ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 47-52

Wavelet Interpolation Networks

Christophe Bernard, Stéphane Mallat, Jean-Jacques Slotine*

Centre de Mathématiques Appliquées
Ecole Polytechnique
91128 Palaiseau Cedex, France

Abstract. We describe a new approach to real time learning of unknown functions based on an interpolating wavelet estimation. We choose a subfamily of a wavelet basis relying on *nested hierarchical allocation* and update in real time our estimate of the unknown function. Such an interpolation process can be used for real time applications like neural network adaptive control, where learning an unknown function very fast is critical.

1. Introduction

Our purpose is to approximate an unknown function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ from scattered samples $(x_\tau, y_\tau = f(x_\tau))_{\tau=1\ldots t}$, where

- we have little a priori knowledge on the unknown function f: it lives in some infinite dimensional smooth function space.
- the function approximation process is performed iteratively: each new measure on the function $(x_t, f(x_t))$ is used to compute a new estimate f_t as an update of a previous estimate f_{t-1}.
- the above update computations and the data storage should be efficient, to fit in a real time learning framework.

A classical way to deal with such an ill–posed problem is regularization [4]. Regularization consists in minimizing some functional $M[f] + \lambda S[f]$ where the matching functional $M[f]$ and the smoothness functional $S[f]$ are defined as

$$M[f] = \sum_{\tau=1}^{t} \|f(x_\tau) - y_\tau\|^2 \quad S[f] = \|f\|_R^2 \tag{1}$$

R is some regularization Hilbert space such that the Dirac functional is continuous with regard to the underlying norm $\|.\|_R$. Suitable regularization spaces R are for example Sobolev spaces H_s for any s larger than $n/2$.

Regularization also guarantees existence and uniqueness of the solution, and the optimum depends continuously on the input measures.

*Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 47-52

Road map

In section 2, we recall why wavelets are efficient for storing, retrieving and updating the estimate of an unknown function. Then, we show how a regularization approach behaves in a wavelet basis, and how this behavior can be mimicked with a wavelet coefficient update to reduce the computational cost of the function estimation.

In section 3, we describe the two main parts of our algorithm: allocation and coefficient update. Section 4 is devoted to simulations.

2. Wavelets and regularization

2.1. Wavelet networks

Wavelet networks have been already used as single layer perceptron by Zhang & Benveniste [11], Pati & Krishnaprasad [7], Slotine & *al.* [8]. The nodes of this neural network are the wavelet coefficients of the function expansion that have a significant value.

The reason why wavelets are used instead of other transfer functions is triple:

— Wavelets have high compression abilities. Because wavelets are localized in time, wavelet coefficients will have significant values only around singularities. Keeping only such coefficients (by thresholding) consists in choosing an adaptive grid to store the function.

— Computing the value at a single point involves a small subset of coefficients

— Updating the function estimate from a new local measure also does.

2.2. Wavelet coefficient analysis

Let $(\psi_{jk})_{j\in\mathbb{N},k\in\mathbb{Z}^n}$ be an orthogonal wavelet basis, defined as $\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k)$ if $k > 0$, and $\psi_{0k} = \sqrt{2}\phi(2x-k)$ where ϕ and ψ are a scaling function and a base wavelet (see e.g. [6] for more details). We write any expansion of a function f in $L_2(\mathbb{R}^n)$ as $f = \sum_{jk} c_{jk}(f)\psi_{jk}$.

Provided that the wavelet ψ has enough vanishing moments, the following Hilbert norm defined for a given real $s \in \mathbb{R}_*^+$

$$\|f\|_s^2 = \sum_{jk} 2^{2sj}|c_{jk}|^2$$

is equivalent to the usual Sobolev norm $\|f\|_{H_s}$, and thus the set of functions f with finite norm $\|f\|_s$ is the Sobolev space H_s. One can prove that H_s is a *reproducing kernel Hilbert space* (see e.g. [4]), and that the f minimizing $M[f] + \|f\|_s^2$ is some linear combination

$$f(x) = \sum_{\tau=1}^{t} c_\tau K(x, x_\tau) \text{ where } K(x,y) = \sum_{jk} \frac{\psi_{jk}(x)\psi_{jk}(y)}{2^{2sj}} \tag{2}$$

From (2), we see that the coefficients c_{jk} introduced by some measure point x_τ are proportional to both $\psi_{jk}(x_\tau)$ and 2^{-2sj}. The high resolution coefficients are penalized

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 47-52

by the 2^{-2sj} factor. Also, coefficients whose center is close to x_τ are preferred to others.

For each additional measure x_t, we thus want to choose in our wavelet pool a wavelet whose center node $k/2^j$ is close to x_t, and whose resolution index j is as small as possible.

3. Algorithm

Our algorithm is divided into two main parts. The first part of the algorithm is designed to choose a subfamily of our wavelet basis of same cardinality as the number of measures. This is done with hierarchical allocation of the measures to wavelet basis nodes.

The second part consists in maintaining a linear system $AC = Y$ where A is an invertible square sparse matrix, C the vector of wavelet coefficients of f and Y the vector of measures $y_1, \ldots, y_t$. We show that A and A^{-1} can be updated with simple calculations as we get additional measures.

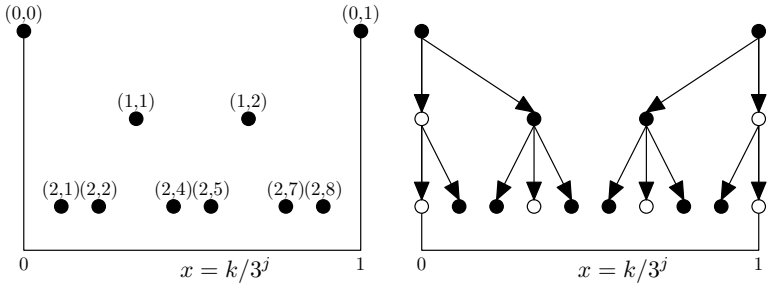
An additional truncation step is then used to limit the size of this growing system.

3.1. Tryadic wavelets

For geometric reasons, an allocation process compliant to the rules we explicited in section 2.2. cannot be seen as a tree descent, unless our wavelet family is an M–band multiresolution, where M is odd. Therefore, we designed a 3–band (tryadic) version of the Deslauriers–Dubuc interpolation process, and the following wavelet system

$$(\psi_{jk})_{(j,k)\in I} \text{ where } \psi_{jk}(x) = \psi(3^j x - k) \tag{3}$$

is a basis of the locally convex function space $C(\mathbb{R}^n, \mathbb{R})$ if the index set I is the set of pairs $(j,k) \in \mathbb{N} \times \mathbb{Z}^n$ where either $j = 0$ and k is any element of $\mathbb{Z}^n$, or $j > 0$ and $k/3 \notin \mathbb{Z}^n$. The set of wavelets is displayed in Fig. 1–a for $n = 1$. Each dot corresponds to a wavelet, and is placed at location $(k/3^j, -j)$, with $k/3^j$ being the center of the corresponding function ψ_{jk}



[a] Tryadic wavelet nodes [b] Tree structure

Figure 1: Tryadic interpolating wavelet tree structure

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 47-52

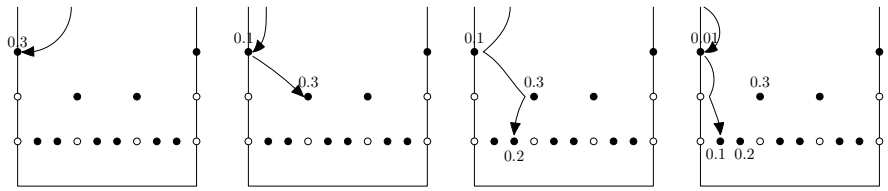
We designed a tree structure based on nodes of two kinds: *normal* and *virtual* nodes. Normal nodes can host a wavelet coefficient, while virtual nodes cannot. Virtual nodes are only here to have sons, so that the tree in Fig. 1–b keeps a simple structure of constant arity 3^n.

3.2. Allocation

If we are given a sequence of measures (x_τ, y_τ), we allocate each of them to one normal node of the tree in Fig. 1–b, somewhat like balls falling down a *pachinko* (Japanese pinball) board.

A new measure (x_t, y_t) is allocated at scale $j = 0$. At scale j, a measure is allocated to the closest node (j, k) of the tree (minimum $|x_t - k/3^j|$). If the node is virtual, the measure goes down to be allocated at scale $j + 1$. If the node is normal and vacant, allocation of this measure is complete. Else, there is a *competition* between the new measure and the measure already allocated at (j, k). The closest measure (in terms of distance between x_τ and k) stays at (j, k), and the other is pushed down to be allocated at scale $j + 1$.

This allocation step is done recursively for $j = 1, \ldots$ until complete. An example allocation sequence is detailed in Fig. 2, with four measure points $x_1, \ldots, x_4$.



[a] $x_1 = 0.3$ [b] $x_2 = 0.1$ [c] $x_3 = 0.2$ [d] $x_4 = 0.01$

Figure 2: Example of the *pachinko*–like allocation

Note that high resolution (large j) wavelets will only be used in areas where the measure points are numerous. Moreover, as the number of coefficients increases, the measures allocated to the wavelets are getting closer and closer to the wavelet centers, which enhances the system conditioning. Also note that thanks to the tryadic structure, the allocation process is a simple tree descent.

3.3. Coefficient update

When we have an additional measure, a new wavelet is chosen from the wavelet basis by the allocation process. Updating A consist in adding one row and one column that are both sparse. A^{-1} is then updated with a Sherman–Morrison–like formula. Updating Y is straightforward, and C can also be updated. As an example, when we add a new column V and a new row $[L\ \theta]$ to A, we must replace A^{-1} with

$$\begin{bmatrix} A & V \\ L & \theta \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + \frac{A^{-1}VLA^{-1}}{\theta - \alpha} & -\frac{A^{-1}V}{\theta - \alpha} \\ -\frac{LA^{-1}}{\theta - \alpha} & \frac{1}{\theta - \alpha} \end{bmatrix}$$

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 47-52

where $\alpha = LA^{-1}V$. Updating matrix A takes $O(\log^2 N)$ operations and updating C only $O(\log N)$ operations, if the number of coefficients stored is N.

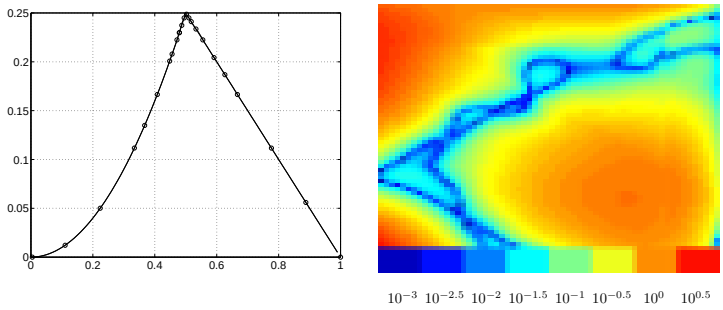
3.4. Truncation

When a computed wavelet coefficient is lower than some threshold at a tip of the tree, the coefficient and the corresponding measure are removed from the system. This consists in crossing out one row and one column from the current estimate of A and of A^{-1}.

This way, the algorithm does not stockpile endlessly measures (x_τ, y_τ) (and the corresponding lines and rows in A and A^{-1}) but only keeps the measures that bring significant additional information on the unknown function, the same way as an adaptive grid algorithm.

4. Simulations

One dimensional and two dimensional experimentations were done with a C++ program on piecewise smooth functions. One dimensional experimentations are done with



[a] 1D experiment [b] 2D experiment

Figure 3: Numerical experimentation

unknown an function f_1 defined as $f_1(x) = x^2$ if $x \leq 1/2$ and $f_1(x) = (1-x)/2$ elsewhere. The measured points are 200 random points. f_1 and the measure points that were not discarded are displayed in Fig. 3–a. The algorithm clearly preferred to keep more points around the singularity, as one would expect from an adaptive grid algorithm. Approximation error is less than 3.10^{-3} around the singularity and less than 3.10^{-5} elsewhere.

Two dimensional experimentations are done with $f_2(x, y) = sin|x - 1.2 \times y|$, on a trajectory

$$\left.\begin{array}{rcl} x_\tau & = & |cos(\tau/100)| \\ y_\tau & = & |1 - cos(\tau/125)| \end{array}\right\} \text{ for } \tau = 1 \ldots 197$$

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 47-52

A chart of the absolute erreur is displayed in Fig. 3–b. It shows that around the trajectory we followed, the approximation error is low (lower than 10^{-2}).

For the one–dimensional case, the system kept 24 measures from a 200 point input. For the 2–dimensional case, the system kept 27 measures from a 197 point input.

Conclusion

We have described here an automated process to adaptively select a wavelet subfamily to approximate an unknown function from random samples, based on the new concept of *allocation*. High resolution and numerous wavelet coefficients will appear only at locations where a high number of samples are available, and where the function has a singularity. This process is thus able to discard measures that can be considered as non informative, because they could have been predicted correctly by the algorithm. As a consequence, we gain several advantages: high precision, low computation and storage requirements.

References

[1] M. Cannon, J-J. Slotine, Space-Frequency Localized Basis Function Networks for Nonlinear System Estimation and Control, *Neurocomputing, 9(3), 1995.*

[2] I. Daubechies, Ten Lectures on Wavelets, *Society for Industrial and Applied Mathematics, Philadelphia, 1992.*

[3] G. Deslauriers and S. Dubuc, Symmetric Iterative Interpolation Processes, *Constructive Approximation, Vol. 5, pp49–68, 1989.*

[4] F. Girosi, An Equivalence between Sparse Approximation and Support Vector Machines, *AI Memo No. 1606, MIT, 1997.*

[5] S. G. Mallat, A Theory for Multiresolution Signal Decomposition, *IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 11:7, pp. 674-693, 1989.*

[6] S. G. Mallat, A Wavelet Tour of Signal Processing, *Academic Press, 1997.*

[7] Y. C. Pati and P. S. Krishnaprasad, Analysis and Synthesis of Feedforward Neural Networks Using Discrete Affine Wavelet Transformations, *IEEE Trans. on Neural Networks, vol. 4, no. 1, 1993.*

[8] R. Sanner, J-J. Slotine, Structurally Dynamic Wavelet Networks for the Adaptive Control of Uncertain Robotic Systems, *Proc. IEEE Conf. Decision and Control, New Orleans, 1995.*

[9] A. N. Tikhonov and V. Y. Arsenin, Solutions of ill–posed problems. *W. H. Winston, Washington DC, 1977.*

[10] G. Wahba, Smoothing noisy data by spline functions. *Numer. Math., vol. 24, pp383–393, 1975.*

[11] Q. Zhang and A. Benveniste, Wavelet Networks, *IEEE Trans. on Neural Networks, vol. 3, no 6., 1992.*