A dimension adaptive sparse grid combination technique for machine learning

Jochen Garcke [*]

Abstract

We introduce a dimension adaptive sparse grid combination technique for the machine learning problems of classification and regression. A function over a d-dimensional space, which assumedly describes the relationship between the features and the response variable, is reconstructed using a linear combination of partial functions that possibly depend only on a subset of all features. The partial functions are adaptively chosen during the computational procedure. This approach (approximately) identifies the ANOVA-decomposition of the underlying problem. Experiments on synthetic data, where the structure is known, show the advantages of a dimension adaptive combination technique in run time behaviour, approximation errors, and interpretability.

Contents

[*] Centre for Mathematics and its Applications, Mathematical Sciences Institute, Australian National University, Canberra, AUSTRALIA and Technische Universität Berlin, Institut für Mathematik, Sekretariat MA 3-3, Straße des 17. Juni 136, 10623 Berlin, GERMANY. mailto:garcke@math.tu-berlin.de

1 Introduction

Sparse grids are the basis for efficient high dimensional function approximation. This approach is based on a multiscale tensor product basis where basis functions of small importance are omitted. In the form of the combination technique, sparse grids have successfully been applied to the machine learning problems of classification and regression using a regularisation network approach [3]. Here the problem is discretised and solved on a chosen sequence of anisotropic grids with uniform mesh sizes in each coordinate direction. The sparse grid solution is then obtained from the solutions on these different grids by linear combination.

Although sparse grids cope with the curse of dimensionality to some extent the approach still has high dependence on d, the number of dimensions. But typically the importance of and variance within a dimension vary in real machine learning applications which can exploited by different mesh resolutions for each feature. The degree of interaction between different dimensions also varies; this makes the usage of all dimensions in each partial grid unnecessary.

A large reduction in complexity in regard to d is obtained if one uses a hierarchy starting with a constant together with so-called generalised sparse grids to exploit the above observations. A dimension adaptive algorithm to construct a generalised sparse grid is necessary; one now chooses the grids used in the combination technique during the computation instead of defining them a priori. The aim is to attain function representations for $f(\underline{x})$, where $\underline{x}$ denotes the d-dimensional vector $(x_1, \ldots, x_d)$, of the ANOVA type

$$f(\underline{x}) = \sum_{\{j_1,\ldots,j_q\}\subset\{1,\ldots,d\}} c_{j_1,\ldots,j_q} f_{j_1,\ldots,j_q}(x_{j_1}, \ldots, x_{j_q}),$$

where each $f_{j_1,\ldots,j_q}(x_{j_1}, \ldots, x_{j_q})$ depends only on a subset of size q of the dimensions and may have different refinement levels for each dimension. One especially assumes here that $q < d$, so that the computational complexity depends only on the so-called *superposition* (or *effective*) dimension q. Such an approach was first introduced in a proof-of-concept way for the case of interpolation [6]; here only function evaluations are needed. This was adapted for numerical quadrature and data structures for the efficient handling of the index sets are available [4]. We extend this approach to the case of regularised least squares regression.

In the following we first describe the problem of regression and the approach of regularised least squares. We introduce the dimension adaptive combination technique for this problem and show results on machine learning benchmarks.

2 Regularised least squares regression

In the following we treat regression as a regularisation problem and use sparse grids to discretise the feature space. We consider a dataset of the form

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

and assume that the relation between these data can be described by an unknown function f which belongs to some space V of functions defined over $\mathbb{R}^d$. The aim is now to recover the function f from the given data as accurately as possible. To get a well-posed, uniquely solvable problem we use regularisation theory and impose additional smoothness constraints on the solution of the approximation problem. In our approach this results in the variational problem

$$f_V = \operatorname*{argmin}_{f \in V} R(f) = \operatorname*{argmin}_{f \in V} \frac{1}{M} \sum_{i=1}^{M} (f(\underline{x}_i) - y_i)^2 + \lambda ||\nabla f||^2, \tag{1}$$

where we use the squared error to enforce closeness of f to the data; the second term, called the regularisation term, enforces smoothness of f, and the regularisation parameter λ balances these two terms. Other error measurements or regularisation terms are also suitable.

We now restrict the problem to a finite dimensional subspace $V_N \subset V$. Using basis functions $\{\varphi_j\}_{j=1}^N$ of the function space V_N we represent f as

$$f = \sum_{j=1}^{N} \alpha_j \varphi_j(\underline{x}). \tag{2}$$

Note that the restriction to a suitably chosen finite-dimensional subspace involves additional regularisation (regularisation by discretisation) which depends on the choice of V_N.

After inserting (2) into (1) and differentiating with respect to the α_j we get the linear equation system [3]

$$(\lambda \mathcal{C} + \mathcal{B} \cdot \mathcal{B}^T)\alpha = \mathcal{B}y. \tag{3}$$

Here $\mathcal{C}$ has entries $\mathcal{C}_{j,k} = M \cdot (\nabla\varphi_j, \nabla\varphi_k)_{L_2}$, $j, k = 1, \ldots, N$, and $\mathcal{B}_{j,i} = \varphi_j(\underline{x}_i), i = 1, \ldots, M; j = 1, \ldots, N$. The vector y contains the data labels y_i.

3 Dimension adaptive combination technique

For the discretisation of the function space V we use a generalisation of the sparse grid combination technique [5]. We discretise and solve the problem (1), after rescaling the data to $[0, 1]^d$, on a suitable sequence of small anisotropic grids $\Omega_{\underline{l}} = \Omega_{l_1,\dots,l_d}$, that is grids which have different but uniform mesh sizes in each coordinate direction with $h_t = 2^{-l_t}, t = 1,\dots,d$. The grid points are numbered using the multi-index $\underline{j}, j_t = 0,\dots,2^{l_t}$.

A finite element approach with piecewise d-linear functions

$$\phi_{\underline{l},\underline{j}}(\underline{x}) \ := \ \prod_{t=1}^{d} \phi_{l_t,j_t}(x_t), \quad j_t = 0,\dots,2^{l_t} \tag{4}$$

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

$$\phi_{l,j}(x) = \begin{cases} 1 - |x/h_l - j|, & x \in [(j-1)h_l, (j+1)h_l] \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

now results in the discrete function space $V_{\underline{l}} := \text{span}\{\phi_{\underline{l},\underline{j}}, j_t = 0,\dots,2^{l_t}, t = 1,\dots,d\}$ on grid $\Omega_{\underline{l}}$. A function $f_{\underline{l}} \in V_{\underline{l}}$ is represented as

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

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 $\Omega_{\underline{l}}$.

In the original combination technique [5] one considers all grids $\Omega_{\underline{l}}$ with

$$|\underline{l}|_1 := l_1 + \dots + l_d = n - q, \quad q = 0,\dots,d-1, \quad l_t \geq 0 \tag{6}$$

and uses combination coefficients to add up the partial solutions $f_{\underline{l}}$ to get the solution f_n^c on the corresponding sparse grid in the following way

$$f_n^c = \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{l} \sum_{|\underline{l}|_1 = n-q} f_{\underline{l}}. \tag{7}$$

Hegland generalised the choice of grids used in the combination technique [6]. Instead of using grids which are below a hyperplane after (6) one considers a generalised index set I which fulfils the following *admissibility condition* [4]

$$\underline{k} \in \mathsf{I} \text{ and } \underline{j} \leq \underline{k} \quad \Rightarrow \quad \underline{j} \in \mathsf{I}. \tag{8}$$

In Figure 1 we show examples in two dimensions for such index sets, starting with the grids used by the original combination technique (6), then with some anisotropy, and finally with the extreme case where no real coupling between the dimensions exists. In higher dimensions such effects are

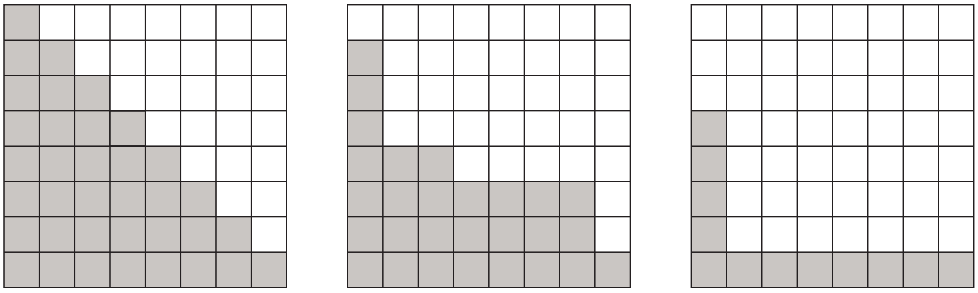

Figure 1: Index sets, where each cell corresponds to an index (i_1, i_2), for the original combination technique (6) and two generalised cases; the employed indices are in grey.

more common and easily achieved and allow a much greater flexibility in the choice of grids than observed in the simple two-dimensional case.

Most important is the choice of a suitable index set I. One might be able to use external knowledge of the properties and the interactions of the dimensions which would allow an a priori choice of the index set. In general the algorithm should choose the grids automatically in a *dimension adaptive* way during the actual computation. We therefore start with the smallest grid with index $\underline{0} = (0, \ldots, 0)$ (i.e., $\mathsf{I} = \{\underline{0}\}$). Step-by-step we add additional indices such that:

(i) the new index set remains admissible;

(ii) the partial result corresponding to the additional index provides a large contribution to the solution of the problem.

To check the admissibility of a new index it is necessary to consider the outer layer of the indices under consideration. We denote by A the set of *active indices* which consists of elements of I, whose forward neighbours have not been considered till now. The set O of *old indices* contains all other elements of I (i.e., $\mathsf{O} := \mathsf{I} \backslash \mathsf{A}$). An index can only be added to the active set A if all backward neighbours are in the old index set O. We denote here by the *backward neighbourhood* of an index $\underline{k}$ the set $\{\underline{k} - \underline{e}_t,\ 1 \leq t \leq d\}$; the set $\{\underline{k} + \underline{e}_t,\ 1 \leq t \leq d\}$ is called the *forward neighbourhood.*

In Figure 2 a few adaptation steps for the two dimensional case are presented. We assume there that, according to suitable error indicators, the indices (1,1), (2,1), (0,3) and (3,1) are chosen in succession. In each case their forward neighbours are considered: in the first step both are admissible and added to A; in the second and third only one each; and in the last step

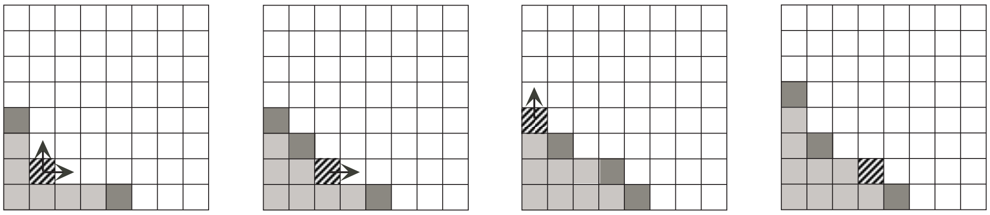

Figure 2: A few steps of the dimension adaptive algorithm. Active indices $\underline{i} \in \mathsf{A}$ are shown in dark grey and old indices $\underline{i} \in \mathsf{O}$ in light grey. The chosen active index (with the largest error indicator) is shown striped. The arrows indicate the admissible forward neighbours which are added to A.

no forward neighbour is admissible since their backward neighbours are not in O.

For the second point we need to measure the contribution of a grid to the overall solution. As an error indicator we compute for each newly added grid the reduction in the functional (1) in comparison to the current solution. Note that although the expression (1) has to be computed in the additive space of all partial grids, its value can still be computed by using just the partial grids [2]. For the data dependent part one computes for each partial function its value on a given data point and adds these using the combination coefficients. The smoothing term of the discrete Laplacian can be expressed as a weighted sum over expressions of the form $\langle \nabla f_{\underline{i}}, \nabla f_{\underline{j}} \rangle$ which can be computed on-the-fly via a grid which includes both $\Omega_{\underline{i}}$ and $\Omega_{\underline{j}}$.

We use a greedy approach for the dimension adaptive grid choice; the algorithm decides, depending on the error indicator (and possibly other values such as the complexity of the computation for a partial solution), which grid provides the highest benefit. This grid is added to the index set and its forward neighbourhood is searched for further candidates. This procedure is followed until a suitable global stopping criterion is reached; currently we stop when the reduction of the residual falls under a given threshold. There is hope that an efficient dimension adaptive algorithm builds an optimal index set in a sense similar to best N-term-approximation. It would be an interesting research topic to look into an underlying theory which could provide results over the quality of the error estimation and a suitable adaptive procedure including bounds in regard to the real error, as is common in the numerical treatment of partial differential equations by adaptive finite elements.

If the computation of the error indicator for $\underline{k} \in \mathsf{A}$ involves the partial solution of the corresponding grid one could directly use this result for the

Algorithm 1: The dimension adaptive algorithm

```
compute partial problem for index $\underline{0}$
$\mathsf{A} := \{\underline{0}\}$                                        ▷ active index set
$\mathsf{O} := \emptyset$                                              ▷ old index set
set $\varepsilon_{\underline{0}}$ to not fulfil global stopping criterion       ▷ for startup
while global stopping criterion not fulfilled do
    choose $\underline{i} \in \mathsf{A}$ with largest $\varepsilon_{\underline{i}}$▷ index with largest contribution
    $\mathsf{O} := \mathsf{O} \cup \{\underline{i}\}$
    $\mathsf{A} := \mathsf{A} \backslash \{\underline{i}\}$
    for $t = 1, \ldots, d$ do                      ▷ look at all neighbours of $\underline{i}$
        $\underline{j} := \underline{i} + \underline{e}_t$
        if $\underline{j} - \underline{e}_l \in \mathsf{O}$ for all $l = 1, \ldots, d$ then        ▷ $\underline{j}$ admissible ?
            $\mathsf{A} := \mathsf{A} \cup \{\underline{j}\}$
            compute partial problem for index $\underline{j}$
        end
    end
    for all $\underline{k} \in \mathsf{A}$ do
        (re-)compute local error indicator $\varepsilon_{\underline{k}}$          ▷ uses opticom
    end
end
```

overall solution, as is the case for numerical integration [4]. But in our experiments for regularised regression the algorithm behaved better when we only used the indices of O for the combination technique.

This adaptive computational procedure is sketched in Algorithm 1.

Computing the sparse grid solution now involves solving the partial problems and combining them after (7). When one generalises the original combination technique with dimensional adaptivity the resulting coefficients, which are related to the “inclusion/exclusion” principle from combinatorics, depend only on the grids involved [6, 7]. But this ansatz leads to instabilities in our machine learning application [2, 7]. Instead we use so-called optimal combination coefficients $c_{\underline{l}}$; these now also depend on the function to be represented. They are optimal in the sense that the sum of the partial functions minimises the error against the actual sparse grid solution computed directly in the joint function space. We use the scalar product

$$\langle u, v \rangle_{RLS} = \frac{1}{M} \sum_{i=1}^{M} u(\underline{x}_i) v(\underline{x}_i) + \lambda \langle \nabla u, \nabla v \rangle_2$$

corresponding to the variational problem (1) and compute the optimal coefficients according to

$$\begin{bmatrix} \langle f_1, f_1\rangle_{RLS} & \cdots & \langle f_1, f_k\rangle_{RLS} \\ \langle f_2, f_1\rangle_{RLS} & \cdots & \langle f_2, f_k\rangle_{RLS} \\ \vdots & \ddots & \vdots \\ \langle f_k, f_1\rangle_{RLS} & \cdots & \langle f_k, f_k\rangle_{RLS} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} = \begin{bmatrix} \|f_1\|^2_{RLS} \\ \|f_2\|^2_{RLS} \\ \vdots \\ \|f_k\|^2_{RLS} \end{bmatrix}$$

using a suitable numbering of the $f_{\underline{l}}$ [2, 7].

The regression function is now built via

$$f_n^c(\underline{x}) := \sum_{\underline{l}\in\mathsf{O}} c_{\underline{l}} f_{\underline{l}}(\underline{x}) \tag{9}$$

using the above optimal coefficients.

A further generalisation of the original combination technique consists in the use of a slightly different level hierarchy. Let us formally define the one-dimensional basis functions $\widetilde{\phi}_{l,j}(x)$ as

$$\begin{aligned} \widetilde{\phi}_{-1,0} &:= 1, \\ \widetilde{\phi}_{0,0} &:= \phi_{0,1}, \\ \widetilde{\phi}_{l,j} &:= \phi_{l,j} \quad \text{for } l \geq 1, \end{aligned}$$

with $\phi_{l,j}$ as in (5). Note that it holds $\phi_{0,0} = \widetilde{\phi}_{-1,0} - \widetilde{\phi}_{0,0}$. We build the tensor product between a constant in one dimension and a $(d-1)$-linear function; the resulting d-dimensional function is still $(d-1)$-linear, so we gain no additional degrees of freedom. But formally introducing a level -1, and using this as coarsest level in the dimension adaptive procedure, allows us to build a combined function in the ANOVA-style, in other words each partial function possibly depends only on a subset of all features.

Using this hierarchy we now start in Algorithm 1 with the constant function of grid $\Omega_{\underline{-1}}$ (i.e., start with $\mathsf{A} := \{\underline{-1}\}$) and look in the first step at all grids which are linear in only one dimension, that is all $\Omega_{\underline{-1}+\underline{e}_j}$ with $j = 1, \ldots, d$. After one of these one-dimensional grids is chosen in the adaptive step the algorithm starts to branch out to grids which can involve two dimensions. Since each partial grid is now much smaller it allows us to treat even higher dimensional problems than before. Furthermore, the information about which dimensions are refined and in which combination allows an interpretation of the combined solution by the end-user, for example one can easily see which input dimensions are significant.

Table 1: Results for the synthetic Friedman data sets using the dimension adaptive combination technique (below) in comparison with the optimised combination technique, SVM, and MARS. Given is the mean squared error (MSE) on the test data, the timings are 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

dimension adaptive combination technique

	tol.	MSE	time	max. level per dim of $\underline{i} \in \mathsf{A}$
Friedman1	0.001	1.035	68.1	3 3 4 1 1 0 0 0 0 0
Friedman2 ($\times 10^3$)	0.0025	15.39	12.2	2 1 3 1
Friedman3 ($\times 10^{-3}$)	0.0001	11.57	35.9	2 4 6 0

4 Numerical experiments

For our experiments we use the well known synthetic data sets Friedman1 to Friedman3 [1]. We randomly generate 100,000 data points for training and another 10,000 for testing, where the positions are uniformly distributed over the domain. For the optimised combination technique (opticom) and the dimension adaptive combination technique (using optimal combination coefficients and starting with constants in each dimension) we employ a 2:1 split of the data for the parameter fitting of λ and n or the threshold of the stopping criterion, respectively. We compare with ϵ-support vector regression (SVM) as a state-of-art method using a Gaussian-RBF-kernel and perform a grid search over its parameters on a small subset of the training data. As a simple and fast baseline method we use multivariate adaptive regression splines (MARS) with the highest degree of interaction useful. Results are shown in Table 1, we compare the mean squared error (MSE) on the test data.

First let us look at the four dimensional data sets Friedman2 and Friedman3 which have data in $0 \leq x_1 \leq 100$, $40\pi \leq x_2 \leq 560\pi$, $0 \leq x_3 \leq 1$, and $1 \leq x_4 \leq 11$. The outputs for Friedman2 are created according to the formula $y = (x_1^2 + (x_2 x_3 - 1/(x_2 x_4))^2)^{0.5} + e$ where e is the normal distribution $N(0, 125)$. We achieve with the new method more or less the same results as the other algorithms, but reduce the time in comparison with the opticom approach resulting in the fastest method for this data set. For Friedman3 one has $y = \mathrm{atan}((x_2x_3 - 1/(x_2x_4))/x_1) + e$ where e is $N(0, 0.1)$. Here we significantly improve, both in run time and accuracy, relative to the opticom approach, which before gave the best accuracy of the three compared routines. Note that here the fourth dimension is not refined and therefore can be viewed as not significant although it is used in the generating formula.

The data set Friedman1 is generated with $y = 10\sin(\pi x_1x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + e$ where e is $N(0, 1)$ and all ten variables, including five as noise, are in $[0, 1]^d$. On this benchmark the standard opticom method is negatively affected by the five noise variables. The dimension adaptive approach perfectly captures the behaviour of the data, all five noise variables are considered not significant, the fourth and fifth are refined once and therefore recognised as having a linear contribution, the first two variables are refined jointly and the third variable is highly refined. The run time for the dimension adaptive approach is about 40 times smaller than that for opticom; furthermore we achieve the best testing accuracy with our new method.

5 Conclusions

The dimension adaptive combination technique for regression is an approach to achieve good approximation results in high dimensions with small computational effort. It results in a non-linear function describing the relationship between predictor and response variables and (approximately) identifies the ANOVA-decomposition of the problem. We currently employ a simple greedy approach in the adaptive procedure. More sophisticated adaptation strategies and error estimators are worthwhile investigating, especially in regard to an underlying theory which could provide robustness and efficiency of the approach similar to the numerical solution of partial differential equations with adaptive finite elements. Note that the concept of weighted Sobolev spaces $H(k_{\underline{a}})$ with weights of finite order [8] could be used as a theoretical framework in this context.

The dimension adaptive procedure allows an interpretation of the results, especially important for real data. By examining which dimensions are not refined, one can pick out features which are not significant for the prediction. Studying which dimensions are chosen for concurrent refinement on partial grids gives information on which features interact in some way. Depending on the application from which the data stems, this can be fruitful information. Investigations on real life data in more than 15 dimensions (the limit of the normal combination technique) are being planned.

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

References

[1] Jerome H. Friedman. Multivariate adaptive regression splines. *Ann. Statist.*, 19(1):1–141, 1991. `http://projecteuclid.org/euclid.aos/1176347963`.

[2] J. Garcke. Regression with the optimised combination technique. In W. Cohen and A. Moore, editors, *Proceedings of the 23rd ICML*, pages 321–328, 2006. `http://dx.doi.org/10.1007/s006070170007`.

[3] J. Garcke, M. Griebel, and M. Thess. Data mining with sparse grids. *Computing*, 67(3):225–253, 2001. `http://dx.doi.org/10.1007/s006070170007`.

[4] T. Gerstner and M. Griebel. Dimension–Adaptive Tensor–Product Quadrature. *Computing*, 71(1):65–87, 2003. `http://dx.doi.org/10.1007/s00607-003-0015-5`.

[5] M. Griebel, M. Schneider, and C. Zenger. A combination technique for the solution of sparse grid problems. In P. de Groen and R. Beauwens, editors, *Iterative Methods in Linear Algebra*, pages 263–281. IMACS, Elsevier, North Holland, 1992. `http://wissrech.ins.uni-bonn.de/research/pub/griebel/griesiam.ps.gz`.

[6] M. Hegland. Adaptive sparse grids. In K. Burrage and Roger B. Sidje, editors, *Proc. of 10th Computational Techniques and Applications Conference CTAC-2001*, volume 44 of *ANZIAM J.*, pages C335–C353, 2003. `http://anziamj.austms.org.au/V44/CTAC2001/Hegl`.

[7] M. Hegland, J. Garcke, and V. Challis. The combination technique and some generalisations. *Linear Algebra and its Applications*, 420(2–3):249–275, 2007. `http://dx.doi.org/doi:10.1016/j.laa.2006.07.014`.

[8] Ian H. Sloan, Xiaoqun Wang, and Henryk Woźniakowski. Finite-order weights imply tractability of multivariate integration. *Journal of Complexity*, 20:46–74, 2004. `http://dx.doi.org/10.1016/j.jco.2003.11.003`.