Sparse Multiscale Gaussian Process Regression

Christian Walder CHRISTIAN.WALDER@TUEBINGEN.MPG.DE
Kwang In Kim KIMKI@TUEBINGEN.MPG.DE
Bernhard Schölkopf BERHARD.SCHOELKOPF@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics
Spemannstr. 38, 72076 Tuebingen, Germany

Abstract

Most existing sparse Gaussian process (g.p.) models seek computational advantages by basing their computations on a set of m basis functions that are the covariance function of the g.p. with one of its two inputs fixed. We generalise this for the case of Gaussian covariance function, by basing our computations on m Gaussian basis functions with arbitrary diagonal covariance matrices (or length scales). For a fixed number of basis functions and any given criteria, this additional flexibility permits approximations no worse and typically better than was previously possible. We perform gradient based optimisation of the marginal likelihood, which costs $O(m^2n)$ time where n is the number of data points, and compare the method to various other sparse g.p. methods. Although we focus on g.p. regression, the central idea is applicable to all kernel based algorithms, and we also provide some results for the support vector machine (s.v.m.) and kernel ridge regression (k.r.r.). Our approach outperforms the other methods, particularly for the case of very few basis functions, *i.e.* a very high sparsity ratio.

1. Introduction

The Gaussian process (g.p.) is a popular non-parametric model for supervised learning problems. Although g.p.'s have been shown to perform well on a wide range of tasks, their usefulness is severely limited by the $O(n^3)$ time and $O(n^2)$ storage requirements where n is the number of data points. A large amount of work has been done to alleviate this problem, either by approximating the posterior distribution, or constructing degenerate covariance functions for which the exact posterior is less expensive to evaluate (Smola & Bartlett, 2000; Csató & Opper, 2002; Lawrence et al., 2002; Seeger et al., 2003; Snelson & Ghahramani, 2006) — for a unifying overview see (Quiñonero-Candela & Rasmussen, 2005). The majority of such methods achieve an $O(m^2n)$ time complexity for training where $m \ll n$ is the number of points on which the computations are based.

The g.p. can be interpreted as a linear (in the parameters) model which, due to its non-parametric nature, has potentially as many parameters to estimate as there are training points. An exception is the case where the covariance function has finite rank, such as the linear covariance function on $\mathbb{R}^d \times \mathbb{R}^d$ given by $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \boldsymbol{x}'$, which has rank d. In this case the g.p. collapses to a parametric method and it is possible to derive algorithms with $O(d^2n)$ time complexity by basing the computations on d basis functions.

For non-degenerate covariance functions, most existing sparse g.p. algorithms all have in common that they base their computations on m basis functions of the form $k(\boldsymbol{v}_i, \cdot)$. Typically the set $\mathcal{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m\}$ is taken to be a subset of the training set (Smola & Bartlett, 2000; Csató & Opper, 2002; Seeger et al., 2003). For example Seeger *et al.* (Seeger et al., 2003) employ a highly efficient approximate information gain criteria to incrementally select points from the training set in a greedy manner.

More recently Snelson and Ghahramani (2006) have shown that further improvements in the quality of the model for a given m can be made — especially for small m — by removing the restriction that $\mathcal{V}$ be a subset of the training set. For this they introduced a new sparse g.p. model which has the advantage of being closer to the full g.p., and also of being more amenable to gradient based optimisation of the marginal likelihood with respect to the set $\mathcal{V}$. A further advantage of their con-

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

tinuous optimisation of $\mathcal{V}$ is that the hyper-parameters of the model can be optimised at the same time — this is more difficult when $\mathcal{V}$ is taken to be a subset of the training set, since choosing such a subset is a hard combinatoric problem.

In this paper we take a logical step forward in the development of sparse g.p. algorithms. We also base our computations on a finite set of basis functions, but remove the restriction that the basis functions be of the form $k(\boldsymbol{v}_i, \cdot)$ where k is the covariance of the g.p. This will require computing integrals involving the basis and covariance functions, and so cannot always be done in closed form. Fortunately however, closed form expressions can be obtained for arguably the most useful scenario, namely that of Gaussian covariance function (with arbitrary diagonal covariance matrix) along with Gaussian basis functions (again each with their own arbitrary diagonal covariance matrix).

The central idea is that, under some mild restrictions, we can compute the prior probability density — under the g.p. model with Gaussian covariance — of arbitrary Gaussian mixtures. Our analysis is new, but there is a precedent for it in the literature. In particular, Walder et al. (2006) employ a similar idea, but from an reproducing kernel Hilbert space (r.k.h.s.) rather than a g.p. perspective, and for a different basis and covariance function. Also related is (Gehler & Franz, 2006), which analyses from a g.p. perspective with arbitrary basis and covariance function, but with the difference that they do not take infinite limits.

Our idea has a direct r.k.h.s. analogy. Indeed the main idea is applicable to any kernel machine, but in this paper we focus on the g.p. framework. The main reason for this is that it allows us to build on the sparse g.p. model of Snelson and Ghahramani (2006), which has been shown to be amenable to gradient based optimisation of the marginal likelihood. Nonetheless we do provide some experimental results for the kernel ridge regression (k.r.r.) case, as well as an animated toy example of the support vector machine (s.v.m.), in the accompanying video.

The paper is structured as follows. Section 2 provides an introduction to g.p. regression. In Section 3 we derive the likelihood of arbitrary Gaussian mixtures under the g.p. model with Gaussian covariance, and clarify the link to r.k.h.s.'s. In Section 4 we discuss and motivate the precise probabilistic model which we use to make practical use of our theoretical results. Experimental results and conclusions are presented in Sections 5 and 6, respectively.

2. Gaussian Process Regression

We assume that we are given an independent and identically distributed (i.i.d.) sample

$$\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R}$$

drawn from an unknown distribution, and the goal is to estimate $p(y|\boldsymbol{x})$. We introduce a latent variable $u \in \mathbb{R}$, and make the assumption that $p(y|u, \boldsymbol{x}) = p(y|u)$. Hence we can think of y as a noisy realisation of u, which we model by $p(y|u) = \mathcal{N}(y|u, \sigma_n^2)$ where σ_n is a hyper-parameter.[1]

The relationship $\boldsymbol{x} \to u$ is a random process $u(\cdot)$, namely a zero mean g.p. with covariance function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Typically k will be defined in terms of further hyper-parameters. We shall denote such a g.p. as $\mathcal{G}(k)$, which is defined by the fact that its joint evaluation at a finite number of input points is a zero mean Gaussian random variable with covariance

$$\mathbb{E}_{f\sim\mathcal{G}(k)}\left[f(\boldsymbol{x})f(\boldsymbol{z})\right] = k(\boldsymbol{x}, \boldsymbol{z}).$$

One can show that given the hyper-parameters, the posterior $p(\boldsymbol{u}|\mathcal{S})$, where[2] $[\boldsymbol{u}]_i = u(\boldsymbol{x}_i)$, is

$$\begin{aligned} p(\boldsymbol{u}|\mathcal{S}) &\propto p(\boldsymbol{u})\mathcal{N}(\boldsymbol{y}|\boldsymbol{u}, \sigma_n^2 I) \\ &\propto \mathcal{N}\left(\boldsymbol{u}|K_{xx}(K_{xx}+\sigma_n^2 I)^{-1}y, \sigma_n^2 K_{xx}(K_{xx}+\sigma_n^2 I)^{-1}\right), \end{aligned} \tag{1}$$

where $[K_{xx}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Like many authors we neglect to notate the conditioning upon the hyper-parameters, both in the above expression and for the remainder of the paper. Now, it can also be shown that the latent function $u_* = u(\boldsymbol{x}_*)$ at an arbitrary test point $\boldsymbol{x}_*$ is distributed according to $p(u_*|\boldsymbol{x}_*, \mathcal{S}) = \int p(u_*|\boldsymbol{x}_*, \mathcal{S}, \boldsymbol{u})p(\boldsymbol{u}|\boldsymbol{x}_*, \mathcal{S})\,\mathrm{d}\boldsymbol{u} = \mathcal{N}(u_*|\mu_*, \sigma_*^2)$, where

$$\mu_* = \boldsymbol{y}^\top (K_{xx} + \sigma_n^2 I)^{-1}\boldsymbol{k}_*, \tag{2}$$

$$\sigma_*^2 = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^\top (K_{xx} + \sigma_n^2 I)^{-1}\boldsymbol{k}_*, \tag{3}$$

and we have defined $[\boldsymbol{k}_*]_i = k(\boldsymbol{x}_*, \boldsymbol{x}_i)$.

In a Bayesian setting, one places priors over the hyper-parameters and computes the hyper-posterior, but this usually involves costly numerical integration techniques. Alternatively one may fix the hyper-parameters to those obtained by maximising some criteria such as the marginal likelihood conditioned upon them, $p(\boldsymbol{y}|\mathcal{X}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, K_y)$, where $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$

[1] We adopt the common convention of writing $\mathcal{N}(x|\mu, \sigma)$ for the probability density at x of the Gaussian random variable with mean μ and variance σ.

[2] Square brackets with subscripts denote elements of matrices and vectors, and a colon subscript denotes an entire row or column of a matrix.

and $K_{yy} = K_{xx} + \sigma_n^2 I$ is the covariance matrix for $\boldsymbol{y}$. This can be computed using the result that

$$\log\left(p(\boldsymbol{y}|\mathcal{X})\right) \propto -\boldsymbol{y}^\top K_{yy}^{-1}\boldsymbol{y} - \log|K_{yy}| + c, \quad (4)$$

where c is a term independent of the hyper-parameters. Even when one neglects the cost of choosing the hyper-parameters however, it typically costs $O(n)$ and $O(n^2)$ time to evaluate the posterior mean and variance respectively, after an initial setup cost of $O(n^3)$.

3. Sparse Multiscale Gaussian Process Regression

In this section we – loosely speaking – derive the likelihood of a mixture of Gaussians with arbitrary diagonal covariance matrices, under a g.p. prior with a covariance function that is also a Gaussian with arbitrary diagonal covariance matrix. Let u be drawn from $\mathcal{G}(k)$. As we mentioned previously, this means that the vector of joint evaluations at an arbitrary ordered set of points $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is a random variable, call it $\boldsymbol{u}_\mathcal{X}$, distributed according to

$$p_{\boldsymbol{u}_\mathcal{X}}(\boldsymbol{u}) = \mathcal{N}\left(\boldsymbol{u}|\boldsymbol{0}, K_{xx}\right). \quad (5)$$

Hence by definition

$$p_{\boldsymbol{u}_\mathcal{X}}(\textstyle\sum_{i=1}^m c_i\boldsymbol{u}_i) = \left|2\pi K_{xx}^{-1}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\sum_{i,j=1}^m c_i c_j \boldsymbol{u}_i^\top K_{xx}^{-1}\boldsymbol{u}_j\right),$$

where $|\cdot|$ denotes the matrix determinant. Note that this is simply the probability density function (p.d.f.) of $\boldsymbol{u}_\mathcal{X}$ where we have set the argument to be $\sum_{i=1}^m c_i\boldsymbol{u}_i$, for some $c_i \in \mathbb{R}$. We have done this because later we will wish to determine the likelihood of a function expressed as a summation of fixed basis functions. To this end we now consider an infinite limit of the above case. Taking the limit $n \to \infty$ of uniformly distributed points[3] $\boldsymbol{x}_i$ leads to the following p.d.f. for $\mathcal{G}(k)$,

$$p_{\mathcal{G}(k)}(\textstyle\sum_{i=1}^m c_i u_i) = \left|2\pi k^{-1}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\sum_{i,j=1}^m c_i c_j \Psi_k(u_i, u_j)\right), \quad (6)$$

where

$$\Psi_k(u_i, u_j) \triangleq \int\int k^{-1}(\boldsymbol{x}, \boldsymbol{y}) u_i(\boldsymbol{x}) u_j(\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}. \quad (7)$$

We will discuss the factor of $\left|2\pi k^{-1}\right|^{-\frac{1}{2}}$ shortly. Note that in the previous case of finite n, if we let $\boldsymbol{u} = K_{xx}\boldsymbol{\alpha}$ and assume that K_{xx} is invertible, then $\boldsymbol{\alpha} = K_{xx}^{-1}\boldsymbol{u}$. Following this finite analogy, by k^{-1} we now intend a sloppy notation for the function which, for $u = \int \alpha(\boldsymbol{x}) k(\boldsymbol{x}, \cdot)\,\mathrm{d}\boldsymbol{x}$, satisfies $\int u(\boldsymbol{x}) k^{-1}(\boldsymbol{x}, \cdot)\,\mathrm{d}\boldsymbol{x} = \alpha(\cdot)$. Hence if we define

$$M_k : \alpha \mapsto M_k\alpha = \int \alpha(\boldsymbol{x}) k(\boldsymbol{x}, \cdot)\,\mathrm{d}\boldsymbol{x},$$

then k^{-1} is by definition the Green's function (Roach, 1970) of M_k, as it satisfies

$$\int \big(M_k\alpha\big)(\boldsymbol{x}) k^{-1}(\boldsymbol{x}, \cdot)\,\mathrm{d}\boldsymbol{x} = \alpha(\cdot). \quad (8)$$

Let us now consider the covariance function given by $k(\boldsymbol{x}, \boldsymbol{y}) = cg(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma})$, where $c > 0$, $\boldsymbol{\sigma} > \boldsymbol{0} \in \mathbb{R}^d$ and g is a normalised Gaussian on $\mathbb{R}^d \times \mathbb{R}^d$ with diagonal covariance matrix, that is[4]

$$g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma}) \triangleq |2\pi\mathrm{diag}\,(\boldsymbol{\sigma})|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^d \frac{([\boldsymbol{x} - \boldsymbol{y}]_i)^2}{[\boldsymbol{\sigma}]_i}\right). \quad (9)$$

If we assume furthermore that our function is an arbitrary mixture of such Gaussians, so that

$$u_i(\boldsymbol{x}) = g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i), \quad (10)$$

then the well known integral (for the convolution of two Gaussians)

$$\int g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i) g(\boldsymbol{x}, \boldsymbol{v}_j, \boldsymbol{\sigma}_j)\,\mathrm{d}\boldsymbol{x} = g(\boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j), \quad (11)$$

leads to

$$\left(\frac{1}{c} M_{cg(\cdot,\cdot,\boldsymbol{\sigma})} g(\cdot, \boldsymbol{v}_i, \boldsymbol{\sigma}_i - \boldsymbol{\sigma})\right)(\boldsymbol{x}) = g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i) = u_i(\boldsymbol{x}). \quad (12)$$

As the covariance function and the basis functions are all Gaussian we can obtain in closed form

$$\begin{aligned}\Psi_k(u_i, u_j) &\overset{(7,10,12)}{=} \int\int k^{-1}(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i) \\ &\quad \cdot \left(\frac{1}{c} M_{cg(\cdot,\cdot,\boldsymbol{\sigma})} g(\cdot, \boldsymbol{v}_j, \boldsymbol{\sigma}_j - \boldsymbol{\sigma})\right)(\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} \\ &\overset{(8)}{=} \frac{1}{c}\int g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i) g(\boldsymbol{x}, \boldsymbol{v}_j, \boldsymbol{\sigma}_j - \boldsymbol{\sigma})\,\mathrm{d}\boldsymbol{x} \\ &\overset{(11)}{=} \frac{1}{c}\, g(\boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j - \boldsymbol{\sigma}).\end{aligned}$$

[3] Although any non-vanishing distribution leads to the same result.

[4] We use diag in a sloppy fashion with two meanings — for $\boldsymbol{a} \in \mathbb{R}^n$, $\mathrm{diag}(\boldsymbol{a}) \in \mathbb{R}^{n\times n}$ is a diagonal matrix satisfying $[\mathrm{diag}(\boldsymbol{a})]_{ii} = [\boldsymbol{a}]_i$. But for $A \in \mathbb{R}^{n\times n}$, $\mathrm{diag}(A) \in \mathbb{R}^n$ is a column vector with $[\mathrm{diag}(A)]_i = [A]_{ii}$

For clarity we have noted above each equals sign the number of the equation which implies the corresponding logical step. The following expression summarises the main idea of the present section

$$p_{\mathcal{G}(cg(\cdot,\cdot,\boldsymbol{\sigma}))}\Big(\sum_{i=1}^{m} c_i g(\cdot,\boldsymbol{v}_i,\boldsymbol{\sigma}_i)\Big)$$
$$\propto \exp\Big(-\frac{1}{2}\sum_{i,j=1}^{m}\frac{1}{c}c_i c_j g(\boldsymbol{v}_i,\boldsymbol{v}_j,\boldsymbol{\sigma}_i+\boldsymbol{\sigma}_j-\boldsymbol{\sigma})\Big). \quad (13)$$

We give only an unnormalised form by neglecting the factor $\left|2\pi k^{-1}\right|^{-\frac{1}{2}}$ in (6). The neglected factor is equal to the inverse of the integral of the right hand side of the above expression with respect to all functions $\sum_{i=1}^{m} c_i g(\cdot,\boldsymbol{v}_i,\boldsymbol{\sigma}_i)$. We need not concern ourselves with choosing a measure with respect to which this integral is finite, due to the fact that, since we will be working only with ratios of the above likelihood (*i.e.* for *maximum a posteriori* (m.a.p.) estimation and marginal likelihood maximisation), we need only the unnormalised form. Note that this peculiarity is not particular to our proposed sparse approximation to the g.p., but is a property of g.p.'s in general.

Interpretation We now make two remarks regarding the expression (13). **i)** If $\boldsymbol{\sigma}_1=\boldsymbol{\sigma}_2=\cdots=\boldsymbol{\sigma}_n=\boldsymbol{\sigma}$ and we reparameterise $c_i=cc_i'$ then it simplifies to (5). **ii)** Let $c=1$ and $h(\boldsymbol{x})=\exp\left(-\frac{1}{2}\boldsymbol{x}^\top \mathrm{diag}\left(\boldsymbol{\sigma}_1\right)^{-1}\boldsymbol{x}\right)$, an unnormalised Gaussian. Using (9) and (13) we can derive the log-likelihood of h under the g.p. prior,

$$\log\Big(p_{\mathcal{G}(g(\cdot,\cdot,\boldsymbol{\sigma}))}\big(h(\cdot)\big)\Big)\propto -\sqrt{\frac{|\mathrm{diag}\left(\boldsymbol{\sigma}_1\right)|}{|\mathrm{diag}\left(2\boldsymbol{\sigma}_1-\boldsymbol{\sigma}\right)|}}. \quad (14)$$

Simple analysis of this expression shows that the most likely such function h is that with $\boldsymbol{\sigma}_1=\boldsymbol{\sigma}$. From this extremal point, as any component of $\boldsymbol{\sigma}_1$ increases, the log likelihood of h decreases without bound. Similarly decreasing any component of $\boldsymbol{\sigma}_1$ also decreases the log likelihood, and as any component of $\boldsymbol{\sigma}_1$ approaches half the value of the corresponding component of $\boldsymbol{\sigma}$, then the log likelihood decreases without bound. To be more precise, we have for all $j=1,2,\ldots,d$ that

$$\lim_{[\boldsymbol{\sigma}_1]_j\to\left(\frac{1}{2}[\boldsymbol{\sigma}]_j\right)^+}\log\Big(p_{\mathcal{G}(g(\cdot,\cdot,\boldsymbol{\sigma}))}\big(h(\cdot)\big)\Big)=-\infty.$$

An interesting consequence of the second remark is that, roughly speaking, it is not possible to recover a Gaussian function using a g.p. with Gaussian covariance, if the covariance function is more than twice as broad as the function to be recovered. Although this may at first appear to contradict proven consistency results for the Gaussian covariance function (for example (Steinwart, 2002)), this is not the case. On the contrary, such results hold only for compact domains, and our analysis is for $\mathbb{R}^d$.

An r.k.h.s. Analogy We note that (13) has a direct analogy in the theory of r.k.h.s.'s, as made clear by the following lemma. The lemma follows from (13) and the well understood relationship between every g.p. and the corresponding r.k.h.s. of functions.

Lemma 3.1. *Let $\mathcal{H}$ be the r.k.h.s. with reproducing kernel $g(\cdot,\cdot,\boldsymbol{\sigma})$. If the conditions $\boldsymbol{\sigma}_i>\frac{1}{2}\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}_j>\frac{1}{2}\boldsymbol{\sigma}$ are satisfied component-wise, then*

$$\langle g(\cdot,\boldsymbol{v}_i,\boldsymbol{\sigma}_i),g(\cdot,\boldsymbol{v}_j,\boldsymbol{\sigma}_j)\rangle_{\mathcal{H}}=g(\boldsymbol{v}_i,\boldsymbol{v}_j,\boldsymbol{\sigma}_i+\boldsymbol{\sigma}_j-\boldsymbol{\sigma}). \quad (15)$$

If either condition is not satisfied, then the corresponding function on the left hand side is not in $\mathcal{H}$.

Naturally this can also be proven directly, but doing so for the general case is more involved and we omit the details due to space limitations.[5] However, by assuming that the conditions $\boldsymbol{\sigma}_i>\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}_j>\boldsymbol{\sigma}$ are satisfied component-wise, then it is straightforward to obtain the main result. The basic idea is as follows. Using (11) we substitute $g(\cdot,\boldsymbol{v}_p,\boldsymbol{\sigma}_p)=\int g(\cdot,\boldsymbol{x}_p,\boldsymbol{\sigma})g(\boldsymbol{x}_p,\boldsymbol{v}_p,\boldsymbol{\sigma}_p-\boldsymbol{\sigma})\,\mathrm{d}\boldsymbol{x}_p$ for $p=i,j$ into the l.h.s. of (15). By linearity we can write the two integrals outside the inner product. Next we use the r.k.h.s. reproducing property — the fact that $\langle f(\cdot),g(\cdot,\boldsymbol{x},\boldsymbol{\sigma})\rangle_{\mathcal{H}}=f(\boldsymbol{x}),\forall f\in\mathcal{H},\boldsymbol{x}\in\mathbb{R}^d$ — to evaluate the inner product. Using (11) we integrate to obtain the r.h.s. of (15).

4. Inference with the Sparse Model

4.1. A Simple Approach

In the previous section we derived the g.p. likelihood over a certain restricted function space. This likelihood defines a distribution over functions of the form $\sum_{i=1}^{m} c_i g(\cdot,\boldsymbol{v}_i,\boldsymbol{\sigma}_i)$ where g as given previously is deterministic and the c_i are, by inspection of (13), normally distributed according to

$$\boldsymbol{c}\sim\mathcal{N}\left(\boldsymbol{0},U_\Psi^{-1}\right), \quad (16)$$

where $[U_\Psi]_{i,j}=\Psi_k(u_i,u_j)$. Let us write $\mathcal{U}=\{u_1,\ldots,u_m\}$ (which we refer to as the basis) and refer to the random process thus defined as $\mathcal{G}_{\mathcal{U}}(k)$. This new random process is equivalent to a full g.p. with

[5] For ICML reviewing, we can provide proof on request.

covariance function of rank at most m given by

$$\mathbb{E}_{f\sim\mathcal{G}_{\mathcal{U}}(k)}\left[f(\boldsymbol{x})f(\boldsymbol{z})\right] = \mathbb{E}_{\boldsymbol{c}\sim\mathcal{N}\left(\boldsymbol{0},U_\Psi^{-1}\right)}\left[\left(\boldsymbol{u}_{vx}^\top\boldsymbol{c}\right)\left(\boldsymbol{u}_{vz}^\top\boldsymbol{c}\right)^\top\right] = \boldsymbol{u}_{vx}^\top U_\Psi^{-1}\boldsymbol{u}_{vz}, \tag{17}$$

where $[\boldsymbol{u}_{vx}]_i = g(\boldsymbol{x},\boldsymbol{v}_i,\boldsymbol{\sigma}_i)$ and $[\boldsymbol{u}_{vz}]_i = g(\boldsymbol{z},\boldsymbol{v}_i,\boldsymbol{\sigma}_i)$.

As an aside, note that if we choose as the basis $\mathcal{U} = \{g(\cdot,\boldsymbol{x},\boldsymbol{\sigma}), g(\cdot,\boldsymbol{z},\boldsymbol{\sigma})\}$, then it is easy to verify using (17) that $\mathbb{E}_{f\sim\mathcal{G}_{\mathcal{U}}(g(\cdot,\cdot,\boldsymbol{\sigma}))}\left[f(\boldsymbol{x})f(\boldsymbol{z})\right] = \mathbb{E}_{f\sim\mathcal{G}(k)}\left[f(\boldsymbol{x})f(\boldsymbol{z})\right]$. This is analogous to a special case of the representer theorem from the theory of r.k.h.s.'s, and agrees with the interpretation that (16) is such that $\mathcal{G}_{\mathcal{U}}(k)$ approximates $\mathcal{G}(k)$ well in some sense, for the given basis $\mathcal{U}$.

Returning to the main thread, the new posterior can be derived as it was at the end of Section 2 for the exact g.p., but using the new covariance function (17). Hence after some algebra we have from (2) and (3) that, conditioned again upon the hyper-parameters, the latent function $u_* = u(\boldsymbol{x}_*)$ at an arbitrary test point is distributed according to $p_{u\sim\mathcal{G}_{\mathcal{U}}(k)}(u_*|\boldsymbol{x}_*,\mathcal{S}) = \mathcal{N}(u_*|\mu_*,\sigma_*^2)$, where

$$\mu_* = \left(U_{vx}\boldsymbol{y}\right)^\top\left(U_{vx}U_{vx}^\top + \sigma_n^2 U_\Psi\right)^{-1}\boldsymbol{u}_{v*}, \tag{18}$$

$$\sigma_*^2 = \sigma_n^2\boldsymbol{u}_{v*}^\top\left(U_{vx}U_{vx}^\top + \sigma_n^2 U_\Psi\right)^{-1}\boldsymbol{u}_{v*}, \tag{19}$$

and we have defined $[U_{vx}]_{i,j} = g(\boldsymbol{x}_j,\boldsymbol{v}_i,\boldsymbol{\sigma}_i)$, *etc.* Note that these expressions can be evaluated in $O(m)$ and $O(m^2)$ time respectively, after an initial setup or training cost of $O(m^2n)$. This is the usual improvement over the full g.p. obtained by such sparse approximation schemes. It turns out however that by employing an idea introduced by Snelson and Ghahramani (2006), we can retain these computational advantages while switching to a different model that is closer to the full g.p.

4.2. Inference with Improved Variance

A fair criticism of the previous model is that the predictive variance approaches zero far away from the basis function centres $\boldsymbol{v}_i$, as can be seen from (19). It turns out that this is particularly problematic to gradient based methods for choosing the basis (the $\boldsymbol{v}_i$ and $\boldsymbol{\sigma}_i$) by maximising the marginal likelihood (Snelson & Ghahramani, 2006). An effective but still computationally attractive way of healing the model is to switch to a different g.p. — which we denote $\tilde{\mathcal{G}}_{\mathcal{U}}(k)$ — whose covariance function satisfies

$$\mathbb{E}_{\tilde{\mathcal{G}}_{\mathcal{U}}(k)}\left[f(\boldsymbol{x})f(\boldsymbol{z})\right] = \delta_{\boldsymbol{x},\boldsymbol{z}}k(\boldsymbol{x},\boldsymbol{z}) + \overline{\delta}_{\boldsymbol{x},\boldsymbol{z}}\boldsymbol{u}_{vx}^\top U_\Psi^{-1}\boldsymbol{u}_{vz}, \tag{20}$$

where $\delta_{\boldsymbol{a},\boldsymbol{b}}$ is the Kronecker delta function and $\overline{\delta}_{\boldsymbol{a},\boldsymbol{b}} = 1 - \delta_{\boldsymbol{a},\boldsymbol{b}}$. Note that if $\boldsymbol{x} = \boldsymbol{z}$ then the covariance is that of the original g.p. $\mathcal{G}(k)$, otherwise it is that of $\mathcal{G}_{\mathcal{U}}(k)$. Unlike (17), the prior variance in this case is the same as that of the full g.p., even though in general the covariance is not. Once again the posterior can be found as before by replacing the covariance function in (2) and (3) with the right hand side of (20). In this case we obtain after some algebra the expression $p_{u\sim\tilde{\mathcal{G}}_{\mathcal{U}}(k)}(u_*|\boldsymbol{x}_*,\mathcal{S}) = \mathcal{N}(u_*|\mu_*,\sigma_*^2)$, where

$$\mu_* = \boldsymbol{u}_{v*}^\top Q^{-1}U_{vx}\left(\Lambda + \sigma_n^2 I\right)^{-1}\boldsymbol{y}, \tag{21}$$

$$\sigma_*^2 = k(\boldsymbol{x}_*,\boldsymbol{x}_*) - \boldsymbol{u}_{v*}^\top\left(U_\Psi^{-1} - Q^{-1}\right)\boldsymbol{u}_{v*}, \tag{22}$$

$\Lambda = \operatorname{diag}(\boldsymbol{\lambda})$, and

$$[\boldsymbol{\lambda}]_i = k(\boldsymbol{v}_i,\boldsymbol{v}_i) - [U_{vv}]_{:,i}^\top U_\Psi^{-1}[U_{vv}]_{:,i},$$
$$Q = U_\Psi + U_{vx}\left(\Lambda + \sigma_n^2 I\right)^{-1}U_{vx}^\top.$$

To compute the marginal likelihood we can use the expression (4). Note that it can be computed efficiently using Cholesky decompositions. In order to optimize the marginal likelihood, we also need its gradients with respect to the various parameters. Our derivation of the gradients (which closely follows (Seeger et al., 2003)) is long and tedious, and has been omitted due to space limitations. Note that by factorising appropriately, all of the required gradients can be obtained in $O(m^2n + mnd)$.

4.3. A Unifying View

We now briefly outline how the method of the previous section fits into the unifying framework of sparse g.p.'s provided by Quiñonero-Candela and Rasmussen (2005). Using Bayes rule and marginalising out the training set latent variables $\boldsymbol{u}$, we obtain the posterior

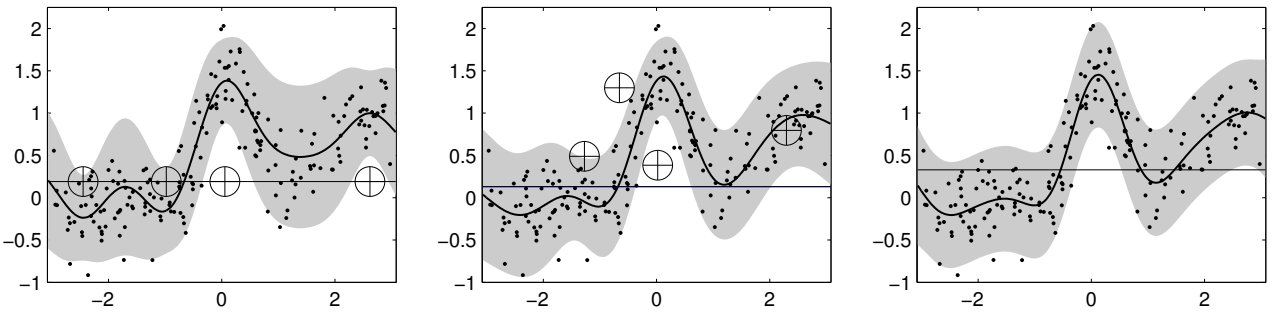
$$p(u_*|\boldsymbol{y}) = \frac{1}{p(\boldsymbol{y})}\int p(\boldsymbol{y}|\boldsymbol{u})p(\boldsymbol{u},u^*)\,\mathrm{d}\boldsymbol{u}.$$

Here we have neglected to notate conditioning on $\boldsymbol{x}^*$ and $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$, and have written p instead of the more precise $p_{u\sim\mathcal{G}(k)}$. Our algorithm can be interpreted as employing two separate approximations. The first is conditional independence of $\boldsymbol{u}$ and u^* given $\boldsymbol{a}$, *i.e.*

$$p(\boldsymbol{u},u^*) = \int p(\boldsymbol{u},u^*|\boldsymbol{a})p(\boldsymbol{a})\,\mathrm{d}\boldsymbol{a} \approx \int p(\boldsymbol{u}|\boldsymbol{a})p(u^*|\boldsymbol{a})p(\boldsymbol{a})\,\mathrm{d}\boldsymbol{a},$$

where $\boldsymbol{a}$ (which is marginalised out) is taken to be

$$\left(\langle u_1,u\rangle_{\mathcal{H}}\ \ \langle u_2,u\rangle_{\mathcal{H}}\ \ \cdots\ \ \langle u_m,u\rangle_{\mathcal{H}}\right)^\top,$$



(a) *spgp-full* (b) Our *vsgp-full* (c) Exact g.p.

Figure 1. Predictive distributions (mean curve with ± two standard deviations shaded). For the *spgp-full* and *vsgp-full* algorithms, we plot the $(\boldsymbol{v}_i, \boldsymbol{\sigma}_i) \in \mathbb{R} \times \mathbb{R}$ of the basis as crossed circles. The horizontal lines denote the resulting $\boldsymbol{\sigma} \in \mathbb{R}$ of the covariance function $cg(\cdot, \cdot, \boldsymbol{\sigma})$.

the vector of inner products between the basis functions u_i and the latent function u, in the r.k.h.s. $\mathcal{H}$ associated with $k(\cdot, \cdot)$. The second approximation is

$$\begin{aligned} p(\boldsymbol{u}|\boldsymbol{a}) &= \mathcal{N}\left(U_{xv}U_\Psi^{-1}\boldsymbol{v}, K_{xx} - U_{xv}U_\Psi^{-1}U_{xv}^\top\right) \\ &\approx \mathcal{N}\left(U_{xv}U_\Psi^{-1}\boldsymbol{v}, \mathrm{diag}'\left(K_{xx} - U_{xv}U_\Psi^{-1}U_{xv}^\top\right)\right). \end{aligned}$$

where $\mathrm{diag}'(A)$ is a diagonal matrix matching A on the diagonal, and $[K_{xv}]_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{v}_j)$, *etc.* Note that the first line can be shown with some algebra, whereas the second is an approximation. One can show that this leads to the result of Section 4.2, but we omit the details for brevity. Of the algorithms considered in (Quiñonero-Candela & Rasmussen, 2005), ours is closest to that of Snelson and Ghahramani (2006), however there the basis functions take the form $u_i = k(\boldsymbol{v}_i, \cdot)$, which has two implications. Firstly, $\boldsymbol{a}$ simplifies to

$$\left(u(\boldsymbol{v}_1)\ u(\boldsymbol{v}_2)\ \cdots\ u(\boldsymbol{v}_m)\right)^\top,$$

the vector of the values of u at $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$. Secondly, U_Ψ and U_{xv} simplify to K_{vv} and K_{xv}, respectively.

5. Experiments

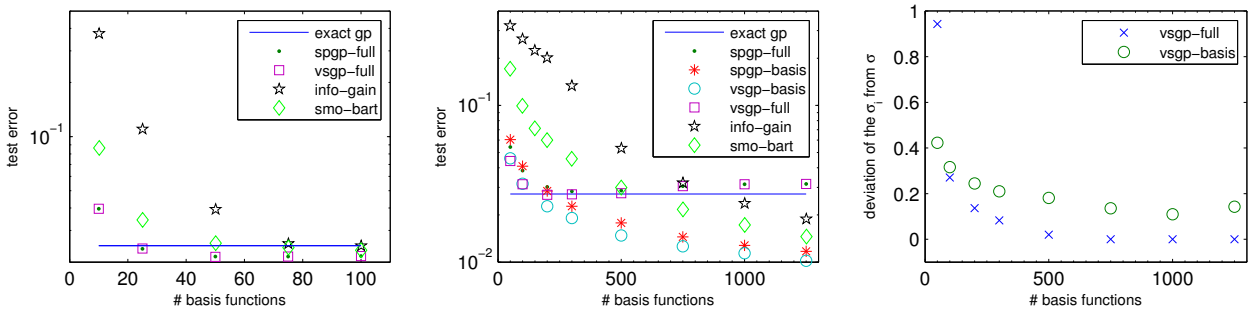
Our main goal is to demonstrate the value of being able to vary the $\boldsymbol{\sigma}_i$ individually. Note that the chief advantage of our method is in producing highly sparse solutions, and the results represent the state of the art in this respect. As such, and since the prediction cost is $O(md)$, we analyse the predictive performance of the model as a function of the number of basis functions m. Note that neither our method nor the most closely related method of Snelson and Ghahramani (2006) are particularly competitive in terms of training time. Nonetheless, there is a demand for algorithms which sacrifice training speed for testing speed, such as real-time vision and control systems, and web services in which the number of queries is large.

Let us clarify the terminology we use to refer to the various algorithms under comparison. Our new method is the variable sigma Gaussian process (v.s.g.p.). The *vsgp-full* variant consists of optimising the marginal likelihood with respect to the m basis centers $\boldsymbol{v}_i \in \mathbb{R}^d$ and length scales $\boldsymbol{\sigma}_i \in \mathbb{R}^d$ of our basis functions $u_i = g(\cdot, \boldsymbol{v}_i, \boldsymbol{\sigma}_i)$ where g is defined in (9). Also optimised are the following hyper parameters — the noise variance $\sigma_n \in \mathbb{R}$ of (1), and the parameters $c \in \mathbb{R}$ and $\boldsymbol{\sigma} \in \mathbb{R}^d$ of our original covariance function $cg(\cdot, \cdot, \boldsymbol{\sigma})$. The *vsgp-basis* variant is identical to *vsgp-full* except that σ_n, c and $\boldsymbol{\sigma}$ are determined by optimising the marginal likelihood of a full g.p. trained on a subset of the training data, and then held fixed while the $\boldsymbol{\sigma}_i$ and $\boldsymbol{v}_i$ are optimised as before. Both v.s.g.p. variants use the $\tilde{\mathcal{G}}_{\mathcal{U}}(k)$ probabilistic model of Section 4.2, where $k = cg(\cdot, \cdot, \boldsymbol{\sigma})$. For the optimisation of the sparse pseudo-input Gaussian process (s.p.g.p.) and v.s.g.p. methods we used a standard conjugate gradient type optimiser.[6]

spgp-full and *spgp-basis* correspond to the work of Snelson and Ghahramani (2006), and are identical to their v.s.g.p. counterparts except that — as with all sparse g.p. methods prior to the present work — they are forced to satisfy the constraints $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}, i = 1 \ldots m$. To initialise the marginal likelihood optimisation we take the $\boldsymbol{v}_i$ to be a *k-means* clustering of the training data. The other parameters are always initialised to the same sensible starting values, which is reasonable due to the preprocessing we employ (which is identical to that of (Seeger et al., 2003)) in order to standardise the data sets.

Figure 1 demonstrates the basic idea on a one dimensional toy problem. Using $m = 4$ basis functions is not

[6] Carl Rasmussen's `minimize.m`, which is freely available from `http://www.kyb.mpg.de/~carl`.



(a) *pumadyn-32nm* test error (b) *kin-40k* test error (c) *kin-40k* $\boldsymbol{\sigma}_i$ deviation

Figure 2. Plots (a) and (b) depict the test error as a function of basis size m. In (c) we plot against m the deviation of the $\boldsymbol{\sigma}_i$ from $\boldsymbol{\sigma}$, measured by the mean squared difference (see the text), for the *kin-40k* data set.

enough for *spgp-full* to infer a posterior similar to that of the full g.p. trained on the depicted $n = 200$ training points. The v.s.g.p. achieves a posterior closer to that of the full g.p. by employing — in comparison to the full g.p. — larger $\boldsymbol{\sigma}_i$'s and a smaller $\boldsymbol{\sigma}$. This leads to an effective covariance function — that of $\tilde{\mathcal{G}}_{\mathcal{U}}(k)$ as given by (17) — which better matches that of the full g.p. depicted in Figure 1 (c). In addition to merely observing the similarity between Figures 1 (b) and (c), we verified this last statement directly by visualising $\mathbb{E}_{\tilde{\mathcal{G}}_{\mathcal{U}}(k)}[f(\boldsymbol{x})f(\boldsymbol{z})]$ of (20) as a function of $\boldsymbol{x}$ and $\boldsymbol{z}$, but we omit the plot due to space limitations.

Figure 2 shows our experiments which, as in (Seeger et al., 2003) and (Snelson & Ghahramani, 2006), were performed on the *pumadyn-32nm* and *kin-40k* data sets.[7] Optimising the v.s.g.p. methods from a random initialisation tended to lead to inferior local optima, so we used the s.p.g.p. to find a starting point for the optimisation. This is possible because both methods optimise the same criteria, while the s.p.g.p. merely searches a subset of the space permitted by the v.s.g.p. framework. To ensure a fair comparison, we optimised the s.p.g.p. for 4000 iterations, whereas for the v.s.g.p. we optimised first the s.p.g.p. for 2000 iterations (*i.e.* fixing $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}, i = 1 \dots m$), took the result as a starting point, and optimised the v.s.g.p. for a further 2000 iterations (with the $\boldsymbol{\sigma}_i$ unconstrained).

We have also reproduced with kind permission the results of Seeger *et al.* (Seeger et al., 2003), and hence have used exactly the experimental methodology described therein. The results we reproduce are from the *info-gain* and *smo-bart* methods. *info-gain* is their own method which is extremely cheap to train for a given set of hyper parameters. The method uses greedy subset selection based on a criteria which can be evaluated efficiently. *smo-bart* is similar but is based on a criteria which is more expensive to compute (Smola & Bartlett, 2000). We also show the result of training a full g.p. on a subset of the data of size 2000 and 1024 for *kin-40k* and *pumadyn-32nm*, respectively.

Neither *info-gain* nor *smo-bart* estimate the hyper-parameters, but rather fix them to the values determined by optimising the marginal likelihood of the full g.p. Hence they are most directly comparable to *spgp-basis* and *vsgp-basis*. However, *spgp-full* and *vsgp-full* correspond to the more difficult task of estimating the hyper parameters at the same time as the basis.

For *pumadyn-32nm* we do not plot *spgp-basis* and *vsgp-basis* as the results are practically identical to *spgp-full* and *vsgp-full*. This differs from (Snelson & Ghahramani, 2006), where local minima problems with *spgp-full* on the *pumadyn-32nm* data set are reported. It is unclear why our experiments did not suffer in this way — possible explanations are the choice of initial starting point, as well as the choice of optimisation algorithm. The results of the s.p.g.p. and v.s.g.p. methods on the *pumadyn-32nm* data set very similar, but both outperform the *info-gain* and *smo-bart* approaches.

The *kin-40k* results are rather different. While the $\boldsymbol{\sigma}_i$ deviated little from $\boldsymbol{\sigma}$ on the *pumadyn-32nm* data set, this was not the case for *kin-40k*, particularly for small m, as seen in Figure 2 (c) where we plot $\frac{1}{md}\sum_{i=1}^{m}\sum_{j=1}^{d}([\boldsymbol{\sigma}_i - \boldsymbol{\sigma}]_j)^2$. Our results are in agreement with those of (Snelson & Ghahramani, 2006) — our *vsgp-full* outperforms *spgp-full* for small m, which in turn outperforms both *info-gain* and *smo-bart*. However for large m both *spgp-full* and *vsgp-full*

[7] *kin-40k*: 10000 training, 30000 test, 9 attributes, see www.igi.tugraz.at/aschwaig/data.html.
pumadyn-32nm: 7168 training, 1024 test, 33 attributes, see www.cs.toronto/delve.

tend to over-fit. This is to be expected due to the use of marginal likelihood optimisation, as the choice of basis $\mathcal{U}$ is equivalent to the choice of the order of md hyper parameters for the covariance function of $\tilde{\mathcal{G}}_{\mathcal{U}}(k)$. Happily, and somewhat surprisingly, the *vsgp-full* method tends not to over-fit more than the *spgp-full*, in spite of its having roughly twice as many basis parameters. Neither *vsgp-basis* nor *spgp-basis* suffered from over-fitting however, and while they both outperform *info-gain* and *smo-bart*, our *vsgp-basis* clearly demonstrates the advantage of our new s.p.g.p. framework by consistently outperforming *spgp-basis*.

Finally, to emphasise the applicability of our idea to other kernel algorithms, we provide an accompanying video which visualises the optimisation of an s.v.m. using multiscale gaussian basis functions.

6. Conclusions

Sparse g.p. regression is an important topic which has received a lot of attention in recent years. Previous methods have based their computations on subsets of the data or pseudo input points. To relate this to our method, this is analogous to basing the computations on a set of basis functions of the form $k(\boldsymbol{v}_i, \cdot)$ where k is the covariance function and the $\boldsymbol{v}_i$ are for example the pseudo input points. We have generalised this for the case of Gaussian covariance function, by basing our computations on a set of Gaussian basis functions whose bandwidth parameters may vary independently.

This provides a new avenue for approximations, applicable to all kernel based algorithms, including g.p.'s and the s.v.m., for example. To demonstrate the utility of this new degree of freedom, we have constructed sparse g.p. and k.r.r. algorithms which outperform previous methods, particularly for very sparse solutions. As such, our approach yields state of the art performance as a function of prediction time.

References

Csató, L., & Opper, M. (2002). Sparse on-line gaussian processes. *Neural Comp.*, *14*, 641–668.

Gehler, P., & Franz, M. (2006). *Implicit wiener series, part ii: Regularised estimation* (Technical Report 148). Max Planck Institute for Biological Cybernetics.

Lawrence, N., Seeger, M., & Herbrich, R. (2002). Fast sparse gaussian process methods: The informative vector machine. *Advances in Neural Information Processing Systems 15* (pp. 609–616).

Quiñonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, *6*, 1935–1959.

Roach, G. F. (1970). *Green's functions*. Cambridge, UK: Cambridge University Press.

Seeger, M., Williams, C., & Lawrence, N. D. (2003). Fast forward selection to speed up sparse gaussian process regression. In C. M. Bishop and B. J. Frey (Eds.), *Workshop on ai and statistics 9*. Society for Artificial Intelligence and Statistics.

Smola, A. J., & Bartlett, P. L. (2000). Sparse greedy gaussian process regression. In T. K. Leen, T. G. Dietterich and V. Tresp (Eds.), *Advances in neural information processing systems 13*, 619–625. Cambridge, MA: MIT Press.

Snelson, E., & Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf and J. Platt (Eds.), *Advances in neural information processing systems 18*, 1257–1264. Cambridge, MA: MIT Press.

Steinwart, I. (2002). On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, *2*, 67–93.

Walder, C., Schölkopf, B., & Chapelle, O. (2006). Implicit surface modelling with a globally regularised basis of compact support. *Proc. EUROGRAPHICS*, *25*, 635–644.