

with

$$
\mathbf{Z}'(l,k) = \delta(l-k)\frac{\partial Z_\phi(\theta)}{d\theta_l}, \quad E'_\theta(l) = \frac{\partial E_\theta}{\partial \theta_l}. \tag{431}
$$

For compensating hyperprior $E_\theta = -\ln Z_\phi(\theta)$ the right hand side of Eq. (430) vanishes.

Finally, we want to remark that in case function evaluation of $p(\phi,\theta)$ is much cheaper than calculating the gradient (430), minimization methods not using the gradient should be considered, like for example the downhill simplex method [196].

5.2 Adapting prior means

5.2.1 General considerations

A prior mean or template function t represents a prototype, reference function or base line for ϕ. It may be a typical expected pattern in time series prediction or a reference image in image reconstruction. Consider, for example, the task of completing an image ϕ given some pixel values (training data) [137]. Expecting the image to be that of a face the template function t may be chosen to be some prototypical image of a face. We have seen in Section 3.5 that a single template t could be eliminated for Gaussian (specific) priors by solving for $\phi - t$ instead for ϕ. Restricting, however, to only a single template may be a very bad choice. Indeed, faces for example appear on images in many variations, like in different scales, translated, rotated, various illuminations, and other kinds of deformations. We may now describe such variations by a family of templates $t(\theta)$, the parameter θ describing scaling, translations, rotations, and more general deformations. Thus, we expect a function to be similar to only one of the templates $t(\theta)$ and want to implement a (soft, probabilistic) OR, approximating $t(\theta_1)$ OR $t(\theta_2)$ OR $\cdots$ (See also [133, 134, 135, 136]).

A (soft, probabilistic) AND of approximation conditions, on the other hand, is implemented by adding error terms. For example, classical error functionals where data and prior terms are added correspond to an approximation of training data AND *a priori* data.

Similar considerations apply for *model selection*. We could for example expect ϕ to be well approximated by a neural network or a decision tree. In that case $t(\theta)$ spans, for example, a space of neural networks or decision trees. Finally, let us emphasize again that the great advantage and practical feasibility of adaptive templates for regression problems comes from the fact that no additional normalization terms have to be added to the error functional.

5.2.2 Density estimation

The general case with adaptive means for Gaussian prior factors and hyperparameter energy E_θ yields an error functional

$$E_{\theta,\phi} = -(\ln P(\phi),\, N) + \frac{1}{2}\Big(\phi - t(\theta),\, \mathbf{K}\,(\phi - t(\theta))\Big) + (P(\phi),\, \Lambda_X) + E_\theta. \quad (432)$$

Defining

$$\mathbf{t}'(l;x,y) = \frac{\partial t(x,y;\theta)}{\partial \theta_l}, \quad (433)$$

the stationarity equations of (432) obtained from the functional derivatives with respect to ϕ and hyperparameters θ become

$$\mathbf{K}(\phi - t) \;=\; \mathbf{P}'(\phi)\mathbf{P}^{-1}(\phi)N - \mathbf{P}'(\phi)\Lambda_X, \quad (434)$$

$$\mathbf{t}'\mathbf{K}(\phi - t) \;=\; E'_\theta. \quad (435)$$

Inserting Eq. (434) in Eq. (435) gives

$$\mathbf{t}'\mathbf{P}'(\phi)\mathbf{P}^{-1}(\phi)N = \mathbf{t}'\mathbf{P}'(\phi)\Lambda_X + E'_\theta. \quad (436)$$

Eq.(436) becomes equivalent to the parametric stationarity equation (356) with vanishing prior term in the deterministic limit of vanishing prior covariances $\mathbf{K}^{-1}$, i.e., under the assumption $\phi = t(\theta)$, and for vanishing E'_θ. Furthermore, a non–vanishing prior term in (356) can be identified with the term E_θ. This shows, that parametric methods can be considered as deterministic limits of (prior mean) hyperparameter approaches. *In particular, a parametric solution can thus serve as reference template t, to be used within a specific prior factor.* Similarly, such a parametric solution is a natural initial guess for a nonparametric ϕ when solving a stationarity equation by iteration.

If working with parameterized $\phi(\xi)$ extra prior terms Gaussian in some function $\psi(\xi)$ can be included as discussed in Section 4.2. Then, analogously to templates t for ϕ, also parameter templates t_ψ can be made adaptive with hyperparameters θ_ψ. Furthermore, prior terms E_θ and E_{θ_ψ} for the hyperparameters θ, θ_ψ can be added. Including such additional error terms yields

$$\begin{aligned} E_{\theta,\theta_\psi,\phi(\xi),\psi(\xi)} &= -(\ln P(\,\phi(\xi)\,),\, N) + (P(\,\phi(\xi)\,),\, \Lambda_X) \\ &+\frac{1}{2}\Big(\phi(\xi)-t(\theta),\, \mathbf{K}\,(\phi(\xi)-t(\theta))\Big) \\ &+\frac{1}{2}\Big(\psi(\xi)-t_\psi(\theta_\psi),\, \mathbf{K}_\psi\,(\psi(\xi)-t_\psi(\theta_\psi))\Big) \\ &+E_\theta + E_{\theta_\psi}, \end{aligned} \tag{437}$$

and Eqs.(434) and (434) change to

$$\Phi'\mathbf{K}(\phi-t) + \Psi'\mathbf{K}_\psi(\psi-t_\psi) = \mathbf{P}'_\xi\mathbf{P}^{-1}N - \mathbf{P}'_\xi\Lambda_X, \tag{438}$$

$$\mathbf{t}'\mathbf{K}(\phi-t) = E'_\theta, \tag{439}$$

$$\mathbf{t}'_\psi\mathbf{K}_\psi(\psi-t_\psi) = E'_{\theta_\psi}, \tag{440}$$

where $\mathbf{t}'_\psi$, E'_{θ_ψ}, E'_θ , denote derivatives with respect to the parameters θ_ψ or θ, respectively. Parameterizing E_θ and E_{θ_ψ} the process of introducing hyperparameters can be iterated.

5.2.3 Unrestricted variation

To get a first understanding of the approach (432) let us consider the extreme example of completely unrestricted t–variations. In that case the template function $t(x,y)$ itself represents the hyperparameter. (Such function hyperparameters or hyperfields are also discussed in Sect. 5.6.) Then, $\mathbf{t}' = \mathbf{I}$ and Eq. (435) gives $\mathbf{K}(\phi-t) = 0$ (which for invertible $\mathbf{K}$ is solved uniquely by $t=\phi$), resulting according to Eq. (229) in

$$\Lambda_X = N_X. \tag{441}$$

The case of a completely free prior mean t is therefore equivalent to a situation without prior. Indeed, for invertible $\mathbf{P}'$, projection of Eq. (436) into the x–data space by $\mathbf{I}_D$ of Eq. (260) yields

$$P_D = \mathbf{\Lambda}^{-1}_{X,D}N, \tag{442}$$

where $\mathbf{\Lambda}_{X,D} = \mathbf{I}_D\mathbf{\Lambda}_X\mathbf{I}_D$ is invertible and $P_D = \mathbf{I}_D P$. Thus for x_i for which y_i are available

$$P(x_i,y_i) = \frac{N(x_i,y_i)}{N_X(x_i)} \tag{443}$$

is concentrated on the data points. Comparing this with solutions of Eq. (228), for $\theta = t$ fixed, we see that adaptive means tend to lower the influence of prior terms.

5.2.4 Regression

Consider now the case of regression according to functional (247) with an adaptive template $t_0(\theta)$. The system of stationarity equations for the regression function $h(x)$ (corresponding to $\phi(x,y)$) and θ becomes

$$\mathbf{K}_0(h - t_0) = \mathbf{K}_D(t_D - h), \tag{444}$$

$$\mathbf{t}_0'\mathbf{K}_0(h - t_0) = 0. \tag{445}$$

It will also be useful to insert Eq. (444) in Eq. (445), yielding

$$0 = \mathbf{t}_0'\mathbf{K}_D(h - t_D). \tag{446}$$

For fixed t Eq. (444) is solved by the template average t

$$h = t = (\mathbf{K}_0 + \mathbf{K}_D)^{-1} (\mathbf{K}_0 t_0 + \mathbf{K}_D t_D), \tag{447}$$

so that Eq. (445) or Eq. (446), respectively, become

$$0 = \mathbf{t}_0'\mathbf{K}_0(t - t_0), \tag{448}$$

$$0 = \mathbf{t}_0'\mathbf{K}_D(t - t_D). \tag{449}$$

It is now interesting to note that if we replace in Eq. (449) the full template average t by t_0 we get

$$0 = \mathbf{t}_0'\mathbf{K}_D(t_0 - t_D), \tag{450}$$

which is equivalent to the stationarity equation

$$0 = \mathbf{H}'\mathbf{K}_D(h - t_D), \tag{451}$$

(with derivative matrix $\mathbf{H}'$ being the analogue to Φ' for h) of an error functional

$$E_{D,h(\xi)} = \frac{1}{2}(\, h(\xi) - t_D,\, \mathbf{K}_D(h(\xi) - t_D)\,) \tag{452}$$

without prior terms but with parameterized $h(\xi)$, e.g., a neural network. The approximation $h = t = t_0$ can, for example, be interpreted as limit $\lambda \to \infty$,

$$\lim_{\lambda\to\infty} h = \lim_{\lambda\to\infty} t = t_0, \tag{453}$$

after replacing $\mathbf{K}_0$ by $\lambda\mathbf{K}_0$ in Eq. (447). The setting $h = t_0$ can then be used as initial guess h^0 for an iterative solution for h. For existing $\mathbf{K}_0^{-1}$ $h = t_0$ is also obtained after one iteration step of the iteration scheme $h^i = t_0 + \mathbf{K}_0^{-1}\mathbf{K}_D(t_D - h^{i-1})$ starting with initial guess $h^0 = t_D$.

For comparison with Eqs.(449,450,451) we give the stationarity equations for parameters ξ for a parameterized regression functional including an additional prior term with hyperparameters

$$E_{\theta,h(\xi)} = \frac{1}{2}(\, h(\xi) - t_D,\, \mathbf{K}_D(h(\xi) - t_D)\,) + \frac{1}{2}(\, h(\xi) - t_0(\theta),\, \mathbf{K}_0(\theta)(h(\xi) - t_0(\theta))\,), \tag{454}$$

which are

$$0 = \mathbf{H}'\mathbf{K}_D(h - t_D) + \mathbf{H}'\mathbf{K}_0(h - t_0). \tag{455}$$

Let us now compare the various regression functionals we have met up to now. The non–parameterized and regularized regression functional E_h (247) implements prior information explicitly by a regularization term.

A parameterized and regularized functional $E_{h(\xi)}$ of the form (353) corresponds to a functional of the form (454) for θ fixed. It imposes restrictions on the regression function h in two ways, by choosing a specific parameterization and by including an explicit prior term. If the number of data is large enough, compared to the flexibility of the parameterization, the data term of $E_{h(\xi)}$ alone can have a unique minimum. Then, at least technically, no additional prior term would be required. This corresponds to the classical error minimization methods used typically for parametric approaches. Nevertheless, also in such situations the explicit prior term can be useful if it implements useful prior knowledge over h.

The regularized functional with prior– or hyperparameters $E_{\theta,h}$ (432) implements, compared to E_h, effectively weaker prior restrictions. The prior term corresponds to a *soft restriction* of h to the space spanned by the parameterized $t(\theta)$. In the limit where the parameterization of $t(\theta)$ is rich enough to allow $t(\theta^*) = h^*$ at the stationary point the prior term vanishes completely.

The parameterized and regularized functional $E_{\theta,h(\xi)}$ (454), including prior parameters θ, implements prior information explicitly by a regularization term and implicitly by the parameterization of $h(\xi)$. The explicit prior term vanishes if $t(\theta^*) = h(\xi^*)$ at the stationary point. The functional combines a *hard restriction* of h with respect to the space spanned by the parameterization $h(\xi)$ and a *soft restriction* of h with respect to the space spanned by the parameterized $t(\theta)$. Finally, the parameterized and non–regularized functional $E_{D,h(\xi)}$ (452) implements prior information only implicitly by parameterizing $h(\xi)$. In contrast to the functionals $E_{\theta,h}$ and $E_{\theta,h(\xi)}$ it implements only a *hard restriction* for h. The following table summarizes the discussion:

Functional	Eq.	prior implemented
E_h	(247)	explicitly
$E_{h(\xi)}$	(353)	explicitly and implicitly
$E_{\theta,h}$	(432)	explicitly
		no prior for $t(\theta^*) = h^*$
$E_{\theta,h(\xi)}$	(454)	explicitly and implicitly
		no expl. prior for $t(\theta^*) = h(\xi^*)$
$E_{D,h(\xi)}$	(452)	implicitly

5.3 Adapting prior covariances

5.3.1 General case

Parameterizing covariances $\mathbf{K}^{-1}$ is often desirable in practice. It includes for example adapting the trade–off between data and prior terms (i.e., the determination of the regularization factor), the selection between different symmetries, smoothness measures, or in the multidimensional situation the determination of directions with low variance. As far as the normalization depends on $\mathbf{K}(\theta)$ one has to consider the error functional

$$E_{\theta,\phi} = -(\ln P(\phi),\, N) + \frac{1}{2}\Big(\phi - t,\, \mathbf{K}(\theta)\,(\phi - t)\Big) + (P(\phi),\, \Lambda_X) + \ln Z_\phi(\theta) + E_\theta, \tag{456}$$

with

$$Z_\phi(\theta) = (2\pi)^{\frac{d}{2}} (\det \mathbf{K}(\theta))^{-\frac{1}{2}}, \tag{457}$$

for a d–dimensional Gaussian specific prior, and stationarity equations

$$\mathbf{K}(\phi - t) = \mathbf{P}'(\phi)\mathbf{P}^{-1}(\phi)N - \mathbf{P}'(\phi)\Lambda_X, \tag{458}$$

$$\frac{1}{2}\Big(\phi - t,\, \frac{\partial \mathbf{K}(\theta)}{\partial \theta}\,(\phi - t)\Big) = \frac{1}{2}\mathrm{Tr}\left(\mathbf{K}^{-1}(\theta)\frac{\partial \mathbf{K}(\theta)}{\partial \theta}\right) - E'_\theta. \tag{459}$$

Here we used

$$\frac{\partial}{\partial \theta} \ln\det \mathbf{K} = \frac{\partial}{\partial \theta}\mathrm{Tr}\,\ln \mathbf{K} = \mathrm{Tr}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta}\right). \tag{460}$$

In case of an unrestricted variation of the matrix elements of $\mathbf{K}$ the hyperparameters become $\theta_l = \theta(x,y;x',y') = \mathbf{K}(x,y;x',y')$. Then, using

$$\frac{\partial \mathbf{K}(x,y;x',y')}{\partial \theta(x'',y'';x''',y''')} = \delta(x-x'')\delta(y-y'')\delta(x'-x''')\delta(y'-y'''), \tag{461}$$

Eqs.(459) becomes the inhomogeneous equation

$$\frac{1}{2}(\phi - t)\,(\phi - t)^T = \operatorname{Tr}\left(\mathbf{K}^{-1}(\theta)\frac{\partial \mathbf{K}(\theta)}{\partial \theta}\right) - E'_\theta. \tag{462}$$

We will in the sequel consider the two special cases where the determinant of the covariance is θ–independent so that the trace term vanishes, and where θ is just a multiplicative factor for the specific prior energy, i.e., a so called regularization parameter.

5.3.2 Automatic relevance detection

A useful application of hyperparameters is the identification of sensible directions within the space of x and y variables. Consider the general case of an inverse covariance, decomposed into components $\mathbf{K}_0 = \sum_i \theta_i \mathbf{K}_i$. Treating the coefficient vector θ (with components θ_i) as hyperparameter with hyperprior $p(\theta)$ results in a prior energy (error) functional

$$\frac{1}{2}(\phi - t,\ (\sum_i \theta_i \mathbf{K}_i)(\phi - t)\,) - \ln p(\theta) + \ln Z_\phi(\theta). \tag{463}$$

The θ–dependent normalization $\ln Z_\phi(\theta)$ has to be included to obtain the correct stationarity condition for θ. The components $\mathbf{K}_i$ can be the components of a negative Laplacian, for example, $\mathbf{K}_i = -\partial^2_{x_i}$ or $\mathbf{K}_i = -\partial^2_{y_i}$. In that case adapting the hyperparameters means searching for sensible directions in the space of x or y variables. This technique has been called *Automatic Relevance Determination* by MacKay and Neal [170]. The positivity constraint for a can be implemented explicitly, for example by using $\mathbf{K}_0 = \sum_i \theta_i^2 \mathbf{K}_i$ or $\mathbf{K}_0 = \sum_i \exp(\theta_i)\mathbf{K}_i$.

5.3.3 Local masses and gauge theories

The Bayesian analog of a mass term in quantum field theory is a term proportional to the identity matrix $\mathbf{I}$ in the inverse prior covariance $\mathbf{K}_0$. Consider, for example,

$$\mathbf{K}_0 = \theta^2\,\mathbf{I} - \Delta, \tag{464}$$

with θ real (so that $\theta^2 \ge 0$) representing a mass parameter. For large masses ϕ tends to copy the template t locally, and longer range effects of data points following from smoothness requirements become less important. A constant mass can be replaced by a mass function $\theta(x)$. This allows to adapt locally that interplay between "template copying" and smoothness related influence of training data. As hyperprior, one may use a smoothness constraint on the mass function $\theta(x)$, e.g.,

$$\frac{1}{2}(\phi - t,\ \mathbf{M}^2(\phi - t)) - \frac{1}{2}(\phi - t,\ \Delta(\phi - t)) + \lambda\,(\theta,\ \mathbf{K}_\theta \theta) + \ln Z_\phi(\theta), \quad (465)$$

where $\mathbf{M}$ denotes the diagonal mass operator with diagonal elements $\theta(x)$.

Functional hyperparameters like $\theta(x)$ represent, in the language of physicists, additional fields entering the problem (see also Sect. 5.6). There are similarities for example to gauge fields in physics. In particular, a gauge theory–like formalism can be constructed by decomposing $\theta(x) = \sum_i \theta_i(x)$, so that the inverse covariance

$$\mathbf{K}_0 = \sum_i \left(\mathbf{M}_i^2 - \partial_i^2\right) = \sum_i (\mathbf{M}_i + \partial_i)\,(\mathbf{M}_i - \partial_i) = \sum_i D_i^\dagger D_i, \quad (466)$$

can be expressed in terms of a "covariant derivative" $D_i = \partial_i + \theta_i$. Next, one may choose as hyperprior for $\theta_i(x)$

$$\frac{1}{2}\left(\sum_i^{m_x}(\theta_i,\ -\Delta\,\theta_i) - (\sum_i^{m_x}\partial_{x_i}\theta_i,\ \sum_j^{m_x}\partial_{x_j}\theta_j)\right) = \frac{1}{4}\sum_{ij}^{m_x} F_{ij}^2 \quad (467)$$

which can be expressed in terms of a "field strength tensor" (for Abelian fields),

$$F_{ij} = \partial_i\theta_j - \partial_j\theta_i, \quad (468)$$

like, for example, the Maxwell tensor in quantum electrodynamics. (To relate this, as in electrodynamics, to a local $U(1)$ gauge symmetry $\phi \rightarrow e^{i\alpha}\phi$ one can consider complex functions ϕ, with the restriction that their phase cannot be measured.) Notice, that, due to the interpretation of the prior as product $p(\phi|\theta)p(\theta)$, an additional θ–dependent normalization term $\ln Z_\phi(\theta)$ enters the energy functional. Such a term is not present in quantum field theory, where one relates the prior functional directly to $p(\phi, \theta)$, so the norm is independent of ϕ and θ.

5.3.4 Invariant determinants

In this section we discuss parameterizations of the inverse covariance of a Gaussian specific prior which leave the determinant invariant. In that case no θ–dependent normalization factors have to be included which are usually very difficult to calculate. We have to keep in mind, however, that in general a large freedom for $\mathbf{K}(\theta)$ effectively diminishes the influence of the parameterized prior term.

A determinant is, for example, invariant under general similarity transformations, i.e., $\det \tilde{\mathbf{K}} = \det \mathbf{K}$ for $\mathbf{K} \rightarrow \tilde{\mathbf{K}} = \mathbf{OKO}^{-1}$ where $\mathbf{O}$ could be any element of the general linear group. Similarity transformations do not change the eigenvalues, because from $\mathbf{K}\psi = \lambda\psi$ follows $\mathbf{OKO}^{-1}\mathbf{O}\psi = \lambda\mathbf{O}\psi$. Thus, if $\mathbf{K}$ is positive definite also $\tilde{\mathbf{K}}$ is. The additional constraint that $\tilde{\mathbf{K}}$ has to be real symmetric,

$$\tilde{\mathbf{K}} = \tilde{\mathbf{K}}^T = \tilde{\mathbf{K}}^\dagger, \tag{469}$$

requires $\mathbf{O}$ to be real and orthogonal

$$\mathbf{O}^{-1} = \mathbf{O}^T = \mathbf{O}^\dagger. \tag{470}$$

Furthermore, as an overall factor of $\mathbf{O}$ does not change $\tilde{\mathbf{K}}$ one can restrict $\mathbf{O}$ to a special orthogonal group $SO(N)$ with $\det \mathbf{O} = 1$. If $\mathbf{K}$ has degenerate eigenvalues there exist orthogonal transformations with $\mathbf{K} = \tilde{\mathbf{K}}$.

While in one dimension only the identity remains as transformation, the condition of an invariant determinant becomes less restrictive in higher dimensions. Thus, especially for large dimension d of $\mathbf{K}$ (infinite for continuous x) there is a great freedom to adapt inverse covariances without the need to calculate normalization factors, for example to adapt the sensible directions of a multivariate Gaussian.

A positive definite $\mathbf{K}$ can be diagonalized by an orthogonal matrix $\mathbf{O}$ with $\det \mathbf{O} = 1$, i.e., $\mathbf{K} = \mathbf{ODO}^T$. Parameterizing $\mathbf{O}$ the specific prior term becomes

$$\frac{1}{2}\Big(\phi - t,\, \mathbf{K}(\theta)\,(\phi - t)\Big) = \frac{1}{2}\Big(\phi - t,\, \mathbf{O}(\theta)\mathbf{D}\mathbf{O}^T(\theta)\,(\phi - t)\Big), \tag{471}$$

so the stationarity Eq. (459) reads

$$\Big(\phi - t,\, \frac{\partial \mathbf{O}}{\partial \theta}\mathbf{D}\mathbf{O}^T\,(\phi - t)\Big) = -E_\theta^\prime. \tag{472}$$

Matrices $\mathbf{O}$ from $SO(N)$ include rotations and inversion. For a Gaussian specific prior with nondegenerate eigenvalues Eq. (472) allows therefore to adapt the 'sensible' directions of the Gaussian.

There are also transformations which can change eigenvalues, but leave eigenvectors invariant. As example, consider a diagonal matrix $\mathbf{D}$ with diagonal elements (and eigenvalues) $\lambda_i \neq 0$, i.e., $\det \mathbf{D} = \prod_i \lambda_i$. Clearly, any permutation of the eigenvalues λ_i leaves the determinant invariant and transforms a positive definite matrix into a positive definite matrix. Furthermore, one may introduce continuous parameters $\theta_{ij} > 0$ with $i < j$ and transform $\mathbf{D} \to \tilde{\mathbf{D}}$ according to

$$\lambda_i \to \tilde{\lambda}_i = \lambda_i \theta_{ij}, \quad \lambda_j \to \tilde{\lambda}_j = \frac{\lambda_j}{\theta_{ij}}, \tag{473}$$

which leaves the product $\lambda_i \lambda_j = \tilde{\lambda}_i \tilde{\lambda}_j$ and therefore also the determinant invariant and transforms a positive definite matrix into a positive definite matrix. This can be done with every pair of eigenvalues defining a set of continuous parameters θ_{ij} with $i < j$ (θ_{ij} can be completed to a symmetric matrix) leading to

$$\lambda_i \to \tilde{\lambda}_i = \lambda_i \frac{\prod_{j>i} \theta_{ij}}{\prod_{j<i} \theta_{ji}}, \tag{474}$$

which also leaves the determinant invariant

$$\det \tilde{\mathbf{D}} = \prod_i \tilde{\lambda}_i = \prod_i \left(\lambda_i \frac{\prod_{j>i} \theta_{ij}}{\prod_{j<i} \theta_{ji}} \right) = \left(\prod_i \lambda_i \right) \frac{\prod_i \prod_{j>i} \theta_{ij}}{\prod_i \prod_{j<i} \theta_{ji}} = \prod_i \lambda_i = \det \mathbf{D}. \tag{475}$$

A more general transformation with unique parameterization by $\theta_i > 0$, $i \neq i^*$, still leaving the eigenvectors unchanged, would be

$$\tilde{\lambda}_i = \lambda_i \theta_i, \; i \neq i^*; \quad \tilde{\lambda}_{i^*} = \lambda_{i^*} \prod_{i \neq i^*} \theta_i^{-1}. \tag{476}$$

This techniques can be applied to a general positive definite $\mathbf{K}$ after diagonalizing

$$\mathbf{K} = \mathbf{O}\mathbf{D}\mathbf{O}^T \to \tilde{\mathbf{K}} = \mathbf{O}\tilde{\mathbf{D}}\mathbf{O}^T \Rightarrow \det \mathbf{K} = \det \tilde{\mathbf{K}}. \tag{477}$$

As example consider the transformations (474, 476) for which the specific prior term becomes

$$\frac{1}{2}\Big(\phi - t,\, \mathbf{K}(\theta)\,(\phi - t)\Big) = \frac{1}{2}\Big(\phi - t,\, \mathbf{O}\mathbf{D}(\theta)\mathbf{O}^T\,(\phi - t)\Big), \tag{478}$$

and stationarity Eq. (459)

$$\frac{1}{2}\Big(\phi - t,\, \mathbf{O}\frac{\partial \mathbf{D}}{\partial \theta}\mathbf{O}^T\,(\phi - t)\Big) = -E_\theta', \tag{479}$$

and for (474), with $k < l$,

$$\frac{\partial \mathbf{D}(i,j)}{\partial \theta_{kl}} = \delta(i-j)\left(\delta(k-i)\,\lambda_k \frac{\prod_{l\neq n>k}\theta_{kn}}{\prod_{n<k}\theta_{nk}} + \;\delta(l-i)\,\lambda_l \frac{\prod_{n>l}\theta_{ln}}{\prod_{k\neq n<l}\theta_{nl}}\right), \tag{480}$$

or, for (476), with $k \neq i^*$,

$$\frac{\partial \mathbf{D}(i,j)}{\partial \theta_k} = \delta(i-j)\left(\delta(k-i)\,\lambda_k + \;\delta(i-i^*)\,\lambda_{i^*}\frac{1}{\theta_k \prod_{l\neq i^*}\theta_l}\right). \tag{481}$$

If, for example, $\mathbf{K}$ is a translationally invariant operator it is diagonalized in a basis of plane waves. Then also $\tilde{\mathbf{K}}$ is translationally invariant, but its sensitivity to certain frequencies has changed. The optimal sensitivity pattern is determined by the given stationarity equations.

5.3.5 Regularization parameters

Next we consider the example $\mathbf{K}(\gamma) = \gamma\mathbf{K}_0$ where $\theta \geq 0$ has been denoted γ, representing a regularization parameter or an inverse temperature variable for the specific prior. For a d–dimensional Gaussian integral the normalization factor becomes $Z_\phi(\gamma) = (\frac{2\pi}{\gamma})^{\frac{d}{2}}(\det \mathbf{K}_0)^{-1/2}$. For positive (semi–)definite $\mathbf{K}$ the dimension d is given by the rank of $\mathbf{K}$ under a chosen discretization. Skipping constants results in a normalization energy $E_N(\gamma) = -\frac{d}{2}\ln\gamma$. With

$$\frac{\partial \mathbf{K}}{\partial \gamma} = \mathbf{K}_0 \tag{482}$$

we obtain the stationarity equations

$$\gamma\mathbf{K}_0(\phi - t) \;=\; \mathbf{P}'(\phi)\mathbf{P}^{-1}(\phi)N - \mathbf{P}'(\phi)\Lambda_X, \tag{483}$$

$$\frac{1}{2}(\phi - t,\, \mathbf{K}_0\,(\phi - t)) \;=\; \frac{d}{2\,\gamma} - E_\gamma'. \tag{484}$$

For compensating hyperprior the right hand side of Eq. (484) vanishes, giving thus no stationary point for γ. Using however the condition $\gamma \geq 0$ one sees that for positive definite $\mathbf{K}_0$ Eq. (483) is minimized for $\gamma = 0$ corresponding to the 'prior–free' case. For example, in the case of Gaussian regression the solution would be the data template $\phi = h = t_D$. This is also known as "δ–catastrophe". To get a nontrivial solution for γ a noncompensating hyperparameter energy $E_\gamma = E_\theta$ must be used so that $\ln Z_\phi + E_N$ is nonuniform [16, 24].

The other limiting case is a vanishing E'_γ for which Eq. (484) becomes

$$\gamma = \frac{d}{(\phi - t,\, \mathbf{K}_0\,(\phi - t))}. \tag{485}$$

For $\phi \to t$ one sees that $\gamma \to \infty$. Moreover, in case $P[t]$ represents a normalized probability, $\phi = t$ is also a solution of the first stationarity equation (483) in the limit $\gamma \to \infty$. Thus, for vanishing E'_γ the 'data–free' solution $\phi = t$ is a selfconsistent solution of the stationarity equations (483,484).

Fig.6 shows a posterior surface for uniform and for compensating hyperprior for a one–dimensional regression example. The Maximum A Posteriori Approximation corresponds to the highest point of the joint posterior over γ, h in that figures. Alternatively one can treat the γ–integral by Monte–Carlo–methods [236].

Finally we remark that in the setting of empirical risk minimization, due to the different interpretation of the error functional, regularization parameters are usually determined by cross–validation or similar techniques [166, 6, 230, 216, 217, 81, 39, 211, 228, 54, 83].

5.4 Exact posterior for hyperparameters

In the previous sections we have studied saddle point approximations which lead us to maximize the joint posterior $p(h, \theta|D, D_0)$ simultaneously with respect to the hidden variables h and θ

$$p(y|x, D, D_0) = p(y_D|x_D, D_0)^{-1}\!\int\! dh \int\! d\theta\, p(y|x, h)\, \underbrace{p(y_D|x_D, h)p(h|D_0, \theta)p(\theta)}_{\propto p(h,\theta|D,D_0),\ \max \text{w.r.t. } \theta \text{ and h}}, \tag{486}$$

assuming for the maximization with respect to h a slowly varying $p(y|x, h)$ at the stationary point.

This simultaneous maximization with respect to both variables is consistent with the usual asymptotic justification of a saddle point approximation. For example, for a function $f(h, \theta)$ of two (for example, one–dimensional)

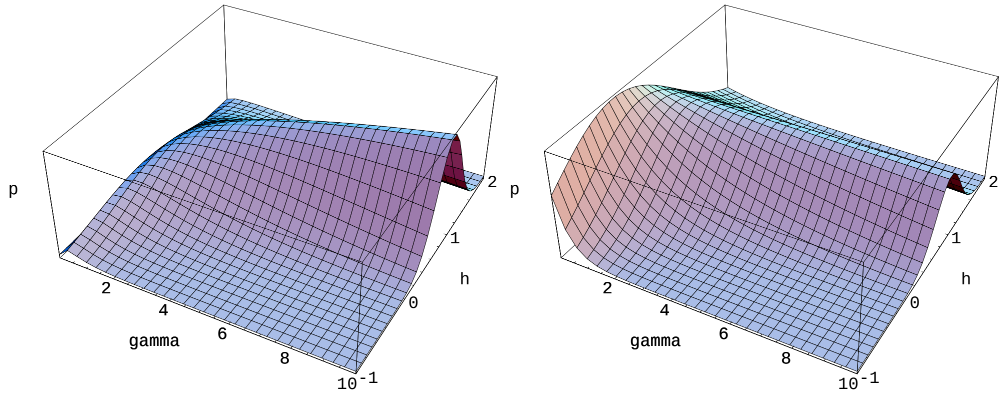

Figure 6: Shown is the joint posterior density of h and γ, i.e., $p(h, \gamma|D, D_0) \propto p(y_D|h)p(h|\gamma, D_0)p(\gamma)$ for a zero–dimensional example of Gaussian regression with training data $y_D = 0$ and prior data $y_{D_0} = 1$. L.h.s: For uniform prior $p(\gamma) \propto 1$ so that the joint posterior becomes $p \propto e^{-\frac{1}{2}h^2 - \frac{\gamma}{2}(h-1)^2 + \frac{1}{2}\ln\gamma}$, having its maximum is at $\gamma = \infty$, $h = 1$. R.h.s.: For compensating hyperprior $p(\gamma) \propto 1/\sqrt{\gamma}$ so that $p \propto e^{-\frac{1}{2}h^2 - \frac{\gamma}{2}(h-1)^2}$ having its maximum is at $\gamma = 0$, $h = 0$.

variables h, θ

$$\int dh\, d\theta\, e^{-\beta f(h,\theta)} \approx e^{-\beta f(h^*,\theta^*) - \frac{1}{2}\ln\det(\beta\mathbf{H}/2\pi)} \tag{487}$$

for large enough β (and a unique maximum). Here $f(h^*, \theta^*)$ denotes the joint minimum and $\mathbf{H}$ the Hessian of f with respect to h and θ. For θ–dependent determinant of the covariance and the usual definition of β, results in a function f of the form $f(h, \theta) = E(h, \theta) + (1/2\beta)\ln\det(\beta\mathbf{K}(\theta)/2\pi)$, where both terms are relevant for the minimization of f with respect to θ. For large β, however, the second term becomes small compared to the first one. (Of course, there is the possibility that a saddle point approximation is not adequate for the θ integration. Also, we have seen that the condition of a positive definite covariance may lead to a solution for θ on the boundary where the (unrestricted) stationarity equation is not fulfilled.)

Alternatively, one might think of performing the two integrals stepwise. This seems especially useful if one integral can be calculated analytically.

Consider, for example

$$\int dh\, d\theta\, e^{-\beta f(h,\theta)} \approx \int d\theta\, e^{-\beta f(\theta, h^*(\theta)) - \frac{1}{2}\ln\det\left(\frac{\beta}{2\pi}\frac{\partial^2 f(h^*(\theta))}{\partial h^2}\right)} \tag{488}$$

which would be exact for a Gaussian h–integral. One sees now that minimizing the complete negative exponent $\beta f(\theta, h^*) + \frac{1}{2}\ln\det(\beta(\partial^2 f/\partial h^2)/2\pi)$ with respect to θ is different from minimizing only f in (487), if the second derivative of f with respect to h depends on θ (which is not the case for a Gaussian θ integral). Again this additional term becomes negligible for large enough β. Thus, at least asymptotically, this term may be altered or even be skipped, and differences in the results of the variants of saddle point approximation will be expected to be small.

Stepwise approaches like (488) can be used, for example to perform Gaussian integrations analytically, and lead to somewhat simpler stationarity equations for θ–dependent covariances [236].

In particular, let us look at the case of Gaussian regression in a bit more detail. The following discussion, however, also applies to density estimation if, as in (488), the Gaussian first step integration is replaced by a saddle point approximation including the normalization factor. (This requires the calculation of the determinant of the Hessian.) Consider the two step procedure for Gaussian regression

$$\begin{aligned} p(y|x, D, D_0) &= p(y_D|x_D, D_0)^{-1} \int d\theta\, \underbrace{p(\theta) \underbrace{\int dh\, p(y|x,h) p(y_D|x_D,h) p(h|D_0,\theta)}_{\text{exact}},}_{p(\theta)p(y,y_D|x,x_D,D_0,\theta)\propto p(y,\theta|x,D,D_0)\ \max \text{ w.r.t. } \theta} \\ &= \int d\theta\, \underbrace{\underbrace{p(\theta|D, D_0)}_{\propto\text{exact}} \underbrace{p(y|x, D, D_0, \theta)}_{\text{exact}}}_{p(y,\theta|x,D,D_0),\ \max \text{ w.r.t. } \theta} \end{aligned} \tag{489}$$

where in a first step $p(y, y_D|x, x_D, D_0, \theta)$ can be calculated analytically and in a second step the θ integral is performed by Gaussian approximation around a stationary point. Instead of maximizing the joint posterior $p(h, \theta|D, D_0)$ with respect to h and θ this approach performs the h–integration analytically and maximizes $p(y, \theta|x, D, D_0)$ with respect to θ. The disadvantage of this approach is the y–, and x–dependency of the resulting solution.

Thus, assuming a slowly varying $p(y|x, D, D_0, \theta)$ at the stationary point it appears simpler to maximize the h–marginalized posterior, $p(\theta|D, D_0) =$

$\int dh\, p(h,\theta|D,D_0)$, if the h–integration can be performed exactly,

$$p(y|x,D,D_0) = \int d\theta \; \underbrace{\underbrace{p(\theta|D,D_0)}_{\text{exact}}}_{\max \text{w.r.t.}\,\theta} \underbrace{p(y|x,D,D_0,\theta)}_{\text{exact}}. \tag{490}$$

Having found a maximum posterior solution θ^* the corresponding analytical solution for $p(y|x,D,D_0,\theta^*)$ is then given by Eq. (321). The posterior density $p(\theta|D,D_0)$ can be obtained from the likelihood of θ and a specified prior $p(\theta)$

$$p(\theta|D,D_0) = \frac{p(y_D|x_D,D_0,\theta)p(\theta)}{p(y_D|x_D,D_0)}. \tag{491}$$

Thus, in case the θ–likelihood can be calculated analytically, the θ–integral is calculated in saddle point approximation by maximizing the posterior for θ with respect to θ. In the case of a uniform $p(\theta)$ the optimal θ^* is obtained by maximizing the θ–likelihood. This corresponds technically to an *empirical Bayes approach* [35]. As h is integrated out in $p(y_D|x_D,D_0,\theta)$ the θ–likelihood is also called *marginalized likelihood*.

Indeed, for Gaussian regression, the θ-likelihood can be integrated analytically. Analogously to Section 3.7.2 one finds [228, 237, 236],

$$\begin{aligned}
p(y_D|x_D,D_0,\theta) &= \int dh\, p(y_D|x_D,h)\, p(h|D_0,\theta) \\
&= \int dh\, e^{-\frac{1}{2}\sum_{i=0}^n \left(h-t_i,\,\mathbf{K}_i(h-t_i)\right)+\frac{1}{2}\sum_{i=0}^n \ln\det_i(\mathbf{K}_i/2\pi)} \\
&= e^{-\frac{1}{2}\sum_{i=0}^n \left(t_i,\,\mathbf{K}_i t_i\right)+\frac{1}{2}\left(t,\,\mathbf{K}t\right)+\frac{1}{2}\ln\det_D(\widetilde{\mathbf{K}}/2\pi)} \\
&= e^{-\frac{1}{2}\left(t_D-t_0,\,\widetilde{\mathbf{K}}(t_D-t_0)\right)+\frac{1}{2}\ln\det_D\widetilde{\mathbf{K}}-\frac{\tilde{n}}{2}\ln(2\pi)} \\
&= e^{-\widetilde{E}+\frac{1}{2}\ln\det_D\widetilde{\mathbf{K}}-\frac{\tilde{n}}{2}\ln(2\pi)},
\end{aligned} \tag{492}$$

where $\widetilde{E} = \frac{1}{2}\Big(t_D - t_0,\, \widetilde{\mathbf{K}}(t_D - t_0)\Big)$, $\widetilde{\mathbf{K}} = (\mathbf{K}_D^{-1} + \mathbf{K}_{0,DD}^{-1}(\theta))^{-1} = \mathbf{K}_D + \mathbf{K}_D\mathbf{K}^{-1}\mathbf{K}_D$, $\det_D$ the determinant in data space, and we used that from $\mathbf{K}_i^{-1}\mathbf{K}_j = \delta_{ij}$ for $i,j>0$ follows $\sum_{i=0}^n (t_i,\,\mathbf{K}_i t_i) = (t_D,\,\mathbf{K}_D t_D) + (t_0,\,\mathbf{K}_0 t_0)$ $= (t_D,\,\mathbf{K}t)$, with $\mathbf{K} = \sum_{i=0}^n \mathbf{K}_i$. In cases where the marginalization over h, necessary to obtain the evidence, cannot be performed analytically and all h–integrals are calculated in saddle point approximation, we get the same result as for a direct simultaneous MAP for h and θ for the predictive density as indicated in (486).

Now we are able to compare the three resulting stationary equations for θ–dependent mean $t_0(\theta)$, inverse covariance $\mathbf{K}_0(\theta)$ and prior $p(\theta)$. Setting the derivative of the joint posterior $p(h,\theta|D,D_0)$ with respect to θ to zero yields

$$
\begin{aligned}
0 \;=\; & \left(\frac{\partial t_0}{\partial \theta},\, \mathbf{K}_0(t_0-h)\right) \;+\frac{1}{2}\Big(h-t_0,\, \frac{\partial \mathbf{K}_0(\theta)}{\partial \theta}\,(h-t_0)\Big) \\
& -\frac{1}{2}\mathrm{Tr}\left(\mathbf{K}_0^{-1}\frac{\partial \mathbf{K}_0}{\partial \theta}\right) - \frac{1}{p(\theta)}\frac{\partial p(\theta)}{\partial \theta}.
\end{aligned}
\tag{493}
$$

This equation which we have already discussed has to be solved simultaneously with the stationarity equation for h. While this approach is easily adapted to general density estimation problems, its difficulty for θ–dependent covariance determinants lies in calculation of the derivative of the determinant of $\mathbf{K}_0$. Maximizing the h–marginalized posterior $p(\theta|D,D_0)$, on the other hand, only requires the calculation of the derivative of the determinant of the $\tilde{n}\times\tilde{n}$ matrix $\widetilde{\mathbf{K}}$

$$
\begin{aligned}
0 \;=\; & \left(\frac{\partial t_0}{\partial \theta},\, \widetilde{\mathbf{K}}(t_0-t_D)\right) \;+\frac{1}{2}\left((t_D-t_0),\, \frac{\partial \widetilde{\mathbf{K}}}{\partial \theta}(t_D-t_0)\right) \\
& -\frac{1}{2}\mathrm{Tr}\left(\widetilde{\mathbf{K}}^{-1}\frac{\partial \widetilde{\mathbf{K}}}{\partial \theta}\right) - \frac{1}{p(\theta)}\frac{\partial p(\theta)}{\partial \theta}.
\end{aligned}
\tag{494}
$$

Evaluated at the stationary $h^* = t_0 + \mathbf{K}_0^{-1}\widetilde{\mathbf{K}}(t_D - t_0)$, the first term of Eq. (493), which does not contain derivatives of the inverse covariances, becomes equal to the first term of Eq. (494). The last terms of Eqs. (493) and (494) are always identical. Typically, the data–independent $\mathbf{K}_0$ has a more regular structure than the data–dependent $\widetilde{\mathbf{K}}$. Thus, at least for one or two dimensional x, a straightforward numerical solution of Eq. (493) by discretizing x can also be a good choice for Gaussian regression problems.

Analogously, from Eq. (321) follows for maximizing $p(y,\theta|x,D,D_0)$ with respect to θ

$$
\begin{aligned}
0 \;=\; & \left(\frac{\partial t}{\partial \theta},\, \mathbf{K}_y(t-y)\right) \;+\frac{1}{2}\left((y-t),\, \frac{\partial \mathbf{K}_y}{\partial \theta}(y-t)\right) \\
& -\frac{1}{2}\mathrm{Tr}\left(\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta}\right) - \frac{1}{p(\theta|D,D_0)}\frac{\partial p(\theta|D,D_0)}{\partial \theta},
\end{aligned}
\tag{495}
$$

which is y–, and x–dependent. Such an approach may be considered if interested only in specific test data x, y.

We may remark that also in Gaussian regression the θ–integral may be quite different from a Gaussian integral, so a saddle point approximation does not necessarily have to give satisfactory results. In cases one encounters problems one can, for example, try variable transformations $\int f(\theta)d\theta = \int \det(\partial\theta/\partial\theta')f(\theta(\theta'))d\theta'$ to obtain a more Gaussian shape of the integrand. Due to the presence of the Jacobian determinant, however, the asymptotic interpretation of the corresponding saddle point approximation is different for the two integrals. The variability of saddle point approximations results from the freedom to add terms which vanish asymptotically but remains finite in the nonasymptotic region. Similar effects are known in quantum many body theory (see for example [172], chapter 7.) Alternatively, the θ–integral can be solved numerically by Monte Carlo methods[237, 236].

5.5 Integer hyperparameters

The hyperparameters θ considered up to now have been real numbers, or vector of real numbers. Such hyperparameters can describe continuous transformations, like the translation, rotation or scaling of template functions and the scaling of inverse covariance operators. For real θ and differentiable posterior, stationarity conditions can be found by differentiating the posterior with respect to θ.

Instead of a class of continuous transformations a finite number of alternative template functions or inverse covariances may be given. For example, an image to be reconstructed might be expected to show a digit between zero and nine, a letter from some alphabet, or the face of someone who is a member of known group of people. Similarly, a particular times series may be expected to be either in a high or in a low variance regime. In all these cases, there exist a finite number of classes i which could be represented by specific templates t_i or inverse covariances $\mathbf{K}_i$. Such "class" variables i are nothing else than hyperparameters θ with integer values.

Binary parameters, for example, allow to select from two reference functions or two inverse covariances that one which fits the data best. E.g., for $i = \theta \in \{0, 1\}$ one can write

$$t(\theta) = (1-\theta)t_1 + \theta t_2, \tag{496}$$

$$\mathbf{K}(\theta) = (1-\theta)\mathbf{K}_1 + \theta\mathbf{K}_2. \tag{497}$$

For integer θ the integral $\int d\theta$ becomes a sum $\sum_\theta$ (we will also use the letter i and write $\sum_i$ for integer hyperparameters), so that prior, posterior, and predictive density have the form of a *finite mixture* with components θ.

For a moderate number of components one may be able to include all of the mixture components. Such prior mixture models will be studied in Section 6.

If the number of mixture components is too large to include them all explicitly, one again must restrict to some of them. One possibility is to select a random sample using Monte–Carlo methods. Alternatively, one may search for the θ^* with maximal posterior. In contrast to typical optimization problems for real variables, the corresponding integer optimization problems are usually not very smooth with respect to θ (with smoothness defined in terms of differences instead of derivatives), and are therefore often much harder to solve.

There exists, however, a variety of deterministic and stochastic integer optimization algorithms, which may be combined with ensemble methods like genetic algorithms [98, 79, 44, 157, 121, 209, 160], and with homotopy methods, like simulated annealing [114, 156, 199, 43, 1, 203, 243, 68, 244, 245]. Annealing methods are similar to (Markov chain) Monte–Carlo methods, which aim in sampling many points from a specific distribution (i.e., for example at fixed temperature). For them it is important to have (nearly) independent samples and the correct limiting distribution of the Markov chain. For annealing methods the aim is to find the correct minimum (i.e., the ground state having zero temperature) by smoothly changing the temperature from a finite value to zero. For them it is less important to model the distribution for nonzero temperatures exactly, but it is important to use an adequate cooling scheme for lowering the temperature.

Instead of an integer optimization problem one may also try to solve a similar problem for real θ. For example, the binary $\theta \in \{0,1\}$ in Eqs. (496) and (497) may be extended to real $\theta \in [0,1]$. By smoothly increasing an appropriate additional hyperprior $p(\theta)$ one can finally enforce again binary hyperparameters $\theta \in \{0,1\}$.

5.6 Local hyperfields

Most, but not all hyperparameters θ considered so far have been real or integer *numbers*, or *vectors* with real or integer components θ_i. With the unrestricted template functions of Sect. 5.2.3 or the functions parameterizing the inverse covariance in Section 5.3.3, we have, however, also already encountered *function hyperparameters* or *hyperfields*. In this section we will now discuss function hyperparameters in more detail.

Functions can be seen as continuous vectors, the function values $\theta(u)$ being the (continuous) analogue of vector components θ_i. In numerical calculations, in particular, functions usually have to be discretized, so, numerically, functions stand for high dimensional vectors.

Typical arguments of function hyperparameters are the independent variables x and, for general density estimation, also the dependent variables y. Such functions $\theta(x)$ or $\theta(x,y)$ will be called *local hyperparameters* or *local hyperfields*. Local hyperfields $\theta(x)$ can be used, for example, to adapt templates or inverse covariances locally. (For general density estimation problems replace here and in the following x by (x,y).)

The price to be paid for the additional flexibility of function hyperparameters is a large number of additional degrees of freedom. This can considerably complicate calculations and, requires a sufficient number of training data and/or a sufficiently restrictive hyperprior to be able to determine the hyperfield and not to make the prior useless.

To introduce local hyperparameters $\theta(x)$ we express real symmetric, positive (semi–)definite inverse covariances by square roots or "filter operators" $\mathbf{W}$, $\mathbf{K} = \mathbf{W}^T\mathbf{W} = \int dx \; W_x W_x^T$ where W_x represents the vector $\mathbf{W}(x,\cdot)$. Thus, in components

$$\mathbf{K}(x,x') = \int dx'' \, \mathbf{W}^T(x,x'')\mathbf{W}(x'',x'), \tag{498}$$

and therefore

$$\begin{aligned}
\Big(\phi - t\,,\,\mathbf{K}(\phi - t)\Big) &= \int dx\, dx' \, [\phi(x) - t(x)] \, \mathbf{K}^T(x,x') \, [\phi(x') - t(x')] \\
&= \int dx\, dx'\, dx'' \, [\phi(x) - t(x)] \mathbf{W}^T(x,x') \\
&\qquad \times \mathbf{W}(x',x'')[\phi(x'') - t(x'')] \\
&= \int dx \, |\omega(x)|^2,
\end{aligned} \tag{499}$$

where we defined the "filtered differences"

$$\omega(x) = (\, W_x\,,\, \phi - t\,) = \int dx' \, \mathbf{W}(x,x')[\phi(x') - t(x')]. \tag{500}$$

Thus, for a Gaussian prior for ϕ we have

$$p(\phi) \propto e^{-\frac{1}{2}\left(\phi - t\,,\,\mathbf{K}(\phi - t)\right)} = e^{-\frac{1}{2}\int dx\, |\omega(x)|^2}. \tag{501}$$

A real local hyperfield $\theta(x)$ mixing, for instance, locally two alternative filtered differences may now be introduced as follows

$$p(\phi|\theta) = e^{-\frac{1}{2}\int dx |\omega(x;\theta)|^2 - \ln Z_\phi(\theta)} = e^{-\frac{1}{2}\int dx\, |[1-\theta(x)]\omega_1(x)+\theta(x)\omega_2(x)|^2 - \ln Z_\phi(\theta)}, \tag{502}$$

where

$$\omega(x;\theta) = [1-\theta(x)]\,\omega_1(x) + \theta(x)\,\omega_2(x), \tag{503}$$

and, say, $\theta(x) \in [0,1]$. For unrestricted real $\theta(x)$ an arbitrary real $\omega(x;\theta)$ can be obtained. For a binary local hyperfield with $\theta(x) \in \{0,1\}$ we have $\theta^2 = \theta$, $(1-\theta)^2 = (1-\theta)$, and $\theta(1-\theta) = 0$, so Eq. (502) becomes

$$p(\phi|\theta) = e^{-\frac{1}{2}\int dx |\omega(x;\theta)|^2 - \ln Z_\phi(\theta)} = e^{-\frac{1}{2}\int dx\, \left([1-\theta(x)]|\omega_1(x)|^2 + \theta(x)|\omega_2(x)|^2\right) - \ln Z_\phi(\theta)}. \tag{504}$$

For real $\theta(x)$ in Eq. (503) terms with $\theta^2(x)$, $[1-\theta(x)]^2$, and $[1-\theta(x)]\theta(x)$ would appear in Eq. (504). A binary θ variable can be obtained from a real θ with the help of a step function $\Theta(x)$ and a threshold ϑ by replacing

$$B_\theta(x) = \Theta(\theta(x) - \vartheta) \to \theta(x). \tag{505}$$

Clearly, if both prior and hyperprior are formulated in terms of such $B_\theta(x)$ this is equivalent to using directly a binary hyperfield.

For a local hyperfield $\theta(x)$ a local adaption of the functions $\omega(x;\theta)$ as in Eq. (503) can be achieved by switching locally between alternative templates or alternative filter operators $\mathbf{W}$

$$t_x(x';\theta) = [1-\theta(x)]\, t_{1,x}(x') + \theta(x)\, t_{2,x}(x'), \tag{506}$$

$$\mathbf{W}(x,x';\theta) = [1-\theta(x)]\, \mathbf{W}_1(x,x') + \theta(x)\, \mathbf{W}_2(x,x'). \tag{507}$$

In Eq. (506) it is important to notice that "local" templates $t_x(x';\theta)$ for fixed x are still functions of an x' variable. Indeed, to obtain $\omega(x;\theta)$, the function t_x is needed for all x' for which $\mathbf{W}$ has nonzero entries,

$$\omega(x;\theta) = \int dx'\, \mathbf{W}(x,x')[\phi(x') - t_x(x';\theta)]. \tag{508}$$

That means that the template is adapted individually for every local filtered difference. Thus, Eq. (506) has to be distinguished from the choice

$$t(x';\theta) = [1-\theta(x')]\, t_1(x') + \theta(x')\, t_2(x'). \tag{509}$$

The unrestricted adaption of templates discussed in Sect. 5.2.3, for instance, can be seen as an approach of the form of Eq. (509) with an unbounded real hyperfield $\theta(x)$.

Eq. (507) corresponds for binary θ to an inverse covariance

$$\mathbf{K}(\theta) = \int\! dx\; \mathbf{K}_x(\theta) = \int\! dx\; \Big([1-\theta(x)] W_{1,x} W_{1,x}^T + \theta(x) W_{2,x} W_{2,x}^T \Big) , \quad (510)$$

where $\mathbf{K}_x(\theta) = W_x(\theta) W_x^T(\theta)$ and $W_{i,x} = \mathbf{W}_i(x,\cdot)$, $W_x(\theta) = \mathbf{W}(x,\cdot\,;\theta)$. We remark that θ–dependent inverse covariances require to include the normalization factors when integrating over θ or solving for the optimal θ in MAP. If we consider two binary hyperfields θ, θ', one for t and one for $\mathbf{W}$, we get a prior

$$p(\phi|\theta,\theta') \propto e^{-\frac{1}{2}\int dx\, \left(\phi - t_x(\theta)\,,\, \mathbf{K}_x(\theta')\,[\phi - t_x(\theta)]\right)} \propto e^{-E(\phi|\theta,\theta')}. \quad (511)$$

Up to a ϕ–independent constant (which still depends on θ, θ') the corresponding prior energy can again be written in the form

$$E(\phi|\theta,\theta') = \frac{1}{2} \Big(\phi - t(\theta,\theta')\,,\, \mathbf{K}(\theta')[\phi - t(\theta,\theta')] \Big). \quad (512)$$

Indeed, the corresponding effective template $t(\theta,\theta')$ and effective inverse covariance $\mathbf{K}(\theta')$ are according to Eqs. (247,252) given by

$$t(\theta,\theta') \;=\; \mathbf{K}(\theta')^{-1} \int\! dx\, \mathbf{K}_x(\theta')\, t_x(\theta), \quad (513)$$

$$\mathbf{K}(\theta') \;=\; \int\! dx\, \mathbf{K}_x(\theta'). \quad (514)$$

Hence, one may rewrite

$$\int\! dx\, |\omega(x;\theta,\theta')|^2 \;=\; \Big(\phi - t(\theta,\theta'),\; \mathbf{K}(\theta')\,[\phi - t(\theta,\theta')]\Big) \quad (515)$$

$$\qquad + \sum_x \Big(t_x(\theta),\; \mathbf{K}_x(\theta')\, t_x(\theta) \Big) - \Big(t(\theta,\theta'),\; \mathbf{K}(\theta')\, t(\theta,\theta') \Big).$$

The MAP solution of Gaussian regression for a prior corresponding to (515) at optimal θ^*, θ'^* is according to Section 3.7 therefore given by

$$\phi^*(\theta^*,\theta'^*) = [\mathbf{K}_D + \mathbf{K}(\theta'^*)]^{-1} \Big(\mathbf{K}_D t_D + \mathbf{K}(\theta'^*)\, t(\theta^*,\theta'^*) \Big) . \quad (516)$$

One may avoid dealing with “local” templates $t_x(\theta)$ by adapting templates in prior terms where $\mathbf{K}$ is equal to the identity $\mathbf{I}$. In that case $(t_0)_x(x';\theta)$ is only needed for $x = x'$ and we may thus directly write $(t_0)_x(x';\theta) = t_0(x';\theta)$. As example, consider the following prior energy, where the θ–dependent template is located in a term with $\mathbf{K} = \mathbf{I}$ and another, say smoothness, prior is added with zero template

$$E(\phi|\theta) = \frac{1}{2}\Big(\phi - t_0(\theta),\, (\phi - t_0(\theta))\Big) + \frac{1}{2}\Big(\phi,\, \mathbf{K}_0\, \phi\Big). \tag{517}$$

Combining both terms yields

$$E(\phi|\theta) = \frac{1}{2}\Big(\Big(\phi - t(\theta),\, \mathbf{K}[\phi - t(\theta)]\Big) + \Big(t_0(\theta),\, \Big(\mathbf{I} - \mathbf{K}^{-1}\Big)\, t_0(\theta)\Big)\Big), \tag{518}$$

with effective template and effective inverse covariance

$$t(\theta) = \mathbf{K}^{-1} t_0(\theta), \quad \mathbf{K} = \mathbf{I} + \mathbf{K}_0. \tag{519}$$

For differential operators $\mathbf{W}$ the effective $t(\theta)$ is thus a smoothed version of $t_0(\theta)$.

The extreme case would be to treat t and $\mathbf{W}$ itself as unrestricted hyperparameters. Notice, however, that increasing flexibility tends to lower the influence of the corresponding prior term. That means, using completely free templates and covariances without introducing additional restricting hyperpriors, just eliminates the corresponding prior term (see Section 5.2.3).

Hence, to restrict the flexibility, typically a smoothness hyperprior may be imposed to prevent highly oscillating functions $\theta(x)$. For real $\theta(x)$, for example, a smoothness prior like $(\theta, -\Delta\theta)$ can be used in regions where it is defined. (The space of ϕ–functions for which a smoothness prior $(\phi - t,\, \mathbf{K}(\phi - t))$ with discontinuous $t(\theta)$ is defined depends on the locations of the discontinuities.) An example of a non–Gaussian hyperprior is,

$$p(\theta) \propto e^{-\frac{\kappa}{2}\int dx\, C_\theta(x)}, \tag{520}$$

where κ is some constant and

$$C_\theta(x) = \Theta\left(\left(\frac{\partial\theta}{\partial x}\right)^2 - \vartheta_\theta\right). \tag{521}$$

is zero at locations where the square of the first derivative is smaller than a certain threshold $0 \le \vartheta_\theta < \infty$, and one otherwise. (The step function Θ is defined as $\Theta(x) = 0$ for $x \le 0$ and $\Theta(x) = 1$ for $0 < x \le \infty$.) To enable differentiation the step function Θ could be replaced by a sigmoidal function. For discrete x one can analogously count the number of jumps larger than a given threshold. Similarly, one may penalize the number $N_d(\theta)$ of discontinuities where $\left(\frac{\partial\theta}{\partial x}\right)^2 = \infty$ and use

$$p(\theta) \propto e^{-\frac{\kappa}{2}N_d(\theta)}. \tag{522}$$

In the case of a binary field this corresponds to counting the number of times the field changes its value.

The expression C_θ of Eq. (521) can be generalized to

$$C_\theta(x) = \Theta\left(|\omega_\theta(x)|^2 - \vartheta_\theta\right), \tag{523}$$

where, analogously to Eq. (500),

$$\omega_\theta(x) = \int dx'\,\mathbf{W}_\theta(x,x')[\theta(x') - t_\theta(x')], \tag{524}$$

and $\mathbf{W}_\theta$ is some filter operator acting on the hyperfield and $t_\theta(x')$ is a template for the hyperfield.

Discontinuous functions ϕ can either be approximated by using discontinuous templates $t(x;\theta)$ or by eliminating matrix elements of the inverse covariance which connect the two sides of the discontinuity. For example, consider the discrete version of a negative Laplacian with periodic boundary conditions,

$$\mathbf{K} = \mathbf{W}^T\mathbf{W} = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}, \tag{525}$$

and possible square root,

$$\mathbf{W} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}. \tag{526}$$

The first three points can be disconnected from the last three points by setting $\mathbf{W}(3,\cdot)$ and $\mathbf{W}(6,\cdot)$ to zero, namely,

$$\tilde{\mathbf{W}} = \left(\begin{array}{ccc|ccc} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right). \tag{527}$$

so that the smoothness prior with inverse covariance

$$\tilde{\mathbf{K}} = \tilde{\mathbf{W}}^T\tilde{\mathbf{W}} = \left(\begin{array}{ccc|ccc} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{array}\right), \tag{528}$$

is ineffective between points from different regions, In contrast to using discontinuous templates, the height of the jump at the discontinuity has not to be given in advance when using such disconnected Laplacians (or other inverse covariances). On the other hand training data are then required for all separated regions to determine the free constants which correspond to the zero modes of the Laplacian.

Non–Gaussian priors, which will be discussed in more detail in the next Section, often provide an alternative to the use of function hyperparameters. Similarly to Eq. (521) one may for example define a binary function $B(x)$ in terms of ϕ,

$$B(x) = \Theta\left(|\omega_1(x)|^2 - |\omega_2(x)|^2 - \vartheta\right), \tag{529}$$

like, for a negative Laplacian prior,

$$B(x) = \Theta\left(\left|\frac{\partial(\phi - t_1)}{\partial x}\right|^2 - \left|\frac{\partial(\phi - t_2)}{\partial x}\right|^2 - \vartheta\right). \tag{530}$$

Here $B(x)$ is directly determined by ϕ and is not considered as an independent hyperfield. Notice also that the functions $\omega_i(x)$ and $B(x)$ may be nonlocal with respect to $\phi(x)$, meaning they may depend on more than one $\phi(x)$ value.

The threshold ϑ has to be related to the prior expectations on ω_i. A possible non–Gaussian prior for ϕ formulated in terms of B can be,

$$p(\phi) \propto e^{-\frac{1}{2}\int dx\,\left(|\omega_1(x)|^2(1-B(x))+|\omega_2(x)|^2 B(x)-\frac{\kappa}{2}N_d(B)\right)}, \tag{531}$$

with $N_d(B)$ counting the number of discontinuities of $B(x)$. Alternatively to N_d one may for a real B define, similarly to (523),

$$C(x) = \Theta\left(|\omega_B(x)|^2 - \vartheta_B\right), \tag{532}$$

with

$$\omega_B(x) = \int dx'\,\mathbf{W}_B(x,x')[B(x') - t_B(x')], \tag{533}$$

and some filter operator $\mathbf{W}_B$ and template t_B. Similarly to the introduction of hyperparameters, one can treat $B(x)$ formally as an independent function by including a term $\lambda\left(B(x) - \Theta\left(|\omega_1(x)|^2 - |\omega_2(x)|^2 - \vartheta\right)\right)$ in the prior energy and taking the limit $\lambda \to \infty$.

Eq. (531) looks similar to the combination of the prior (504) with the hyperprior (522),

$$p(\phi,\theta) \propto e^{-\frac{1}{2}\int dx\,\left(|\omega_1(x)|^2(1-B_\theta(x))+|\omega_2(x)|^2 B_\theta(x)-\frac{\kappa}{2}N_d(B_\theta)-\ln Z_\phi(\theta)\right)}. \tag{534}$$

Notice, however, that the definition (505) of the hyperfield B_θ (and $N_d(B_\theta)$ or C_θ, respectively), is different from that of B (and $N_d(B)$ or C), which are direct functions of ϕ. If the ω_i differ only in their templates, the normalization term can be skipped. Then, identifying B_θ in (534) with a binary θ and assuming $\vartheta = 0$, $\vartheta_\theta = \vartheta_B$, $\mathbf{W}_\theta = \mathbf{W}_B$, the two equations are equivalent for $\theta = \Theta\left(|\omega_1(x)|^2 - |\omega_2(x)|^2\right)$. In the absence of hyperpriors, it is indeed easily seen that this is a selfconsistent solution for θ, given ϕ. In general, however, when hyperpriors are included, another solution for θ may have a larger posterior. Non–Gaussian priors will be discussed in Section 6.5.

Hyperpriors or non–Gaussian prior terms are useful to enforce specific global constraints for $\theta(x)$ or $B(x)$. In images, for example, discontinuities are expected to form closed curves. Hyperpriors, organizing discontinuities along lines or closed curves, are thus important for image segmentation [70, 153, 66, 67, 238, 247].

6 Non–Gaussian prior factors

6.1 Mixtures of Gaussian prior factors

Complex, non–Gaussian prior factors, for example being multimodal, may be constructed or approximated by using mixtures of simpler prior components. In particular, it is convenient to use as components or "building blocks" Gaussian densities, as then many useful results obtained for Gaussian processes survive the generalization to mixture models [132, 133, 134, 135, 136]. We will therefore in the following discuss applications of mixtures of Gaussian priors. Other implementations of non–Gaussian priors will be discussed in Section 6.5.

In Section 5.1 we have seen that hyperparameters label components of mixture densities. Thus, if j labels the components of a mixture model, then j can be seen as hyperparameter. In Section 5 we have treated the corresponding hyperparameter integration completely in saddle point approximation. In this section we will assume the hyperparameters j to be discrete and try to calculate the corresponding summation exactly.

Hence, consider a discrete hyperparameter j, possibly in addition to continuous hyperparameters θ. In contrast to the θ–integral we aim now in treating the analogous sum over j exactly, i.e., we want to study *mixture models*

$$p(\phi, \theta|\tilde{D}_0) = \sum_j^m p(\phi, \theta, j|\tilde{D}_0) = \sum_j^m p(\phi|\tilde{D}_0, \theta, j)p(\theta, j). \tag{535}$$

In the following we concentrate on *mixtures of Gaussian specific priors*. Notice that such models do *not* correspond to Gaussian mixture models for ϕ as they are often used in density estimation. Indeed, the form of ϕ may be completely unrestricted, it is only its prior or posterior density which is modeled by a mixture. We also remark that a strict asymptotical justification of a saddle point approximation would require the introduction of a parameter $\tilde{\beta}$ so that $p(\phi, \theta|\tilde{D}_0) \propto e^{\tilde{\beta} \ln \sum_j p_j}$. If the sum is reduced to a single term, then $\tilde{\beta}$ corresponds to β.

We already discussed shortly in Section 5.2 that, in contrast to a product of probabilities or a sum of error terms implementing a probabilistic AND of approximation conditions, a sum over j implements a probabilistic OR. Those alternative approximation conditions will in the sequel be represented by alternative templates t_j and inverse covariances $\mathbf{K}_j$. A prior (or posterior) density in form of a probabilistic OR means that the optimal solution does not necessarily have to approximate all but possibly only one of the t_j (in a metric defined by $\mathbf{K}_j$). For example, we may expect in an image reconstruction task blue or brown eyes whereas a mixture between blue and brown might not be as likely. Prior mixture models are potentially useful for

1. Ambiguous (prior) data. Alternative templates can for example represent different expected trends for a time series.

2. Model selection. Here templates represent alternative reference models (e.g., different neural network architectures, decision trees) and determining the optimal θ corresponds to training of such models.

3. Expert knowledge. Assume *a priori* knowledge to be formulated in terms of conjunctions and disjunctions of simple components or building blocks (for example verbally). E.g., an image of a face is expected to contain certain constituents (eyes, mouth, nose; AND) appearing in various possible variants (OR). Representing the simple components/building blocks by Gaussian priors centered around a typical example (e.g.,of an eye) results in Gaussian mixture models. This constitutes a possible interface between symbolic and statistical methods. Such an application of prior mixture models has some similarities with the quantification of "linguistic variables" by fuzzy methods [118, 119].

For a discussion of possible applications of prior mixture models see also [132, 133, 134, 135, 136]. An application of prior mixture models to image completion can be found in [137].

6.2 Prior mixtures for density estimation

The mixture approach (535) leads in general to non–convex error functionals. For Gaussian components Eq. (535) results in an error functional

$$\begin{aligned} E_{\theta,\phi} &= -(\ln P(\phi),\, N) + (P(\phi),\, \Lambda_X) \\ &\quad - \ln \sum_j e^{-\left(\frac{1}{2}(\phi - t_j(\theta),\, \mathbf{K}_j(\theta)\,(\phi - t_j(\theta))) + \ln Z_\phi(\theta,j) + E_{\theta,j}\right)}, \qquad (536) \\ &= -\ln \sum_j e^{-E_{\phi,j} - E_{\theta,j} + c_j}, \qquad (537) \end{aligned}$$

where

$$E_{\phi,j} = -(\ln P(\phi),\, N) + (P(\phi),\, \Lambda_X) + \frac{1}{2}\Big(\phi - t_j(\theta),\, \mathbf{K}_j(\theta)\,(\phi - t_j(\theta))\Big), \quad (538)$$

and

$$c_j = -\ln Z_\phi(\theta, j). \tag{539}$$

The stationarity equations for ϕ and θ

$$\begin{aligned} 0 &= \sum_j^m \frac{\delta E_{\phi,j}}{\delta \phi} e^{-E_{\phi,j}-E_{\theta,j}+c_j}, & (540) \\ 0 &= \sum_j^m \left(\frac{\partial E_{\phi,j}}{\partial \theta} + \frac{\partial E_{\theta,j}}{\partial \theta} + \mathbf{Z}'_j Z_\phi^{-1}(\theta, j) \right) e^{-E_{\phi,j}-E_{\theta,j}+c_j}, & (541) \end{aligned}$$

can also be written

$$\begin{aligned} 0 &= \sum_j^m \frac{\delta E_{\phi,j}}{\delta \phi} p(\phi, \theta, j|\tilde{D}_0), & (542) \\ 0 &= \sum_j^m \left(\frac{\partial E_{\phi,j}}{\partial \theta} + \frac{\partial E_{\theta,j}}{\partial \theta} + \mathbf{Z}'_j Z_\phi^{-1}(\theta, j) \right) p(\phi, \theta, j|\tilde{D}_0). & (543) \end{aligned}$$

Analogous equations are obtained for parameterized $\phi(\xi)$.

6.3 Prior mixtures for regression

For regression it is especially useful to introduce an inverse temperature multiplying the terms depending on ϕ, i.e., likelihood and prior.[3] As in regression ϕ is represented by the regression function $h(x)$ the temperature–dependent error functional becomes

$$E_{\theta,h} = -\ln \sum_j^m e^{-\beta E_{h,j} - E_{\theta,\beta,j} + c_j} = -\ln \sum_j^m e^{-E_j + c_j}, \tag{544}$$

with

$$E_j = E_D + E_{0,j} + E_{\theta,\beta,j}, \tag{545}$$

$$E_D = \frac{1}{2}\left(h - t_D,\, \mathbf{K}_D\,(h - t_D)\right), \quad E_{0,j} = \frac{1}{2}\left(h - t_j(\theta),\, \mathbf{K}_j(\theta)\,(h - t_j(\theta))\right), \tag{546}$$

[3]The variable β is, together with the regression function $\phi(x)$, part of the hidden variables $\tilde{\phi} = (\phi, \beta)$ which determine the likelihood and thus not a hyperparameter. Technically, however, β can be treated similar to a regularization factor, being the reason that it is included in this chapter on hyperparameters.

some hyperprior energy $E_{\theta,\beta,j}$, and

$$\begin{aligned} c_j(\theta,\beta) &= -\ln Z_h(\theta,j,\beta) + \frac{n}{2}\ln\beta - \frac{\beta}{2}V_D - c \\ &= \frac{1}{2}\ln\det\left(\mathbf{K}_j(\theta)\right) + \frac{d+n}{2}\ln\beta - \frac{\beta}{2}V_D \end{aligned} \tag{547}$$

with some constant c. If we also maximize with respect to β we have to include the (h–independent) training data variance $V_D = \sum_i^n V_i$ where $V_i = \sum_k^{n_i} y(x_k)^2/n_i - t_D^2(x_i)$ is the variance of the n_i training data at x_i. In case every x_i appears only once V_D vanishes. Notice that c_j includes a contribution from the n data points arising from the β–dependent normalization of the likelihood term. Writing the stationarity equation for the hyperparameter β separately, the corresponding three stationarity conditions are found as

$$0 = \sum_j^m \Big(\mathbf{K}_D\,(h-t_D) + \mathbf{K}_j\,(h-t_j)\Big)\, e^{-\beta E_{h,j} - E_{\theta,\beta,j} + c_j}, \tag{548}$$

$$0 = \sum_j^m \left(E'_{h,j} + E'_{\theta,\beta,j} + \mathrm{Tr}\left(\mathbf{K}_j^{-1}\frac{\partial \mathbf{K}_j}{\partial\theta}\right)\right) e^{-\beta E_{h,j} - E_{\theta,\beta,j} + c_j}, \tag{549}$$

$$0 = \sum_j^m \left(E_{0,j} + \frac{\partial E_{\theta,\beta,j}}{\partial\beta} + \frac{d+n}{2\beta}\right) e^{-\beta E_{h,j} - E_{\theta,\beta,j} + c_j}. \tag{550}$$

As β is only a one–dimensional parameter and its density can be quite non–Gaussian it is probably most times more informative to solve for varying values of β instead to restrict to a single 'optimal' β^*. Eq. (548) can also be written

$$h = \left(\mathbf{K}_D + \sum_j^m a_j \mathbf{K}_j\right)^{-1} \left(\mathbf{K}_D t_D + \sum_l^m a_j \mathbf{K}_j t_j\right), \tag{551}$$

with

$$\begin{aligned} a_j &= p(j|h,\theta,\beta,D_0) = \frac{e^{-E_j+c_j}}{\sum_k^m e^{-E_k+c_k}} = \frac{e^{-\beta E_{0,j} - E_{\theta,\beta,j} + \frac{1}{2}\ln\det\mathbf{K}_j}}{\sum_k^m e^{-\beta E_{0,k} - E_{\theta,\beta,k} + \frac{1}{2}\ln\det\mathbf{K}_k}} \\ &= \frac{p(h|j,\theta,\beta,D_0)p(j|\theta,\beta,D_0)}{p(h|\theta,\beta,D_0)} = \frac{p(h|j,\theta,\beta,D_0)p(j,\theta|\beta,D_0)}{p(h,\theta|\beta,D_0)}, \end{aligned} \tag{552}$$

being thus still a nonlinear equation for h.

6.3.1 High and low temperature limits

It are the limits of large and small β which make the introduction of this additional parameter useful. The reason being that the high temperature limit $\beta \to 0$ gives the convex case, and statistical mechanics provides us with high and low temperature expansions. Hence, we study the high temperature and low temperature limits of Eq. (551).

In the *high temperature limit* $\beta \to 0$ the exponential factors a_j become h–independent

$$a_j \stackrel{\beta\to 0}{\longrightarrow} a_j^0 = \frac{e^{-E_{\theta,\beta,j}+\frac{1}{2}\ln\det \mathbf{K}_j}}{\sum_k^m e^{-E_{\theta,\beta,k}+\frac{1}{2}\ln\det \mathbf{K}_k}}. \tag{553}$$

In case one chooses $E_{\theta,\beta,j} = E_{\beta,j} + \beta E_\theta$ one has to replace $E_{\theta,\beta,j}$ by $E_{\beta,j}$. The high temperature solution becomes

$$h = \bar{t} \tag{554}$$

with (generalized) 'complete template average'

$$\bar{t} = \left(\mathbf{K}_D + \sum_j^m a_j^0 \mathbf{K}_j\right)^{-1} \left(\mathbf{K}_D t_D + \sum_l^m a_j^0 \mathbf{K}_j t_j\right). \tag{555}$$

Notice that $\bar{t}$ corresponds to the minimum of the quadratic functional

$$E_{(\beta=\infty)} = \Big(h - t_D,\, \mathbf{K}_D(h - t_D)\Big) + \sum_j^m a_j^0 \Big(h - t_j,\, \mathbf{K}_j(h - t_j)\Big). \tag{556}$$

Thus, in the infinite temperature limit a combination of quadratic priors by OR is effectively replaced by a combination by AND.

In the *low temperature limit* $\beta \to \infty$ we have, assuming $E_{\theta,\beta,j} = E_\beta + E_j + \beta E_\theta$,

$$\sum_j e^{-\beta(E_{0,j}+E_\theta)-E_\beta-E_j} = e^{-\beta(E_{0,j^*}+E_\theta)-E_\beta} \sum_j e^{-\beta(E_{0,j}-E_{0,j^*})-E_j} \tag{557}$$

$$\stackrel{\beta\to\infty}{\longrightarrow} e^{-\beta(E_{0,j^*}+E_\theta)-E_\beta-E_j} \qquad \text{for} \quad E_{0,j^*} < E_{0,j},\ \forall j \ne j^*,\ p(j^*) \ne 0,, \tag{558}$$

meaning that

$$a_j \stackrel{\beta\to\infty}{\longrightarrow} a_j^\infty = \left\{ \begin{array}{lcl} 1 & : & j = \text{argmin}_j E_{0,j} = \text{argmin}_j E_{h,j} \\ 0 & : & j \ne \text{argmin}_j E_{0,j} = \text{argmin}_j E_{h,j} \end{array} \right. . \tag{559}$$

Henceforth, all (generalized) 'component averages' $\bar{t}_j$ become solutions

$$h = \bar{t}_j, \tag{560}$$

with

$$\bar{t}_j = (\mathbf{K}_D + \mathbf{K}_j)^{-1} (\mathbf{K}_D t_D + \mathbf{K}_j t_j), \tag{561}$$

provided the $\bar{t}_j$ fulfill the stability condition

$$E_{h,j}(h = \bar{t}_j) < E_{h,j'}(h = \bar{t}_j), \quad \forall j' \ne j, \tag{562}$$

i.e.,

$$V_j < \frac{1}{2}\Big(\bar{t}_j - \bar{t}_{j'},\, (\mathbf{K}_D + \mathbf{K}_{j'})\,(\bar{t}_j - \bar{t}_{j'})\Big) + V_{j'}, \quad \forall j' \ne j, \tag{563}$$

where

$$V_j = \frac{1}{2}\Bigg(\Big(t_D,\, \mathbf{K}_D\, t_D\Big) + \Big(t_j,\, \mathbf{K}_j\, t_j\Big) - \Big(\bar{t}_j,\, (\mathbf{K}_D + \mathbf{K}_j)\, \bar{t}_j\Big)\Bigg). \tag{564}$$

That means single components become solutions at zero temperature $1/\beta$ in case their (generalized) 'template variance' V_j, measuring the discrepancy between data and prior term, is not too large. Eq. (551) for h can also be expressed by the (potential) low temperature solutions $\bar{t}_j$

$$h = \left(\sum_j^m a_j(\mathbf{K}_D + \mathbf{K}_j)\right)^{-1} \sum_j^m a_j\,(\mathbf{K}_D + \mathbf{K}_j)\,\bar{t}_j. \tag{565}$$

Summarizing, in the high temperature limit the stationarity equation (548) becomes linear with a single solution being essentially a (generalized) average of all template functions. In the low temperature limit the single component solutions become stable provided their (generalized) variance corresponding to their minimal error is small enough.

6.3.2 Equal covariances

Especially interesting is the case of j–independent $\mathbf{K}_j(\theta) = \mathbf{K}_0(\theta)$ and θ–independent $\det \mathbf{K}_0(\theta)$. In that case the often difficult to obtain determinants of $\mathbf{K}_j$ do not have to be calculated.

For j–independent inverse covariances the high temperature solution is according to Eqs.(555,561) a linear combination of the (potential) low temperature solutions

$$\bar{t} = \sum_j^m a_j^0 \bar{t}_j. \tag{566}$$

It is worth to emphasize that, as the solution $\bar{t}$ is *not* a mixture of the component templates t_j but of component solutions $\bar{t}_j$, even poor choices for the template functions t_j can lead to good solutions, if enough data are available. That is indeed the reason why the most common choice $t_0 \equiv 0$ for a Gaussian prior can be successful.

Eqs.(565) simplifies to

$$h = \frac{\sum_j^m \bar{t}_j e^{-\beta E_{h,j}(h) - E_{\theta,\beta,j} + c_j}}{\sum_j^m e^{-\beta E_{h,j}(h) - E_{\theta,\beta,j} + c_j}} = \sum_j^m a_j \bar{t}_j = \bar{t} + \sum_j^m (a_j - a_j^0)\, \bar{t}_j, \tag{567}$$

where

$$\bar{t}_j = (\mathbf{K}_D + \mathbf{K}_0)^{-1} (\mathbf{K}_D t_D + \mathbf{K}_0 t_j), \tag{568}$$

and (for j–independent d)

$$a_j = \frac{e^{-E_j}}{\sum_k e^{-E_k}} = \frac{e^{-\beta E_{h,j} - E_{\theta,\beta,j}}}{\sum_k e^{-\beta E_{h,k} - E_{\theta,\beta,k}}} = \frac{e^{-\frac{\beta}{2} a B_j a + d_j}}{\sum_k e^{-\frac{\beta}{2} a B_k a + d_k}}, \tag{569}$$

introducing vector a with components a_j, $m \times m$ matrices

$$B_j(k,l) = \Big(\bar{t}_k - \bar{t}_j,\, (\mathbf{K}_D + \mathbf{K}_0)\, (\bar{t}_l - \bar{t}_j)\Big) \tag{570}$$

and constants

$$d_j = -\beta V_j - E_{\theta,\beta,j}, \tag{571}$$

with V_j given in (564). Eq. (567) is still a nonlinear equation for h, it shows however that the solutions must be convex combinations of the h–independent $\bar{t}_j$. Thus, it is sufficient to solve Eq. (569) for m mixture coefficients a_j instead of Eq. (548) for the function h.

The high temperature relation Eq. (553) becomes

$$a_j \stackrel{\beta \to 0}{\longrightarrow} a_j^0 = \frac{e^{-E_{\theta,\beta,j}}}{\sum_k^m e^{-E_{\theta,\beta,k}}}, \tag{572}$$

or $a_j^0 = 1/m$ for a hyperprior $p(\theta, \beta, j)$ uniform with respect to j. The low temperature relation Eq. (559) remains unchanged.

For $m = 2$ Eq. (567) becomes

$$h = \sum_j^2 a_j \bar{t}_j = \frac{\bar{t}_1 + \bar{t}_2}{2} + (a_1 - a_2)\frac{\bar{t}_1 - \bar{t}_2}{2} = \frac{\bar{t}_1 + \bar{t}_2}{2} + (\tanh \Delta)\,\frac{\bar{t}_1 - \bar{t}_2}{2}, \quad (573)$$

with $(\bar{t}_1 + \bar{t}_2)/2 = \bar{t}$ in case $E_{\theta,\beta,j}$ is uniform in j so that $a_j^0 = 0.5$, and

$$\begin{aligned}\Delta &= \frac{E_2 - E_1}{2} = \beta \frac{E_{h,2} - E_{h,1}}{2} + \frac{E_{\theta,\beta,2} - E_{\theta,\beta,1}}{2} \\ &= -\frac{\beta}{4} a(B_1 - B_2)a + \frac{d_1 - d_2}{2} = \frac{\beta}{4}\, b(2a_1 - 1) + \frac{d_1 - d_2}{2}, \quad (574)\end{aligned}$$

because the matrices B_j are in this case zero except $B_1(2,2) = B_2(1,1) = b$. The stationarity Eq. (569) can be solved graphically (see Figs.7, 8), the solution being given by the point where $a_1 e^{-\frac{\beta}{2}ba_1^2 + d_2} = (1 - a_1)e^{-\frac{\beta}{2}b(1-a_1)^2 + d_1}$, or, alternatively,

$$a_1 = \frac{1}{2}\left(\tanh \Delta + 1\right). \quad (575)$$

That equation is analogous to the celebrated mean field equation of the ferromagnet.

We conclude that in the case of equal component covariances, in addition to the linear low–temperature equations, only a $m-1$–dimensional nonlinear equation has to be solved to determine the 'mixing coefficients' $a_1, \cdots, a_{m-1}$.

6.3.3 Analytical solution of mixture models

For regression under a Gaussian mixture model the predictive density can be calculated analytically for fixed θ. The predictive density can be expressed in terms of the likelihood of θ and j, marginalized over h,

$$p(y|x, D, D_0) = \sum_j \int d\theta\, \frac{p(\theta, j)\, p(y_D|x_D, D_0, \theta, j)}{\sum_j \int d\theta\, p(\theta, j) p(y_D|x_D, D_0, \theta, j)} p(y|x, D, D_0, \theta, j). \quad (576)$$

(Here we concentrate on θ. The parameter β can be treated analogously.) According to Eq. (492) the likelihood can be written

$$p(y_D|x_D, D_0, \theta, j) = e^{-\beta \widetilde{E}_{0,j}(\theta) + \frac{1}{2}\ln\det(\frac{\beta}{2\pi}\widetilde{\mathbf{K}}_j(\theta))}, \quad (577)$$

with

$$\widetilde{E}_{0,j}(\theta) = \frac{1}{2}(t_D - t_j(\theta),\, \widetilde{\mathbf{K}}_j(\theta)(t_D - t_j(\theta))) = V_j, \quad (578)$$

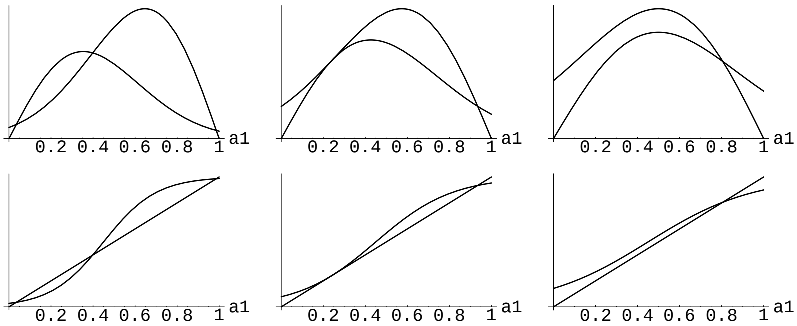

Figure 7: The solution of stationary equation Eq. (569) is given by the point where $a_1 e^{-\frac{\beta}{2} b a_1^2 + d_2} = (1-a_1) e^{-\frac{\beta}{2} b (1-a_1)^2 + d_1}$ (upper row) or, equivalently, $a_1 = \frac{1}{2}(\tanh \Delta + 1)$ (lower row). Shown are, from left to right, a situation at high temperature and one stable solution ($\beta = 2$), at a temperature ($\beta = 2.75$) near the bifurcation, and at low temperature with two stable and one unstable solutions $\beta = 4$. The values of $b = 2$, $d_1 = -0.2025\beta$ and $d_2 = -0.3025\beta$ used for the plots correspond for example to the one–dimensional model of Fig.9 with $t_1 = 1$, $t_2 = -1$, $t_D = 0.1$. Notice, however, that the shown relation is valid for $m = 2$ at arbitrary dimension.

and $\widetilde{\mathbf{K}}_j(\theta) = (\mathbf{K}_D^{-1} + \mathbf{K}_{j,DD}^{-1}(\theta))^{-1}$ being a $\tilde{n} \times \tilde{n}$–matrix in data space. The equality of V_j and $\widetilde{E}_{0,j}$ can be seen using $\mathbf{K}_j - \mathbf{K}_j(\mathbf{K}_D + \mathbf{K}_j)^{-1}\mathbf{K}_j = \mathbf{K}_D - \mathbf{K}_D(\mathbf{K}_D + \mathbf{K}_{j,DD})^{-1}\mathbf{K}_D = \mathbf{K}_{j,DD} - \mathbf{K}_{j,DD}(\mathbf{K}_D + \mathbf{K}_{j,DD)}^{-1}\mathbf{K}_{j,DD} = \widetilde{\mathbf{K}}$. For the predictive mean, being the optimal solution under squared–error loss and log–loss (restricted to Gaussian densities with fixed variance) we find therefore

$$\bar{y}(x) = \int dy\, y\, p(y|x, D, D_0) = \sum_j \int d\theta\; b_j(\theta)\, \bar{t}_j(\theta), \tag{579}$$

with, according to Eq. (327),

$$\bar{t}_j(\theta) = t_j + \mathbf{K}_j^{-1}\widetilde{\mathbf{K}}_j(t_D - t_j), \tag{580}$$

and mixture coefficients

$$\begin{aligned} b_j(\theta) &= p(\theta, j|D) = \frac{p(\theta, j)p(y_D|x_D, D_0, \theta, j)}{\sum_j \int d\theta p(\theta, j)p(y_D|x_D, D_0, \theta, j)} \\ &\propto e^{-\beta \widetilde{E}_j(\theta) - E_{\theta,j} + \frac{1}{2}\ln\det(\widetilde{K}_j(\theta))}, \end{aligned} \tag{581}$$

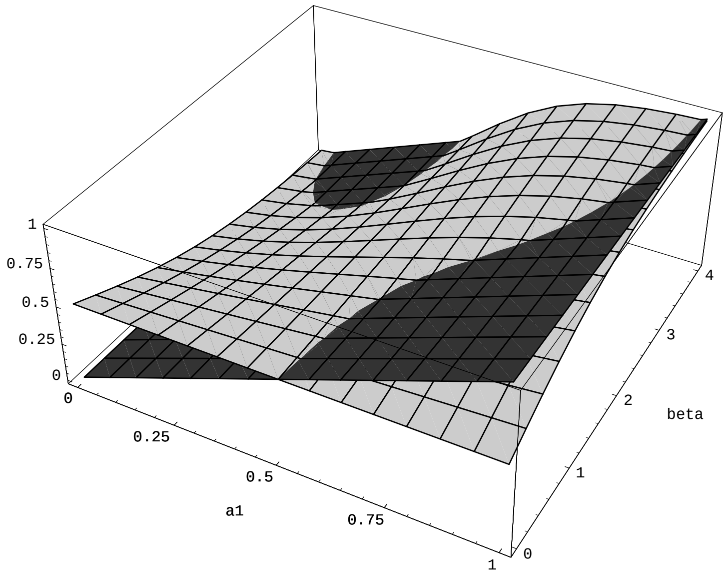

Figure 8: As in Fig.7 the plots of $f_1(a_1) = a_1$ and $f_2(a_1) = \frac{1}{2}(\tanh\Delta + 1)$ are shown within the inverse temperature range $0 \le \beta \le 4$.

which defines $\widetilde{E}_j = \beta\widetilde{E}_{0,j} + E_{\theta,j}$. For solvable θ–integral the coefficients can therefore be obtained exactly.

If b_j is calculated in saddle point approximation at $\theta = \theta^*$ it has the structure of a_j in (552) with $E_{0,j}$ replaced by $\widetilde{E}_j$ and $\mathbf{K}_j$ by $\widetilde{\mathbf{K}}_j$. (The inverse temperature β could be treated analogously to θ. In that case $E_{\theta,j}$ would have to be replaced by $E_{\theta,\beta,j}$.)

Calculating also the likelihood for j, θ in Eq. (581) in saddle point approximation, i.e., $p(y_D|x_D, D_0, \theta^*, j) \approx p(y_D|x_D, h^*)p(h^*|D_0, \theta^*, j)$, the terms $p(y_D|x_D, h^*)$ in numerator and denominator cancel, so that, skipping D_0 and β,

$$b_j(\theta^*) = \frac{p(h^*|j, \theta^*)p(j, \theta^*)}{p(h^*, \theta^*)} = a_j(h^*, \theta^*), \tag{582}$$

becomes equal to the $a_j(\theta^*)$ in Eq. (552) at $h = h^*$.

Eq. (581) yields as stationarity equation for θ, similarly to Eq. (494)

$$\begin{aligned} 0 &= \sum_j b_j \left(\frac{\partial \widetilde{E}_j}{\partial \theta} - \mathrm{Tr}\left(\widetilde{\mathbf{K}}_j^{-1} \frac{\partial \widetilde{\mathbf{K}}_j}{\partial \theta} \right) \right) \\ &= \sum_j b_j \left(\left(\frac{\partial t_j(\theta)}{\partial \theta},\, \widetilde{\mathbf{K}}_j(\theta)(t_j(\theta) - t_D) \right) \right. \\ &\quad \left. + \frac{1}{2} \left((t_D - t_j(\theta)),\, \frac{\partial \widetilde{\mathbf{K}}_j(\theta)}{\partial \theta}(t_D - t_j(\theta)) \right) \right. \end{aligned} \tag{583}$$

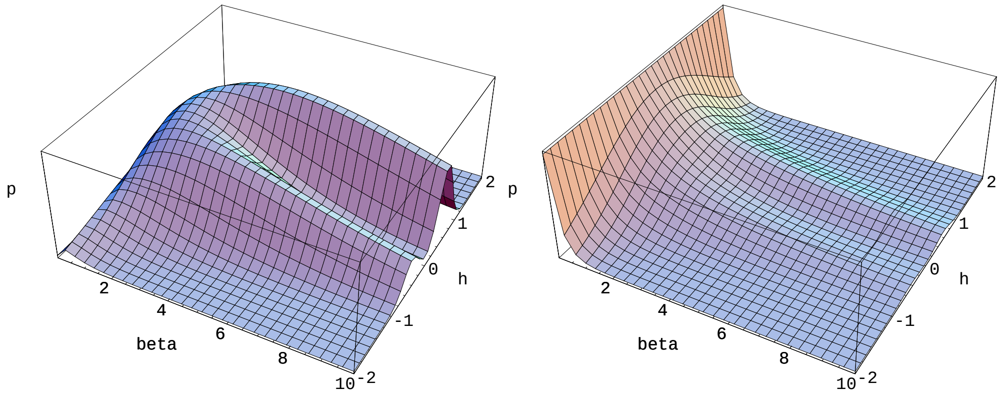

Figure 9: Shown is the joint posterior density of h and β, i.e., $p(h, \beta|D, D_0) \propto p(y_D|h, \beta)p(h|\beta, D_0)p(\beta)$ for a zero–dimensional example of a Gaussian prior mixture model with training data $y_D = 0.1$ and prior data $y_{D_0} = \pm 1$ and inverse temperature β. L.h.s.: For uniform prior (middle) $p(\beta) \propto 1$ with joint posterior $p \propto e^{-\frac{\beta}{2}h^2 + \ln \beta} \left(e^{-\frac{\beta}{2}(h-1)^2} + e^{-\frac{\beta}{2}(h+1)^2} \right)$ the maximum appears at finite β. (Here no factor 1/2 appears in front of $\ln \beta$ because normalization constants for prior and likelihood term have to be included.) R.h.s.: For compensating hyperprior $p(\beta) \propto 1/\sqrt{\beta}$ with $p \propto e^{-\frac{\beta}{2}h^2 - \frac{\beta}{2}(h-1)^2} + e^{-\frac{\beta}{2}h^2 - \frac{\beta}{2}(h+1)^2}$ the maximum is at $\beta = 0$.

$$-\mathrm{Tr}\left(\widetilde{\mathbf{K}}_j^{-1}(\theta) \frac{\partial \widetilde{\mathbf{K}}_j(\theta)}{\partial \theta} \right) - \frac{1}{p(\theta, j)} \frac{\partial p(\theta, j)}{\partial \theta} \Bigg). \tag{584}$$

For fixed θ and j–independent covariances the high temperature solution is a mixture of component solutions weighted by their prior probability

$$\bar{y} \stackrel{\beta \to 0}{\longrightarrow} \sum_j p(j)\, \bar{t}_j = \sum_j a_j^0\, \bar{t}_j = \bar{t}. \tag{585}$$

The low temperature solution becomes the component solution $\bar{t}_j$ with minimal distance between data and prior template

$$\bar{y} \stackrel{\beta \to \infty}{\longrightarrow} \bar{t}_{j^*} \quad \text{with } j^* = \ \mathrm{argmin}_j (t_D - t_j,\, \widetilde{\mathbf{K}}_j (t_D - t_j)). \tag{586}$$

Fig.11 compares the exact mixture coefficient b_1 with the dominant solution of the maximum posterior coefficient a_1 (see also [132]) which are related

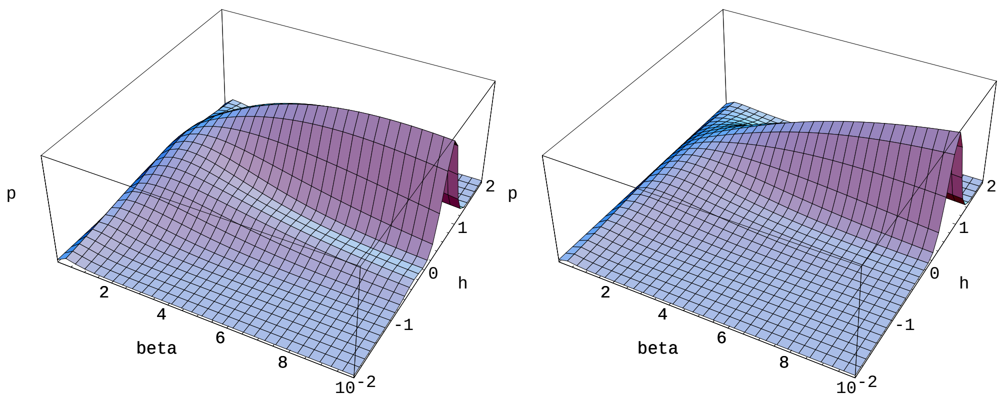

Figure 10: Same zero–dimensional prior mixture model for uniform hyperprior on β as in Fig.9, but for varying data $x_d = 0.3$ (left), $x_d = 0.5$ (right).

according to (569)

$$a_j = \frac{e^{-\frac{\beta}{2} a B_j a - \widetilde{E}_j}}{\sum_k e^{-\frac{\beta}{2} a B_k a - \widetilde{E}_k}} = \frac{b_j \, e^{-\frac{\beta}{2} a B_j a}}{\sum_k b_k \, e^{-\frac{\beta}{2} a B_k a}}. \tag{587}$$

6.4 Local mixtures

Global mixture components can be obtained by combining local mixture components. Predicting a time series, for example, one may allow to switch locally (in time) between two or more possible regimes, each corresponding to a different local covariance or template.

The problem which arises when combining local alternatives is the fact that the total number of mixture components grows exponentially in the number local components which have to be combined for a global mixture component.

Consider a local prior mixture model, similar to Eq. (531),

$$p(\phi|\theta) = e^{-\int dx ; |\omega(x;\theta(x))|^2 - \ln Z_\phi(\theta)} \tag{588}$$

where $\theta(x)$ may be a binary or an integer variable. The local mixture variable $\theta(x)$ labels local alternatives for filtered differences $\omega(x;\theta(x))$ which may differ in their templates $t(x;\theta(x))$ and/or their local filters $\mathbf{W}(x;\theta(x))$. To

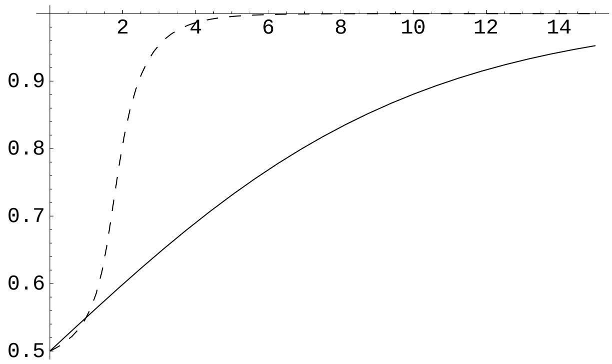

Figure 11: Exact b_1 and a_1 (dashed) vs. β for two mixture components with equal covariances and $B_1(2,2) = b = 2$, $\tilde{E}_1 = 0.405$, $\tilde{E}_2 = 0.605$.

avoid infinite products, we choose a discretized x variable (which may include the y variable for general density estimation problems), so that

$$p(\phi) = \sum_\theta p(\theta) e^{-\sum_x |\omega(x;\theta(x))|^2 - \ln Z_\phi(\theta)}, \tag{589}$$

where the sum $\sum_\theta$ is over all local integer variables $\theta(x)$, i.e.,

$$\sum_\theta = \sum_{\theta(x_1)} \cdots \sum_{\theta(x_l)} = \left(\prod_x \sum_{\theta(x_1)} \right). \tag{590}$$

Only for factorizing hyperprior $p(\theta) = \prod_x p(\theta(x))$ the complete posterior factorizes

$$\begin{aligned} p(\phi) &= \left(\prod_{x'} \sum_{\theta(x')} \right) \prod_x \left(p(\theta(x)) e^{-|\omega(x;\theta(x))|^2 - \ln Z_\phi(x,\theta(x))} \right) \\ &= \prod_x \sum_{\theta(x)} \left(p(\theta(x)) e^{-|\omega(x;\theta(x))|^2 - \ln Z_\phi(x,\theta(x))} \right), \end{aligned} \tag{591}$$

because

$$Z_\phi = \prod_x \sum_{\theta(x)} \left(e^{-|\omega(x;\theta(x))|^2} \right) = \prod_x Z_\phi(x, \theta(x)). \tag{592}$$

Under that condition the mixture coefficients a_θ of Eq. (552) can be obtained from the equations, local in $\theta(x)$,

$$a_\theta = a_{\theta(x_1)\cdots\theta(x_l)} = p(\theta|\phi) = \prod_x a_{\theta(x)} \tag{593}$$

with

$$a_{\theta(x)} = \frac{p(\theta(x))e^{-|\omega(x;\theta(x))|^2 - \ln Z_\phi(x;\theta(x))}}{\sum_{\theta'(x)} p(\theta'(x))e^{-|\omega(x;\theta'(x))|^2 - \ln Z_\phi(x;\theta'(x))}}. \tag{594}$$

For equal covariances this is a nonlinear equation within a space of dimension equal to the number of local components. For non–factorizing hyperprior the equations for different $\theta(x)$ cannot be decoupled.

6.5 Non–quadratic potentials

Solving learning problems numerically by discretizing the x and y variables allows in principle to deal with arbitrary non–Gaussian priors. Compared to Gaussian priors, however, the resulting stationarity equations are intrinsically nonlinear.

As a typical example let us formulate a prior in terms of nonlinear and non–quadratic "potential" functions ψ acting on "filtered differences" $\omega = \mathbf{W}(\phi - t)$, defined with respect to some positive (semi–)definite inverse covariance $\mathbf{K} = \mathbf{W}^T\mathbf{W}$. In particular, consider a prior factor of the following form

$$p(\phi) = e^{-\int dx\, \psi(\omega(x)) - \ln Z_\phi} = \frac{e^{-E(\phi)}}{Z_\phi}, \tag{595}$$

where $E(\phi) = \int dx\, \psi(\omega(x))$. For general density estimation problems we understand x to stand for a pair (x, y). Such priors are for example used for image restoration [70, 28, 168, 71, 248, 246].

For differentiable ψ function the functional derivative with respect to $\phi(x)$ becomes

$$\delta_{\phi(x)} p(\phi) = -e^{-\int dx'\, \psi(\omega(x')) - \ln Z_\phi} \int dx''\, \psi'(\omega(x''))\mathbf{W}(x'', x), \tag{596}$$

with $\psi'(s) = d\psi(z)/dz$, from which follows

$$\delta_\phi E(\phi) = -\delta_\phi \ln p(\phi) = \mathbf{W}^T \psi'. \tag{597}$$

For nonlinear filters acting on $\phi - t$, $\mathbf{W}$ in Eq. (597) must be replaced by $\omega'(x) = \delta_{\phi(x)}\omega(x)$. Instead of one $\mathbf{W}$ a "filter bank" $\mathbf{W}_\alpha$ with corresponding $\mathbf{K}_\alpha$, ω_α, and ψ_α may be used, so that

$$e^{-\sum_\alpha \int dx\, \psi_\alpha(\omega_\alpha(x)) - \ln Z_\phi}, \tag{598}$$

and

$$\delta_\phi E(\phi) = \sum_\alpha \mathbf{W}_\alpha^T \psi'_\alpha. \tag{599}$$

The potential functions ψ may be fixed in advance for a given problem. Typical choices to allow discontinuities are symmetric "cup" functions with minimum at zero and flat tails for which one large step is cheaper than many small ones [238]). Examples are shown in Fig. 12 (a,b). The cusp in (b), where the derivative does not exist, requires special treatment [246]. Such functions can also be interpreted in the sense of robust statistics as flat tails reduce the sensitivity with respect to outliers [100, 101, 67, 26].

Inverted "cup" functions, like those shown in Fig. 12 (c), have been obtained by optimizing a set of ψ_α with respect to a sample of natural images [246]. (For statistics of natural images their relation to wavelet–like filters and sparse coding see also [175, 176].)

While, for $\mathbf{W}$ which are differential operators, cup functions promote smoothness, inverse cup functions can be used to implement structure. For such $\mathbf{W}$ the gradient algorithm for minimizing $E(\phi)$,

$$\phi_{\rm new} = \phi_{\rm old} - \eta \delta_\phi E(\phi_{\rm old}), \tag{600}$$

becomes in the continuum limit a nonlinear parabolic partial differential equation,

$$\phi_\tau = -\sum_\alpha \mathbf{W}_\alpha^T \psi'_\alpha(\mathbf{W}_\alpha(\phi - t)). \tag{601}$$

Here a formal time variable τ have been introduced so that $(\phi_{\rm new} - \phi_{\rm old})/\eta \rightarrow \phi_\tau = d\phi/d\tau$. For cup functions this equation is of diffusion type [173, 188], if also inverted cup functions are included the equation is of reaction–diffusion type [246]. Such equations are known to generate a great variety of patterns.

Alternatively to fixing ψ in advance or, which is sometimes possible for low–dimensional discrete function spaces like images, to approximate ψ by sampling from the prior distribution, one may also introduce hyperparameters and adapt potentials $\psi(\theta)$ to the data.

For example, attempting to adapt a unrestricted function $\psi(x)$ with hyperprior $p(\psi)$ by Maximum A Posteriori Approximation one has to solve the stationarity condition

$$0 = \delta_{\psi(s)} \ln p(\phi, \psi) = \delta_{\psi(s)} \ln p(\phi|\psi) + \delta_{\psi(s)} \ln p(\psi). \tag{602}$$

From

$$\delta_{\psi(s)} p(\phi|\psi) = -p(\phi|\psi) \int dx\, \delta\left(s - \omega(x)\right) - \frac{1}{Z_\phi^2} \delta_{\psi(s)} Z_\phi, \tag{603}$$

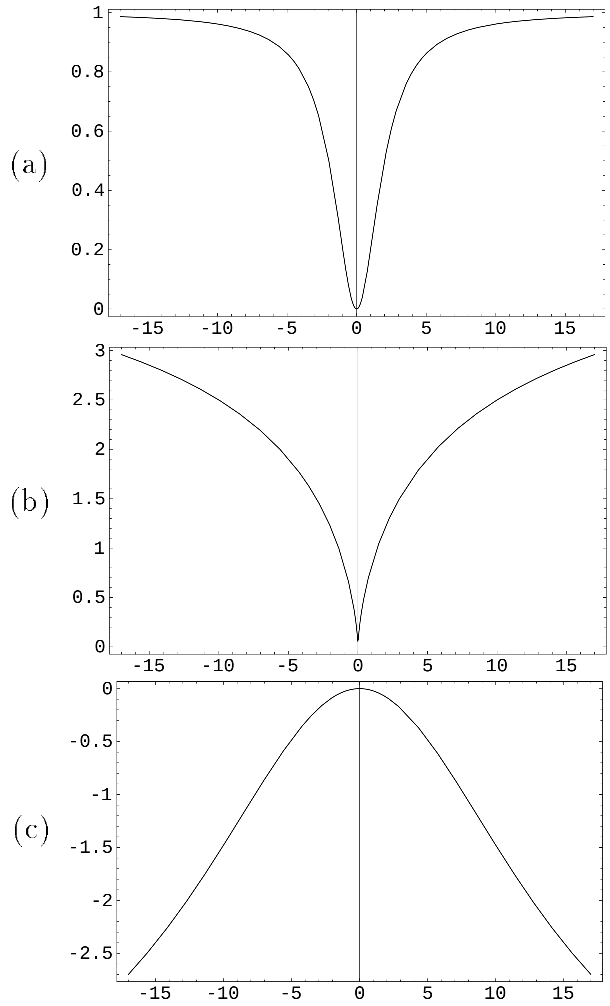

Figure 12: Non–quadratic potentials of the form $\psi(x) = a(1.0 - 1/(1 + (|x - x_0|/b)^\gamma))$, [246]: "Diffusion terms": (a) Winkler's cup function [238] (a= 5, $b = 10$, $\gamma = 0.7$, $x_0 = 0$), (b) with cusp (a= 1, $b = 3$, $\gamma = 2$, $x_0 = 0$), (c) "Reaction term" ($a = -4.8$, $b = 15$, $\gamma = 2.0$ $x_0 = 0$).

it follows

$$-\delta_{\psi(s)} \ln p(\phi|\psi) = n(s) - < n(s) >, \tag{604}$$

with integer

$$n(s) = \int dx\, \delta\left(s - \omega(x)\right), \tag{605}$$

being the histogram of the filtered differences, and average histogram

$$< n(s) > = \int d\phi\, p(\phi|\psi)\, n(s). \tag{606}$$

The right hand side of Eq. (604) is zero at ϕ^* if, e.g., $p(\phi|\psi) = \delta(\phi - \phi^*)$, which is the case for $\psi(\omega(x;\phi)) = \beta\left(\omega(x;\phi) - \omega(x;\phi^*)\right)^2$ in the $\beta \to \infty$ limit.

Introducing hyperparameters one has to keep in mind that the resulting additional flexibility must be balanced by the number of training data and the hyperprior to be useful in practice.

7 Iteration procedures: Learning

7.1 Numerical solution of stationarity equations

Due to the presence of the logarithmic data term $-(\ln P, N)$ and the normalization constraint in density estimation problems the stationary equations are in general nonlinear, even for Gaussian specific priors. An exception are Gaussian regression problems discussed in Section 3.7 for which $-(\ln P, N)$ becomes quadratic and the normalization constraint can be skipped. However, the nonlinearities arising from the data term $-(\ln P, N)$ are restricted to a finite number of training data points and for Gaussian specific priors one may expect them, like those arising from the normalization constraint, to be numerically not very harmful. Clearly, severe nonlinearities can appear for general non–Gaussian specific priors or general nonlinear parameterizations $P(\xi)$.

As nonlinear equations the stationarity conditions have in general to be solved by iteration. In the context of empirical learning iteration procedures to minimize an error functional represent possible *learning algorithms*.

In the previous sections we have encountered stationarity equations

$$0 = \frac{\delta(-E_\phi)}{\delta\phi} = G(\phi), \tag{607}$$

for error functionals E_ϕ, e.g., $\phi = L$ or $\phi = P$, written in a form

$$\mathbf{K}\phi = T. \tag{608}$$

with ϕ–dependent T (and possibly $\mathbf{K}$). For the stationarity Eqs. (143), (172), and (193) the operator $\mathbf{K}$ is a ϕ–independent inverse covariance of a Gaussian specific prior. It has already been mentioned that for existing (and not too ill–conditioned) $\mathbf{K}^{-1}$ (representing the covariance of the prior process) Eq. (608) suggests an iteration scheme

$$\phi^{(i+1)} = \mathbf{K}^{-1}T(\phi^{(i)}), \tag{609}$$

for discretized ϕ starting from some initial guess $\phi^{(0)}$. In general, like for the non–Gaussian specific priors discussed in Section 6, $\mathbf{K}$ can be ϕ–dependent. Eq. (368) shows that general nonlinear parameterizations $P(\xi)$ lead to nonlinear operators $\mathbf{K}$.

Clearly, if allowing ϕ–dependent T, the form (608) is no restriction of generality. One always can choose an arbitrary invertible (and not too ill–conditioned) $\mathbf{A}$, define

$$T_{\mathbf{A}} = G(\phi) + \mathbf{A}\phi, \tag{610}$$

write a stationarity equation as

$$\mathbf{A}\phi \;= T_{\mathbf{A}}, \tag{611}$$

discretize and iterate with $\mathbf{A}^{-1}$. To obtain a numerical iteration scheme we will choose a linear, positive definite *learning matrix* $\mathbf{A}$. The learning matrix may depend on ϕ and may also change during iteration.

To connect a stationarity equation given in form (608) to an arbitrary iteration scheme with a learning matrix $\mathbf{A}$ we define

$$\mathbf{B} = \mathbf{K} - \mathbf{A}, \quad \mathbf{B}_\eta = \mathbf{K} - \frac{1}{\eta}\mathbf{A}, \tag{612}$$

i.e., we split $\mathbf{K}$ into two parts

$$\mathbf{K} = \mathbf{A} + \mathbf{B} = \frac{1}{\eta}\mathbf{A} + \mathbf{B}_\eta, \tag{613}$$

where we introduced η for later convenience. Then we obtain from the stationarity equation (608)

$$\phi = \eta\mathbf{A}^{-1}(T - \mathbf{B}_\eta\phi). \tag{614}$$

To iterate we start by inserting an approximate solution $\phi^{(i)}$ to the right–hand side and obtain a new $\phi^{(i+1)}$ by calculating the left hand side. This can be written in one of the following equivalent forms

$$\begin{aligned}
\phi^{(i+1)} &= \eta \mathbf{A}^{-1}\left(T^{(i)} - \mathbf{B}_\eta \phi^{(i)}\right) && (615)\\
&= (1-\eta)\phi^{(i)} + \eta \mathbf{A}^{-1}\left(T^{(i)} - \mathbf{B}\phi^{(i)}\right) && (616)\\
&= \phi^{(i)} + \eta \mathbf{A}^{-1}\left(T^{(i)} - \mathbf{K}\phi^{(i)}\right), && (617)
\end{aligned}$$

where η plays the role of a learning rate or step width, and $\mathbf{A}^{-1} = \left(\mathbf{A}^{(i)}\right)^{-1}$ may be iteration dependent. The update equations (615–617) can be written

$$\Delta\phi^{(i)} = \eta \mathbf{A}^{-1} G(\phi^{(i)}), \tag{618}$$

with $\Delta\phi^{(i)} = \phi^{(i+1)} - \phi^{(i)}$. Eq. (617) does not require the calculation of $\mathbf{B}$ or $\mathbf{B}_\eta$ so that instead of $\mathbf{A}$ directly $\mathbf{A}^{-1}$ can be given without the need to calculate its inverse. For example operators approximating $\mathbf{K}^{-1}$ and being easy to calculate may be good choices for $\mathbf{A}^{-1}$.

For positive definite $\mathbf{A}$ (and thus also positive definite inverse) convergence can be guaranteed, at least theoretically. Indeed, multiplying with $(1/\eta)\mathbf{A}$ and projecting onto an infinitesimal $d\phi$ gives

$$\frac{1}{\eta}(\, d\phi,\, \mathbf{A}\,\Delta\phi\,) = \left(\, d\phi,\, \left.\frac{\delta(-E_\phi)}{\delta\phi}\right|_{\phi=\phi^{(i)}}\,\right) = d(-E_\phi). \tag{619}$$

In an infinitesimal neighborhood of $\phi^{(i)}$ where $\Delta\phi^{(i)}$ becomes equal to $d\phi$ in first order the left–hand side is for positive (semi–)definite $\mathbf{A}$ larger (or equal) to zero. This shows that at least for η small enough the posterior log–probability $-E_\phi$ increases i.e., the differential dE_ϕ is smaller or equal to zero and the value of the error functional E_ϕ decreases.

Stationarity equation (127) for minimizing E_L yields for (615,616,617),

$$\begin{aligned}
L^{(i+1)} &= \eta \mathbf{A}^{-1}\left(N - \mathbf{\Lambda}_X^{(i)} e^{L^{(i)}} - \mathbf{K}L^{(i)} + \frac{1}{\eta}\mathbf{A}L^{(i)}\right) && (620)\\
&= (1-\eta)L^{(i)} + \eta \mathbf{A}^{-1}\left(N - \mathbf{\Lambda}_X^{(i)} e^{L^{(i)}} - \mathbf{K}L^{(i)} + \mathbf{A}L^{(i)}\right) && (621)\\
&= L^{(i)} + \eta \mathbf{A}^{-1}\left(N - \mathbf{\Lambda}_X^{(i)} e^{L^{(i)}} - \mathbf{K}L^{(i)}\right). && (622)
\end{aligned}$$

The function $\Lambda_X^{(i)}$ is also unknown and is part of the variables we want to solve for. The normalization conditions provide the necessary additional equations, and the matrix $\mathbf{A}^{-1}$ can be extended to include the iteration procedure for Λ_X. For example, we can insert the stationarity equation for Λ_X in (622) to get

$$L^{(i+1)} = L^{(i)} + \eta \mathbf{A}^{-1} \left[N - e^{\mathbf{L}^{(i)}} (N_X - \mathbf{I}_X \mathbf{K} L) - \mathbf{K} L^{(i)} \right]. \tag{623}$$

If normalizing $L^{(i)}$ at each iteration this corresponds to an iteration procedure for $g = L + \ln Z_X$.

Similarly, for the functional E_P we have to solve (166) and obtain for (617),

$$\begin{aligned} P^{(i+1)} &= P^{(i)} + \eta \mathbf{A}^{-1} \left(T_P^{(i)} - \mathbf{K} P^{(i)} \right) & (624) \\ &= P^{(i)} + \eta \mathbf{A}^{-1} \left(\mathbf{P^{(i)}}^{-1} N - \Lambda_X^{(i)} - \mathbf{K} P^{(i)} \right) & (625) \\ &= P^{(i)} + \eta \mathbf{A}^{-1} \left(\mathbf{P^{(i)}}^{-1} N - N_X - \mathbf{I}_X \mathbf{P}^{(i)} \mathbf{K} P^{(i)} - \mathbf{K} P^{(i)} \right). & (626) \end{aligned}$$

Again, normalizing P at each iteration this is equivalent to solving for $z = Z_X P$, and the update procedure for Λ_X can be varied.

7.2 Learning matrices

7.2.1 Learning algorithms for density estimation

There exists a variety of well developed numerical methods for unconstraint as well as for constraint optimization [190, 57, 88, 196, 89, 12, 19, 80, 193]. Popular examples include conjugate gradient, Newton, and quasi–Newton methods, like the variable metric methods DFP (Davidon–Fletcher–Powell) or BFGS (Broyden–Fletcher–Goldfarb–Shanno).

All of them correspond to the choice of specific, often iteration dependent, learning matrices $\mathbf{A}$ defining the learning algorithm. Possible simple choices are:

$$\begin{aligned} \mathbf{A} = \mathbf{I} &: \text{ Gradient} & (627) \\ \mathbf{A} = \mathbf{D} &: \text{ Jacobi} & (628) \\ \mathbf{A} = \mathbf{L} + \mathbf{D} &: \text{ Gauss–Seidel} & (629) \\ \mathbf{A} = \mathbf{K} &: \text{ prior relaxation} & (630) \end{aligned}$$

where $\mathbf{I}$ stands for the identity operator, $\mathbf{D}$ for a diagonal matrix, e.g. the diagonal part of $\mathbf{K}$, and $\mathbf{L}$ for a lower triangular matrix, e.g. the lower triangular part of $\mathbf{K}$. In case $\mathbf{K}$ represents the operator of the prior term in an error functional we will call iteration with $\mathbf{K}^{-1}$ (corresponding to the covariance of the prior process) *prior relaxation*. For ϕ–independent $\mathbf{K}$ and T, $\eta = 1$ with invertible $\mathbf{K}$ the corresponding linear equation is solved by prior relaxation in one step. However, also linear equations are solved by iteration if the size of $\mathbf{K}$ is too large to be inverted. Because of $\mathbf{I}^{-1} = \mathbf{I}$ the gradient algorithm does not require inversion.

On one hand, density estimation is a rather general problem requiring the solution of constraint, inhomogeneous, nonlinear (integro–)differential equations. On the other hand, density estimation problems are, at least for Gaussian specific priors and non restricting parameterization, typically "nearly" linear and have only a relatively simple non–negativity and normalization constraint. Furthermore, the inhomogeneities are commonly restricted to a finite number of discrete training data points. Thus, we expect the inversion of $\mathbf{K}$ to be the essential part of the solution for density estimation problems. However, $\mathbf{K}$ is not necessarily invertible or may be difficult to calculate. Also, inversion of $\mathbf{K}$ is not exactly what is optimal and there are improved methods. Thus, we will discuss in the following basic optimization methods adapted especially to the situation of density estimation.

7.2.2 Linearization and Newton algorithm

For linear equations $\mathbf{K}\phi = T$ where T and $\mathbf{K}$ are no functions of ϕ a spectral radius $\rho(\mathbf{M}) < 1$ (the largest modulus of the eigenvalues) of the *iteration matrix*

$$\mathbf{M} = -\eta \mathbf{A}^{-1}\mathbf{B}_\eta = (1-\eta)\mathbf{I} - \eta\mathbf{A}^{-1}\mathbf{B} = \mathbf{I} - \eta\mathbf{A}^{-1}\mathbf{K} \tag{631}$$

would guarantee convergence of the iteration scheme. This is easily seen by solving the linear equation by iteration according to (615)

$$\begin{aligned}
\phi^{(i+1)} &= \eta\mathbf{A}^{-1}T + \mathbf{M}\phi^{(i)} && (632)\\
&= \eta\mathbf{A}^{-1}T + \eta\mathbf{M}\mathbf{A}^{-1}T + \mathbf{M}^2\phi^{(i-1)} && (633)\\
&= \eta\sum_{n=0}^{\infty}\mathbf{M}^n\mathbf{A}^{-1}T. && (634)
\end{aligned}$$

A zero mode of $\mathbf{K}$, for example a constant function for differential operators without boundary conditions, corresponds to an eigenvalue 1 of $\mathbf{M}$ and would lead to divergence of the sequence $\phi^{(i)}$. However, a nonlinear $T(\phi)$ or $\mathbf{K}(\phi)$, like the nonlinear normalization constraint contained in $T(\phi)$, can then still lead to a unique solution.

A convergence analysis for nonlinear equations can be done in a linear approximation around a fixed point. Expanding the gradient at ϕ^*

$$G(\phi) = \left.\frac{\delta(-E_\phi)}{\delta\phi}\right|_{\phi^*} + (\phi - \phi^*)\ \mathbf{H}(\phi^*) + \cdots \tag{635}$$

shows that the factor of the linear term is the Hessian. Thus in the vicinity of ϕ^* the spectral radius of the iteration matrix

$$\mathbf{M} = \mathbf{I} + \eta\mathbf{A}^{-1}\mathbf{H}, \tag{636}$$

determines the rate of convergence. The Newton algorithm uses the negative Hessian $-\mathbf{H}$ as learning matrix provided it exists and is positive definite. Otherwise it must resort to other methods. In the linear approximation (i.e., for quadratic energy) the Newton algorithm

$$\mathbf{A} = -\mathbf{H} \quad : \quad \text{Newton} \tag{637}$$

is optimal. We have already seen in Sections 3.1.3 and 3.2.3 that the inhomogeneities generate in the Hessian in addition to $\mathbf{K}$ a diagonal part which can remove zero modes of $\mathbf{K}$.

7.2.3 Massive relaxation

We now consider methods to construct a positive definite or at least invertible learning matrix. For example, far from a minimum the Hessian $\mathbf{H}$ may not be positive definite and like a differential operator $\mathbf{K}$ with zero modes, not even invertible. Massive relaxation can transform a non–invertible or not positive definite operator $\mathbf{A}_0$, e.g. $\mathbf{A}_0 = \mathbf{K}$ or $\mathbf{A}_0 = -\mathbf{H}$, into an invertible or positive definite operators:

$$\mathbf{A} = \mathbf{A}_0 + m^2\mathbf{I} \quad : \quad \text{Massive relaxation} \tag{638}$$

A generalization would be to allow $m = m(x, y)$. This is, for example, used in some realizations of Newton‘s method for minimization in regions where $\mathbf{H}$ is not positive definite and a diagonal operator is added to $-\mathbf{H}$, using for example a modified Cholesky factorization [19]. The mass term removes the zero modes of $\mathbf{K}$ if $-m^2$ is not in the spectrum of $\mathbf{A}_0$ and makes it positive definite if m^2 is larger than the smallest eigenvalue of $\mathbf{A}_0$. Matrix elements $(\,\phi,\, (\mathbf{A}_0 - z\mathbf{I})^{-1}\,\phi\,)$ of the resolvent $\mathbf{A}^{-1}(z)$, $z = -m^2$ representing in this case a complex variable, have poles at discrete eigenvalues of $\mathbf{A}_0$ and a cut at the continuous spectrum as long as ϕ has a non–zero overlap with the corresponding eigenfunctions. Instead of multiples of the identity, also other operators may be added to remove zero modes. The Hessian $\mathbf{H}_L$ in (156), for example, adds a x–dependent mass–like, but not necessarily positive definite term to $\mathbf{K}$. Similarly, for example $\mathbf{H}_P$ in (182) has (x, y)–dependent mass $\mathbf{P}^{-2}\mathbf{N}$ restricted to data points.

While full relaxation is the massless limit $m^2 \to 0$ of massive relaxation, a gradient algorithm with η' can be obtained as infinite mass limit $m^2 \to \infty$ with $\eta \to \infty$ and $m^2/\eta = 1/\eta'$.

Constant functions are typical zero modes, i.e., eigenfunctions with zero eigenvalue, for differential operators with periodic boundary conditions. For instance for a common smoothness term $-\Delta$ (kinetic energy operator) as regularizing operator $\mathbf{K}$ the inverse of $\mathbf{A} = \mathbf{K} + m^2\mathbf{I}$ has the form

$$\mathbf{A}^{-1}(x', y'; x, y) = \frac{1}{-\Delta + m^2}. \tag{639}$$

$$= \int_{-\infty}^{\infty} \frac{d^{d_X}k_x \, d^{d_Y}k_y}{(2\pi)^d} \frac{e^{ik_x(x-x')+ik_y(y-y')}}{k_x^2 + k_y^2 + m^2}, \tag{640}$$

with $d = d_X + d_Y$, $d_X = \dim(X)$, $d_Y = \dim(Y)$. This Green's function or matrix element of the resolvent kernel for $\mathbf{A}_0$ is analogous to the (Euclidean) propagator of a free scalar field with mass m, which is its two–point correlation function or matrix element of the covariance operator. According to $1/x = \int_0^\infty dt \, e^{-xt}$ the denominator can be brought into the exponent by introducing an additional integral. Performing the resulting Gaussian integration over $k = (k_x, k_y)$ the inverse can be expressed as

$$\mathbf{A}^{-1}(x', y'; x, y; m) = m^{d-2}\mathbf{A}^{-1}(m(x - x'), m(y - y'); 1)$$

$$= (2\pi)^{-d/2} \left(\frac{m}{|x - x'| + |y - y'|} \right)^{(d-2)/2} K_{(d-2)/2}(m|x - x'| + m|y - y'|), \tag{641}$$

in terms of the modified Bessel functions $K_\nu(x)$ which have the following integral representation

$$K_\nu(2\sqrt{\beta\gamma}) = \frac{1}{2} \left(\frac{\gamma}{\beta} \right)^{\frac{\nu}{2}} \int_0^\infty dt \, t^{\nu-1} e^{\frac{\beta}{t} - \gamma t}. \tag{642}$$

Alternatively, the same result can be obtained by switching to d–dimensional spherical coordinates, expanding the exponential in ultra-spheric harmonic functions and performing the integration over the angle-variables [117]. For the example $d = 1$ this corresponds to Parzens kernel used in density estimation or for $d = 3$

$$\mathbf{A}^{-1}(x', y'; x, y) = \frac{1}{4\pi|x - x'| + 4\pi|y - y'|} e^{-m|x-x'|-m|y-y'|}. \tag{643}$$

The Green's function for periodic, Neumann, or Dirichlet boundary conditions can be expressed by sums over $\mathbf{A}^{-1}(x', y'; x, y)$ [77].

The lattice version of the Laplacian with lattice spacing a reads

$$\hat{\Delta} f(n) = \frac{1}{a^2} \sum_j^d [f(n - a_j) - 2f(n) + f(n + a_j)], \tag{644}$$

writing a_j for a vector in direction j and length a. Including a mass term we get as lattice approximation for $\mathbf{A}$

$$\hat{\mathbf{A}}(n_x, n_y; m_x, m_y) = -\frac{1}{a^2} \sum_{i=1}^{d_X} \delta_{n_y,m_y} (\delta_{n_x+a_i^x,m_x} - 2\delta_{n_x,m_x} + \delta_{n_x-a_i^x,m_x})$$

$$-\frac{1}{a^2} \sum_{j=1}^{d_Y} \delta_{n_x,m_x} (\delta_{n_y+a_j^y,m_y} - 2\delta_{n_y,m_y} + \delta_{n_y-a_j^y,m_y}) + m^2 \delta_{n_x,m_x} \delta_{n_y,m_y} \tag{645}$$

Inserting the Fourier representation (101) of $\delta(x)$ gives

$$\hat{\mathbf{A}}(n_x, n_y; m_x, m_y) = \frac{2d}{a^2} \int_{-\pi}^{\pi} \frac{d^{d_X}k_x \, d^{d_Y}k_y}{(2\pi)^d} e^{ik_x(n_x-m_x)+ik_y(n_y-m_y)}$$

$$\times \left(1 + \frac{m^2 a^2}{2d} - \frac{1}{d} \sum_{i=1}^{d_X} \cos k_{x,i} - \frac{1}{d} \sum_{j=1}^{d_Y} \cos k_{y,j} \right), \tag{646}$$

with $k_{x,i} = k_x a_i^x$, $\cos k_{y,j} = \cos k_y a_j^y$ and inverse

$$\hat{\mathbf{A}}^{-1}(n_x, n_y; m_x, m_y) = \int_{-\pi}^{\pi} \frac{d^{d_X}k_x \, d^{d_Y}k_y}{(2\pi)^d} \hat{\mathbf{A}}^{-1}(k_x, k_y) e^{ik_x(n_x-m_x)+ik_y(n_y-m_y)}$$

$$= \frac{a^2}{2d} \int_{-\pi}^{\pi} \frac{d^{d_X}k_x \, d^{d_Y}k_y}{(2\pi)^d} \frac{e^{ik_x(n_x-m_x)+ik_y(n_y-m_y)}}{1 + \frac{m^2a^2}{2d} - \frac{1}{d} \sum_{i=1}^{d_X} \cos k_{x,i} - \frac{1}{d} \sum_{j=1}^{d_Y} \cos k_{y,j}}. \tag{647}$$

(For $m = 0$ and $d \le 2$ the integrand diverges for $k \to 0$ (infrared divergence). Subtracting formally the also infinite $\hat{\mathbf{A}}^{-1}(0,0;0,0)$ results in finite difference. For example in $d = 1$ one finds $\hat{\mathbf{A}}^{-1}(n_y;m_y) - \hat{\mathbf{A}}^{-1}(0;0) = -(1/2)|n_y - m_y|$ [103]. Using $1/x = \int_0^\infty dt\, e^{-xt}$ one obtains [195]

$$\hat{\mathbf{A}}^{-1}(k_x,k_y) = \frac{1}{2}\int_0^\infty dt\, e^{-\mu t + a^{-2}t\left(\sum_i^{d_X}\cos k_{x,i} + \sum_j^{d_Y}\cos k_{y,j}\right)}, \tag{648}$$

with $\mu = d/a^2 + m^2/2$. This allows to express the inverse $\hat{\mathbf{A}}^{-1}$ in terms of the modified Bessel functions $I_\nu(n)$ which have for integer argument n the integral representation

$$I_\nu(n) = \frac{1}{\pi}\int_0^\pi d\Theta\, e^{n\cos\Theta}\cos(\nu\Theta). \tag{649}$$

One finds

$$\hat{\mathbf{A}}^{-1}(n_x,n_y;m_x,m_y) = \frac{1}{2}\int_0^\infty e^{-\mu t}\prod_{i=1}^{d_X} K_{|n_{x,i}-n'_{x,i}|}(t/a^2)\prod_{j=1}^{d_Y} K_{|m_{y,j}-m'_{y,j}|}(t/a^2). \tag{650}$$

It might be interesting to remark that the matrix elements of the inverse learning matrix or free massive propagator on the lattice $\hat{\mathbf{A}}^{-1}(x',y';x,y)$ can be given an interpretation in terms of (random) walks connecting the two points (x',y') and (x,y) [56, 195]. For that purpose the lattice Laplacian is splitted into a diagonal and a nearest neighbor part

$$-\hat{\Delta} = \frac{1}{a^2}\left(2d\mathbf{I} - \mathbf{W}\right), \tag{651}$$

where the nearest neighbor matrix $\mathbf{W}$ has matrix elements equal one for nearest neighbors and equal to zero otherwise. Thus,

$$\left(-\hat{\Delta} + m^2\right)^{-1} = \frac{1}{2\mu}\left(\mathbf{I} - \frac{1}{2\mu a^2}\mathbf{W}\right)^{-1} = \frac{1}{2\mu}\sum_{n=0}^\infty\left(\frac{1}{2\mu a^2}\right)^n\mathbf{W}^n, \tag{652}$$

can be written as geometric series. The matrix elements $\mathbf{W}^n(x',y';x,y)$ give the number of walks $w[(x',y')\to(x,y)]$ of length $|w| = n$ connecting the two points (x',y') and (x,y). Thus, one can write

$$\left(-\hat{\Delta} + m^2\right)^{-1}(x',y';x,y) = \frac{1}{2\mu}\sum_{w[(x',y')\to(x,y)]}\left(\frac{1}{2\mu a^2}\right)^{|w|}. \tag{653}$$

7.2.4 Gaussian relaxation

As Gaussian kernels are often used in density estimation and also in function approximation (e.g. for radial basis functions [191]) we consider the example

$$\mathbf{A} = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\frac{\mathbf{M}^2}{2\tilde{\sigma}^2} \right)^k = e^{\frac{\mathbf{M}^2}{2\tilde{\sigma}^2}} \quad : \quad \text{Gaussian} \tag{654}$$

with positive semi–definite $\mathbf{M}^2$. The contribution for $k = 0$ corresponds to a mass term so for positive semi–definite $\mathbf{M}$ this $\mathbf{A}$ is positive definite and therefore invertible with inverse

$$\mathbf{A}^{-1} = e^{-\frac{\mathbf{M}^2}{2\tilde{\sigma}^2}}, \tag{655}$$

which is diagonal and Gaussian in $\mathbf{M}$–representation. In the limit $\tilde{\sigma} \to \infty$ or for zero modes of $\mathbf{M}$ the Gaussian $\mathbf{A}^{-1}$ becomes the identity $\mathbf{I}$, corresponding to the gradient algorithm. Consider

$$\mathbf{M}^2(x', y'; x, y) = -\delta(x - x')\delta(y - y')\Delta \tag{656}$$

where the δ–functions are usually skipped from the notation, and

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2},$$

denotes the Laplacian. The kernel of the inverse is diagonal in Fourier representation

$$A(k_x', k_y'; , k_x, k_y) = \delta(k_x - k_x')\delta(k_y - k_y') e^{-\frac{k_x^2 + k_y^2}{2\tilde{\sigma}^2}} \tag{657}$$

and non–diagonal, but also Gaussian in (x, y)–representation

$$\mathbf{A}^{-1}(x', y'; x, y) = e^{-\frac{\Delta}{2\tilde{\sigma}^2}} = \int \frac{dk_x dk_y}{(2\pi)^d} \, e^{-\frac{k_x^2 + k_y^2}{2\tilde{\sigma}^2} + ik_x(x - x') + ik_y(y - y')} \tag{658}$$

$$= \left(\frac{\tilde{\sigma}}{\sqrt{2\pi}} \right)^d e^{-\tilde{\sigma}^2((x-x')^2 + (y-y')^2)} = \frac{1}{\left(\sigma\sqrt{2\pi}\right)^d} \, e^{-\frac{(x-x')^2 + (y-y')^2}{2\sigma^2}}, \tag{659}$$

with $\sigma = 1/\tilde{\sigma}$ and $d = d_X + d_Y$, $d_X = \dim(X)$, $d_Y = \dim(Y)$.

7.2.5 Inverting in subspaces

Matrices considered as learning matrix have to be invertible. Non-invertible matrices can only be inverted in the subspace which is the complement of its zero space. With respect to a symmetric $\mathbf{A}$ we define the projector $\mathbf{Q}_0 = \mathbf{I} - \sum_i \psi_i^T \psi_i$ into its zero space (for the more general case of a normal $\mathbf{A}$ replace ψ_i^T by the hermitian conjugate $\psi_i^\dagger$) and its complement $\mathbf{Q}_1 = \mathbf{I} - \mathbf{Q}_0 = \sum_i \psi_i^T \psi_i$ with ψ_i denoting orthogonal eigenvectors with eigenvalues $a_i \neq 0$ of $\mathbf{A}$, i.e., $\mathbf{A}\psi_i = a_i \psi_i \neq 0$. Then, denoting projected sub-matrices by $\mathbf{Q}_i \mathbf{A} \mathbf{Q}_j = \mathbf{A}_{ij}$ we have $\mathbf{A}_{00} = \mathbf{A}_{10} = \mathbf{A}_{01} = 0$, i.e.,

$$\mathbf{A} = \mathbf{Q}_1 \mathbf{A} \mathbf{Q}_1 = \mathbf{A}_{11}. \tag{660}$$

and in the update equation

$$\mathbf{A}\Delta\phi^{(i)} = \eta\, G \tag{661}$$

only $\mathbf{A}_{11}$ can be inverted. Writing $\mathbf{Q}_j \phi = \phi_j$ for a projected vector, the iteration scheme acquires the form

$$\Delta\phi_1^{(i)} = \eta \mathbf{A}_{11}^{-1} G_1, \tag{662}$$

$$0 = \eta\, G_0. \tag{663}$$

For positive semi–definite $\mathbf{A}$ the sub-matrix $\mathbf{A}_{11}$ is positive definite. If the second equation is already fulfilled or its solution is postponed to a later iteration step we have

$$\phi_1^{(i+1)} = \phi_1^{(i)} + \eta \mathbf{A}_{11}^{-1} \left(T_1^{(i)} - \mathbf{K}_{11}^{(i)} \phi_1^{(i)} - \mathbf{K}_{10}^{(i)} \phi_0^{(i)} \right), \tag{664}$$

$$\phi_0^{(i+1)} = \phi_0^{(i)}. \tag{665}$$

In case the projector $\mathbf{Q}_0 = \mathbf{I}_0$ is diagonal in the chosen representation the projected equation can directly be solved by skipping the corresponding components. Otherwise one can use the Moore–Penrose inverse $\mathbf{A}^\#$ of $\mathbf{A}$ to solve the projected equation

$$\Delta\phi^{(i)} = \eta \mathbf{A}^\# G. \tag{666}$$

Alternatively, an invertible operator $\tilde{\mathbf{A}}_{00}$ can be added to $\mathbf{A}_{11}$ to obtain a complete iteration scheme with $\mathbf{A}^{-1} = \mathbf{A}_{11}^{-1} + \tilde{\mathbf{A}}_{00}^{-1}$

$$\begin{aligned} \phi^{(i+1)} = {} & \phi^{(i)} + \eta \mathbf{A}_{11}^{-1} \left(T_1^{(i)} - \mathbf{K}_{11}^{(i)} \phi_1^{(i)} - \mathbf{K}_{10}^{(i)} \phi_0^{(i)} \right) \\ & + \eta \tilde{\mathbf{A}}_{00}^{-1} \left(T_0^{(i)} - \mathbf{K}_{01}^{(i)} \phi_1^{(i)} - \mathbf{K}_{00}^{(i)} \phi_0^{(i)} \right). \end{aligned} \tag{667}$$

The choice $\mathbf{A}^{-1} = (\mathbf{A}_{11} + \mathbf{I}_{00})^{-1} = \mathbf{A}_{11}^{-1} + \mathbf{I}_{00}, = \mathbf{A}_{11}^{-1} + \mathbf{Q}_0$, for instance, results in a gradient algorithm on the zero space with additional coupling between the two subspaces.

7.2.6 Boundary conditions

For a differential operator invertability can be achieved by adding an operator restricted to a subset $B \subset X \times Y$ (boundary). More general, we consider an projector $\mathbf{Q}_B$ on a space which we will call boundary and the projector on the interior $\mathbf{Q}_I = \mathbf{I} - \mathbf{Q}_B$. We write $\mathbf{Q}_k\mathbf{K}\mathbf{Q}_l = \mathbf{K}_{kl}$ for $k, l \in \{I, B\}$, and require $\mathbf{K}_{BI} = 0$. That means $\mathbf{K}$ is not symmetric, but $\mathbf{K}_{II}$ can be, and we have

$$\mathbf{K} = (\mathbf{I} - \mathbf{Q}_B)\mathbf{K} + \mathbf{Q}_B\mathbf{K}\mathbf{Q}_B = \mathbf{K}_{II} + \mathbf{K}_{IB} + \mathbf{K}_{BB}. \tag{668}$$

For such an $\mathbf{K}$ an equation of the form $\mathbf{K}\phi = T$ can be decomposed into

$$\mathbf{K}_{BB}\phi_B = T_B, \tag{669}$$

$$\mathbf{K}_{IB}\phi_B + \mathbf{K}_{II}\phi_I = T_I, \tag{670}$$

with projected $\phi_k = \mathbf{Q}_k\phi$, $T_k = \mathbf{Q}_k T$, so that

$$\phi_B = \mathbf{K}_{BB}^{-1} T_B, \tag{671}$$

$$\phi_I = \mathbf{K}_{II}^{-1}\left(T_I - \mathbf{K}_{IB}\mathbf{K}_{BB}^{-1}T_B\right). \tag{672}$$

The boundary part is independent of the interior, however, the interior can depend on the boundary. A basis can be chosen so that the projector onto the boundary is diagonal, i.e.,

$$\mathbf{Q}_B = \mathbf{I}_B = \sum_{j:(x_j,y_j)\in B} (\delta(x_j) \otimes \delta(y_j)) \otimes (\delta(x_j) \otimes \delta(y_j))^T.$$

Eliminating the boundary results in an equation for the interior with adapted inhomogeneity. The special case $\mathbf{K}_{BB} = \mathbf{I}_{BB}$, i.e., $\phi_B = T_B$ on the boundary, is known as Dirichlet boundary conditions.

As trivial example of an equation $\mathbf{K}\phi = T$ with boundary conditions, consider a one–dimensional finite difference approximation for a negative

Laplacian $\mathbf{K}$, adapted to include boundary conditions as in Eq. (668),

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \end{pmatrix} = \begin{pmatrix} b \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ b \end{pmatrix}. \tag{673}$$

Then Eq. (669) is equivalent to the boundary conditions, $\phi_1 = b$, $\phi_6 = b$, and the interior equation Eq. (670) reads

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} T_2 \\ T_3 \\ T_4 \\ T_5 \end{pmatrix} + \begin{pmatrix} b \\ 0 \\ 0 \\ b \end{pmatrix}. \tag{674}$$

(Useful references dealing with the numerical solution of partial differential equations are, for example, [8, 161, 87, 196, 84].)

Similarly to boundary conditions for $\mathbf{K}$, we may use a learning matrix $\mathbf{A}$ with boundary conditions (corresponding for example to those used for $\mathbf{K}$):

$$\mathbf{A} = \mathbf{A}_{II} + \mathbf{A}_{IB} + \mathbf{A}_{BB} \quad : \quad \text{Boundary} \tag{675}$$

$$\mathbf{A} = \mathbf{A}_{II} + \mathbf{A}_{IB} + \mathbf{I}_{BB} \quad : \quad \text{Dirichlet boundary} \tag{676}$$

For linear $\mathbf{A}_{BB}$ the form (675) corresponds to general linear boundary conditions. (It is also possible to require nonlinear boundary conditions.) $\mathbf{A}_{II}$ can be chosen symmetric, and therefore positive definite, and the boundary of $\mathbf{A}$ can be changed during iteration. Solving $\mathbf{A}(\phi^{(i+1)} - \phi^{(i)}) = \eta(T^{(i)} - \mathbf{K}^{(i)}\phi^{(i)})$ gives on the boundary and for the interior

$$\phi_B^{(i+1)} = \phi_B^i + \eta \mathbf{A}_{BB}^{-1} \left(T_B^{(i)} - \mathbf{K}_{BB}^{(i)} \phi_B^{(i)} - \mathbf{K}_{BI}^{(i)} \phi_I^{(i)} \right), \tag{677}$$

$$\phi_I^{(i+1)} = \phi_I^i + \eta \mathbf{A}_{II}^{-1} \left(T_I^{(i)} - \mathbf{K}_{II}^{(i)} \phi_I^{(i)} - \mathbf{K}_{IB}^{(i)} \phi_B^{(i)} \right) - \mathbf{A}_{II}^{-1} \mathbf{A}_{IB} \left(\phi_B^{(i+1)} - \phi_B^{(i)} \right), \tag{678}$$

For fulfilled boundary conditions with $\phi_B^{(i)} = \left(\mathbf{K}_{BB}^{(i)}\right)^{-1} T_B^{(i)}$ and $\mathbf{K}_{BI}^{(i)} = 0$, or for $\eta \mathbf{A}_{BB}^{-1} \to 0$ so the boundary is not updated, the term $\phi_B^{(i+1)} - \phi_B^{(i)}$ vanishes. Otherwise, inserting the first in the second equation gives

$$\begin{aligned} \phi_I^{(i+1)} &= \phi_I^i + \eta \mathbf{A}_{II}^{-1} \left(T_I^{(i)} - \mathbf{K}_{II}^{(i)} \phi_I^{(i)} - \mathbf{K}_{IB}^{(i)} \phi_B^{(i)} \right) \\ &\quad - \eta \mathbf{A}_{II}^{-1} \mathbf{A}_{IB} \mathbf{A}_{BB}^{-1} \left(T_B^{(i)} - \mathbf{K}_{BB}^{(i)} \phi_B^{(i)} - \mathbf{K}_{BI}^{(i)} \phi_I^{(i)} \right). \end{aligned} \tag{679}$$

Even if $\mathbf{K}$ is not defined with boundary conditions, an invertible $\mathbf{A}$ can be obtained from $\mathbf{K}$ by introducing a boundary for $\mathbf{A}$. The updating process is then restricted to the interior. In such cases the boundary should be systematically changed during iteration. Block–wise updating of ϕ represent a special case of such learning matrices with variable boundary.

The following table summarizes the learning matrices we have discussed in some detail for the setting of density estimation (for conjugate gradient and quasi–Newton methods see, for example, [196]):

Learning algorithm	Learning matrix
Gradient	$\mathbf{A} = \mathbf{I}$
Jacobi	$\mathbf{A} = \mathbf{D}$
Gauss–Seidel	$\mathbf{A} = \mathbf{L} + \mathbf{D}$
Newton	$\mathbf{A} = -\mathbf{H}$
prior relaxation	$\mathbf{A} = \mathbf{K}$
massive relaxation	$\mathbf{A} = \mathbf{A}_0 + m^2\mathbf{I}$
linear boundary	$\mathbf{A} = \mathbf{A}_{II} + \mathbf{A}_{IB} + \mathbf{A}_{BB}$
Dirichlet boundary	$\mathbf{A} = \mathbf{A}_{II} + \mathbf{A}_{IB} + \mathbf{I}_{BB}$
Gaussian	$\mathbf{A} = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\frac{\mathbf{M}^2}{2\tilde{\sigma}^2}\right)^k = e^{\frac{\mathbf{M}^2}{2\tilde{\sigma}^2}}$

7.3 Initial configurations and kernel methods

7.3.1 Truncated equations

To solve the nonlinear Eq. (609) by iteration one has to begin with an initial configuration $\phi^{(0)}$. In principle any easy to use technique for density estimation could be chosen to construct starting guesses $\phi^{(0)}$.

One possibility to obtain initial guesses is to neglect some terms of the full stationarity equation and solve the resulting simpler (ideally linear) equation first. The corresponding solution may be taken as initial guess $\phi^{(0)}$ for solving the full equation.

Typical error functionals for statistical learning problems include a term $(L,\, N)$ consisting of a discrete sum over a finite number n of training data. For diagonal $\mathbf{P}'$ those contributions result (355) in n δ–peak contributions to the inhomogeneities T of the stationarity equations, like $\sum_i \delta(x-x_i)\delta(y-y_i)$ in Eq. (143) or $\sum_i \delta(x-x_i)\delta(y-y_i)/P(x,y)$ in Eq. (172). To find an initial guess, one can now keep only that δ–peak contributions T_δ arising from the training data and ignore the other, typically continuous parts of T. For (143) and (172) this means setting $\Lambda_X = 0$ and yields a truncated equation

$$\mathbf{K}\phi = \mathbf{P}'\mathbf{P}^{-1}N = T_\delta. \tag{680}$$

Hence, ϕ can for diagonal $\mathbf{P}'$ be written as a sum of n terms

$$\phi(x,y) = \sum_{i=1}^{n} \mathbf{C}(x,y;x_i,y_i)\frac{P'(x_i,y_i)}{P(x_i,y_i)}, \tag{681}$$

with $\mathbf{C} = \mathbf{K}^{-1}$, provided the inverse $\mathbf{K}^{-1}$ exists. For E_L the resulting truncated equation is linear in L. For E_P, however, the truncated equations remains nonlinear. Having solved the truncated equation we restore the necessary constraints for ϕ, like normalization and non–negativity for P or normalization of the exponential for L.

In general, a $\mathbf{C} \neq \mathbf{K}^{-1}$ can be chosen. This is necessary if $\mathbf{K}$ is not invertible and can also be useful if its inverse is difficult to calculate. One possible choice for the kernel is the inverse negative Hessian $\mathbf{C} = -\mathbf{H}^{-1}$ evaluated at some initial configuration $\phi^{(0)}$ or an approximation of it. A simple possibility to construct an invertible operator from a noninvertible $\mathbf{K}$ would be to add a mass term

$$\mathbf{C} = \left(\mathbf{K} + m_C^2\mathbf{I}\right)^{-1}, \tag{682}$$

or to impose additional boundary conditions.

Solving a truncated equation of the form (681) with $\mathbf{C}$ means skipping the term $-\mathbf{C}(\mathbf{P}'\Lambda_X + (\mathbf{K}-\mathbf{C}^{-1})\phi)$ from the exact relation

$$\phi = \mathbf{C}\mathbf{P}'\mathbf{P}^{-1}N - \mathbf{C}(\mathbf{P}'\Lambda_X + (\mathbf{K}-\mathbf{C}^{-1})\phi). \tag{683}$$

A kernel used to create an initial guess $\phi^{(0)}$ will be called an *initializing kernel*.

A similar possibility is to start with an "empirical solution"

$$\phi^{(0)} = \phi_{\text{emp}}, \tag{684}$$

where ϕ_{emp} is defined as a ϕ which reproduces the conditional empirical density P_{emp} of Eq. (236) obtained from the training data, i.e.,

$$P_{\text{emp}} = P(\phi_{\text{emp}}). \tag{685}$$

In case, there are not data points for every x–value, a correctly normalized initial solution would for example be given by $\tilde{P}_{\text{emp}}$ defined in Eq. (238). If zero values of the empirical density correspond to infinite values for ϕ, like in the case $\phi = L$, one can use $P^{\epsilon}_{\rm emp}$ as defined in Eq. (239), with small ϵ, to obtain an initial guess.

Similarly to Eq. (681), it is often also useful to choose a (for example smoothing) kernel $\mathbf{C}$ and use as initial guess

$$\phi^{(0)} = \mathbf{C}\phi_{\rm emp}, \tag{686}$$

or a properly normalized version thereof. Alternatively, one may also let the (smoothing) operator $\mathbf{C}$ directly act on $P_{\rm emp}$ and use a corresponding ϕ as initial guess,

$$\phi^{(0)} = (\phi)^{(-1)}(\mathbf{C}P_{\rm emp}), \tag{687}$$

assuming an inverse $(\phi)^{(-1)}$ of the mapping $P(\phi)$ exists.

We will now discuss the cases $\phi = L$ and $\phi = P$ in some more detail.

7.3.2 Kernels for L

For E_L we have the truncated equation

$$L = \mathbf{C}N. \tag{688}$$

Normalizing the exponential of the solution gives

$$L(x,y) = \sum_i^n \mathbf{C}(x,y;x_i,y_i) - \ln \int dy'\, e^{\sum_i^n \mathbf{C}(x,y';x_i,y_i)}, \tag{689}$$

or

$$L = \mathbf{C}N - \ln \mathbf{I}_X \mathbf{e}^{\mathbf{C}N}. \tag{690}$$

Notice that normalizing L according to Eq. (689) after each iteration the truncated equation (688) is equivalent to a one–step iteration with uniform $P^{(0)} = e^{L^{(0)}}$ according to

$$L^1 = \mathbf{C}N + \mathbf{C}\mathbf{P}^{(0)}\Lambda_X, \tag{691}$$

where only $(\mathbf{I} - \mathbf{C}\mathbf{K})L$ is missing from the nontruncated equation (683), because the additional y–independent term $\mathbf{C}\mathbf{P}^{(0)}\Lambda_X$ becomes inessential if L is normalized afterwards.

Lets us consider as example the choice $\mathbf{C} = -\mathbf{H}^{-1}(\phi^{(0)})$ for uniform initial $L^{(0)} = c$ corresponding to a normalized P and $\mathbf{K}L^{(0)} = 0$ (e.g., a differential operator). Uniform $L^{(0)}$ means uniform $P^{(0)} = 1/v_y$, assuming that $v_y = \int dy$ exists and, according to Eq. (137), $\Lambda_X = N_X$ for $\mathbf{K}L^{(0)} = 0$. Thus, the Hessian (161) at $L^{(0)}$ is found as

$$\mathbf{H}(L^{(0)}) = -\left(\mathbf{I} - \frac{\mathbf{I}_X}{v_y}\right)\mathbf{K}\left(\mathbf{I} - \frac{\mathbf{I}_X}{v_y}\right) - \left(\mathbf{I} - \frac{\mathbf{I}_X}{v_y}\right)\frac{\mathbf{N}_X}{v_y} = -\mathbf{C}^{-1}, \qquad (692)$$

which can be invertible due to the presence of the second term.

Another possibility is to start with an approximate empirical log–density, defined as

$$L^{\epsilon}_{\rm emp} = \ln P^{\epsilon}_{\rm emp}, \qquad (693)$$

with $P^{\epsilon}_{\rm emp}$ given in Eq. (239). Analogously to Eq. (686), the empirical log–density may for example also be smoothed and correctly normalized again, resulting in an initial guess,

$$L^{(0)} = \mathbf{C}L^{\epsilon}_{\rm emp} - \ln \mathbf{I}_X \mathbf{e}^{\mathbf{C}L^{\epsilon}_{\rm emp}}. \qquad (694)$$

Similarly, one may let a kernel $\mathbf{C}$, or its normalized version $\tilde{\mathbf{C}}$ defined below in Eq. (698), act on $P_{\rm emp}$ first and then take the logarithm

$$L^{(0)} = \ln(\tilde{\mathbf{C}}P^{\epsilon}_{\rm emp}). \qquad (695)$$

Because already $\tilde{\mathbf{C}}P_{\rm emp}$ is typically nonzero it is most times not necessary to work here with $P^{\epsilon}_{\rm emp}$. Like in the next section $P_{\rm emp}$ may be also be replaced by $\tilde{P}_{\rm emp}$ as defined in Eq. (238).

7.3.3 Kernels for P

For E_P the truncated equation

$$P = \mathbf{C}\mathbf{P}^{-1}N, \qquad (696)$$

is still nonlinear in P. If we solve this equation approximately by a one–step iteration $P^1 = \mathbf{C}(\mathbf{P}^{(0)})^{-1}N$ starting from a uniform initial $P^{(0)}$ and normalizing afterwards this corresponds for a single x–value to the classical kernel methods commonly used in density estimation. As normalized density results

$$P(x,y) = \frac{\sum_i \mathbf{C}(x,y;x_i,y_i)}{\int dy' \sum_i \mathbf{C}(x,y';x_i,y_i)} = \sum_i \bar{\mathbf{C}}(x,y;x_i,y_i), \qquad (697)$$

i.e.,

$$P = \mathbf{N}_{K,X}^{-1}\mathbf{C}N = \bar{\mathbf{C}}N, \tag{698}$$

with (data dependent) normalized kernel $\bar{\mathbf{C}} = \mathbf{N}_{C,X}^{-1}\mathbf{C}$ and $\mathbf{N}_{C,X}$ the diagonal matrix with diagonal elements $\mathbf{I}_X\mathbf{C}N$. Again $\mathbf{C} = \mathbf{K}^{-1}$ or similar invertible choices can be used to obtain a starting guess for P. The form of the Hessian (182) suggests in particular to include a mass term on the data.

It would be interesting to interpret Eq. (698) as stationarity equation of a functional $\hat{E}_P$ containing the usual data term $\sum_i \ln P(x_i, y_i)$. Therefore, to obtain the derivative $\mathbf{P}^{-1}N$ of this data term we multiply for existing $\bar{\mathbf{C}}^{-1}$ Eq. (698) by $\mathbf{P}^{-1}\bar{\mathbf{C}}^{-1}$, where $P \neq 0$ at data points, to obtain

$$\widetilde{\mathbf{C}}^{-1}P = \mathbf{P}^{-1}N, \tag{699}$$

with data dependent

$$\widetilde{\mathbf{C}}^{-1}(x, y; x', y') = \frac{\bar{\mathbf{C}}^{-1}(x, y; x', y')}{\sum_i \bar{\mathbf{C}}(x, y; x_i, y_i)}. \tag{700}$$

Thus, Eq. (698) is the stationarity equation of the functional

$$\hat{E}_P = -(\, N,\, \ln P\,) + \frac{1}{2}(\, P,\, \widetilde{\mathbf{C}}^{-1}\, P\,). \tag{701}$$

To study the dependence on the number n of training data for a given $\mathbf{C}$ consider a normalized kernel with $\int dy\, \mathbf{C}(x, y; x', y') = \lambda$, $\forall x, x', y'$. For such a kernel the denominator of $\bar{\mathbf{C}}$ is equal to $n\lambda$ so we have

$$\bar{\mathbf{C}} = \frac{\mathbf{C}}{n\lambda}, \quad P = \frac{\mathbf{C}N}{n\lambda} \tag{702}$$

Assuming that for large n the empirical average $(1/n)\sum_i \mathbf{C}(x, y; x_i, y_i)$ in the denominator of $\widetilde{\mathbf{C}}^{-1}$ becomes n independent, e.g., converging to the true average $n\int dx'dy'\, p(x', y')\mathbf{C}(x, y; x', y')$, the regularizing term in functional (701) becomes proportional to n

$$\widetilde{\mathbf{C}}^{-1} \propto n\lambda^2, \tag{703}$$

According to Eq. (76) this would allow to relate a saddle point approximation to a large n–limit.

Again, a similar possibility is to start with the empirical density $\tilde{P}_{\text{emp}}$ defined in Eq. (238). Analogously to Eq. (686), the empirical density can for example also be smoothed and correctly normalized again, so that

$$P^{(0)} = \tilde{\mathbf{C}}\tilde{P}_{\text{emp}}. \tag{704}$$

with $\tilde{\mathbf{C}}$ defined in Eq. (698).

Fig. 13 compares the initialization according to Eq. (697), where the smoothing operator $\tilde{C}$ acts on N, with an initialization according to Eq. (704), where the smoothing operator $\tilde{C}$ acts on the correctly normalized $\tilde{P}_{\rm emp}$.

7.4 Numerical examples

7.4.1 Density estimation with Gaussian specific prior

In this section we look at some numerical examples and discuss implementations of the nonparametric learning algorithms for density estimation we have discussed in this paper.

As example, consider a problem with a one–dimensional X–space and a one–dimensional Y–space, and a smoothness prior with inverse covariance

$$\mathbf{K} = \lambda_x \left(\mathbf{K}_X \otimes 1_Y\right) + \lambda_y \left(1_X \otimes \mathbf{K}_Y\right), \tag{705}$$

where

$$\begin{aligned} \mathbf{K}_X &= \lambda_0 \mathbf{I}_X - \lambda_2 \Delta_x + \lambda_4 \Delta_x^2 - \lambda_6 \Delta_x^3 && (706) \\ \mathbf{K}_Y &= \lambda_0 \mathbf{I}_Y - \lambda_2 \Delta_y + \lambda_4 \Delta_y^2 - \lambda_6 \Delta_y^3, && (707) \end{aligned}$$

and Laplacian

$$\Delta_x(x,x') = \delta''(x-x') = \delta(x-x')\frac{d^2}{dx^2}, \tag{708}$$

and analogously for Δ_y. For $\lambda_2 \neq 0 = \lambda_0 = \lambda_4 = \lambda_6$ this corresponds to the two Laplacian prior factors Δ_x for x and Δ_y for y. (Notice that also for $\lambda_x = \lambda_y$ the λ_4– and λ_6–terms do not include all terms of an iterated 2–dimensional Laplacian, like $\Delta^2 = (\Delta_x + \Delta_y)^2$ or Δ^4, as the mixed derivatives $\Delta_x\Delta_y$ are missing.)

We will now study nonparametric density estimation with prior factors being Gaussian with respect to L as well as being Gaussian with respect to P.

The error or energy functional for a Gaussian prior factor in L is given by Eq. (108). The corresponding iteration procedure is

$$L^{(i+1)} = L^{(i)} + \eta \mathbf{A}^{-1}\left(N - \mathbf{K}L^{(i)} - \mathbf{e}^{\mathbf{L}^{(i)}}\left[N_X - \mathbf{I}_X \mathbf{K} L^{(i)}\right]\right). \tag{709}$$

Written explicitly for $\lambda_2 = 1$, $\lambda_0 = \lambda_4 = \lambda_6 = 0$ Eq. (709) reads,

$$L^{(i+1)}(x,y) = L^{(i)}(x,y) + \eta \sum_j \mathbf{A}^{-1}(x,y;x_j,y_j) \tag{710}$$

$$+\eta \int dx'dy' \mathbf{A}^{-1}(x,y;x',y') \left[\frac{d^2}{d(x')^2} L^{(i)}(x',y') + \frac{d^2}{d(y')^2} L^{(i)}(x',y') \right.$$

$$\left. - \left(\sum_j \delta(x'-x_j) + \int dy'' \frac{d^2}{d(x')^2} L^{(i)}(x',y'') + \int dy'' \frac{d^2}{d(y'')^2} L^{(i)}(x',y'') \right) e^{L^{(i)}(x',y')} \right].$$

Here

$$\int_{y_A}^{y_B} dy'' \, \frac{d^2}{d(y'')^2} L^{(i)}(x',y'') = \left. \frac{d}{d(y'')} L^{(i)}(x',y'') \right|_{y_A}^{y_B}$$

vanishes if the first derivative $\frac{d}{dy} L^{(i)}(x,y)$ vanishes at the boundary or if periodic.

Analogously, for error functional E_P (164) the iteration procedure

$$P^{(i+1)} = P^{(i)} + \eta \mathbf{A}^{-1} \left[(\mathbf{P}^{(i)})^{-1} N - N_X - \mathbf{I}_X \mathbf{P}^{(i)} \mathbf{K} P^{(i)} - \mathbf{K} P^{(i)} \right]. \tag{711}$$

becomes for $\lambda_2 = 1$, $\lambda_0 = \lambda_4 = \lambda_6 = 0$

$$P^{(i+1)}(x,y) = P^{(i)}(x,y) + \eta \sum_j \frac{\mathbf{A}^{-1}(x,y;x_j,y_j)}{P^{(i)}(x_j,y_j)} \tag{712}$$

$$+\eta \int dx'dy' \mathbf{A}^{-1}(x,y;x',y') \left[\frac{d^2}{d(x')^2} P^{(i)}(x',y') + \frac{d^2}{d(y')^2} P^{(i)}(x',y') \right.$$

$$- \left(\sum_j \delta(x'-x_j) + \int dy'' P^{(i)}(x',y'') \frac{d^2 P^{(i)}(x',y'')}{d(x')^2} \right.$$

$$\left. \left. + \int dy'' P^{(i)}(x',y'') \frac{d^2 P^{(i)}(x',y'')}{d(y'')^2} \right) \right].$$

Here

$$\int_{y_A}^{y_B} dy'' P^{(i)}(x',y'') \frac{d^2 P^{(i)}(x',y'')}{d(y'')^2} =$$

$$\left. P^{(i)}(x',y'') \frac{dP^{(i)}(x',y'')}{d(y'')} \right|_{y_A}^{y_B} - \int_{y_A}^{y_B} dy'' \left(\frac{dP^{(i)}(x',y'')}{dy''} \right)^2, \tag{713}$$

where the first term vanishes for $P^{(i)}$ periodic or vanishing at the boundaries. (This has to be the case for $i\,d/dy$ to be hermitian.)

We now study density estimation problems numerically. In particular, we want to check the influence of the nonlinear normalization constraint. Furthermore, we want to compare models with Gaussian prior factors for L with models with Gaussian prior factors for P.

The following numerical calculations have been performed on a mesh of dimension $10\times$ 15, i.e., $x \in [1, 10]$ and $y \in [1, 15]$, with periodic boundary conditions on y and sometimes also in x. A variety of different iteration and initialization methods have been used.

Figs. 14 – 17 summarize results for density estimation problems with only two data points, where differences in the effects of varying smoothness priors are particularly easy to see. A density estimation with more data points can be found in Fig. 21.

For Fig. 14 a Laplacian smoothness prior on L has been implemented. The solution has been obtained by iterating with the negative Hessian, as long as positive definite. Otherwise the gradient algorithm has been used. One iteration step means one iteration according to Eq. (617) with the optimal η. Thus, each iteration step includes the optimization of η by a line search algorithm. (For the figures the Mathematica function FindMinimum has been used to optimize η.)

As initial guess in Fig. 14 the kernel estimate $L^{(0)} = \ln(\tilde{\mathbf{C}}\tilde{P}_{\rm emp})$ has been employed, with $\tilde{\mathbf{C}}$ defined in Eq. (698) and $\mathbf{C} = (\mathbf{K}+m_C^2\mathbf{I})$ with squared mass $m_C^2 = 0.1$. The fast drop–off of the energy E_L within the first two iterations shows the quality of this initial guess. Indeed, this fast convergence seems to indicate that the problem is nearly linear, meaning that the influence of the only nonlinear term in the stationarity equation, the normalization constraint, is not too strong. Notice also, that the reconstructed regression shows the typical piecewise linear approximations well known from one–dimensional (normalization constraint free) regression problems with Laplacian prior.

Fig. 15 shows a density estimation similar to Fig. 14, but for a Gaussian prior factor in P and thus also with different λ_2, different initialization, and slightly different iteration procedure. For Fig. 15 also a kernel estimate $P^{(0)}$ $= (\tilde{\mathbf{C}}\tilde{P}_{\rm emp})$ has been used as initial guess, again with $\tilde{\mathbf{C}}$ as defined in Eq. (698) and $\mathbf{C} = (\mathbf{K}+m_C^2\mathbf{I})$ but with squared mass $m_C^2 = 1.0$. The solution has been obtained by prior relaxation $\mathbf{A} = \mathbf{K} + m^2\mathbf{I}$ including a mass term with m^2 $= 1.0$ to get for a Laplacian $\mathbf{K} = -\Delta$ and periodic boundary conditions an invertible $\mathbf{A}$. This iteration scheme does not require to calculate the Hessian

$\mathbf{H}_P$ at each iteration step. Again the quality of the initial guess (and the iteration scheme) is indicated by the fast drop–off of the energy E_P during the first iteration.

Because the range of P–values, being between zero and one, is smaller than that of L–values, being between minus infinity and zero, a larger Laplacian smoothness factor λ_2 is needed for Fig. 15 to get similar results than for Fig. 14. In particular, such λ_2 values have been chosen for the two figures that the maximal values of the the two reconstructed probability densities P turns out to be nearly equal.

Because the logarithm particularly expands the distances between small probabilities one would expect a Gaussian prior for L to be especially effective for small probabilities. Comparing Fig. 14 and Fig. 15 this effect can indeed be seen. The deep valleys appearing in the L–landscape of Fig. 15 show that small values of L are not smoothed out as effectively as in Fig. 14. Notice, that therefore also the variance of the solution $p(y|x,h)$ is much smaller for a Gaussian prior in P at those x which are in the training set.

Fig. 16 resumes results for a model similar to that presented in Fig. 14, but with a $(-\Delta^3)$–prior replacing the Laplacian $(-\Delta)$–prior. As all quadratic functions have zero third derivative such a prior favors, applied to L, quadratic log–likelihoods, corresponding to Gaussian probabilities P. Indeed, this is indicated by the striking difference between the regression functions in Fig. 16 and in Fig. 14: The $(-\Delta^3)$–prior produces a much rounder regression function, especially at the x values which appear in the data. Note however, that in contrast to a pure Gaussian regression problem, in density estimation an additional non–quadratic normalization constraint is present.

In Fig. 17 a similar prior has been applied, but this time being Gaussian in P instead of L. In contrast to a $(-\Delta^3)$–prior for L, a $(-\Delta^3)$–prior for P implements a tendency to quadratic P. Similarly to the difference between Fig. 14 and Fig. 16, the regression function in Fig. 17 is also rounder than that in Fig. 15. Furthermore, smoothing in Fig. 17 is also less effective for smaller probabilities than it is in Fig. 16. That is the same result we have found comparing the two priors for L shown in Fig. 15 and Fig. 14. This leads to deeper valleys in the L–landscape and to a smaller variance especially at x which appear in the training data.

Fig. 21 depicts the results of a density estimation based on more than two data points. In particular, fifty training data have been obtained by sampling with uniform $p(x)$ from the "true" density

$$P_{\text{true}}(x,y) = p(y|x,h_{\text{true}}) = \frac{1}{2\sqrt{2\pi}\sigma_0}\left(e^{-\frac{(y-h_a(x))^2}{2\sigma_0^2}} + e^{-\frac{(y-h_b(x))^2}{2\sigma_0^2}}\right), \tag{714}$$

with $\sigma_0 = 1.5$, $h_a(x) = 125/18 + (5/9)x$, $h_b(x) = 145/18 - (5/9)x$, shown in the top row of Fig. 18. The sampling process has been implemented using the transformation method (see for example [196]). The corresponding empirical density N/n (235) and conditional empirical density P_{emp} of Eq. (236), in this case equal to the extended $\tilde{P}_{\text{emp}}$ defined in Eq. (238), can be found in Fig. 20.

Fig. 21 shows the maximum posterior solution $p(y|x,h^*)$ and its logarithm, the energy E_L during iteration, the *regression function*

$$h(x) = \int dy\, y\, p(y|x,h_{\text{true}}) = \int dy\, y\, P_{\text{true}}(x,y), \tag{715}$$

(as reference, the regression function for the true likelihood $p(y|x,h_{\text{true}})$ is given in Fig. 19), the *average training error* (or *empirical (conditional) log–loss*)

$$< -\ln p(y|x,h) >_D = -\frac{1}{n}\sum_{i=1}^{n} \ln p(y_i|x_i,h), \tag{716}$$

and the *average test error* (or *true expectation of (conditional) log–loss*) for uniform $p(x)$

$$< -\ln p(y|x,h) >_{P_{\text{true}}} = -\int dy\, dx\, p(x) p(y|x,h_{\text{true}}) \ln p(y|x,h), \tag{717}$$

which is, up to a constant, equal to the expected Kullback–Leibler distance between the actual solution and the true likelihood,

$$\text{KL}\Big(p(x,y|h_{\text{true}}), p(y|x,h)\Big) = -\int dy\, dx\, p(x,y|h_{\text{true}}) \ln \frac{p(y|x,h)}{p(y|x,h_{\text{true}})}. \tag{718}$$

The test error measures the quality of the achieved solution. It has, in contrast to the energy and training error, of course not been available to the learning algorithm.

The maximum posterior solution of Fig. 21 has been calculated by minimizing E_L using massive prior iteration with $\mathbf{A} = \mathbf{K} + m^2\mathbf{I}$, a squared mass $m^2 = 0.01$, and a (conditionally) normalized, constant $L^{(0)}$ as initial guess.

Convergence has been fast, the regression function is similar to the true one (see Fig. 19).

Fig. 22 compares some iteration procedures and initialization methods Clearly, all methods do what they should do, they decrease the energy functional. Iterating with the negative Hessian yields the fastest convergence. Massive prior iteration is nearly as fast, even for uniform initialization, and does not require calculation of the Hessian at each iteration. Finally, the slowest iteration method, but the easiest to implement, is the gradient algorithm.

Looking at Fig. 22 one can distinguish data–oriented from prior–oriented initializations. We understand data–oriented initial guesses to be those for which the training error is smaller at the beginning of the iteration than for the final solution and prior–oriented initial guesses to be those for which the opposite is true. For good initial guesses the difference is small. Clearly, the uniform initializations is prior–oriented, while an empirical log–density $\ln(N/n + \epsilon)$ and the shown kernel initializations are data–oriented.

The case where the test error grows while the energy is decreasing indicates a misspecified prior and is typical for overfitting. For example, in the fifth row of Fig. 22 the test error (and in this case also the average training error) grows again after having reached a minimum while the energy is steadily decreasing.

7.4.2 Density estimation with Gaussian mixture prior

Having seen Bayesian field theoretical models working for Gaussian prior factors we will study in this section the slightly more complex prior mixture models. Prior mixture models are an especially useful tool for implementing complex and unsharp prior knowledge. They may be used, for example, to translate verbal statements of experts into quantitative prior densities [132, 133, 134, 135, 136], similar to the quantification of "linguistic variables" by fuzzy methods [118, 119].

We will now study a prior mixture with Gaussian prior components in L. Hence, consider the following energy functional with mixture prior

$$E_L = -\ln \sum_j p_j e^{-E_j} = -(L, N) + (e^L, \Lambda_X) - \ln \sum_j p_j e^{-\lambda E_{0,j}} \tag{719}$$

with mixture components

$$E_j = -(L, N) + \lambda E_{0,j} + (e^L, \Lambda_X). \tag{720}$$

We choose Gaussian component prior factors with equal covariances but differing means

$$E_{0,j} = \frac{1}{2}\Big(L - t_j,\, \mathbf{K}(L - t_j)\Big). \tag{721}$$

Hence, the stationarity equation for Functional (719) becomes

$$0 = N - \lambda \mathbf{K}\left(L - \sum_j a_j t_j\right) - \mathbf{e}^{\mathbf{L}} \Lambda_X, \tag{722}$$

with Lagrange multiplier function

$$\Lambda_X = N_X - \lambda \mathbf{I}_X \mathbf{K}\left(L - \sum_j a_j t_j\right), \tag{723}$$

and mixture coefficients

$$a_j = \frac{p_j e^{-\lambda E_{0,j}}}{\sum_k p_k e^{-\lambda E_{0,k}}}. \tag{724}$$

The parameter λ plays here a similar role as the inverse temperature β for prior mixtures in regression (see Sect. 6.3). In contrast to the β–parameter in regression, however, the "low temperature" solutions for $\lambda \to \infty$ are the pure prior templates t_j, and for $\lambda \to 0$ the prior factor is switched off.

Typical numerical results of a prior mixture model with two mixture components are presented in Figs. 23 – 28. Like for Fig. 21, the true likelihood used for these calculations is given by Eq. (714) and shown in Fig. 18. The corresponding true regression function is thus that of Fig. 19. Also, the same training data have been used as for the model of Fig. 21 (Fig. 20). The two templates t_1 and t_2 which have been selected for the two prior mixture components are (Fig. 18)

$$t_1(x,y) = \frac{1}{2\sqrt{2\pi}\sigma_t}\left(e^{-\frac{(y-\mu_a)^2}{2\sigma_t^2}} + e^{-\frac{(y-\mu_b)^2}{2\sigma_t^2}}\right), \tag{725}$$

$$t_2(x,y) = \frac{1}{\sqrt{2\pi}\sigma_t} e^{-\frac{(y-\mu_2)^2}{2\sigma_t^2}}, \tag{726}$$

with $\sigma_t = 2$, $\mu_a = \mu_2 + 25/9 = 10.27$, $\mu_b = \mu_2 - 25/9 = 4.72$, and $\mu_2 = 15/2$. Both templates capture a bit of the structure of the true likelihood, but not too much, so learning remains interesting. The average test error of

t_1 is equal to 2.56 and is thus lower than that of t_2 being equal to 2.90. The minimal possible average test error 2.23 is given by that of the true solution P_{true}. A uniform P, being the effective template in the zero mean case of Fig. 21, has with 2.68 an average test error between the two templates t_1 and t_2.

Fig. 23 proves that convergence is fast for massive prior relaxation when starting from t_1 as initial guess $L^{(0)}$. Compared to Fig. 21 the solution is a bit smoother, and as template t_1 is a better reference than the uniform likelihood the final test error is slightly lower than for the zero–mean Gaussian prior on L. Starting from $L^{(0)} = t_2$ convergence is not much slower and the final solution is similar, the test error being in that particular case even lower (Fig. 24). Starting from a uniform $L^{(0)}$ the mixture model produces a solution very similar to that of Fig. 21 (Fig. 25).

The effect of changing the λ parameter of the prior mixture can be seen in Fig. 26 and Fig. 27. Larger λ means a smoother solution and faster convergence when starting from a template likelihood (Fig. 26). Smaller λ results in a more rugged solution combined with a slower convergence. The test error in Fig. 27 already indicates overfitting.

Prior mixture models tend to produce metastable and approximately stable solutions. Fig. 28 presents an example where starting with $L^{(0)} = t_2$ the learning algorithm seems to have produced a stable solution after a few iterations. However, iterating long enough this decays into a solution with smaller distance to t_1 and with lower test error. Notice that this effect can be prevented by starting with another initialization, like for example with $L^{(0)} = t_1$ or a similar initial guess.

We have seen now that, and also how, learning algorithms for Bayesian field theoretical models can be implemented. In this paper, the discussion of numerical aspects was focussed on general density estimation problems. Other Bayesian field theoretical models, e.g., for regression and inverse quantum problems, have also been proved to be numerically feasible. Specifically, prior mixture models for Gaussian regression are compared with so–called Landau–Ginzburg models in [132]. An application of prior mixture models to image completion, formulated as a Gaussian regression model, can be found in [137]. Furthermore, hyperparameter have been included in numerical calculations in [133] and also in [137]. Finally, learning algorithms for inverse quantum problems are treated in [143] for inverse quantum statistics, and, in combination with a mean field approach, in [142] for inverse quantum many–body theory. Time–dependent inverse quantum problems will be the topic of [138].

In conclusion, we may say that many different Bayesian field theoretical models have already been studied numerically and proved to be computationally feasible. This also shows that such nonparametric Bayesian approaches are relatively easy to adapt to a variety of quite different learning scenarios. Applications of Bayesian field theory requiring further studies include, for example, the prediction of time–series and the interactive implementation of unsharp *a priori* information.

Acknowledgements The author wants to thank Federico Girosi, Tomaso Poggio, Jörg Uhlig, and Achim Weiguny for discussions.

References

[1] Aarts, E. & Korts, J. (1989) *Simulated Annealing and Boltzmann Machines.* New York: Wiley.

[2] Abu–Mostafa, Y. (1990) Learning from Hints in Neural Networks. *Journal of Complexity* **6**, 192–198.

[3] Abu–Mostafa, Y. (1993) Hints and the VC Dimension. *Neural Computation* **5**, 278–288.

[4] Abu–Mostafa, Y. (1993b) A method for learning from hints. *Advances in Neural Information Processing Systems* **5**, S. Hanson et al (eds.), 73–80, San Mateo, CA: Morgan Kauffmann.

[5] Aida, T. (1999) Field Theoretical Analysis of On-line Learning of Probability Distributions. *Phys. Rev. Lett.* **83**, 3554-3557, `arXiv:cond-mat/9911474`.

[6] Allen, D.M. (1974) The relationship between variable selection and data augmentation and a method of prediction. *Technometrics* **16**, 125.

[7] Amari, S., Cichocki, A., & Yang, H.H.(1996) A New Learning Algorithm for Blind Signal Separation. in *Advances in Neural Information Processing Systems* **8**, D.S. Touretzky et al (eds.), 757–763, Cambridge, MA: MIT Press.

[8] Ames, W.F. (1977) *Numerical Methods for Partial Differential Equations.* (2nd. ed.) New York: Academic Press.

[9] Balian, R. (1991) *From Microphysics to Macrophysics.* Vol. I. Berlin: Springer Verlag.

[10] Ballard, D.H. (1997) *An Introduction to Natural Computation.* Cambridge, MA: MIT Press.

[11] Bayes, T.R. (1763) An Essay Towards Solving a Problem in the Doctrine of Chances. *Phil. Trans. Roy. Soc. London* **53**, 370. (Reprinted in *Biometrika* (1958) **45**, 293)

[12] Bazaraa, M.S., Sherali, H.D., & Shetty, C.M. (1993) *Nonlinear Programming.* (2nd ed.) New York: Wiley.

[13] Beck, C. & Schlögl, F. (1993) *Thermodynamics of chaotic systems.* Cambridge: Cambridge University Press.

[14] Bell, A.J. & Sejnowski, T.J. (1995) *Neural Computation* **7**(6), 1129–1159.

[15] Ben–Israel, A. & Greville, T.N.E. (1974) *Generalized Inverses: Theory and Applications* New York: Wiley.

[16] Berger, J.O. (1980) *Statistical Decision Theory and Bayesian Analysis.* New York: Springer Verlag.

[17] Berger, J.O. & Wolpert R. (1988) *The Likelihood Principle.* (2nd ed.) Hayward, CA: IMS Lecture Notes — Monograph Series **9**.

[18] Bernado, J.M. & Smith, A.F. (1994) *Bayesian Theory.* New York: John Wiley.

[19] Bertsekas, D.P. (1995) *Nonlinear Programming.* Belmont, MA: Athena Scientific.

[20] Bialek, W., Callan, C.G., & Strong, S.P. (1996) Field Theories for Learning Probability Distributions. *Phys. Rev. Lett.* **77**, 4693-4697, `arXiv:cond-mat/9607180`.

[21] Binder, K. & Heermann, D.W. (1988) *Monte Carlo simulation in statistical physics: An introduction.* Berlin: Springer Verlag.

[22] Bishop, C.M. (1993) Curvature–driven smoothing: A learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks* **4**(5),882–884.

[23] Bishop, C.M. (1995) Training with noise is equivalent to Tikhonov regularization. *Neural Computation* **7** (1), 108–116.

[24] Bishop, C.M. (1995) *Neural Networks for Pattern Recognition.* Oxford: Oxford University Press.

[25] Bishop, E. & Bridges, D. (1985) *Constructive Analysis.* Grundlehren der Mathematischen Wissenschaften, Vol. 279. Berlin: Springer Verlag.

[26] Black, M.J. & Rangarajan, A. (1996) On the Unification of Line Processes, Outlier Rejection, and Robust Statistics With Applications in Early Vision. *Int'l J. Computer Vision* **19** (1).

[27] Blaizot, J.–P. & Ripka, G. (1986) *Quantum Theory of Finite Systems.* Cambridge, MA: MIT Press.

[28] Blake, A. & Zisserman, A. (1987) *Visual reconstruction* Cambridge, MA: MIT Press.

[29] Blanchard, P. & Bruening, E. (1982) *Variational Methods in Mathematical Physics.* Berlin: Springer Verlag.

[30] Bleistein, N. & Handelsman, N. (1986) *Asymptotic Expansions of Integrals.* (Originally published in 1975 by Holt, Rinehart and Winston, New York) New York: Dover.

[31] Breiman, L. (1993) Hinging hyperplanes for regression, classification, and function approximation. *IEEE Trans. Inform. Theory* **39**(3), 999–1013.

[32] Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. (1993) *Classification and Regression Trees.*, New York: Chapman & Hall.

[33] Bretthorst, G.L. (1988) *Bayesian spectrum analysis and parameter estimation.* Lecture Notes in Statistics, Vol. 48. Berlin: Springer Verlag. (Available at `http://bayes.wustl.edu/glb/book.pdf`)

[34] Cardy, J. (1996) *Scaling and Renormalization in Statistical Physics.* Cambridge: Cambridge University Press.

[35] Carlin, B.P. & Louis T.A. (1996) *Bayes and Empirical Bayes Methods for Data Analysis.* Boca Raton: Chapman & Hall/CRC.

[36] Choquet–Bruhat Y., DeWitt–Morette, C., & Dillard–Bleick, M. (1982) *Analysis, Manifolds, and Physics.* Part I. Amsterdam: North–Holland.

[37] Collins, J. (1984) *Renormalization.* Cambridge: Cambridge University Press.

[38] Cox, D.R. & Hinkley, D.V. (1974) *Theoretical Statistics.* London: Chapman & Hall.

[39] Craven, P. & Wahba, G. (1979) Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross–validation. *Numer. Math.* **31**, 377–403.

[40] Cressie, N.A.C. (1993) *Statistics for Spatial Data.* New York, Wiley.

[41] Creutz, M. (1983) *Quarks, gluons and lattices.* Cambridge: Cambridge University Press.

[42] D'Agostini, G. (1999) *Bayesian Reasoning in High Energy Physics. — Principles and Applications* — CERN Yellow Report 99–03 (Available at `http://www-zeus.roma1.infn.it/~agostini/prob+stat.html`)

[43] Davis, L. (ed.) (1987) *Genetic Algorithms and Simulated Annealing.* San Mateo, CA: Morgan Kaufmann.

[44] Davis, L. (ed.) (1991) *Handbook of Genetic Algorithms.* New York: Van Nostrand Reinhold.

[45] De Bruijn, N.G. (1981) *Asymptotic Methods in Analysis.* (Originally published in 1958 by the North–Holland Publishing Co., Amsterdam) New York: Dover.

[46] Deco, G. & Obradovic, D. (1996) *An Information–Theoretic Approach to Neural Computing.* New York: Springer Verlag.

[47] Devroye, L., Györfi, L., & Lugosi, G. (1996) *A Probabilistic Theory of Pattern recognition.* New York: Springer Verlag.

[48] Di Castro, C. & Jona-Lasinio, G. (1976) The Renormalization Group Approach to Critical Phenomena. In: Domb, C. & Green M.S. (eds.) *Phase Transitions and Critical Phenomena.* London: Academic Press.

[49] Dietrich, R., Opper, M., & Sompolinsky, H. (1999) Statistical Mechanics of Support Vector Networks. *Physical Review Letters* **82**(14), 2975–2978.

[50] Donoho, D.L. & Johnstone, I.M. (1989) Projection–based approximation and a duality with kernel methods. *Ann. Statist.* **17**(1),58–106.

[51] Doob, J.L. (1953) *Stochastic Processes.* (New edition 1990) New York: Wiley.

[52] Dudley, R.M. (1984) A course on empirical processes. *Lecture Notes in Mathematics* 1097,2-142.

[53] Ebeling, W., Freund, J., & Schweitzer, F. (1998) *Komplexe Strukturen: Entropie und Information.* Stuttgart: Teubner.

[54] Efron, B. & Tibshirani R.J. (1993) *An Introduction to the Bootstrap.* New York: Chapman & Hall.

[55] Eisenberg, J. & Greiner, W. (1972) *Microscopic Theory of the Nucleus.* North–Holland, Amsterdam.

[56] Fernández, R., Fröhlich, J., & Sokal, A.D. (1992) *Random Walks, Critical Phenomena, and Triviality in Quantum Field Theory.* Berlin: Springer Verlag.

[57] Fletcher, R. (1987) *Practical Methods of Optimization.* New York: Wiley.

[58] Fredholm I. (1903) *Acta Math.* **27**.

[59] Friedman, J.H. & Tukey, J.W. (1974) A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.* **24**, 1000–1006.

[60] Friedman, J.H. & Stuetzle, W. (1981) Projection pursuit regression. *J. Am. Statist. Assoc.* **76**(376), 817–823.

[61] Fukunaga, K. (1990) *Statistical Pattern Recognition.* Boston: Academic Press.

[62] Gardiner, C.W. (1990) *Handbook of Stochastic Methods.* (2nd ed.) Berlin: Springer Verlag.

[63] Gardner, E. (1987) Maximum Storage Capacity in Neural Networks. *Europhysics Letters* **4** 481–485.

[64] Gardner, E. (1988) The Space of Interactions in Neural Network Models. *Journal of Physics* **A 21** 257–270.

[65] Gardner, E. & Derrida B. (1988) Optimal Storage Properties of Neural Network Models. *Journal of Physics* **A 21** 271–284.

[66] Geiger, D. & Girosi, F. (1991) Parallel and Deterministic Algortihms for MRFs: Surface Reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **13** (5), 401–412.

[67] Geiger, D. & Yuille, A.L. (1991) A Common Framework for Image Segmentation. *Int'l J. Computer Vision* **6** (3), 227–243.

[68] Gelfand, S.B. & Mitter, S.K. (1993) *On Sampling Methods and Annealing Algorithms. Markov Random Fields – Theory and Applications.* New York: Academic Press.

[69] Gelman, A., Carlin, J.B., Stern, H.S., & Rubin, D.B. (1995) *Bayesian Data Analysis.* New York: Chapman & Hall.

[70] Geman, S. & Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **6**, 721–741. Reprinted in Shafer & Pearl (eds.) (1990) Readings in Uncertainty Reasoning. San Mateo, CA: Morgan Kaufmann.

[71] Geman, D. & Reynoids, G. (1992) Constraint restoration and the Recover of Discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence.* **14**, 367–383.

[72] Giraud, B.G., Lapedes, A., Liu, L.C., & Lemm, J.C. (1995) Lorentzian Neural Nets. *Neural Networks* **8** (5), 757-767.

[73] Girosi, F. (1991) *Models of noise and robust estimates.* A.I.Memo 1287, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[74] Girosi, F. (1997) An equivalence between sparse approximation and support vector machines. A.I. Memo No.1606, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[75] Girosi, F., Poggio, T., & Caprile, B. (1991) Extensions of a theory of networks for approximations and learning: Outliers and negative examples. In Lippmann, R., Moody, J., & Touretzky, D. (eds.) *Advances in Neural Information Processing Systems* **3**, San Mateo, CA: Morgan Kaufmann.

[76] Girosi, F., Jones, M., & Poggio, T. (1995) Regularization Theory and Neural Networks Architectures. *Neural Computation* **7** (2), 219–269.

[77] Glimm, J. & Jaffe, A. (1987) *Quantum Physics. A Functional Integral Point of View.* New York: Springer Verlag.

[78] Goeke, K., Cusson, R.Y., Gruemmer, F., Reinhard, P.–G., Reinhardt, H., (1983) Time–Dependent Hartree–Fock and Beyond: A Review. *Prog. Theor. Physics [Suppl.]* **74** & **75**, 33.

[79] Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning.* Redwood City, CA: Addison–Wesley.

[80] Golden, R.M. (1996) *Mathematical Methods for Neural Network Analysis and Design.* Cambridge, MA: MIT Press.

[81] Golup, G., Heath, M., & Wahba, G.(1979) Generalized cross validation as a method for choosing a good ridge parameter. *Technometrics* **21**, 215–224.

[82] Good, I.J. & Gaskins, R.A. (1971) Nonparametric roughness penalties for probability densities. *Biometrika* **58**, 255–277.

[83] Green, P.J. & Silverman, B.W. (1994) *Nonparametric Regression and Generalized Linear Models.* London: Chapman & Hall.

[84] Großman, Ch. & Roos H.–G. (1994) *Numerik partieller Differentialgleichungen.* Stuttgart: Teubner.

[85] Gull, S.F. (1988) Bayesian data analysis – straight line fitting. In Skilling, J., (ed.) *Maximum Entropy and Bayesian Methods. Cambridge*, 511 –518, Dordrecht: Kluwer.

[86] Gull, S.F. (1989) Developments in maximum entropy data analysis. In Skilling, J, (ed.) *Maximum Entropy and Bayesian Methods. Cambridge 1988*, 53 – 71, Dordrecht: Kluwer.

[87] Hackbusch, W. (1985) *Multi–grid Methods and Applications.* New York: Springer Verlag.

[88] Hackbusch, W. (1989) *Integralgleichungen.* Teubner Studienbücher. Stuttgart: Teubner.

[89] Hackbusch, W. (1993) *Iterative Lösung großer schwachbesetzter Gleichungssysteme.* Teubner Studienbücher. Stuttgart: Teubner.

[90] Härdle, W. (1990) *Applied nonparametric regression.* Cambridge: Cambridge University Press.

[91] Hammersley, J.M. & Handscomb, D.C. (1964) *Monte Carlo Methods.* London: Chapman & Hall.

[92] Hastie, T.J. & Tibshirani, R.J. (1986) Generalized Additive Models. *Statist. Sci.* **1**, 297–318.

[93] Hastie, T.J. & Tibshirani, R.J. (1987) Generalized Additive Models: Some applications. *J. Am. Statist. Assoc.* **82**,371–386.

[94] Hastie, T.J. & Tibshirani, R.J. (1990) *Generalized Additive Models.* London: Chapman & Hall.

[95] Hastings, W.K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109.

[96] Hertz, J., Krogh, A. & Palmer, R.G. (1991) *Introduction to the Theory of Neural Computation.* Santa Fe Institute, Lecture Notes Volume I, Addison–Wesley.

[97] Hilbert, D. & Courant, R. (1989) *Methods of Mathematical Physics.* Vol.1&2,(1st German editions 1924,1937, Springer Verlag) New York: Wiley.

[98] Holland, J.H. (1975) *Adaption in Natural and Artificial Systems.* University of Michigan Press. (2nd ed. MIT Press, 1992.)

[99] Horst, R., Pardalos, M., & Thoai, N.V. (1995) *Introduction to Global Optimization.* Dordrecht: Kluwer.

[100] Huber, P.J. (1979) Robust Smoothing. In Launer, E. & Wilkinson G. (eds.) *Robustness in Statistics* New York: Academic Press.

[101] Huber, P.J. (1981) *Robust Statistics.* New York: Wiley.

[102] Huber, P.J. (1985) Projection Pursuit. *Ann. Statist.* **13**(2),435–475.

[103] Itzkyson, C. & Drouffe, J.–M. (1989) *Statistical Field Theory.* (Vols. 1 and 2) Cambridge: Cambridge University Press.

[104] Jaynes, E.T. (in preparation) *Probability Theory: The Logic Of Science.* (Available at `http://bayes.wustl.edu/etj/prob.html`)

[105] Jeffrey, R. (1999). *Probabilistic Thinking.* (Available at `http://www.princeton.edu/~bayesway/`)

[106] Jeggle, H. (1979) *Nichtlineare Funktionalanalysis.* Stuttgart: Teubner.

[107] Jensen, F.V. (1996) *An Introduction to Bayesian Networks.* New York: Springer Verlag.

[108] Jones, M.C. & Sibson, R. (1987) What is Projection Pursuit? *J. Roy. Statist. Soc. A* **150**, 1–36.

[109] Kaku, M. (1993) *Quantum Field Theory.* Oxford: Oxford University Press.

[110] van Kampen, N.G. (1992) *Stochastic Processes in Physics and Chemistry.* Amsterdam: North–Holland.

[111] Kant, I. (1911) *Kritik der reinen Vernunft.*(2nd ed.) Werke, Vol.3 Berlin: Königliche Akademie der Wissenschaften.

[112] Kimmeldorf, G.S. & Wahba, G. (1970) A correspondence between Bayesian estimation on stochastic processes and smoothing splines. *Ann. Math. Stat.* **41**, 495–502.

[113] Kimmeldorf, G.S. & Wahba, G. (1970) Spline functions and stochastic processes. *Sankhya Ser. A* **32**, Part 2, 173–180.

[114] Kirkpatrick, S., Gelatt Jr., C.D., & Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science* **220**, 671–680.

[115] Kirsch, A. (1996) *An Introduction to the Mathematical Theory of Inverse Problems.* New York: Springer Verlag.

[116] Kitagawa, G., Gersch, W. (1996) *Smoothness Priors Analysis of Time Series* New York: Springer Verlag.

[117] Kleinert, H.(1993) *Pfadintegrale.* Mannheim: Wissenschaftsverlag.

[118] Klir, G.J. & Yuan, B. (1995) *Fuzzy Sets and Fuzzy Logic.* Prentice Hall.

[119] Klir, G.J. & Yuan, B. (eds.) (1996) *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems.* World Scientific.

[120] Koecher, M. (1985) *Lineare Algebra und analytische Geometrie.* Berlin: Springer Verlag.

[121] Koza, J.R. (1992) *Genetic Programming* Cambridge, MA: MIT Press.

[122] Kullback, S. & Leibler R.A. (1951) On Information and Sufficiency. *Ann.Math.Statist.* **22**, 79–86.

[123] Kullback, S. (1951) *Information Theory and Statistics.* New York: Wiley.

[124] Lapedes, A. & Farber, R. (1988) How neural nets work. in *Neural Information Processing Systems*, D.Z.Anderson, (ed.),442–456. New York: American Institute of Physics.

[125] Lauritzen, S.L. (1996) *Graphical Models.* Oxford: Clarendon Press.

[126] Lawson, C. & Hanson, R. (1974) *Solving Least Squares Problems.* Englewood Cliffs, NJ: Prentice-Hall.

[127] Le Bellac, M. (1991) *Quantum and Statistical Field Theory.* Oxford Science Publications, Oxford: Clarendon Press.

[128] Le Cam, L. (1986) *Asymptotic Methods in Statistical Decision Theory.* New York: Springer Verlag.

[129] Leen, T.K. (1995) From Data Distributions to Regularization in Invariant Learning. *Neural Computation* **7**, 974–981.

[130] Lemm, J.C. (1995) Inhomogeneous Random Phase Approximation for Nuclear and Atomic Reactions. *Annals of Physics* **244** (1), 136–200, 1995.

[131] Lemm, J.C. (1995) Inhomogeneous Random Phase Approximation: A Solvable Model. *Annals of Physics* **244** (1), 201–238, 1995.

[132] Lemm, J.C. (1996) *Prior Information and Generalized Questions.* A.I.Memo No. 1598, C.B.C.L. Paper No. 141, Massachusetts Institute of Technology. (Available at `http://pauli.uni-muenster.de/~lemm`)

[133] Lemm, J.C. (1998) *How to Implement A Priori Information: A Statistical Mechanics Approach.* Technical Report MS-TP1-98-12, Münster University, `arXiv:cond-mat/9808039`.

[134] Lemm, J.C. (1998) Fuzzy Interface with Prior Concepts and Non-Convex Regularization. In Wilfried Brauer (Ed.), *Proceedings of the 5. International Workshop "Fuzzy-Neuro Systems '98" (FNS '98)*, March 19-20, 1998, Munich, Germany, Sankt Augustin: Infix.

[135] Lemm, J.C. (1998) Quadratic Concepts. In Niklasson L., Bodén, M., & Ziemke, T. (eds.) *Proceedings of the 8th International Conference on Artificial Neural Networks. (ICANN98)* New York: Springer Verlag.

[136] Lemm, J.C. (1998) Fuzzy Rules and Regularization Theory. In ELITE European Laboratory for Intelligent Techniques Engineering (ed.): *Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT '98)*, Aachen, Germany, September 7-10, 1998, Mainz, Aachen.

[137] Lemm, J.C. (1999) Mixtures of Gaussian Process Priors. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*

(ICANN99), IEEE Conference Publication No. 470. London: Institution of Electrical Engineers.

[138] Lemm, J.C. (2000) *Inverse Time–dependent Quantum Mechanics.* Technical Report, MS-TP1-00-1, Münster University, `arXiv:quant-ph/0002010`.

[139] Lemm, J.C., Beiu, V., & Taylor, J.G. (1995) Density Estimation as a Preprocessing Step for Constructive Algorithms. In Kappen B., Gielen, S. (eds.): *Proceedings of the 3rd SNN Neural Network Symposium.* The Netherlands, Nijmegen, 14–15 September 1995, Berlin, Springer Verlag.

[140] Lemm, J.C., Giraud, B.G., & Weiguny, A. (1990) Mean field approximation versus exact treatment of collisions in few–body systems. *Z. Phys. A – Atomic Nuclei* **336**, 179–188.

[141] Lemm, J.C., Giraud, B.G., & Weiguny, A. (1994) Beyond the time independent mean field theory for nuclear and atomic reactions: Inclusion of particle-hole correlations in a generalized random phase approximation. *Phys. Rev. Lett.* **73**, 420, `arXiv:nucl-th/9911056`.

[142] Lemm, J.C. & Uhlig, J. (1999) *Hartree-Fock Approximation for Inverse Many-Body Problems.* Technical Report, MS-TP1-99-10, Münster University, `arXiv:nucl-th/9908056`.

[143] Lemm, J.C., Uhlig J., & Weiguny, A. (2000) Bayesian Approach to Inverse Quantum Statistics. *Phys. Rev. Lett.* **84**, 2068. `arXiv:cond-mat/9907013`.

[144] Lifshits, M.A. (1995) *Gaussian Random Functions.* Dordrecht: Kluwer.

[145] Loredo T. (1990) From Laplace to Supernova SN 1987A: Bayesian Inference in Astrophysics. In Fougère, P.F. (ed.) *Maximum-Entropy and Bayesian Methods, Dartmouth, 1989*, 81–142. Dordrecht: Kluwer. (Available at `http://bayes.wustl.edu/gregory/gregory.html`)

[146] Louis, A.K. (1989) *Inverse und schlecht gestellte Probleme.* Stuttgart: Teubner.

[147] MacKay, D.J.C. (1992) The evidence framework applied to classification networks. *Neural Computation* **4** (5), 720–736.

[148] MacKay, D.J.C. (1992) A practical Bayesian framework for backpropagation networks. *Neural Computation* **4** (3), 448–472.

[149] MacKay, D.J.C. (1994) Hyperparameters: optimise or integrate out? In Heidbreder, G. (ed.) *Maximum Entropy and Bayesian Methods, Santa Barbara 1993.* Dordrecht: Kluwer.

[150] MacKay, D.J.C. (1998) Introduction to Gaussian processes. In Bishop, C., (ed.) *Neural Networks and Machine Learning.* NATO Asi Series. Series F, Computer and Systems Sciences, Vol. 168.

[151] Marquardt, D.W. (1970) Generalized Inverses, Ridge Regression, Biased Linear Estimation, and Nonlinear Regression. *Technometrics* **12**, 591-613.

[152] Marquardt, D.W. & Snee R.D. (1975) Ridge Regression in Practice *The American Statistician* **29**, 3–20.

[153] Marroquin, J.L., Mitter, S., & Poggio, T. (1987) Probabilistic solution of ill–posed problems in computational vision. *J. Am. Stat. Assoc.* **82**, 76–89.

[154] McCullagh, P. & Nelder, J.A. (1989) *Generalized Linear Models* London: Chapman & Hall.

[155] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., & Teller, E. (1953) Equation of state calculations by fast computing machines. *Journal of Chemical Physics* **21**, 1087–1092.

[156] Mezard, M., Parisi, G., & Virasoro, M.A. (1987) *Spin Glass Theory and Beyond.* Singapore: World Scientific.

[157] Michalewicz, Z. (1992) *Genetic Algorithms + Data Structures = Evolution Programs.* Berlin: Springer Verlag.

[158] Michie, D., Spiegelhalter, D.J., & Taylor, C.C. (Eds.) (1994) *Machine Learning, Neural and Statistical Classification.* New York: Ellis Horwood.

[159] Minski, M.L. & Papert, S.A. (1990) *Perceptrons.* (Expanded Edition, Original edition, 1969) Cambridge, MA: MIT Press.

[160] Mitchell, M. (1996) *An Introduction to Genetic Algorithms.* Cambridge, MA: MIT Press.

[161] Mitchell, A.R. & Griffiths, D.F. (1980) *The Finite Difference Method in Partial Differential Equations.* New York: Wiley.

[162] Molgedey, L. & Schuster, H.G. (1994) Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.* **72**(23), 3634–3637.

[163] Montvay, I. & Münster, G. (1994) *Quantum Fields on a Lattice.* Cambridge: Cambridge University Press.

[164] On the Reciprocal of the General Algebraic Matrix. Moore, E.H. (1920) *Bull.Amer.Math.Soc.* **26**, 394–395.

[165] Morozov, V.A. (1984) *Methods for Solving Incorrectly Posed problems.* New York: Springer Verlag.

[166] Mosteller, F. & Wallace, D. (1963) Inference in an authorship problem. A comparative study of discrimination methods applied to authorships of the disputed Federalist papers. *J. Amer. Statist. Assoc.* **58**, 275–309.

[167] Müller, B. & Reinhardt, J. (1991) *Neural Networks.* (2nd printing) Berlin, Springer Verlag.

[168] Mumford, D. & Shah, J. (1989) Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems. *Comm. Pure Applied Math.* **42**, 577–684.

[169] Nadaraya, E.A. (1965) On nonparametric estimates of density functions and regression curves. *Theor.Prob.Appl.* **10**,186–190.

[170] Neal, R.M. (1996) *Bayesian Learning for Neural Networks.* New York: Springer Verlag.

[171] Neal, R.M. (1997) Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. Technical Report No. 9702, Dept. of Statistics, Univ. of Toronto, Canada.

[172] Negele, J.W. & Orland, H. (1988) *Quantum Many–Particle Systems.* Frontiers In Physics Series (Vol. 68), Redwood City, CA: Addison–Wesley.

[173] Nitzberg, M. & Shiota T. (1992) Nonlinear Image Filtering With Edge and Corner Enhancement. *IEEE Trans. on Pattern Analysis and Machine Intelligence.* **14**, (8) 862-833.

[174] O'Hagen, A. (1994) *Kendall's advanced theory of statistics*, Vol. 2B: *Bayesian inference.* London: Edward Arnold.

[175] Olshausen, B.A. & Field, D.J. (1995) Natural Image Statistics and Efficient Coding. *Workshop on Information Theory and the Brain,* Sept. 4–5, 1995, University of Stirling. Proceedings published in *Network* **7**, 333-339.

[176] Olshausen, B.A. & Field, D.J. (1996) Emergence of simple–cell receptive field properties by learning a spares code for natural images. *Nature* **381**, 607–609.

[177] Opper, M. (1999) *Gaussian Processes for Classification: Mean Field Algorithms.* Tech Report NCRG/1999/030, Neural Computing Research Group at Aston University, UK.

[178] Opper, M. & Kinzel, W. (1996) Statistical Mechanics of Generalization. In Domany, E., van Hemmen, J.L., & Schulten, K. (eds.) *Models of Neural Networks III.* New York: Springer Verlag.

[179] Opper, M., & Winther, O. (1999) Mean field methods for classification with Gaussian processes. In Kearns, M.S., Solla, S.S., & Cohn D.A. (eds.) *Advances in Neural Information Processing Systems* **11**, 309–315, Cambridge, MA: MIT Press.

[180] Ó Ruanaidh, J.J.K. & Fitzgerald W.J. (1996) *Numerical Bayesian Methods Applied to Signal Processing.* New York: Springer Verlag.

[181] Parzen, E. (1962) An approach to time series analysis. *Ann.Math.Statist.* **32**, 951–989.

[182] Parzen, E. (1962) On the estimation of a probability function and mode. *Ann.Math.Statist.* **33**(3).

[183] Parzen, E. (1963) Probability density functionals and reproducing kernel Hilbert spaces. In Rosenblatt, M.(ed.) *Proc. Symposium on Time Series Analysis*, 155–169, New York: Wiley.

[184] Parzen, E. (1970) Statistical inference on time series by rkhs methods. In Pyke, R.(ed.) *Proc. 12th Biennal Seminar*, 1–37, Montreal, Canada: Canadian Mathematical Congress.

[185] Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems.* San Mateo, CA: Morgan Kauffmann.

[186] Penrose, R., (1955) A generalized inverse for matrices. *Proc. Cambridge Philos. Soc.* **51**, 406-413.

[187] Penrose, R., (1956) On Best Approximate Solutions of Linear Matrix Equations. *Proc. Cambridge Philos. Soc.* **52**, 17–19.

[188] Perona, P. & Malik J. (1990) Scale–Space and Edge Detection Using Anisotropic Diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence.* **12**(7), 629–639.

[189] Perskin, M.E. & Schroeder, D.V. (1995) *An Introduction to Quantum Field Theory.* Reading, MA, Addison–Wesley.

[190] Pierre, D.A. (1986) *Optimization Theory with Applications.* New York: Dover. (Original edition Wiley, 1969).

[191] Poggio, T. & Girosi, F. (1990) Networks for Approximation and Learning. *Proceedings of the IEEE*, Vol 78, No. 9.

[192] Poggio, T., Torre, V., & Koch, C. (1985) Computational vision and regularization theory. *Nature* **317**, 314–319.

[193] Polak, E. (1997) *Optimization.* New York: Springer Verlag.

[194] Pollard, D. (1984) Convergence of Stochastic Processes. New York: Springer Verlag.

[195] Pordt, A. (1998) Random Walks in Field Theory In Meyer–Ortmanns, H, Klümper A. (eds.) (1998) *Field Theoretical Tools for Polymer and Particle Physics.* Berlin: Springer Verlag.

[196] Press, W.H., Teukolsky, S.A., Vetterling, W.T., & Flannery, B.P. (1992) *Numerical Recipes in C.* Cambridge: Cambridge University Press.

[197] Ring, P., & Schuck, P. (1980) *The Nuclear Many–Body Problem.* New York: Springer Verlag.

[198] Ripley, B.D. (1977) Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society series B* **39**, 172–212.

[199] Ripley, B.D. (1987) *Stochastic Simulation.* New York: Wiley.

[200] Ripley, B.D. (1996) *Pattern Recognition and Neural Networks.* Cambridge: Cambridge University Press.

[201] Robert, C.P. (1994) *The Bayesian Choice.* New York: Springer Verlag.

[202] Rodriguez, C.C. (1997) *Cross validated Non Parametric Bayesianism by Markov Chain Monte Carlo.* `arXiv:physics/9712041`.

[203] Rose, K., Gurewitz, E., & Fox, G.C. (1990) Statistical mechanics and phase transitions in clustering. *Phys. Rev. Lett.* **65**, 945–948.

[204] Rothe, H.J. (1992) *Lattice Gauge Theories.* Singapore: World Scientific.

[205] Rumelhart, D.E., McClelland, J.L., and the PDP Research Group (1986) *Parallel Distributed Processing*, vol.1& 2, Cambridge, MA: MIT Press.

[206] Ryder, L.H. (1996) *Quantum Field Theory.* Cambridge: Cambridge University Press.

[207] Schervish, M.J. (1995) *Theory of Statistics.* New York: Springer Verlag.

[208] Schölkopf, B., Burges C., & Smola, A. (1998) *Advances in Kernel Methods: Support Vector Machines.* Cambridge, MA: MIT Press.

[209] Schwefel, H.–P. (1995) *Evolution and Optimum Seeking.* New York: Wiley.

[210] Silverman, B.W. (1984) Spline smoothing: The equivalent variable kernel method. *Ann. Statist.* **12**, 898–916.

[211] Silverman, B.W. (1986) *Density Estimation for Statistics and Data Analysis.* London: Chapman & Hall.

[212] Sivia, D.S. (1996) *Data Analysis: A Bayesian Tutorial.* Oxford: Oxford University Press.

[213] Skilling, J. (1991) On parameter estimation and quantified MaxEnt. In Grandy, W.T. & Schick, L.H. (eds.) *Maximum Entropy and Bayesian Methods. Laramie, 1990*, 267 –273, Dordrecht: Kluwer.

[214] Smola, A.J. & Schölkopf, B., (1998) From regularization operators to support vector kernels. In: Jordan, M.I., Kearns, M.J., & Solla S.A. (Eds.): *Advances in Neural Information Processing Systems* **10**. Cambridge, MA: MIT Press.

[215] Smola, A.J., Schölkopf, B., & Müller, K–R. (1998) The connection between regularization operators and support vector kernels. *Neural Networks* **11**, 637–649.

[216] Stone, M. (1974) Cross–validation choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B* **36**, 111-147.

[217] Stone, M. (1977) An asymptotic equivalence of choice of model by cross–validation and Akaike's criterion. *Journal of the Royal Statistical Society B* **39**, 44.

[218] Stone, C.J. (1985) Additive regression and other nonparametric models. *Ann. Statist.* **13**,689–705.

[219] Tierney, L. (1994) Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics* **22**, 1701–1762.

[220] Tikhonov, A.N. (1963) Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.* **4**, 1035–1038.

[221] Tikhonov, A.N. & Arsenin, V.Y. (1977) *Solution of Ill–posed Problems.* Washington, DC: W.H.Winston.

[222] Uhlig, J. (2000) PhD Thesis, Münster University.

[223] Uhlig, J., Lemm, J., & Weiguny, A. (1998) Mean field methods for atomic and nuclear reactions: The link between time–dependent and time–independent approaches. *Eur. Phys.* A **2**, 343–354.

[224] Vapnik, V.N. (1982) *Estimation of dependencies based on empirical data.* New York: Springer Verlag.

[225] Vapnik, V.N. (1995) *The Nature of Statistical Learning Theory.* New York: Springer Verlag.

[226] Vapnik, V.N. (1998) *Statistical Learning Theory.* New York: Wiley.

[227] Vico, G. (1858, original 1710) *De antiquissima Italorum sapientia* Naples: Stamperia de' Classici Latini.

[228] Wahba, G. (1990) *Spline Models for Observational Data.* Philadelphia: SIAM.

[229] Wahba, G. (1997) *Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV.* Technical Report 984, University of Wisconsin.

[230] Wahba, G. & Wold, S. 1975) A completely automatic French curve. *Commun. Statist.* **4**, 1–17.

[231] Watkin, T.L.H., Rau, A., & Biehl, M. (1993) The statistical mechanics of learning a rule. *Rev. Mod. Phys.* **65**, 499–556.

[232] Watzlawick, P. (ed.) (1984) *The Invented Reality.* New York: Norton.

[233] Weinstein, S. (1995) *The Quantum Theory of Fields.* Vol.1 Cambridge: Cambridge University Press.

[234] Weinstein, S. (1996) *The Quantum Theory of Fields.* Vol.2 Cambridge: Cambridge University Press.

[235] West, M. & Harrison, J. (1997) *Bayesian Forecasting and Dynamic Models.* New York, Springer Verlag.

[236] Williams, C.K.I. & Barber, D. (1998) Bayesian Classification With Gaussian Processes *IEEE Trans. on Pattern Analysis and Machine Intelligence.* **20**(12), 1342–1351.

[237] Williams, C.K.I. & Rasmussen, C.E. (1996) Gaussian Processes for Regression. in *Advances in Neural Information Processing Systems* **8**, D.S. Touretzky et al (eds.), 515–520, Cambridge, MA: MIT Press.

[238] Winkler, G. (1995) *Image Analysis, Random Fields and Dynamic Monte Carlo Methods.* Berlin: Springer Verlag.

[239] Wolpert, D.H. (ed.) (1995) *The Mathematics of Generalization.* The Proceedings of the SFI/CNLS Workshop on Formal Approaches to Supervised Learning. Santa Fe Institute, Studies in the Sciences of Complexity. Reading, MA: Addison–Wesley.

[240] Wolpert, D.H. (1996) The Lack of A Priori Distinctions between Learning Algorithms. *Neural Computation* **8** (7), 1341-1390.

[241] Wolpert, D.H. (1996) The Existence of A Priori Distinctions between Learning Algorithms. *Neural Computation* **8** (7), 1391-1420.

[242] Yakowitz, S.J. & Szidarovsky, F. (1985) A Comparison of Kriging With Nonparametric Regression Methods. *J.Multivariate Analysis.* **16**, 21-53.

[243] Yuille, A.L., (1990) Generalized deformable models, statistical physics and matching problems. *Neural Computation*, **2**, (1) 1–24.

[244] Yuille, A.L. & Kosowski, J.J. (1994) Statistical Physics Algorithm That Converge. *Neural Computation* **6** (3), 341–356.

[245] Yuille, A.L., Stolorz, P., & Utans, J. (1994) Statistical Physics, Mixtures of Distributions, and EM Algorithm. *Neural Computation*, **6** (2), 334–340.

[246] Zhu, S.C. & Mumford, D. (1997) Prior Learning and Gibbs Reaction–Diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **19** (11), 1236–1250.

[247] Zhu, S.C. & Yuille, A.L. (1996) Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **18** (9), 884–900.

[248] Zhu, S.C., Wu, Y.N., & Mumford, D. (1997) Minimax Entropy principle and Its Application to Texture Modeling. *Neural Computation*, **9** (8).

[249] Zinn–Justin, J. (1989) *Quantum Field Theory and Critical Phenomena.* Oxford: Oxford Science Publications.

just for the URZ Printer

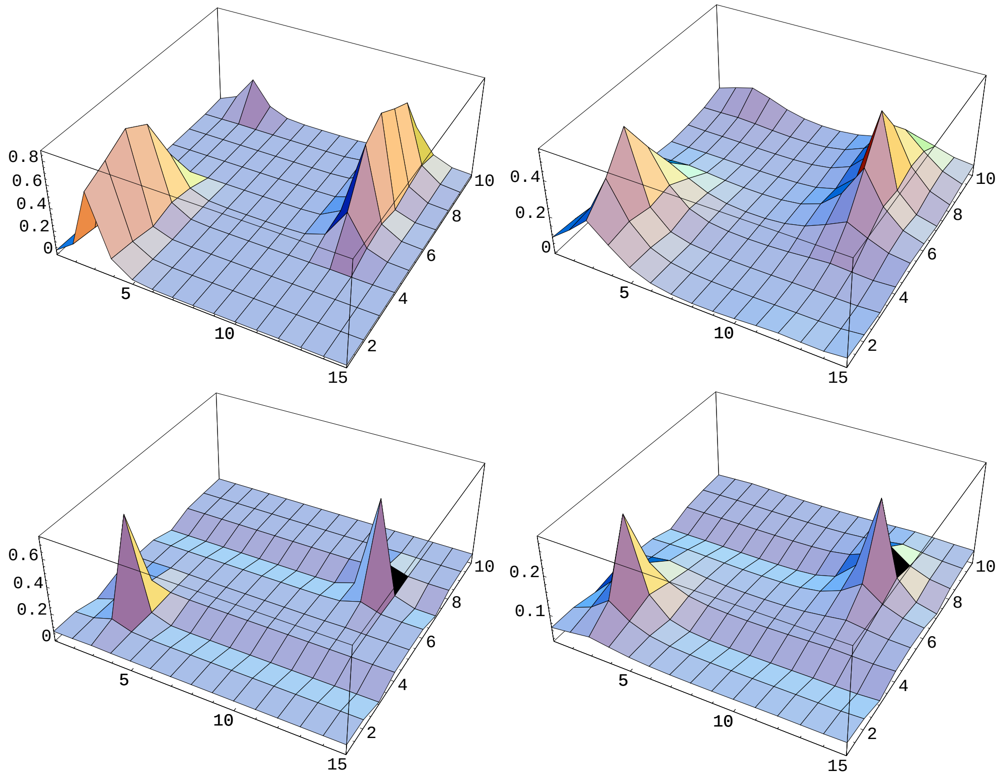

Figure 13: Comparison of initial guesses $P^{(0)}(x, y)$ for a case with two data points located at $(3, 3)$ and $(7, 12)$ within the intervals $y \in [1, 15]$ and $x \in [1, 10]$ with periodic boundary conditions. First row: $P^{(0)} = \tilde{\mathbf{C}}N$. (The smoothing operator acts on the unnormalized N. The following conditional normalization changes the shape more drastically than in the example shown in the second row.) Second row: $P^{(0)} = \tilde{\mathbf{C}}\tilde{P}_{\text{emp}}$. (The smoothing operator acts on the already conditionally normalized $\tilde{P}_{\text{emp}}$.) The kernel $\tilde{\mathbf{C}}$ is given by Eq. (698) with $\mathbf{C} = (\mathbf{K} + m_C^2\mathbf{I})$, $m_C^2 = 1.0$, and a $\mathbf{K}$ of the form of Eq. (705) with $\lambda_0 = \lambda_4 = \lambda_6 = 0$, and $\lambda_2 = 0.1$ (figures on the l.h.s.) or $\lambda_2 = 1.0$ (figures on the r.h.s.), respectively.

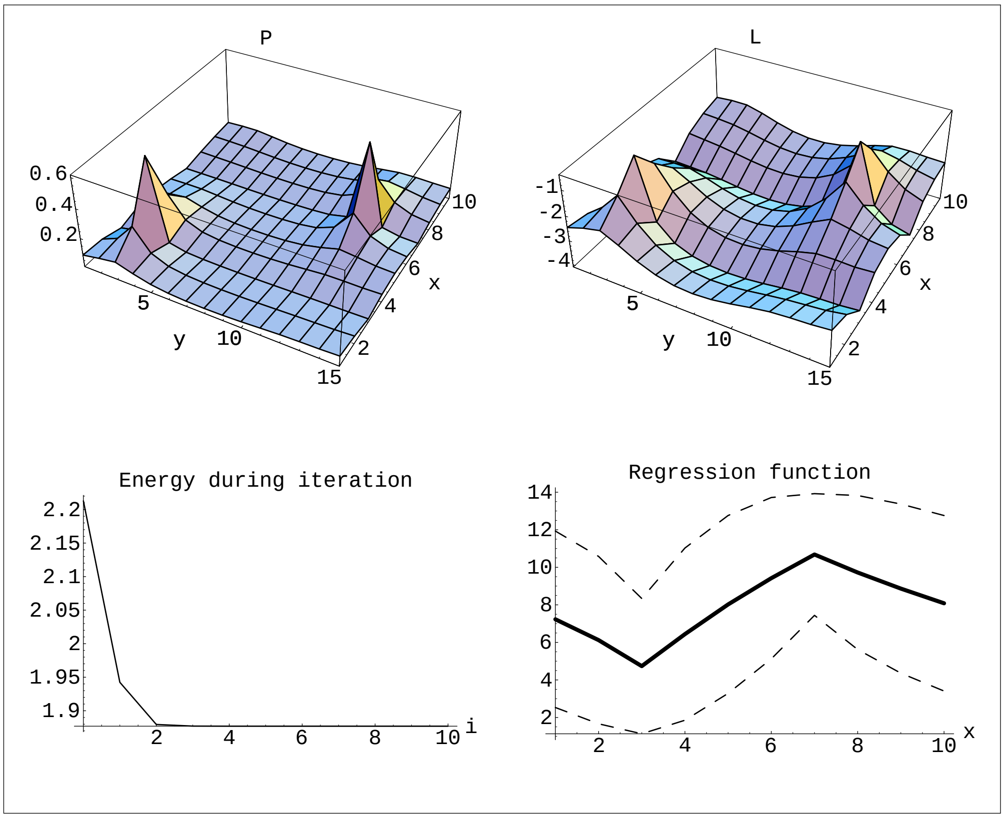

Figure 14: Density estimation with 2 data points and a Gaussian prior factor for the log–probability L. First row: Final P and L. Second row: The l.h.s. shows the energy E_L (108) during iteration, the r.h.s. the regression function $h(x) = \int dy\, yp(y|x, h_{\rm true}) = \int dy\, y P_{\rm true}(x, y)$. The dotted lines indicate the range of one standard deviation above and below the regression function (ignoring periodicity in x). The fast convergence shows that the problem is nearly linear. The asymmetry of the solution between the x– and y–direction is due to the normalization constraint, only required for y. (Laplacian smoothness prior $\mathbf{K}$ as given in Eq. (705) with $\lambda_x = \lambda_y = 1$, $\lambda_0 = 0$, $\lambda_2 = 0.025$, $\lambda_4 = \lambda_6 = 0$. Iteration with negative Hessian $\mathbf{A} = -\mathbf{H}$ if positive definite, otherwise with the gradient algorithm, i.e., $\mathbf{A} = \mathbf{I}$. Initialization with $L^{(0)} = \ln(\tilde{\mathbf{C}}\tilde{P}_{\rm emp})$, i.e., $L^{(0)}$ normalized to $\int dy\, e^L = 1$, with $\tilde{\mathbf{C}}$ of Eq. (698) and $\mathbf{C} = (\mathbf{K} + m_C^2\mathbf{I})$, $m_C^2 = 0.1$. Within each iteration step the optimal step width η has been found by a line search. Mesh with 10 points in x-direction and 15 points in y–direction, periodic boundary conditions in x and y. The 2 data points are $(3, 3)$ and $(7, 12)$.)

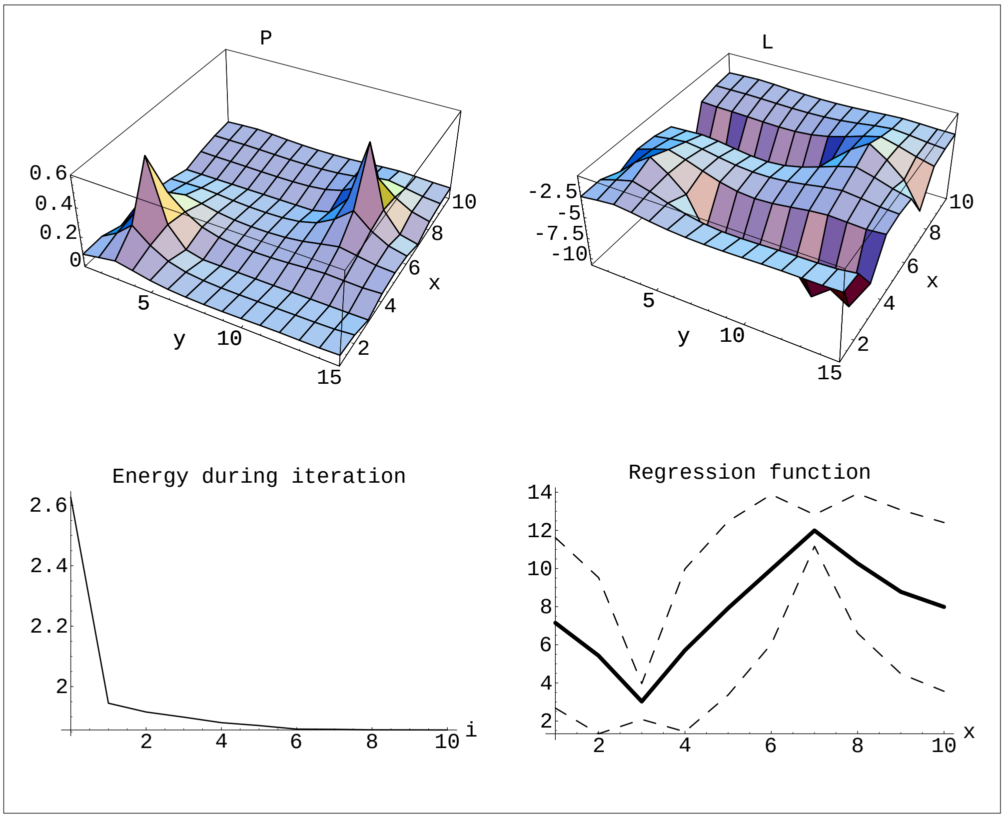

Figure 15: Density estimation with 2 data points, this time with a Gaussian prior factor for the probability P, minimizing the energy functional E_P (164). To make the figure comparable with Fig. 14 the parameters have been chosen so that the maximum of the solution P is the same in both figures ($\max P = 0.6$). Notice, that compared to Fig. 14 the smoothness prior is less effective for small probabilities. (Same data, mesh and periodic boundary conditions as for Fig. 14. Laplacian smoothness prior $\mathbf{K}$ as in Eq. (705) with $\lambda_x = \lambda_y = 1$, $\lambda_0 = 0$, $\lambda_2 = 1$, $\lambda_4 = \lambda_6 = 0$. Iterated using massive prior relaxation, i.e., $\mathbf{A} = \mathbf{K} + m^2\mathbf{I}$ with $m^2 = 1.0$. Initialization with $P^{(0)} = \tilde{\mathbf{C}}\tilde{P}_{\rm emp}$, with $\tilde{\mathbf{C}}$ of Eq. (698) so $P^{(0)}$ is correctly normalized, and $\mathbf{C} = (\mathbf{K} + m_C^2\mathbf{I})$, $m_C^2 = 1.0$. Within each iteration step the optimal factor η has been found by a line search algorithm.)

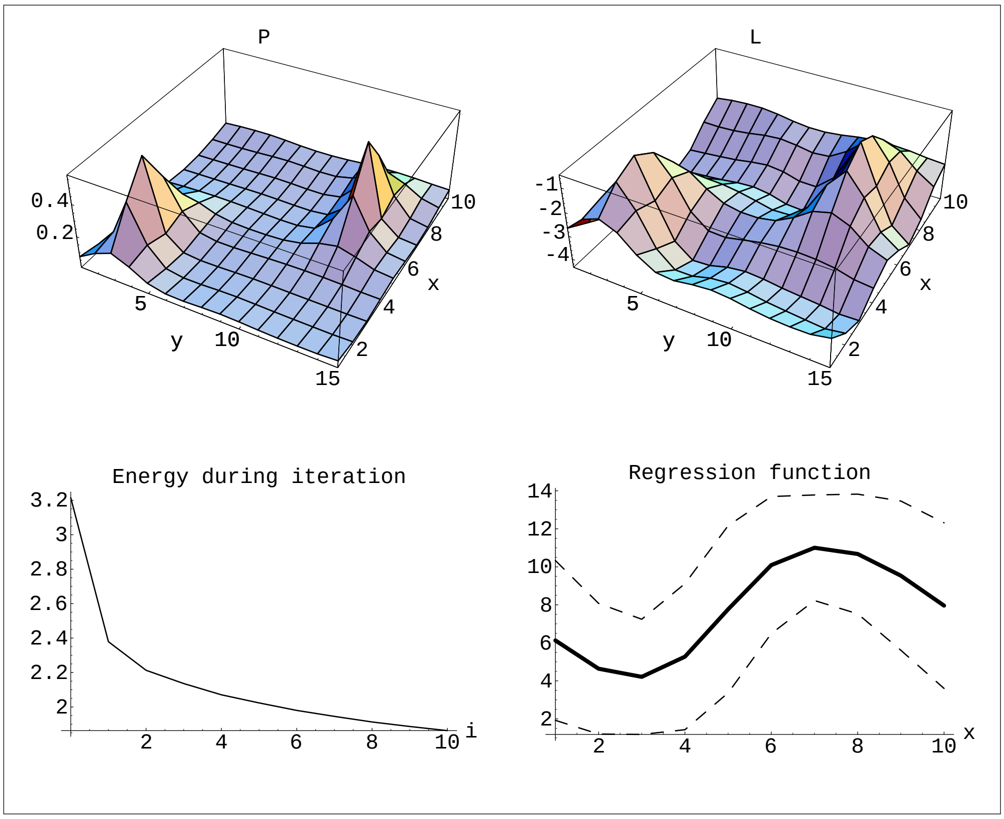

Figure 16: Density estimation with a $(-\Delta^3)$ Gaussian prior factor for the log–probability L. Such a prior favors probabilities of Gaussian shape. (Smoothness prior $\mathbf{K}$ of the form of Eq. (705) with $\lambda_x = \lambda_y = 1$, $\lambda_0 = 0$, $\lambda_2 = 0$, $\lambda_4 = 0$, $\lambda_6 = 0.01$. Same iteration procedure, initialization, data, mesh and periodic boundary conditions as for Fig. 14.)

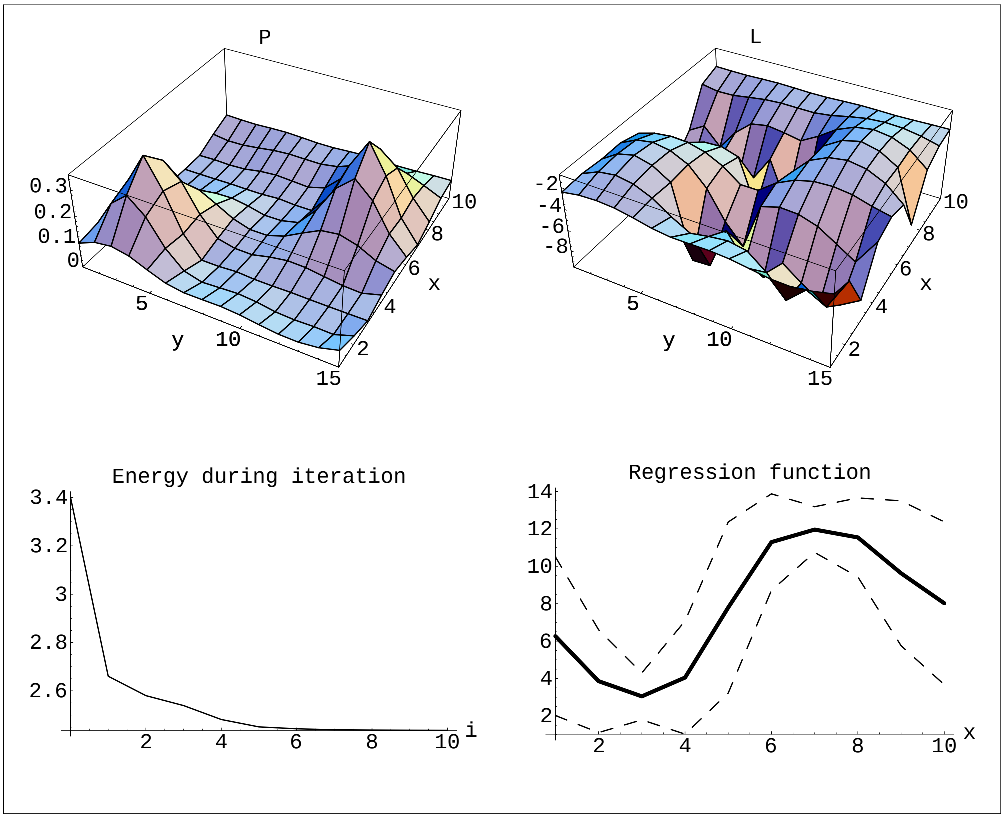

Figure 17: Density estimation with a $(-\Delta^3)$ Gaussian prior factor for the probability P. As the variation of P is smaller than that of L, a smaller λ_6 has been chosen than in Fig. 17. The Gaussian prior in P is also relatively less effective for small probabilities than a comparable Gaussian prior in L. (Smoothness prior $\mathbf{K}$ of the form of Eq. (705) with $\lambda_x = \lambda_y = 1$, $\lambda_0 = 0$, $\lambda_2 = 0$, $\lambda_4 = 0$, $\lambda_6 = 0.1$. Same iteration procedure, initialization, data, mesh and periodic boundary conditions as for Fig. 15.)

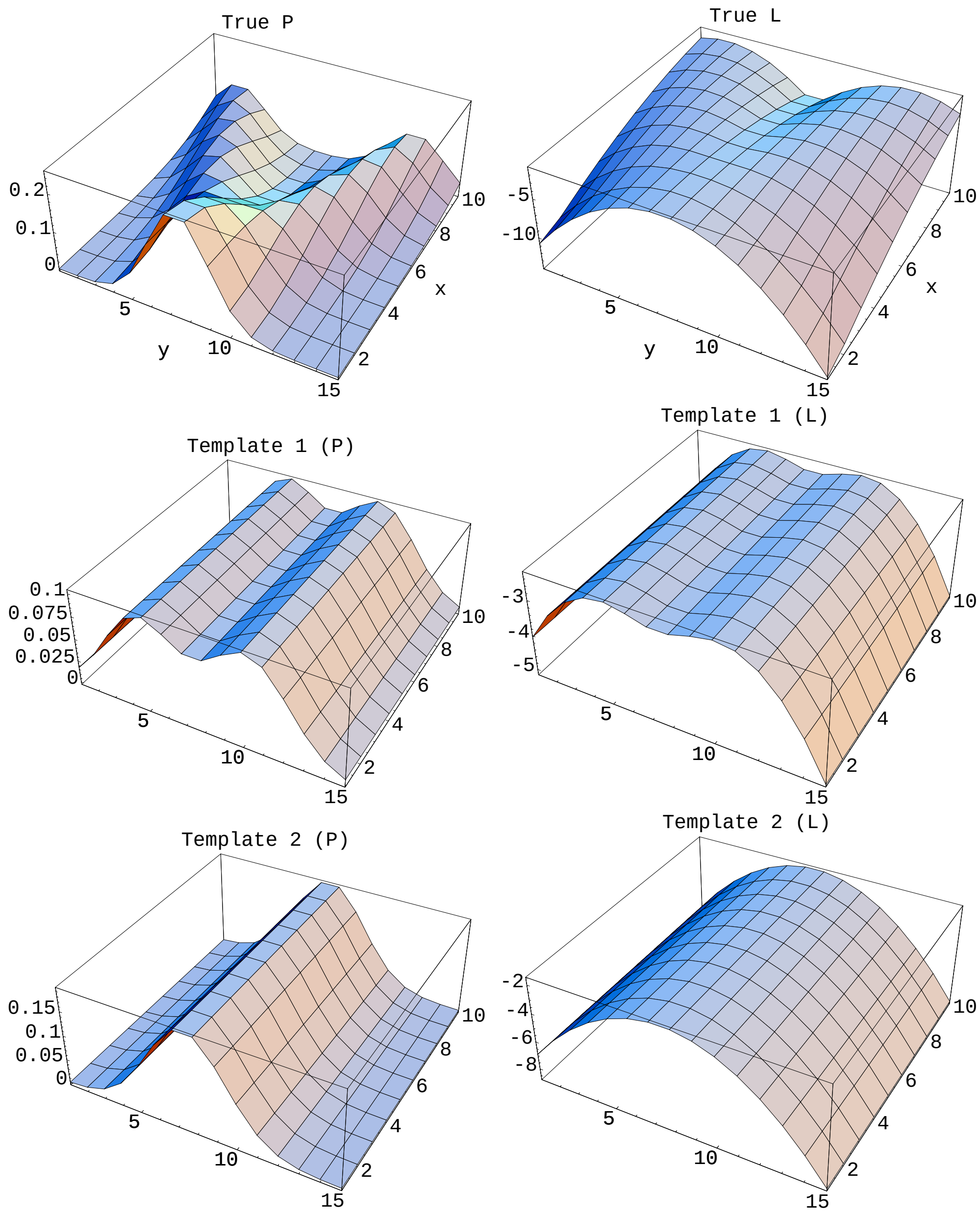

Figure 18: First row: True density P_{true} (l.h.s.) true log–density $L_{\text{true}} = \ln P_{\text{true}}$ (r.h.s.) used for Figs. 21–28. Second and third row: The two templates t_1 and t_2 of Figs. 23–28 for P (t_i^P, l.h.s.) or for L (t_i^L, r.h.s.), respectively, with $t_i^L = \ln t_i^P$. As reference for the following figures we give the expected test error $\int dy\, dx\, p(x) p(y|x, h_{\text{true}}) \ln p(y|x, h)$ under the true $p(y|x, h_{\text{true}})$ for uniform $p(x)$. It is for h_{true} equal to 2.23 for template t_1 equal to 2.56, for template t_2 equal 2.90 and for a uniform P equal to 2.68.

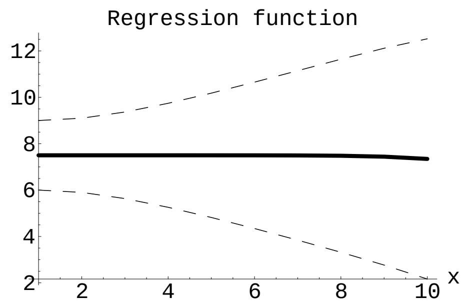

Figure 19: Regression function $h_{\rm true}(x)$ for the true density $P_{\rm true}$ of Fig. 18, defined as $h(x) = \int dy\, y p(y|x, h_{\rm true}) = \int dy\, y P_{\rm true}(x, y)$. The dashed lines indicate the range of one standard deviation above and below the regression function.

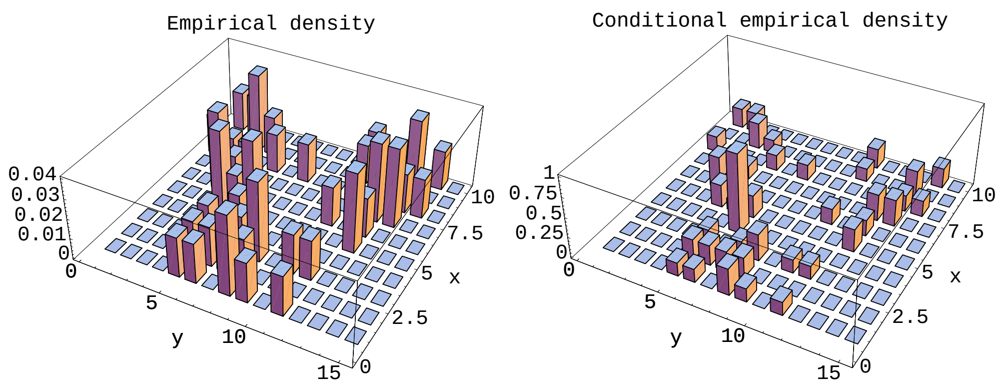

Figure 20: L.h.s.: Empirical density $N(x, y)/n = \sum_i \delta(x - x_i)\delta(y - y_i)/\sum_i 1$. sampled from $p(x, y|h_{\rm true}) = p(y|x, h_{\rm true})p(x)$ with uniform $p(x)$. R.h.s.: Corresponding conditional empirical density $P_{\rm emp}(x, y) = (\mathbf{N}_X^{-1} N)(x, y) = \sum_i \delta(x - x_i) \sum_i \delta(y - y_i) \sum_i / \sum_i \delta(x - x_i)$. Both densities are obtained from the 50 data points used for Figs. 21–28.

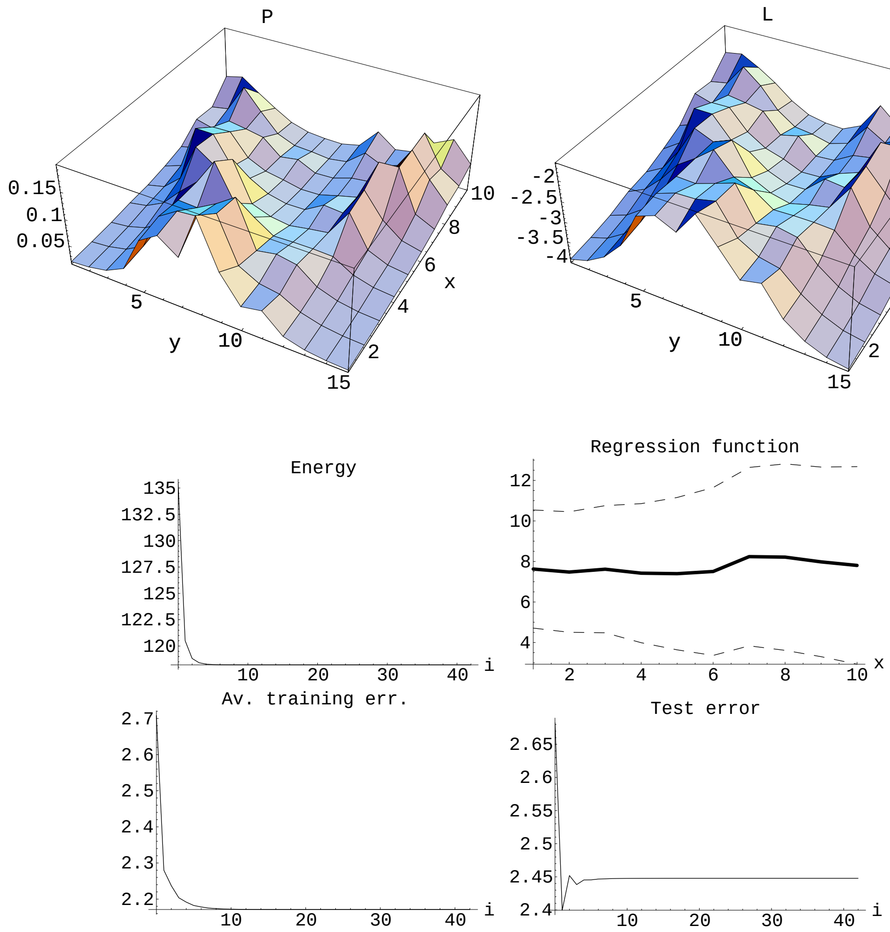

Figure 21: Density estimation with Gaussian prior factor for log–probability L with 50 data points shown in Fig. 20. Top row: Final solution $P(x,y) = p(y|x,h)$ and $L = \ln P$. Second row: Energy E_L (108) during iteration and final regression function. Bottom row: Average training error $-(1/n)\sum_{i=1}^{n} \ln p(y_i|x_i,h)$ during iteration and average test error $-\int dy\, dx\, p(x)p(y|x,h_{\text{true}}) \ln p(y|x,h)$ for uniform $p(x)$. (Parameters: Zero mean Gaussian smoothness prior with inverse covariance $\lambda\mathbf{K}$, $\lambda = 0.5$ and $\mathbf{K}$ of the form (705) with $\lambda_x = 2$, $\lambda_y = 1$, $\lambda_0 = 0$, $\lambda_2 = 1$, $\lambda_4 = \lambda_6 = 0$, massive prior iteration with $\mathbf{A} = \mathbf{K} + m^2\mathbf{I}$ and squared mass $m^2 = 0.01$. Initialized with normalized constant L. At each iteration step the factor η has been adapted by a line search algorithm. Mesh with 10 points in x-direction and 15 points in y–direction, periodic boundary conditions in y.)

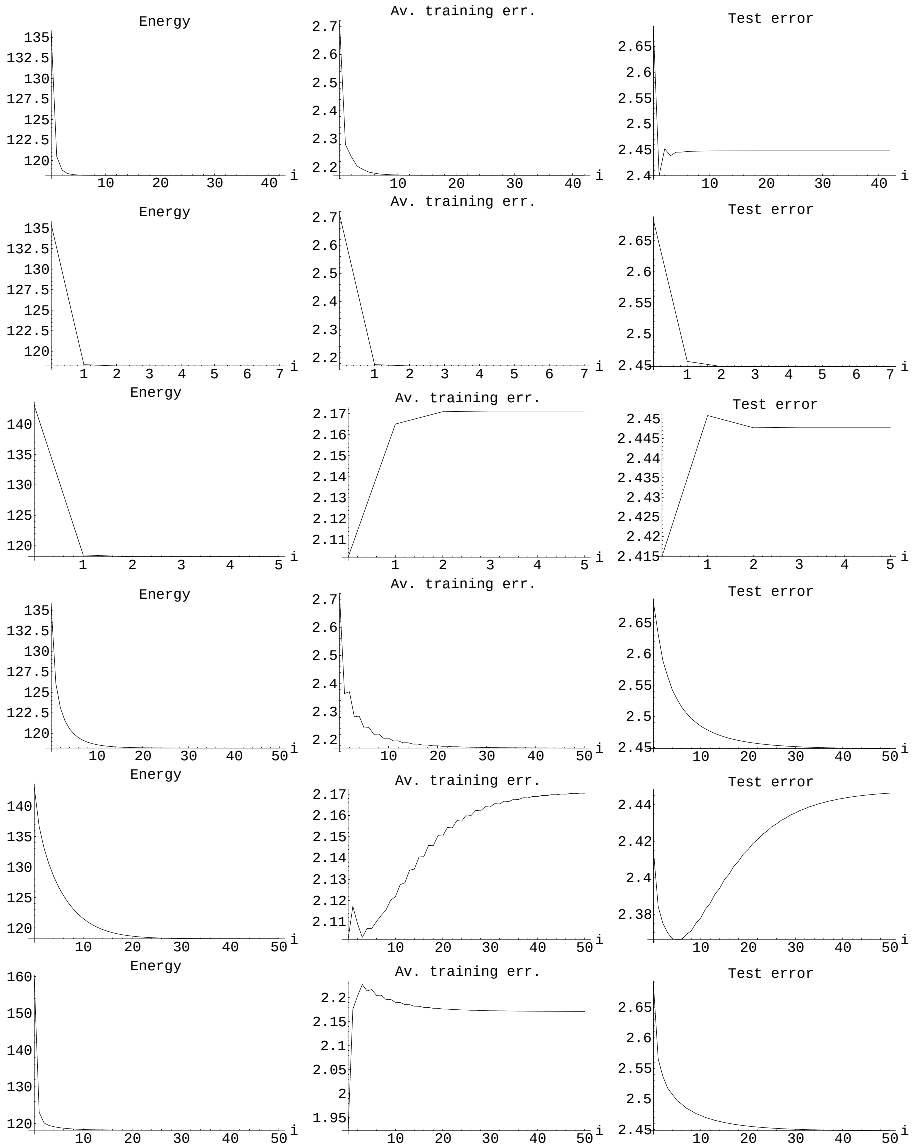

Figure 22: Comparison of iteration schemes and initialization. First row: Massive prior iteration (with $\mathbf{A} = \mathbf{K} + m^2\mathbf{I}$, $m^2 = 0.01$) and uniform initialization. Second row: Hessian iteration ($\mathbf{A} = -\mathbf{H}$) and uniform initialization. Third row: Hessian iteration and kernel initialization (with $\mathbf{C} = \mathbf{K} + m_C^2\mathbf{I}$, $m_C^2 = 0.01$ and normalized afterwards). Forth row: Gradient ($\mathbf{A} = \mathbf{I}$) with uniform initialization. Fifth row: Gradient with kernel initialization. Sixth row: Gradient with delta–peak initialization. (Initial L equal to $\ln(N/n + \epsilon)$, $\epsilon = 10^{-10}$, conditionally normalized. For N/n see Fig. 20). Minimal number of iterations 4, maximal number of iterations 50, iteration stopped if $|L^{(i)} - L^{(i-1)}| < 10^{-8}$. Energy functional and parameters as for Fig. 21.

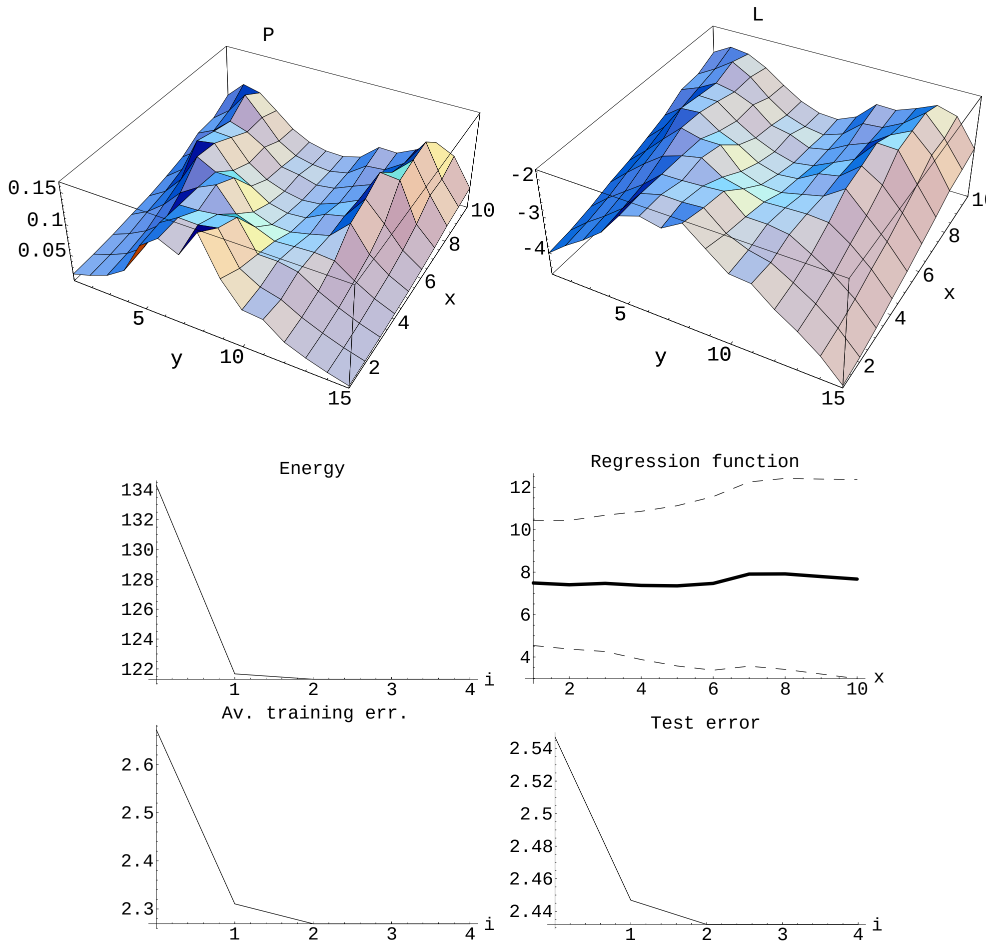

Figure 23: Density estimation with a Gaussian mixture prior for log–probability L with 50 data points, Laplacian prior and the two template functions shown in Fig. 18. Top row: Final solution $P(x, y) = p(y|x, h)$ and $L = \ln P$. Second row: Energy Energy E_L (719) during iteration and final regression function. Bottom row: Average training error $-(1/n)\sum_{i=1}^{n} \ln p(y_i|x_i, h)$ during iteration and average test error $-\int dy\, dx\, p(x)p(y|x, h_{\text{true}}) \ln p(y|x, h)$ for uniform $p(x)$. (Two mixture components with $\lambda = 0.5$ and smoothness prior with $\mathbf{K}_1 = \mathbf{K}_2$ of the form (705) with $\lambda_x = 2$, $\lambda_y = 1$, $\lambda_0 = 0$, $\lambda_2 = 1$, $\lambda_4 = \lambda_6 = 0$, massive prior iteration with $\mathbf{A} = \mathbf{K} + m^2\mathbf{I}$ and squared mass $m^2 = 0.01$, initialized with $L = t_1$. At each iteration step the factor η has been adapted by a line search algorithm. Mesh with $l_x = 10$ points in x-direction and $l_y = 15$ points in y–direction, $n = 2$ data points at $(3, 3)$, $(7, 12)$, periodic boundary conditions in y. Except for the inclusion of two mixture components parameters are equal to those for Fig. 21.)

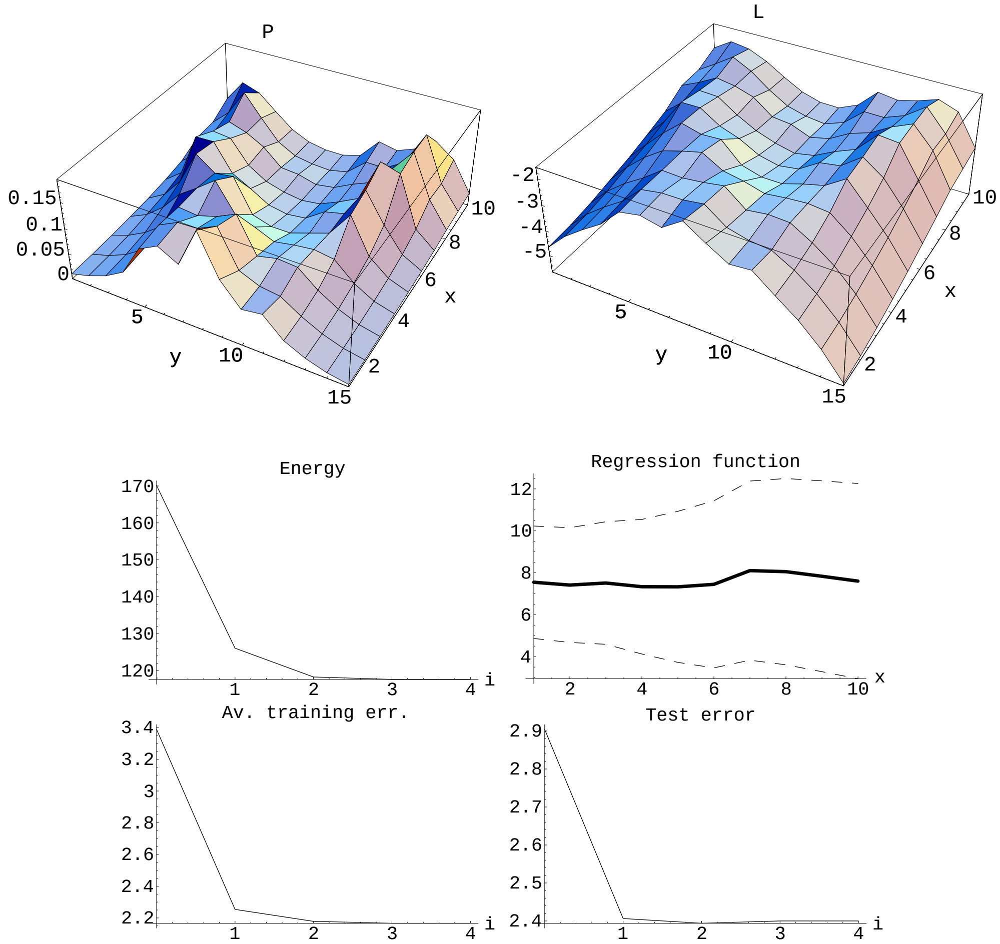

Figure 24: Using a different starting point. (Same parameters as for Fig. 23, but initialized with $L = t_2$.) While the initial guess is worse then that of Fig. 23, the final solution is even slightly better.

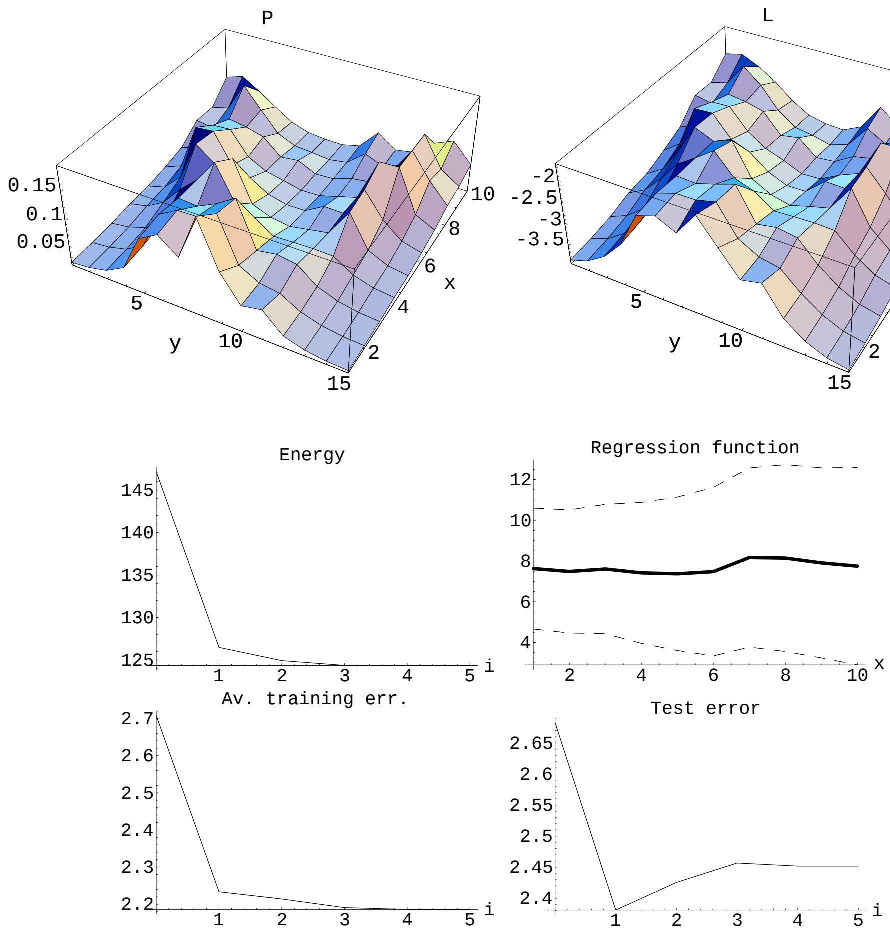

Figure 25: Starting from a uniform initial guess. (Same as Fig. 23, but initialized with uniform L.) The resulting solution is, compared to Figs. 23 and 24, a bit more wiggly, i.e., more data oriented. One recognizes a slight "overfitting", meaning that the test error increases while the training error is decreasing. (Despite the increasing of the test error during iteration at this value of λ, a better solution cannot necessarily be found by just changing λ–value. This situation can for example occur, if the initial guess is better then the implemented prior.)

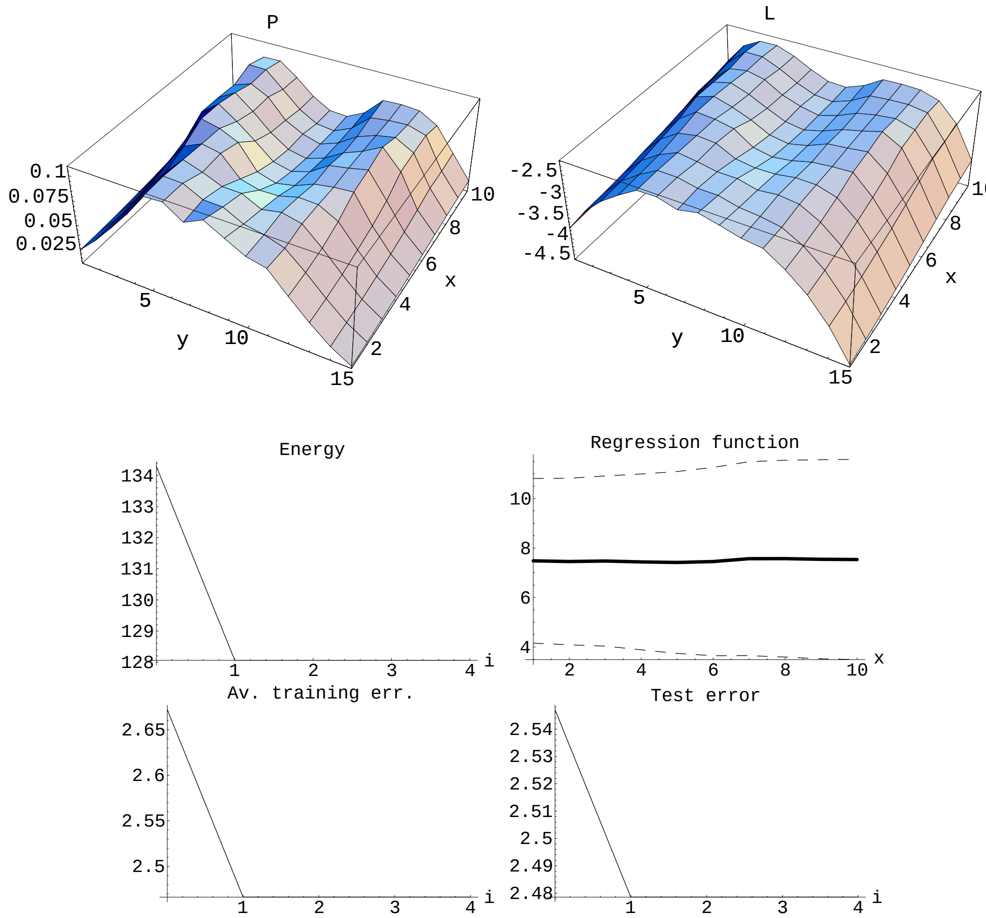

Figure 26: Large λ. (Same parameters as for Fig. 23, except for $\lambda = 1.0$.) Due to the larger smoothness constraint the averaged training error is larger than in Fig. 23. The fact that also the test error is larger than in Fig. 23 indicates that the value of λ is too large. Convergence, however, is very fast.

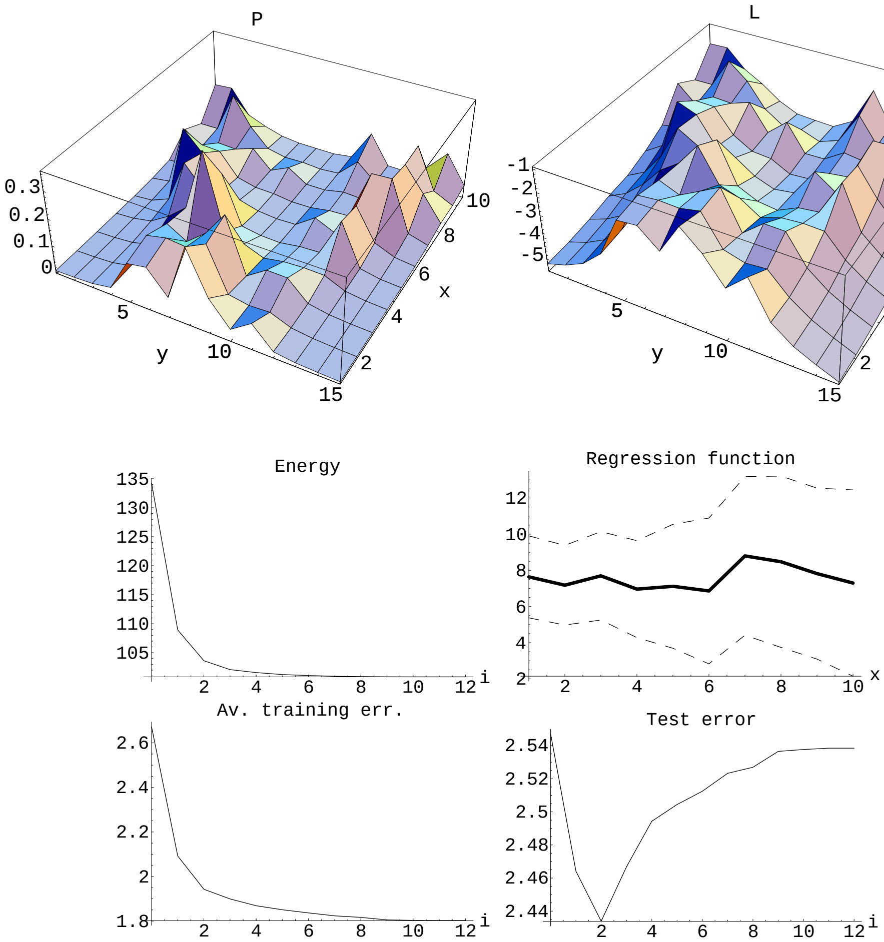

Figure 27: Overfitting due to too small λ. (Same parameters as for Fig. 23, except for $\lambda = 0.1$.) A small λ allows the average training error to become quite small. However, the average test error grows already after two iterations. (Having found at some λ–value during iteration an increasing test error, it is often but not necessarily the case that a better solution can be found by changing λ.)

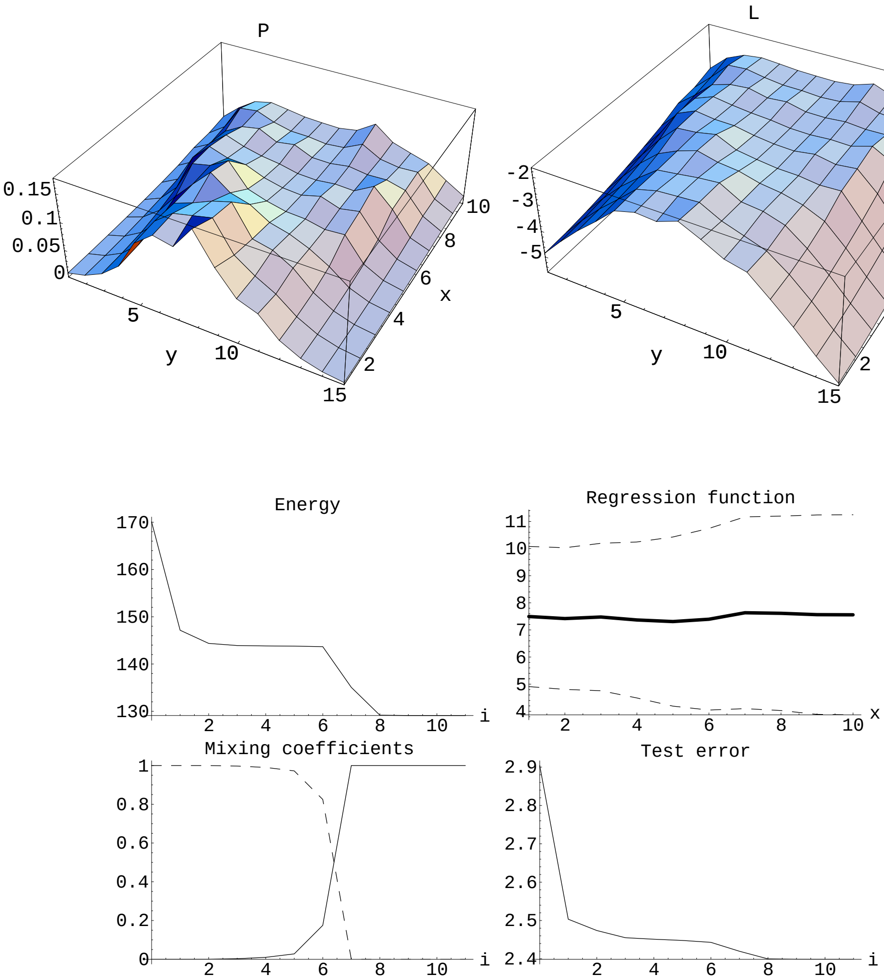

Figure 28: Example of an approximately stable solution. (Same parameters as for Fig. 23, except for $\lambda = 1.2$, $m^2 = 0.5$, and initialized with $L = t_2$.) A nearly stable solution is obtained after two iterations, followed by a plateau between iterations 2 and 6. A better solution is finally found with smaller distance to template t_1. (The plateau gets elongated with growing mass m.) The figure on the l.h.s. in the bottom row shows the mixing coefficients a_j of the components of the prior mixture model for the solution during iteration (a_1, line and a_2, dashed).