Regression with the Optimised Combination Technique

Jochen Garcke JOCHEN.GARCKE@ANU.EDU.AU
Centre for Mathematics and its Applications, Mathematical Sciences Institute
Australian National University, Canberra ACT 0200, Australia

Abstract

We consider the sparse grid combination technique for regression, which we regard as a problem of function reconstruction in some given function space. We use a regularised least squares approach, discretised by sparse grids and solved using the so-called combination technique, where a certain sequence of conventional grids is employed. The sparse grid solution is then obtained by addition of the partial solutions with combination coefficients dependent on the involved grids. This approach shows instabilities in certain situations and is not guaranteed to converge with higher discretisation levels. In this article we apply the recently introduced optimised combination technique, which repairs these instabilities. Now the combination coefficients also depend on the function to be reconstructed, resulting in a non-linear approximation method which achieves very competitive results. We show that the computational complexity of the improved method still scales only linear in regard to the number of data.

1. Introduction

In this paper we consider the regression problem arising in machine learning. A set of data points $\underline{x}_i$ in a d-dimensional feature space is given, together with an associated value y_i. We assume that a function f_* describes the relation between the predictor variables $\underline{x}$ and the response variable y and want to (approximately) reconstruct the function f_* from the given data. This allows us to predict the function value of any newly given data point for future decision-making.

In (Garcke et al., 2001) a discretisation approach to the regularisation network ansatz (Wahba, 1990; Girosi et al., 1995) was introduced. In contrast to other methods which employ mostly global ansatz functions associated with data points to describe the function f_*, here an independent grid with associated local basis functions is used to discretise the regularised minimisation problem. This way the data information is transferred into the discrete function space defined by the grid and its corresponding basis functions. Such a discretisation approach is similar to the numerical treatment of partial differential equations by finite element methods, see e.g. (Braess, 2001). Let $h_n := 2^{-n}$ now be the distance between two grid points in each dimension, i.e. the mesh size of a discretisation of level n. A uniform grid would result in $\mathcal{O}(h_n^{-d})$ grid points. Therefore the complexity of such an approach would grow exponentially with the dimension d and one encounters the curse of dimensionality.

However, so-called sparse grids allow us to cope with the complexity of grid-based discretisation methods to some extent. This method has been originally developed for the numerical solution of partial differential equations (Zenger, 1991; Griebel, 1991) and is now also used successfully for integral equations, interpolation and approximation, eigenvalue problems, and integration problems, see (Bungartz & Griebel, 2004; Garcke, 2005) for detailed references. The underlying concept of a sparse tensor product decomposition has a long tradition in approximation and goes back to the Russian literature (Babenko, 1960; Smolyak, 1963). For a d-dimensional problem, the sparse grid approach employs only $\mathcal{O}(h_n^{-1}(\log(h_n^{-1}))^{d-1})$ grid points in a discretisation of level n. The accuracy of the approximation however is nearly as good as for conventional grid methods, provided that certain additional smoothness requirements are fulfilled. The curse of dimensionality for conventional 'full' grid methods affects sparse grids much less; currently up to around 20 dimensions can be handled.

As in (Garcke et al., 2001) we apply the sparse grid combination technique (Griebel et al., 1992) to the regression problem. Here, the regularisation network problem is discretised and solved on a certain sequence of anisotropic grids, i.e. grids with different mesh sizes

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

in each coordinate direction. The sparse grid solution is then obtained from the (partial) solutions on these different grids by their linear combination using combination coefficients which depend on the employed grids. Thus the regression function is built on sparse grid points and not on data points.

Following empirical results in (Garcke, 2004), which show instabilities of the combination technique in certain situations, we apply here the optimised combination technique introduced in (Hegland, 2003) to the regression problem for the first time. The combination coefficients now not only depend on the grids involved, but on the function to be reconstructed as well, resulting in a non-linear approximation approach. A comparison of experimental results for a number of benchmark data sets with results from (Meyer et al., 2003) suggest that the optimised combination technique is very competitive.

A discussion of the complexity of the method shows that the method scales linearly with the number of instances, i.e. the amount of data to be treated. Therefore, the approach is well suited for applications where the dimension of the feature space is moderately high (e.g. after some preprocessing steps) but the amount of data is very large.

In the following we first describe the discretisation approach to the regularised regression problem, then present the sparse grid combination technique in section 3 followed by the introduction of the optimised combination technique. After experiments in section 5 we conclude with remarks on an extension of our approach.

2. Discretisation of the Regularised Regression Problem

We interpret the regression problem as a scattered data approximation problem in a possibly high-dimensional space. Given is a data set

$$S = \{(\underline{x}_i, y_i)\}_{i=1}^m \quad \underline{x}_i \in \mathbb{R}^d,\ y_i \in \mathbb{R},$$

where we denote with $\underline{\cdot}$ a d-dimensional vector or index with entries $\cdot_1, \ldots, \cdot_d$. We assume that the data has been obtained by sampling an unknown function f_* which belongs to some space V of functions defined over $\mathbb{R}^d$. The aim is to recover the function f_* from the given data as good as possible. To achieve a well-posed (and uniquely solvable) problem Tikhonov-regularisation theory (Tikhonov & Arsenin, 1977; Wahba, 1990) imposes a smoothness constraint on the solution. This leads to the variational problem

$$\min_{f \in V} R(f)$$

with

$$R(f) = \frac{1}{m} \sum_{i=1}^m (f(\underline{x}_i) - y_i)^2 + \lambda ||\mathcal{S} f||_2^2, \qquad (1)$$

where $y_i = f_*(\underline{x}_i)$. Here, $\mathcal{S}$ is a linear operator. The first term in (1) measures the error and therefore enforces closeness of f to the labelled data, the second term $||\mathcal{S}f||_2^2$ enforces smoothness of f, and the regularisation parameter λ balances these two terms. This regularised least squares approach was introduced for machine learning in (Girosi et al., 1995) under the name "regularisation network" (possibly with other error terms instead of the least square error). Note that the corresponding Galerkin equations to (1) are

$$\frac{1}{m} \sum_{i=1}^m f(\underline{x}_i) g(\underline{x}_i) + \lambda \langle \mathcal{S} f, \mathcal{S} g \rangle_2 = \frac{1}{m} \sum_{i=1}^m g(\underline{x}_i) y_i, \quad (2)$$

which hold for the minimum $f \in V$ of $R(f)$ and all $g \in V$.

Let us define the following semi-definite bi-linear form

$$\langle f, g \rangle_{RLS} = \frac{1}{m} \sum_{i=1}^m f(\underline{x}_i) g(\underline{x}_i) + \lambda \langle \mathcal{S} f, \mathcal{S} g \rangle_2 \qquad (3)$$

and choose V so that $\langle \cdot, \cdot \rangle_{RLS}$ is a scalar product on it. With respect to this scalar product the minimisation (1) is an orthogonal projection of f_* into V (Hegland, 2003), i.e. if $\|f - f_*\|_{RLS}^2 \leq \|g - f_*\|_{RLS}^2$ than $R(f) \leq R(g)$.

In the following we restrict the problem explicitly to a finite dimensional subspace $V_N \subset V$ with an appropriate basis $\{\varphi_j\}_{j=1}^N$. A function $f \in V$ is then approximated by

$$f_N(\underline{x}) = \sum_{j=1}^N \alpha_j \varphi_j(\underline{x}). \qquad (4)$$

Note that any discretisation involves additional regularisation by projection (Natterer, 1977) and that there is an interplay between regularisation by projection and Tikhonov-regularisation, see e.g. (Binder et al., 2002).

Such an explicit restriction to a discrete space is fundamentally different from kernel approaches. There, a finite representation of the solution in the infinite dimensional function space induced by the smoothing operator $\mathcal{S}$ is given via the representer theorem as a sum over kernel functions associated with the data points (Wahba, 1990). Thus, kernel based methods can be regarded as working in the data space.

We now plug the representation (4) of a function $f \in V_N$ into (2) and since (2) has to be valid for every basis function $\varphi_j(\cdot)$ directly obtain the linear equation system

$$(\mathcal{B}^\top \mathcal{B} + \lambda M \cdot \mathcal{C})\alpha = \mathcal{B}^\top y \tag{5}$$

and therefore are able to compute the unknown vector α for the solution f_N of (1) in V_N. $\mathcal{C}$ is a symmetric $N \times N$ matrix with entries $\mathcal{C}_{j,k} = \langle \mathcal{S}\varphi_j, \mathcal{S}\varphi_k \rangle_2$, $j,k = 1,\ldots,N$ and corresponds to the smoothness operator. $\mathcal{B}^\top$ is a rectangular $M \times N$ matrix with entries $(\mathcal{B}^\top)_{j,k} = \varphi_j(\underline{x}_k)$, $j = 1,\ldots,N$, $k = 1,\ldots,M$ and transfers the information from the data into the discrete space, $\mathcal{B}$ correspondingly works in the opposite direction. The vector y contains the data labels y_i and has length M.

In the following we also use $\mathcal{G} := \mathcal{B}^\top \cdot \mathcal{B} + \lambda M \cdot \mathcal{C}$ to denote the matrix sum. We now can write the scalar product (3) as $\langle f, g \rangle_{\mathcal{G}} := \langle f, \mathcal{G}g \rangle_2$. Using the corresponding operator matrix one can directly write other variational problems in this form as projections as well.

3. Sparse Grid Combination Technique

For the discretisation of the function space V we use sparse grids (Zenger, 1991; Griebel, 1991; Bungartz & Griebel, 2004; Garcke, 2005), which are based on a hierarchical subspace splitting and a sparse tensor product decomposition. To approximate functions $f \in V$ we apply this approach, as in (Garcke et al., 2001), in the form of the combination technique (Griebel et al., 1992). We discretise and solve the problem (1) on a suitable sequence of small anisotropic grids $\Omega_{\underline{l}} = \Omega_{l_1,\ldots,l_d}$, i.e. grids which have different but uniform mesh sizes in each coordinate direction with $h_t = 2^{-l_t}, t = 1,\ldots,d$. The grid points are numbered using the multi-index $\underline{j}, j_t = 0,\ldots,2^{l_t}$.. For ease of presentation we assume the domain $[0,1]^d$ here and in the following, which can be achieved by a proper rescaling of the data. Note that from here on we employ the gradient as a regularisation operator, i.e. $\mathcal{S} = \nabla$.

A finite element approach with piecewise d-linear functions

$$\phi_{\underline{l},\underline{j}}(\underline{x}) \;:=\; \prod_{t=1}^{d} \phi_{l_t,j_t}(x_t), \quad j_t = 0,\ldots,2^{l_t} \tag{6}$$

on each grid $\Omega_{\underline{l}}$, where the one-dimensional basis functions $\phi_{l,j}(x)$ are the so-called hat functions

$$\phi_{l,j}(x) = \begin{cases} 1 - |\frac{x}{h_l} - j|, & x \in [(j-1)h_l, (j+1)h_l] \\ 0, & \text{otherwise,} \end{cases}$$

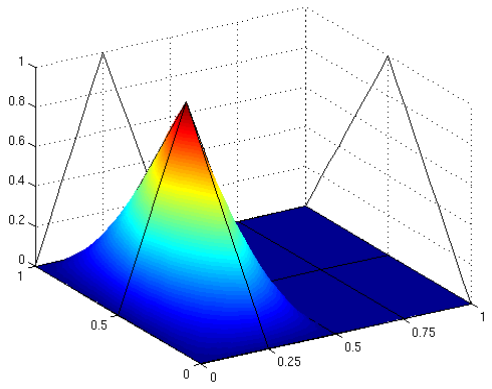

Figure 1. Basis function $\phi_{1,1}$ on grid $\Omega_{2,1}$.

now results in the discrete function space

$$V_{\underline{l}} \;:=\; \text{span}\{\phi_{\underline{l},\underline{j}}, j_t = 0,...,2^{l_t}, t = 1,...,d\} \tag{7}$$

on grid $\Omega_{\underline{l}}$. A function $f_{\underline{l}} \in V_{\underline{l}}$ is represented as

$$f_{\underline{l}}(\underline{x}) = \sum_{j_1=0}^{2^{l_1}} ... \sum_{j_d=0}^{2^{l_d}} \alpha_{\underline{l},\underline{j}} \phi_{\underline{l},\underline{j}}(\underline{x}).$$

Each d-linear function $\phi_{\underline{l},\underline{j}}(\underline{x})$ is one at the grid point $\underline{j}$ and zero at all other points of grid $\Omega_{\underline{l}}$. Its support, i.e. domain where the function is non-zero, is $\otimes_{t=1}^{d}[(j_t - 1)h_{l_t}, (j_t + 1)h_{l_t}]$. See Figure 1 for the basis function at position $1,1$ of the grid $\Omega_2, 1$.

The variational approach (2) now results in the discrete system

$$\left(\mathcal{B}_{\underline{l}}^\top \mathcal{B}_{\underline{l}} + \lambda_A M \cdot \mathcal{C}_{\underline{l}}\right) \alpha_{\underline{l}} = \mathcal{B}_{\underline{l}}^\top y. \tag{8}$$

Note that the matrices on the left hand side can be stored in one $N \times N$ matrix, where $N = \Pi_{t=1}^{d}(2^{l_t} + 1)$, and on the right hand side the evaluation of the matrix $B^\top$ is only needed once for setup. We currently solve these linear equation systems with a diagonally preconditioned conjugate gradient algorithm, see e.g. (Braess, 2001).

For the combination technique we now in particular consider all grids $\Omega_{\underline{l}}$ with

$$|\underline{l}|_1 := l_1 + ... + l_d = n - q, \quad q = 0,..,d-1, \quad l_t \geq 0,$$

set up and solve the associated problems (8). The number of grid points for each these grids is of order $\mathcal{O}(h_n^{-1})$. The original combination technique (Griebel et al., 1992) now linearly combines the resulting dis-

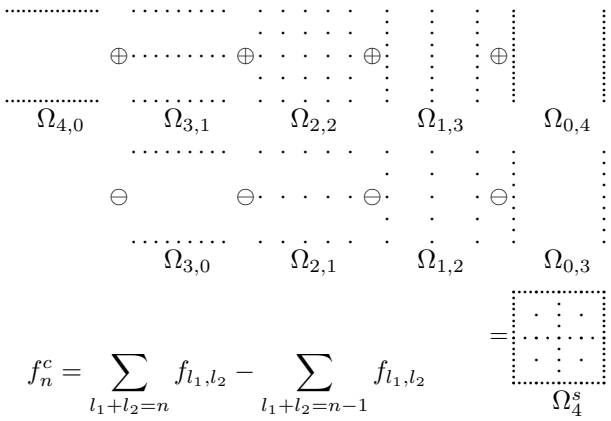

Figure 2. Grids involved for the combination technique of level $n = 4$ in two dimensions.

crete solutions $f_{\underline{l}}(\underline{x})$ from the partial grids $\Omega_{\underline{l}}$ according to the formula

$$f_n^c(\underline{x}) := \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{|\underline{l}|_1 = n-q} f_{\underline{l}}(\underline{x}).$$

Note the varying sign of the combination coefficients, which 'offsets' the fact that some sparse grid points occur several times within the combination technique.

For the two-dimensional case, we display the grids needed in the combination formula of level 4 in Figure 2 and give the resulting sparse grid.

The function f_n^c lives in the sparse grid space

$$V_n^s := \bigoplus_{\substack{|\underline{l}|_1 = n-q \\ q=0,...,d-1 \quad l_t \geq 0}} V_{\underline{l}},$$

with $V_{\underline{l}}$ as in (7). The space V_n^s has dimension of order $\mathcal{O}(h_n^{-1}(\log(h_n^{-1}))^{d-1})$ in contrast to $\mathcal{O}(h_n^d)$ for conventional grid based approaches. It is alternatively spanned by a piecewise d-linear hierarchical tensor product basis. For the approximation of f by a sparse grid function $f_n^s \in V_n^s$ the error relation

$$||f - f_n^s||_{L_p} = \mathcal{O}(h_n^2 \log(h_n^{-1})^{d-1})$$

holds, provided that f fulfils certain smoothness requirements (Bungartz & Griebel, 2004).

Note that the combination technique is only one of the various methods to solve problems on sparse grids. Finite difference and Galerkin finite element approaches which work directly in the hierarchical product basis on the sparse grid also exist, see (Bungartz & Griebel, 2004) for detailed references. But the combination technique is conceptually much simpler and easier to implement. Moreover, it allows the reuse of standard solvers for the anisotropic subgrids $\Omega_{\underline{l}}$.

4. Optimised Combination Technique

The combination technique is an exact projection into the sparse grid space only if the partial projections commute, i.e. the commutator $[P_{V_1}, P_{V_2}] := P_{V_1}P_{V_2} - P_{V_2}P_{V_1}$ is zero for all pairs of involved grids (Hegland et al., 2005). This is the case for interpolation problems (Griebel et al., 1992). Although the commuting property does not hold for the numerical solution of partial differential equations, it was shown that the combination technique has the same approximation order as ordinary sparse grids, as long as a certain error expansion for the partial solutions holds (Griebel et al., 1992).

Recently it was observed empirically (Garcke, 2004) that for regularised regression the combination technique is instable and can actually diverge. To illustrate this we show in Figure 3 the residual and the least square error for a simple problem in two dimensions, note that after level $n = 3$ both error measurements increase on the training data, which cannot happen with a true variational discretisation ansatz. This effect is especially observed for small λ, already with $\lambda = 10^{-4}$ the now stronger influence of the smoothing term results in nearly commuting projectors and a (more) stable approximation method. Note that the instability is more common and significant in higher dimensions.

This observation was recently explained using the context of projections and relating commuting projections to angles between spaces, i.e. two spaces are orthogonal if the projections commute (Hegland et al., 2005).

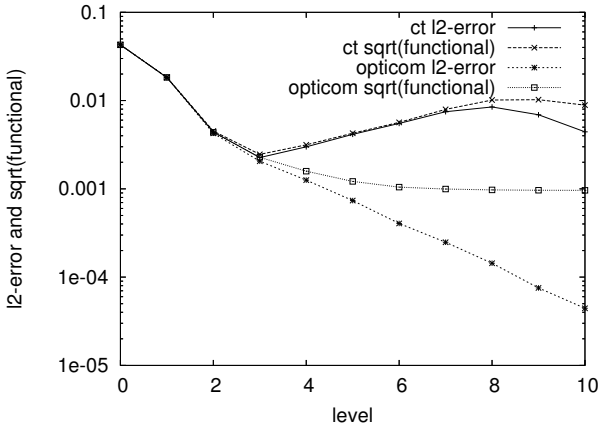

Figure 3. Value of the functional (1) and the least squares error on the data, i.e. $\frac{1}{M}\sum_{i=1}^{M}(f(\underline{x}_i) - y_i)^2$, for the reconstruction of $e^{-x^2} + e^{-y^2}$ from 5000 training data using the combination technique and the optimised combination technique with $\lambda = 10^{-6}$ (right) and level $n = 0, \ldots, 10$.

In the case of regularised regression, which is a projection according to the scalar product (3), strong derivations from the orthogonal angle of $\pi/2$ of up to $\pi/4$ for certain constructed examples could be observed.

In (Hegland, 2003) a modification of the combination technique is introduced where the combination coefficients not only depend on the spaces as before, which gives a linear approximation method, but instead depend on the function to be reconstructed as well, resulting in a non-linear approximation approach. In (Hegland et al., 2005) this ansatz is presented in more detail and the name 'opticom' for this optimised combination technique is suggested. In this paper we apply this approach to the regularised regression problem for the first time in detail.

To compute the optimal combination coefficients c_i one minimises the functional

$$J(c_1, \ldots, c_m) = \|Pf - \sum_{i=1}^{m} c_i P_i f\|_{\mathcal{G}}^2,$$

where one uses the scalar product corresponding to the variational problem $\langle \cdot, \cdot \rangle_{\mathcal{G}}$, defined on V to generate a norm. For ease of presentation we assume a suitable numbering of the involved spaces. By simple expansion one gets

$$\begin{aligned} J(c_1, \ldots, c_m) &= \sum_{i,j=1}^{m} c_i c_j \langle P_i f, P_j f \rangle_{\mathcal{G}} \\ &\quad -2 \sum_{i=1}^{m} c_i \|P_i f\|_{\mathcal{G}}^2 + \|Pf\|_{\mathcal{G}}^2. \end{aligned}$$

While this functional depends on the unknown quantity Pf, the location of the minimum of J does not. By differentiating with respect to the combination coefficients c_i and setting each of these derivatives to zero we see that minimising this norm corresponds to finding c_i which have to satisfy

$$\begin{bmatrix} \|P_1 f\|_{\mathcal{G}}^2 & \cdots & \langle P_1 f, P_m f \rangle_{\mathcal{G}} \\ \langle P_2 f, P_1 f \rangle_{\mathcal{G}} & \cdots & \langle P_2 f, P_m f \rangle_{\mathcal{G}} \\ \vdots & \ddots & \vdots \\ \langle P_m f, P_1 f \rangle_{\mathcal{G}} & \cdots & \|P_m f\|_{\mathcal{G}}^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} \|P_1 f\|_{\mathcal{G}}^2 \\ \|P_2 f\|_{\mathcal{G}}^2 \\ \vdots \\ \|P_m f\|_{\mathcal{G}}^2 \end{bmatrix}$$

The solution of this small system creates little overhead. However, in general an increase in computational complexity is due to the need for the determination of the scalar products $\langle P_i f, P_j f \rangle_{\mathcal{G}}$. Their computation is often difficult as it requires an embedding into a bigger discrete space which contains both V_i and V_j.

To compute the scalar product $\langle P_{\underline{i}} f, P_{\underline{j}} f \rangle_{\mathcal{G}}$ of the two projections into the discrete spaces $V_{\underline{i}}$ and $V_{\underline{j}}$ in our case, the operator matrix $\mathcal{G}$ defining the scalar product $\langle f, \mathcal{G} g \rangle$ has to be computed in the joint space $V_{\underline{k}}$, with $k_t = \max(i_t, j_t)$, into which the partial solutions $P_{\underline{l}} f = f_{\underline{l}}, \underline{l} = \underline{i}, \underline{j}$ have to be interpolated. One observes that $V_{\underline{k}}$ is of size $\mathcal{O}(h_n^{-2})$ in the worst case, as opposed to $\mathcal{O}(h_n^{-1})$ for the $V_{\underline{l}}, \underline{l} = \underline{i}, \underline{j}$. Remember that in our setting the scalar product is defined as

$$\langle f, g \rangle_{RLS} = \frac{1}{m} \sum_{i=1}^{m} f(\underline{x}_i) g(\underline{x}_i) + \lambda \langle \nabla f, \nabla g \rangle_2$$

and the computation of the scalar product can be split according to these two terms. For the first data dependent part the projections $P_{\underline{l}} f$ are evaluated at all data points $\underline{x}_i$ and for a pair of grids the sum over the product of these function values is computed. The second term is computed in the joint space, but since we are using $\mathcal{S} = \nabla$ this can be achieved efficiently on-the-fly, one does not need to explicitly build the matrix $\mathcal{C}$. But a run-time complexity of $\mathcal{O}(h_n^{-2})$ still arises for the latter term. This can be reduced to $\mathcal{O}(h_n^{-1} (\log h_n^{-1})^{d-1})$ following the approach in (Griebel, 1991) which exploits the structural zeros of $\langle \nabla f, \nabla g \rangle_2$ in this setting.

Using these optimal coefficients c_i the combination formula is now

$$f_n^c(\underline{x}) := \sum_{q=0}^{d-1} \sum_{|\underline{l}|_1 = n-q} c_{\underline{l}} f_{\underline{l}}(\underline{x}). \tag{9}$$

Note that we never explicitly assemble the function f_n^c but instead keep the solutions $f_{\underline{l}}$ which arise in the combination formula. Therefore, if we now want to evaluate a newly given set of data points $\{\tilde{\underline{x}}_i\}_{i=1}^{m_n}$ by

$$\tilde{y}_i := f_n^c(\tilde{\underline{x}}_i), \quad i = 1, ..., m_n$$

we just form the combination of the associated values for $f_{\underline{l}}$ according to (9).

In Figure 3 we also show the residual and the least square error for the optimised combination technique and see that the least square error now steadily decreases whereas the residual saturates after some discretisation level.

4.1. Computational Complexity

Let us conclude this section with a discussion of the complexity of our approach. The number of grids we have to consider during the computation of the sparse grid solution is of order $\mathcal{O}(d \cdot \log(h_n^{-1})^{d-1})$ and their size is $\dim(V_{\underline{l}}) = \mathcal{O}(2^{d-1} \cdot h_n^{-1}) = \mathcal{O}(2^{d-1} \cdot 2^n)$. All these problems can be solved independently, which allows for a straightforward parallelisation, for details

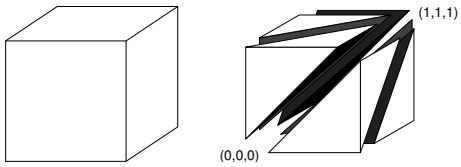

Figure 4. A simplicial discretisation divides each rectangular block formed by grid points into d! simplices.

see (Garcke et al., 2006). Note that the term 2^{d-1} in the above order complexity of $\dim(V_{\underline{l}})$ limits our approach in the number of predictor variables.

Looking at the discrete system (8) we see that only the matrix $\mathcal{B}_{\underline{l}}^\top \mathcal{B}_{\underline{l}}$ with entries $(\mathcal{B}_{\underline{l}}^\top \mathcal{B}_{\underline{l}})_{j,k} = \sum_i^M \varphi_j(\underline{x}_i) \cdot \varphi_k(\underline{x}_i)$ and the right hand side depend on the data. For each data point we have to evaluate all basis functions which are non-zero at this point, or in other words, all basis functions in whose support the data point is situated.

The original derivation of the combination technique is based on d-linear basis functions (6) stemming from a tensor product approach. Each data point is in exactly one finite element cube and this way 2^d basis functions, associated with the nodes of the finite element cube, are non-zero for each data point. To avoid this computational complexity, which is exponential in d, (Garcke & Griebel, 2002) instead employ for each partial grid a simplicial discretisation: now a basis function is one at its corresponding vertex, zero at all others, and linear on each simplex. Figure 4 shows how a finite element cube in three dimensions is divided by $d!$ simplices using the so-called Freudenthal-Kuhn triangulation. With this approach only $d+1$ basis functions have to be evaluated for each data instance, they are associated with the vertices of the simplex in which the data point is situated. This reduces the number of operations needed for the processing of one data point during the computation of the entries of $\mathcal{B}_{\underline{l}}^\top \mathcal{B}_{\underline{l}}$ in (8) from costs that are exponential in d to costs that are only quadratic in d.

But note that the theoretical approximation properties of this variant of the sparse grid combination technique still have to be investigated in more detail. In particular the involved discrete spaces are not nested, and furthermore the assumed error expansion for the approximation properties in the non-commuting case does not hold. However the numerical results warrant its use.

5. Results

We now compare our method with results of the extensive benchmark study (Meyer et al., 2003). We present results for six real-life regression examples; the three others in this study were higher-dimensional (or had categorical attributes) and we abstained from dimension reduction for these experiments. Note that we linearly scale the data into the domain $[0,1]^d$.

We fit the parameters λ and level n on a subset of the data. As in (Meyer et al., 2003) we repeat a ten-fold cross-validation ten times, (i.e. ten partitions with non-overlapping tests sets from ten permutations). In Table 1 we give the mean of the least squares errors on the test data and its standard deviation, the latter computed over the means of the cross validation results, for both the normal combination technique (ct) and the optimised combination technique (opticom). As predicted, the optimised combination technique always gives better results than the normal one, in some cases these are significantly better.

For comparison we give the best result from the benchmark study and note the rank of the optimised combination technique in comparison to the other methods used, these are linear regression, ϵ-support vector regression with a Gaussian-RBF-kernel (svm), neural networks (nnet), regression trees, projection pursuit regression (ppr), multivariate adaptive regression splines, additive spline models by adaptive backfitting, bagging of trees, random forest (rForest), and multivariate adaptive regression trees (mart). Since no timing measurements are given in the benchmark study, one can only compare the quality of the results and not the underlying computational complexity.

These are all small data sets, the largest one has just 8192 examples. One might expect that methods whose complexities scale non-linearly in the number of data would dominate these experiments. Maybe somewhat surprisingly, for four of these data sets the optimised combination technique gives the best result, in two cases even the normal combination technique would suffice to improve on the results of the other methods.

To measure the computational time for larger amounts of data we use the synthetic data from (Friedman, 1991). These are the sets Friedman1 with $y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + e$ where e is the normal distribution $N(0,1)$ and all ten variables, including five as noise, are in $[0,1]^d$. Friedman2 and Friedman3 have data in $0 \le x_1 \le 100$, $40\pi \le x_2 \le 560\pi$, $0 \le x_3 \le 1$, and $1 \le x_4 \le 11$. The outputs for Friedman2 are created according to the formula $y = (x_1^2 + (x_2 x_3 - \frac{1}{x_2 x_4})^2)^{0.5} + e$ where e is $N(0, 125)$ and

Table 1. Results in comparison to the benchmark study (Meyer et al., 2003). We give the mean squared test set errors for the combination technique (ct), the optimised combination technique (opticom) and the results for the best other method. For the opticom approach we also give the used discretisation level.

	ct	opticom		best other		opticom	# algor.
	mean(MSE)	level	mean(MSE)	mean(MSE)		rank	
abalone	4.33 (0.32)	1	4.20 (0.32)	nnet	4.31	1	9
auto-mpg	6.92 (0.34)	2	6.21 (0.46)	svm	7.11	1	9
boston-housing	12.23 (1.31)	1	8.92 (0.72)	svm	9.60	1	9
cpu ($\times 10^3$)	2.30 (0.55)	1	1.73 (0.19)	ppr	3.16	1	9
cpuSmall	56.14 (1.98)	2	8.74 (0.12)	mart	7.55	3	10
SLID	39.67 (0.85)	3	38.64 (0.72)	rForest	34.13	4	9

Table 2. Results for the synthetic Friedman data sets using the optimised combination technique in comparison with svm and mars. The timings are given in seconds.

	opticom			SVM		MARS	
	level	MSE	time	MSE	time	MSE	time
Friedman1	3	1.340	2872	1.148	23604	1.205	10.4
Friedman2 ($\times 10^3$)	3	15.46	35	15.40	3151	15.77	16.9
Friedman3 ($\times 10^{-3}$)	4	13.33	89	27.47	16862	14.45	3.6

for Friedman3 one has $y = \text{atan}((x_2x_3 - \frac{1}{x_2x_4})/x_1) + e$ where e is $N(0, 0.1)$.

We generate 100.000 data for training and another 10.000 for testing, where the data positions are uniformly distributed over their domains. For the optimised combination technique we use a 2:1 split of the data for the parameter fitting of λ and n. We compare with ϵ-support vector regression as a state-of-art method from libsvm[1] using a Gaussian-RBF-kernel, here we perform a grid search over σ and C on a small subset of the training data to find good parameters. As a simple and fast baseline method we use multivariate adaptive regression splines from the R package[2] with the highest degree of interaction useful.

The results are given in Table 2. Note that the machine used was an Intel Pentium 4 (2.4GHz) with about 500 MB of available memory, all of which could be used for kernel caching in the case of the svm. As expected, the fastest method in all cases is MARS and only for the lower dimensional case is the optimised combination technique comparable in time. The opticom method gives between 2% and 7% improvement for the lower dimensional cases and is worse for the ten-dimensional examples in regard to MARS.

As expected the non-linear complexity in respect to the number of data of the svm-approach results in long computational times. With only a fraction of the computation time we achieve for the two lower dimension data sets either almost similar, with a difference of 0.4%, or by 50% significantly better results. Even in the ten-dimensional case our approach is almost an order of magnitude faster, but here the quality of the result is worse due to the influence of the five noise variables.

Note that the computational time for Friedman1 using the optimised combination technique would be significantly reduced if we employed dimension adaptive strategies as in (Garcke, 2004; Kahrs et al., 2005) to realise that five dimensions are just noise. Work in this regard is in progress. If we force our method to only use the five significant attributes we achieve a result of 1.040 in 953.2 seconds with level 5, which again gives the best result of the three methods tested.

6. Conclusions

We introduced the optimised combination technique to compute sparse grid approximations to regularised least square regression problems. We compared our results with a benchmark study and measured the lowest least square error in four out of six cases. Comparing on large synthetic data sets we again achieve good approximations with an order of magnitude less computational time than a SVM-approach. Although slower than MARS we achieve better results in all cases after some preprocessing of the data.

[1] C.-C. Chang and C.-J. Lin, LIBSVM, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[2] `http://www.R-project.org`

To reduce the computational time further we are investigating dimension-adaptive approaches (Garcke, 2004; Kahrs et al., 2005) which will reduce the number of partial grids needed in the combination technique. As indicated, such an approach can also improve the quality of the results. These ideas can be related to ANOVA-style decompositions. Finding the relevant attributes and their combinations will also be worthwhile to interpret results in applications such as the identification of discrete-time, nonlinear, autoregressive models with exogenous inputs (NARX models).

Acknowledgements

The author acknowledges the support of the Australian Research Council. I thank Michael Griebel and Markus Hegland for many stimulating discussions.

References

Babenko, K. I. (1960). Approximation of periodic functions of many variables by trigonometric polynomials. *Dokl. Akad. Nauk SSSR*, *132*, 247–250. Russian, Engl.: Soviet Math. Dokl. 1:513–516, 1960.

Binder, T., Blank, L., Dahmen, W., & Marquardt, W. (2002). On the regularization of dynamic data reconciliation problems. *J. Proc. Cont.*, *12*, 557–567.

Braess, D. (2001). *Finite elements*. Cambridge: Cambridge University Press. Second edition.

Bungartz, H.-J., & Griebel, M. (2004). Sparse grids. *Acta Numerica*, *13*, 147–269.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *Ann. Statist.*, *19*, 1–141.

Garcke, J. (2004). *Maschinelles Lernen durch Funktionsrekonstruktion mit verallgemeinerten dünnen Gittern*. Doktorarbeit, Institut für Numerische Simulation, Universität Bonn.

Garcke, J. (2005). Sparse grid tutorial. `http://wwwmaths.anu.edu.au/~garcke/paper/sparseGridTutorial.pdf`.

Garcke, J., & Griebel, M. (2002). Classification with sparse grids using simplicial basis functions. *Intelligent Data Analysis*, *6*, 483–502. (shortened version appeared in KDD 2001, Proc. of the Seventh ACM SIGKDD, F. Provost and R. Srikant (eds.), pages 87-96, ACM, 2001).

Garcke, J., Griebel, M., & Thess, M. (2001). Data mining with sparse grids. *Computing*, *67*, 225–253.

Garcke, J., Hegland, M., & Nielsen, O. (2006). Parallelisation of sparse grids for large scale data analysis. *ANZIAM Journal*, *48*. to appear.

Girosi, F., Jones, M., & Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, *7*, 219–265.

Griebel, M. (1991). A parallelizable and vectorizable multi-level algorithm on sparse grids. *Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar, Kiel, 1990* (pp. 94–100). Vieweg-Verlag.

Griebel, M., Schneider, M., & Zenger, C. (1992). A combination technique for the solution of sparse grid problems. *Iterative Methods in Linear Algebra* (pp. 263–281). IMACS, Elsevier, North Holland.

Hegland, M. (2003). Additive sparse grid fitting. *Proceedings of the Fifth International Conference on Curves and Surfaces, Saint-Malo, France 2002* (pp. 209–218). Nashboro Press.

Hegland, M., Garcke, J., & Challis, V. (2005). The combination technique and some generalisations. submitted, `http://wwwmaths.anu.edu.au/~garcke/paper/opticom.pdf`.

Kahrs, O., Brendel, M., & Marquardt, W. (2005). Incremental identification of NARX models by sparse grid approximation. *Proceedings of the 16th IFAC World Congress, 3.-8. Juli 2005, Prague*.

Meyer, D., Leisch, F., & Hornik, K. (2003). The support vector machine under test. *Neurocomputing*, *55*, 169–186.

Natterer, F. (1977). Regularisierung schlecht gestellter Probleme durch Projektionsverfahren. *Numer. Math.*, *28*, 329–341.

Smolyak, S. A. (1963). Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, *148*, 1042–1043. Russian, Engl.: Soviet Math. Dokl. 4:240–243, 1963.

Tikhonov, A. N., & Arsenin, V. A. (1977). *Solutions of ill-posed problems*. Washington D.C.: W.H. Winston.

Wahba, G. (1990). *Spline models for observational data*, vol. 59 of *Series in Applied Mathematics*. Philadelphia: SIAM.

Zenger, C. (1991). Sparse grids. *Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar, Kiel, 1990* (pp. 241–251). Vieweg-Verlag.