High dimensional wavelet smoothing

Ole Møller Nielsen*

(Received 7 August 2000)

Abstract

A fundamental issue in Data Mining is the development of algorithms to extract some useful information from very large databases. One important technique is to estimate a smooth surface approximating the data. However, the number of observations can be of the order of millions and there may be hundreds of variables recorded so one has to deal with the so-called "curse of dimensionality". The algorithmic complexity of this process is of the order N^{3d-2} where N is the number of grid points in each dimension and d is the number of dimensions.

*Computer Sciences Laboratory, RSISE, Australian National University, Canberra ACT 0200, Australia. mailto:Ole.Nielsen@anu.edu.au

[0] See http://anziamj.austms.org.au/V42/CTAC99/Neil for this article and ancillary services, © Austral. Mathematical Soc. 2000. Published 27 Nov 2000.

We propose a method for approximating a high dimensional surface by computing a projection onto multiresolution spaces of low density and we demonstrate that the algorithmic complexity of the multiresolution method is proportional to $((\log N)^{d-1}N)^3$—a substantial reduction in computational work. In addition, we show that the approximation error is proportional to d^2J2^{-2J}, the proportionality constant depending on the smoothness of the computed surface.

Contents

1 Data mining

Due to the availability of cheap disk space and automatic data collection mechanisms huge data collections in the terabyte range are becoming common both in business and science. Examples include the customer data bases of health and car insurance companies, banks, business transactions of retailers, taxation office genome data bases and remote sensing data. In business, these data bases have been used to assist in the daily transactions. However, it is seen that the data may also contain a wealth of information about the behaviour of the customers which traditionally has been gathered independently with expensive market surveys. Data mining attempts at getting the benefits out of these large data collections [1].

The algorithms applied in data mining have to deal with two major challenges: First they have to be able to handle data in the Terabyte range and have to be able to scale from smaller to larger data sizes when more data becomes available. Second, they also need to deal with complex data as each record may contain as many as 100 attributes or features.

An important function is to be able to predict the likelihood of a car insurance customer making a claim, a business customer to purchase a product or a resident to commit taxation fraud. This is typically described by a function

$$y = u(x_1, \ldots, x_d)$$

where the x_i are d attributes describing the customer and y the value to be estimated for this customer. This model u is estimated based on the given customer data base. In the following it will be assumed that all the attributes or features x_i used are real values and we set $\boldsymbol{x} = (x_1, \ldots, x_d)^T$. In many applications the response variable is known to depend in a smooth way on the values of the features. Thus smoothing is a natural candidatefor such data mining problems. However, it turns out that the systems of equations which arise from smoothing for data mining applications are very hard to solve due to very large number of unknowns and the denseness of the systems. A general smoothing spline is defined as the minimiser of a quadratic functional of the form

$$J_\alpha(f) = \sum_{i=1}^{n} (f(x^{(i)}) - y^{(i)})^2 + \alpha(Tf, Tf)$$

where n is the number of data points, T is a differential operator which maps real functions of d variables into real vector-valued functions of d variables. Examples include the gradient, Hessian etc. By $(\cdot, \cdot)$ we denote the usual scalar product for the vector-valued functions. The smoothing parameter α controls the trade-off between smoothness and fit.

The functions are chosen from some function space which might be defined by constraints.

As an example consider thin plate splines. They are the solution of a quadratic minimisation problem which trades the goodness of fit for smoothness. It is well known that thin plate splines can be written as the linear combination of radial basis functions [4]. The determination of the coefficients of this linear combination requires the solution of a dense linear system of equations the size of which equals the number of data points. The matrix elements depend on the distances between the data points which can be computed in $O(d)$ time and thus the curse of dimension has been overcome. However, there are $O(n^2)$ matrix elements. Thus just the computation of all the matrix elements is a non-scalable process with complexity $O(n^2)$. Techniques to solve these equations tend to work well for 2 dimensional problems but not for very high dimensional ones.

In a different attempt, finite elements were suggested to solve the underlying variational problem approximately [5]. This leads to a penalised least squares fit for the finite elements. For a given finite element space the data is only required in the assembly of the matrix and every data point needs to be read once only. Thus this method is scalable. However, it is not good in dealing with high dimensional data as it is based on tensor product approximations. It was seen in practical tests that even if the grid points per dimension is moderate (32) it is infeasible to consider higher than 4 dimensional data.

In the following we will develop a new technique based on multiresolution analysis and demonstrate that the algorithmic complexity can be greatly reduced for high dimensional problems.

2 Multiresolution analysis

A natural framework for wavelet theory is multiresolution analysis (MRA) which is a mathematical construction that characterises wavelets in a general way. The goal of MRA is to express an arbitrary function $f \in L^2(\mathbf{R})$ at various levels of detail. MRA is characterised by the following axioms:

$$\begin{aligned}
&\{0\} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \subset L^2(\mathbf{R}) && (a) \\
&\overline{\bigcup_{j=-\infty}^{\infty} V_j} = L^2(\mathbf{R}) && (b) \\
&\{\phi(x-k)\}_{k\in\mathbf{Z}} \text{ is a Riesz basis for } V_0 && (c) \\
&f \in V_j \quad \Leftrightarrow \quad f(2\cdot) \in V_{j+1} && (d)
\end{aligned} \tag{1}$$

This describes a sequence of nested approximation spaces V_j in $L^2(\mathbf{R})$ such that the closure of their union equals $L^2(\mathbf{R})$. Projections of a function $f \in L^2(\mathbf{R})$ onto V_j are approximations to f which converge to f as $j \to \infty$. Furthermore, the space V_0 has a Riesz basis consisting of integral translations of a certain function ϕ. Finally, the spaces are related by the requirement that a function f moves from V_j to V_{j+1} when rescaled by 2. It is usually required that ϕ has unit area [3, p.175], i.e.

$$\int_{-\infty}^{\infty} \phi(x)\, dx = 1 \tag{2}$$

It follows from (1) that the set of functions

$$\{\phi_{j,l}(x) = \phi(2^j x - l)\}_{l\in\mathbf{Z}} \text{ is a Riesz basis for } V_j$$

Given the nested subspaces in (1), define the subspace W_j such that

$$V_{j+1} = V_j \oplus W_j \tag{3}$$

Consider now two spaces V_{J_0} and V_J, where J and J_0 are arbitrarily chosen resolutions with $J \geq J_0$. Oftentimes one takes $J_0 = 0$. Applying (3) recursively we find that

$$V_J = V_{J_0} \oplus \left(\bigoplus_{j=J_0}^{J-1} W_j \right) \tag{4}$$

Thus any function in V_J can be expressed as a linear combination of functions in V_{J_0} and W_j, $j = J_0, J_0 + 1, \ldots, J-1$; hence it can be analysed separately at different scales. Multiresolution analysis has received its name from this separation of scales. It is shown in [3, p.135] that there exists a function $\psi(x)$ such that

$$\{\psi_{j,l}(x) = \psi(2^j x - k)\}_{k\in\mathbf{Z}} \text{ is a Riesz basis for } W_j$$

We call ϕ the **basic scaling function** and ψ the **basic wavelet**[1] and they are the two fundamental functions of the theory.

[1] In the literature ψ is often referred to as the **mother wavelet**.

In order to simplify the notation in the following we will rewrite (4) as

$$\boxed{V_J = \bigoplus_{j=J_0}^{J} S_j,} \qquad S_j = \begin{cases} V_{J_0} & j = J_0 \\ W_{j-1} & j > J_0 \end{cases} \tag{5}$$

2.1 Multiresolution in high dimensions

When moving from one dimensional wavelet decomposition to the d-dimensional decomposition, one may choose the tensor product of the univariate space. In particular, the tensor product spaces

$$U_j = \bigotimes_{s=1}^{d} V_j, \quad j \in \mathbf{Z}$$

are commonly used. The sequence U_j, $j \in \mathbf{Z}$ form a multiresolution of $L_2(\mathbf{R}^d)$. The density of this space is 2^{jd}. This is $2^{(j-1)d}$ times more than what was required in the 1D case and this is an appearance of the curse of dimensionality. For large d this type of approximation is not computationally feasible.

Using the MRA decomposition given in (5) we obtain

$$U_J = \bigotimes_{s=1}^{d} \bigoplus_{j_s=J_0}^{J} S_{j_s} = \bigoplus_{j_1,\ldots,j_d=J_0}^{J} \bigotimes_{s=1}^{d} S_{j_s} \tag{6}$$

where we assume that the scales J_0 and J are the same in all dimensions. The density of U_J is still 2^{Jd} but if the function f to be approximated is sufficiently smooth, the projection onto the tensor products $\bigotimes_{s=1}^{d} S_{j_s}$ where some of the j_s are sufficiently large will be very small. In fact, it can be shown [11] that for a sufficiently smooth function f we have the following bound on any of the d-dimensional projections:

$$\left\| \mathcal{P}_{S_{j_1} \otimes \cdots \otimes S_{j_d}} f \right\|_2 \leq C(f, r) 2^{-r \sum_{s=1}^{d} j_s} \tag{7}$$

where r depends on the regularity of the actual wavelet basis - the so-called number of vanishing moments. See e.g. [2], [10], [3] or [8]. Figure 1 shows the norms of the projections onto $S_{j_1} \otimes S_{j_2}$ for $j_s = J_0, \ldots, J$, $s = 1, 2$. It is seen that the projections where $j_1 + j_2 > J + J_0$ contribute little to the reconstruction of f. In fact, it can be verified that the convergence rates are exactly as predicted by Equation (7).

This suggests to remove all basis functions in $\bigotimes_{s=1}^{d} S_{j_s}$ where $\sum_{s}^{d} j_s > J + (d-1)J_0$ from subspace U_J to form a new space T_J. Thus, after compressions, the basis functions in this subspace are inactive. This is data *independent* or a-priori compression as opposed to the more common data dependent compression where wavelet coefficients are discarded based on their magnitude. See e.g. [10].

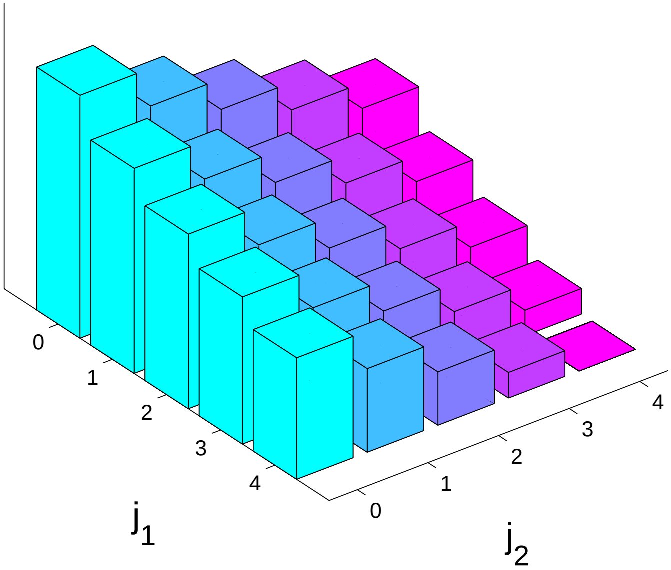


Figure 1: Norms of projections onto detail spaces. The example function is $u(x_1, x_2) = e^{-(x_1^2+x_2^2)}$ with $(x_1, x_2) \in [0, 1] \times [0, 1]$ and $r = 2$.

The truncated subspace T_J of U_J is defined as

$$T_J = \bigoplus_{\substack{J_0 \leq j_1,\ldots,j_d \leq J \\ j_1+\cdots+j_d \leq J+(d-1)J_0}} \bigotimes_{s=1}^{d} S_{j_s} \tag{8}$$

and it can be shown ([11]) that the density of V_J is $J^{d-1}2^J$.

The left graph in Figure 2 shows the spaces that constitute V_J with $J = 4, J_0 = 0, d = 2$ and the right shows the relative sizes of the spaces involved (for arbitrary J and $J_0 = J - 4$, $d = 2$).

2.2 Choosing a suitable basis

In the application at hand, piecewise linear “hat” functions are used in the Finite Element Approximation of u [5]. Therefore it is natural to choose these elements for the scaling functions $\phi_{J,l}(x)$ and then define the wavelets $\psi_{j,l}(x)$ accordingly. More specifically we have

$$\phi(x) = \begin{cases} 1+x & x \in [-1,0] \\ 1-x & x \in [0,1] \\ 0 & \text{otherwise} \end{cases}, \qquad \psi(x) = \phi(2x-1)$$

and the so-called **refinement equations**

$$\begin{aligned} \phi_{j-1,l} &= \tfrac{1}{2}\phi_{j,2l+1} + \phi_{j,2l} + \tfrac{1}{2}\phi_{j,2l-1} \\ \psi_{j-1,l} &= \phi_{j,2l+1} \end{aligned}$$

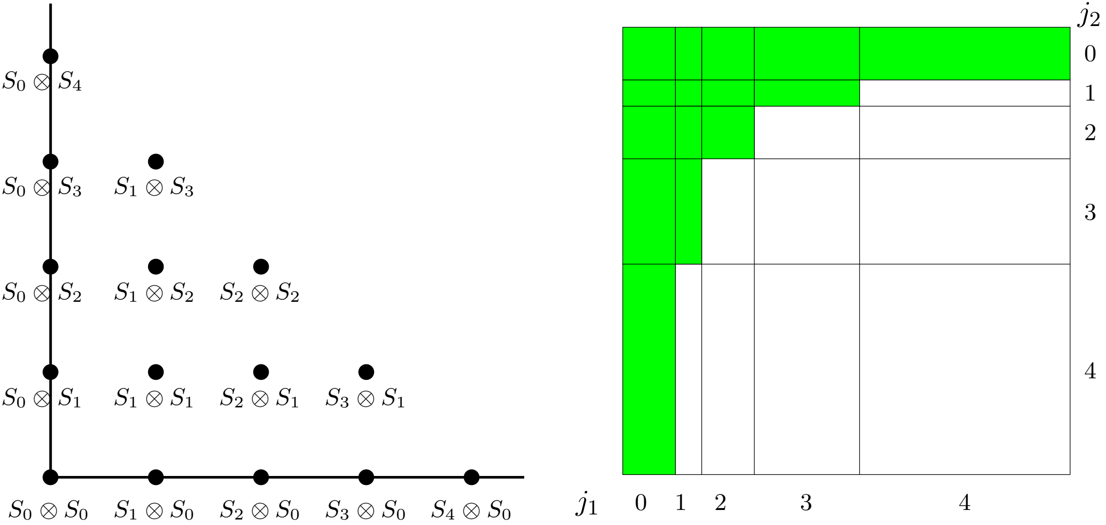


FIGURE 2: The spaces constituting T_4 in 2d (left) and the blocks of the wavelet transformed function u necessary to represent $P_{T_4}u$ (right).

that generate all the necessary functions. This is an example of a *bi-orthogonal* wavelet basis. Figure 3 shows the one used in this paper.

Using the refinement equations one arrives at the recursion

$$\begin{aligned} c_{J,l} &= u_{J,l} = u(l/2^J), \quad l = 0, \ldots, 2^J \\ c_{j-1,l} &= c_{j,2l}, \quad j \leq J,\ l = 0, \ldots, 2^{j-1} \\ d_{j-1,l} &= c_{j,2l+1} - (c_{j,2l} + c_{j,2l+2})/2, \quad j \leq J,\ l = 0, \ldots, 2^{j-1} - 1 \end{aligned}$$

This is a linear transformation so we will denote it by its matrix representation $\boldsymbol{W}$, i.e. we have an invertible mapping from coefficients at the finest level $\boldsymbol{u}$ to the vector of wavelet coefficients denoted by $\boldsymbol{d}$: $\boldsymbol{d} = \boldsymbol{W}\boldsymbol{u}$.

2.3 High dimensional multiresolution basis

We define the multi dimensional basis as

$$\begin{aligned} \boldsymbol{\phi}_{\boldsymbol{j},\boldsymbol{l}} &= \bigoplus_{s=1}^{d} \phi_{j_s,l_s} \\ \boldsymbol{\sigma}_{\boldsymbol{j},\boldsymbol{l}} &= \bigoplus_{s=1}^{d} \sigma_{j_s,l_s}, \qquad \sigma_{j,l} = \begin{cases} \phi_{J_0,l} & j = J_0 \\ \psi_{j-1,l} & j > J_0 \end{cases} \end{aligned}$$

and a high dimensional expansion will then have the form

$$u = \sum_{l_1,\ldots,l_d=0}^{2^J} u_{\boldsymbol{J},\boldsymbol{l}} \boldsymbol{\phi}_{\boldsymbol{J},\boldsymbol{l}}, \qquad u_{\boldsymbol{J},\boldsymbol{l}} = u(l_1/2^J, \ldots, l_d/2^J)$$

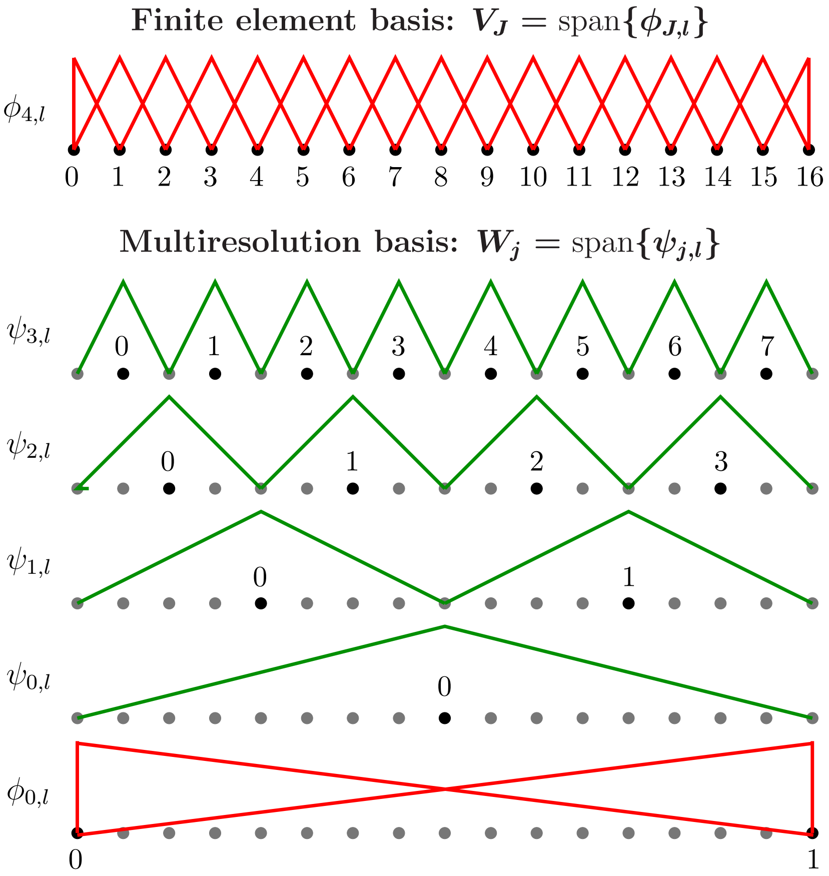


Figure 3: An example of the chosen bi-orthogonal wavelet basis.

$$u \quad = \quad \sum_{j_1,\ldots,j_d=J_0}^{J} \sum_{l_s=0}^{\theta(j_s)} d_{\boldsymbol{j},\boldsymbol{l}} \boldsymbol{\sigma}_{\boldsymbol{j},\boldsymbol{l}}, \qquad \theta(j) = \begin{cases} 2^{J_0} & j = J_0 \\ 2^{j-1} & j > J_0 \end{cases}$$

Similar to the matrix formulation of the 1D transform we have the **fast high dimensional wavelet transform**: $\boldsymbol{u} \to \boldsymbol{d}$ which we represent by the matrix $\boldsymbol{W}$.

3 The wavelet smoothing technique

Given the data set: $(\boldsymbol{x}^{(i)}, y^{(i)}), i = 1, \ldots, n \qquad \boldsymbol{x}^{(i)} \in \mathbf{R}^d,\ y^{(i)} \in \mathbf{R}$ we wish to minimise the functional

$$J_\alpha(u) = \|u(\boldsymbol{x}) - y\|^2 + \alpha \int_{\mathbf{R}^d} |\mathcal{L}u(\boldsymbol{x})|^2 \; d\boldsymbol{x}$$

Using a Galerkin projection we obtain the Matrix formulation

$$J_\alpha(\boldsymbol{u}) = \|\boldsymbol{M}\boldsymbol{u} - \boldsymbol{y}\|^2 + \alpha \boldsymbol{L}\boldsymbol{u}$$

which in turn is equivalent to the linear system

$$\boxed{\left(\boldsymbol{M}^T\boldsymbol{M} + \alpha\boldsymbol{L}\right)\boldsymbol{u} = \boldsymbol{M}^T\boldsymbol{y}}$$

This system can be expressed in our multidimensional wavelet basis as follows

$$\boxed{\left(\widehat{\boldsymbol{M}}^T\widehat{\boldsymbol{M}} + \alpha\widehat{\boldsymbol{L}}\right)\boldsymbol{d} = \widehat{\boldsymbol{M}}^T\boldsymbol{y}}$$

$$\begin{aligned}\widehat{\boldsymbol{M}} &= \boldsymbol{M}\boldsymbol{W}^{-1}\\ \widehat{\boldsymbol{L}} &= (\boldsymbol{W}^{-1})^T\boldsymbol{L}\boldsymbol{W}^{-1}\\ \boldsymbol{d} &= \boldsymbol{W}\boldsymbol{u}\end{aligned}$$

Note that this formulation is used only to describe the process. For computational efficiency these matrices must be derived directly using the Galerkin method with the wavelet basis.

3.1 Reducing the computational work

The system for computing all wavelet coefficients has the form

$$\boldsymbol{A}\boldsymbol{d} = \boldsymbol{v}$$

However the coefficients needed to approximate the compressed u is given by

$$I = \left\{(\boldsymbol{j}, \boldsymbol{l}) : \sum_{s=1}^{d} j_s \le J + (d-1)J_0\right\}$$

Hence we can express the reduced system by discarding equations for inactive coefficients as follows

$$\boldsymbol{A}_{I,I}\boldsymbol{d}_I = \boldsymbol{v}_I$$

$$\mathcal{P}_{T_J}\boldsymbol{u} \approx \boldsymbol{W}^{-1}\boldsymbol{d}$$

Figure 4 shows the coefficient matrix in terms of scaling functions all at scale J, the uncompressed wavelet representation, the wavelet representation with inactive coefficients set to zero, and finally the compressed matrix.

4 Complexity and numerical results

Figure 5 shows the number of non zeros of the full system and the compressed system respectively and Figure 6 shows the actual number of flops involved in the two representations. Figure 7 shows a 2D example on a real data set consisting of 735700 observations of a magnetic field.

5 Conclusion and further work

Our work has indicated that high dimensional wavelet smoothing can lead to

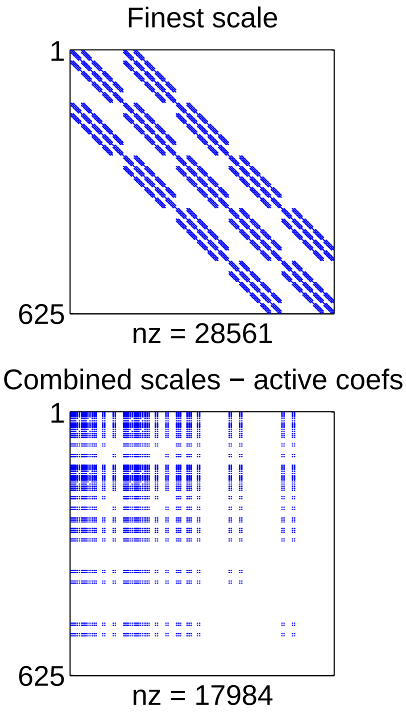
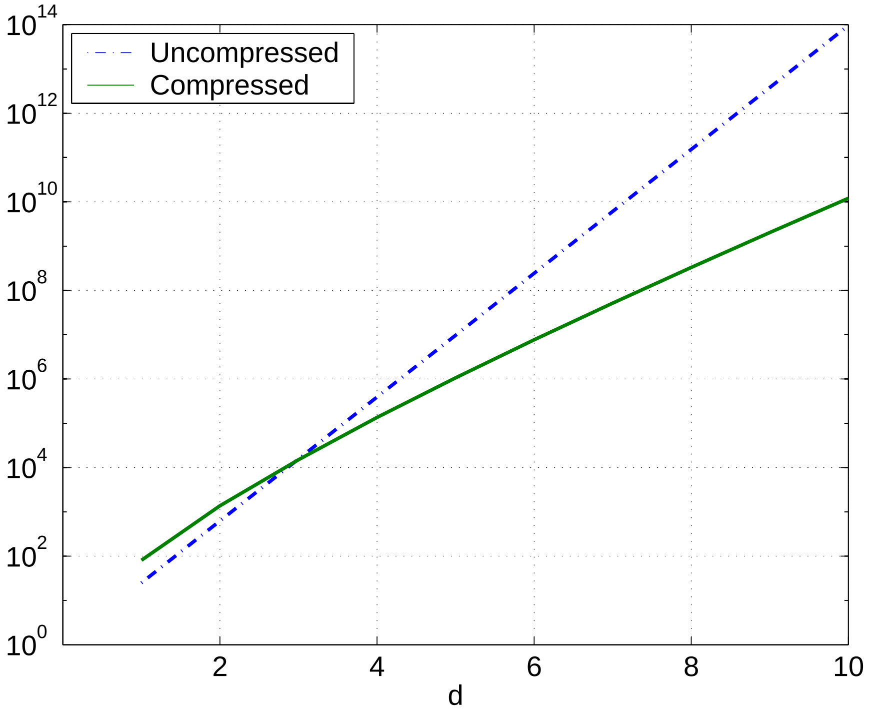
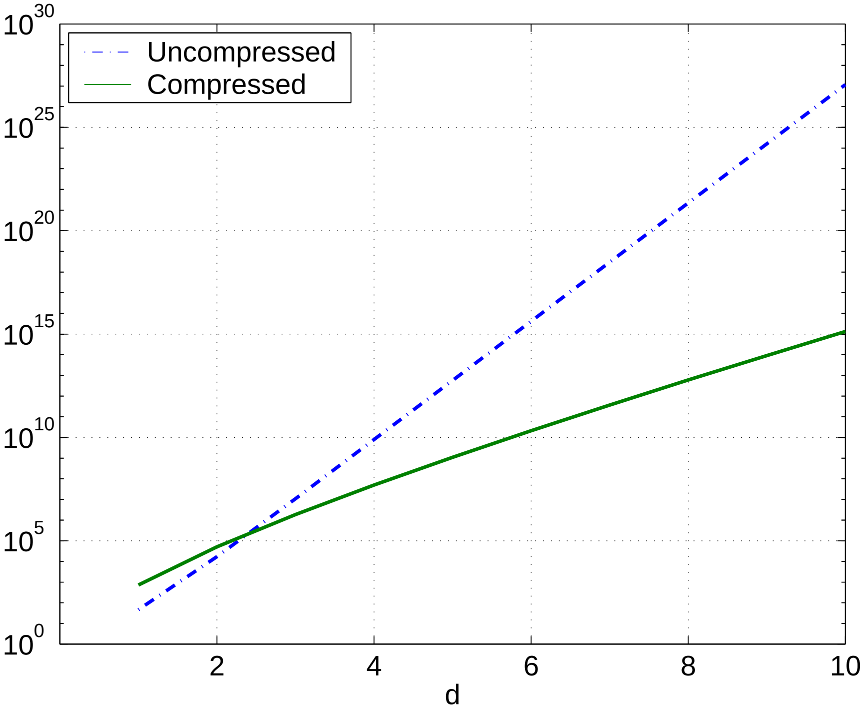


Figure 4: Coefficient matrix: $\widehat{\boldsymbol{M}}^{\boldsymbol{T}}\widehat{\boldsymbol{M}} + \boldsymbol{\alpha}\hat{\boldsymbol{L}} \quad (\boldsymbol{d} = \mathbf{4}, \boldsymbol{J} = \mathbf{2})$



$$
\begin{aligned}
&\text{Uncompressed:} \quad && = (3 \times 2^J - 2)^{d-1} \\
&\text{Compressed:} \quad && = M^2, \quad M = J^{d-1} 2^J
\end{aligned}
$$

FIGURE 5: Nonzeros, $\boldsymbol{J = 3}$



$$\begin{array}{lll} \text{Uncompressed:} & \approx B^2 2^{Jd}, & B = 3 \times 2^{J(d-1)} \\ \text{Compressed:} & = M^3, & M = J^{d-1} 2^J \end{array}$$

Figure 6: Floating point operations, $\boldsymbol{J = 3}$

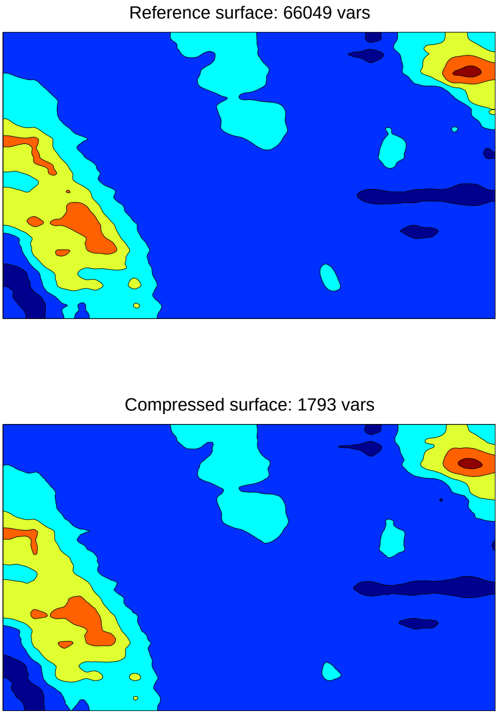


Figure 7: A real example on a 2D data set consisting of 735700 measurements of a magnetic field. The two surfaces are very similar but the lower one uses only a fraction of the coefficients used to represent the upper one.

- Very large reduction of complexity for three dimensions and higher
- Good approximation of smooth functions

Our plans for the future is to

- Incorporate this technology for more complicated penalty terms (e.g. in the in the TPSFEM data mining tool described in [5]).
- Apply to more real-world data mining problems
- Extend method to incorporate a-posteriori compression and automatic detection of “events” of interest.

Acknowledgement: This work was sponsored by the Danish Research Council.

References

[1] M.J.A. Berry and G.S. Linoff. *Data Mining Techniques for Marketing, Sales and Customer Support.* John Wiley and Sons, 1997. C1036

[2] C.K. Chui. *Wavelets: A Mathematical Tool for Signal Analysis*. SIAM Monographs on Mathematical Modeling and Computation. SIAM, 1997. C1042

[3] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992. C1039, C1040, C1042

[4] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. *Lecture Notes in Math.*, Vol. 571, pages 85–100, 1977. C1037

[5] M. Hegland, S. Roberts, and I. Altas. Finite element thin plate splines for data mining applications. In M. Daehlen, T. Lyche and L.L. Schumaker, editors, *Mathematical Methods for Curves & Surfaces II*, pages 245–253, Nashville, TN, 1998, Vanderbilt University Press. C1038, C1044, C1055

[6] M. Hegland. Real and complex fast Fourier transforms on the Fujitsu VPP500. *Parallel Computing*, 22:539–553, 1996.

[7] Y. Meyer. *Wavelets: Algorithms and Applications*. SIAM, 1993.

[8] O. M. Nielsen. *Wavelet in Scientific Computing.* PhD thesis, Technical University of Denmark, 1998. http://www.bigfoot.com/~Ole.Nielsen/thesis.html C1042

[9] O. M. Nielsen, G. Mercer, and M. Hegland. Vector-parallel fast wavelet transforms. In *PCW97 - Parallel Computing Workshop 1997*,

Australian National University, Canberra, ACT, 25-26 September 1997.

[10] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996. C1042, C1042

[11] M. Hegland, O. Nielsen, Z. Shen. A scalable high dimensional smoothing approach based on wavelet decomposition. In preparation. C1042, C1044