Appl. Math. and Comp. Sci., 1993, vol.3, No.4, 713–720

FFT IN CALCULATING NONPARAMETRIC REGRESSION ESTIMATE BASED ON TRIGONOMETRIC SERIES†

EWARYST RAFAJŁOWICZ*, EWA SKUBALSKA-RAFAJŁOWICZ*

In the paper the computational algorithm for nonparametric regression function estimation is proposed. The algorithm is based on using the Fast Fourier Transform method twice and is suited to calculate trigonometric series estimators. Its computational complexity is $0(n \log n)$, which yields essential savings in comparison to direct calculations of this estimator.

1. Introduction

The aim of this paper is to discuss computational aspects of nonparametric estimators for a regression function. The study is motivated by the fact that the theory in this field is now sufficiently rich for supporting possible computer implementations of these estimators. In recent years theoretical results in the fixed design case concentrated on

a) kernel type estimators, investigated by Benedetti (1977), Cheng and Lin (1981), Gasser and Muller (1979), Georgiev (1984; 1986), Georgiev and Greblicki (1986), Priestley and Chao (1972), Schuster and Yakowitz (1979),

b) orthogonal series estimators, discussed by Eubank and Speckman (1991), Greblicki and Pawlak (1985), Rafajłowicz (1984a; 1984b), Rafajłowicz (1987), Rutkowski (1982),

c) smoothing splines developed in Agarwal and Studden (1980), Cox (1984), Nussbaum (1985), among others.

Extensive bibliographies on nonparametric regression estimation in the fixed design case as well as when regressors are random variables can be found in Collomb (1985) and Gyorfi (1981), while spline smoothing is exhaustively reviewed by Eubank (1988).

In the sequel, we concentrate our attention on case b) since computational aspects of smoothing splines deserved much more attention in the literature.

The fast Fourier transform (FFT) algorithm seems to be the most natural tool for calculating the trigonometric series estimate (TSE) of a regression function. We shall discuss computational complexity (CC) of this method. Here CC is defined as the number of floating point additions and multiplications which is necessary for calculating an estimate for a large data set.

It should be mentioned that applications of FFT for calculating nonparametric kernel type estimates have been proposed by Silverman (1982) for probability densities and

† This work has been supported by the grant of the Polish government

* Institute of Engineering Cybernetics, Technical University of Wrocław, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

by Georgiev (1986) for a regression functions. In both cases FFT is applied in order to calculate convolutions more efficiently than by the direct summation. Although formally TSE can be treated as a special case of kernel estimators, we choose a more direct way of calculations. It consists of estimating the Fourier coefficients of the unknown function from given data by FFT, truncating them at a point dependent on the number of measurements and then finding the function estimate using inverse FFT. This approach leads to some computational savings, since values of the Dirichlet kernel are not calculated.

It should be however noticed that Georgiev's method is described in one–dimensional case and it is not clear for the authors, whether it can be generalized to multidimensional domains of independent variable.

The method presented here can be used in multidimensional case if multidimensional version of FFT is at our disposal (see, e.g., Dudgeon and Mersereau (1984)). The paper is organized as follows. In Section 2 main assumptions are listed. Computational version of TSE is presented in Section 3, while its CC is discussed in Section 4. Possible generalizations are mentioned in Section 5.

2. Main Assumptions

Let $X \triangleq [0,1]$ be the interval over which an unknown function $f \in L^2(X)$ is defined. Observations of f are taken at prescribed points

A_1) $x_i \in X; \quad i = 1,2,\ldots,n$ equidistantly spaced in X;
$H \triangleq x_i - x_{i-1}$.

As a result of measurements the data y_i, $i = 1,2,\ldots,n$ are obtained according to the scheme

A_2) $$y_1 = f(x_i) + \varepsilon_i \quad i = 1,2,\ldots,n \tag{1}$$

where the errors ε_i are assumed to be uncorrelated random variables with zero expectations and finite variances. Define $v_1(s) = 1$, $v_{2k}(x) = \sqrt{2}\sin(k\pi x)$, $v_{2k+1}(x) = \sqrt{2}\cos(k\pi x)$ and let $a_k = \int f v_k$ be the Fourier coefficients of f; $k = 1,2,\ldots$. Here and below we omit the domain of integration if X is involved.

We estimate coefficients a_k by

$$\hat{a}_k = H\sum_{i=1}^{n} y_i v_k(x_i), \quad k = 1,2,\ldots \tag{2}$$

where we omit the dependence of $\hat{a}_k$ on n for brevity. The function f is estimated as follows:

$$\hat{f}_n(x) = \sum_{k=1}^{N} \hat{a}_k v_k(x) \tag{3}$$

where $N = N(n)$ is a sequence of positive integers such that

A_3) $$N(n) \longrightarrow \infty \quad \text{as} \quad n \to \infty$$

A few remarks should be made on the above assumptions.

Remark 1. Formulae (2), (3) define the simplest version of TSE. In Rutkowski (1982) and Rafajłowicz (1987) other algorithms of a similar type can be found. Their theoretical properties are better than that of (2), (3), but they are similar from a computational point of view.

Remark 2. Condition A_3) is necessary for convergence of $\hat{f}_n$ to f as $n \longrightarrow \infty$ in any reasonable sense, provided that f does not have a finite expansion into the Fourier series. Sufficient conditions are not difficult to obtain mimicking proofs presented in Greblicki and Pawlak (1985), Rafajłowicz (1987) and Rutkowski (1982).

Remark 3. The fixed design case with equidistantly spaced points is considered here. However, the algorithm can also be applied for nonuniform sampling if every obsevation is rounded off to the nearest x_i. Rounding introduces errors in independent variables but consistency of TSE is in this case retained, as it is recently shown (Rafajłowicz, 1988).

3. Computational Algorithm for Trigonometric Series Estimate

Suppose that the data (x_i, y_i), $i = 1, 2, \ldots, n$ are in our disposal. We are interested in calculating $\hat{f}_n$, defined by (2), (3), at the same points x_i where measurements have been taken.

Algorithm $\mathcal{A}$

Step 1. Calculate the discrete Fourier transform (DFT) of the sequence y_i, $i = 1, 2, \ldots, n$ using FFT algorithm. Denote the result by Y_k; $k = 1, 2, \ldots, n$.

Step 2. Truncate this sequence at the point that corresponds to truncating the sequence $\hat{a}_k$, $k = 1, 2, \ldots$ at N-th element. The remaining elements Y_k multiply by H and denote by $\hat{c}_k$.

Comment. The sequence of complex numbers Y_k, $k = 1, 2, \ldots, n$, being DFT of a real sequence, exhibits symmetric real and antisymmetric imaginary part. Thus, the correspondence between $\hat{a}_k$ and $H \cdot Y_k$; $k = 1, 2, \ldots, n$ reads as follows. Elements $H \cdot (Y_k + Y_{n/2-k})$ are equal to $\hat{a}_k$ before cosine terms, while $H \cdot (Y_k - Y_{n/2-k})$ represent those $\hat{a}_k$ before sine terms, provided that n is divisible by 2.

Step 3. Extend the sequence $\hat{c}_k$ to the length n by putting 0 instead of truncated elements of $H \cdot Y_k$.

Step 4. Find inverse FFT of the extended sequence $\hat{c}_k$; $k = 1, 2, \ldots, n$ to get $\hat{f}_n(x_i)$; $i = 1, 2, \ldots, n$.

Few commets may be useful in implementing the above algorithm.

Remark 4. As it is known, FFT algorithm is efficient when n is a highly factorizable number. In most frequently used implementations of FFT it is required $n = 2^p$ or $n = 3^p$, $p > 0$. We shall not describe known tricks of handling sequences of different lenghts. The same tricks can be used at Step 3 when values of $\hat{f}_n$ are required at a grid other than x_i; $i = 1, 2, \ldots, n$.

Remark 5. In Step 1 we calculate FFT of a real sequence and this fact should be taken into account to make calculations more efficiently. One should also consult (Storn, 1986), where fast discrete Hartley transform (FDHT) is advocated as a somewhat more efficient tool for calculating DFT of real sequences than FFT. Further, however, we discuss CC of the above algorithm without referring to FDHT, since computer codes of FFT are easily available and sufficiently efficient (FDHT saves n multiplications). Furthermore, hardware implementations of FFT are currently available (Mehalik *et al.*, 1985).

Remark 6. For a given data set (x_i, y_i); $i = 1, 2, \ldots, n$ the estimate $\hat{f}_n$ is usually calculated several times in order to choose the truncation parameter N by cross-validation or by the visual inspection. In such a case it suffices to perform Steps 2, 3, 4, reducing CC by half.

Remark 7. Algorithm $\mathcal{A}$ provides a simple interpretation of TSE in terms of filtering higher harmonics in the sequence y_i $i = 1, 2, \ldots, n$.

4. Computational Complexity of the Algorithm

Let T_a, T_m denote execution time of one real addition and multiplication, respectively. By $T_F(n)$ we denote the time of transforming a real sequence of the length n using FFT. As it is known (see, e.g., Cooley *et al.*, 1977),

$$T_F(n) = C \cdot n \cdot \log_2 n \tag{4}$$

where the constant C does not depend on n.
Now, CC of the algorithm $\mathcal{A}$ can be evaluated as follows:

Step 1 requires $T_F(n)$,

Step 2 takes $2 \cdot N \cdot T_m$,

Step 3 is negligibly short,

Step 4 needs $T_F(n)$,

what results in the total time

$$T_1(n) = 2T_F(n) + 2N \cdot T_m, \tag{5}$$

or more roughly $T_1(n) \approx 2T_F(n)$, since usually $N \ll n$. If calculations are repeated for different values of N, then additional computational burden takes, roughly, $T_F(n)$ every time.

5. Discussion

Concluding this paper we indicate that possible generalizations of $\mathcal{A}$ are the following:

1) $\mathcal{A}$ can directly be used in multidimensional case if the domain of independent variables is a hypercube and multidimensional version of FFT is applied (see, e.g., Dudgeon and Mersereau (1984) for its description).

2) Minor changes are required in order to use $\mathcal{A}$ for calculating estimates of the form

$$\tilde{f}_n(x) = \sum_{k=1}^{\infty} w_k(n)\hat{a}_k v_k(x). \tag{6}$$

3) Flexibility of $\mathcal{A}$ can be enlarged if instead of trigonometric functions other orthonormal sequences v_k, $k = 1, 2, \ldots$ are used, for which fast algorithms analoguous to FFT are known.

In Table 1 results of testing algorithm $\mathcal{A}$ are summarized. The last column contains the execution time (in seconds) obtained for sequences of various lengths (given in the first column). The second column, called the Truncation Point, is related to the number of terms N in the regression (3) in the following way:

$$N = \frac{n}{2} - \textit{Truncation Point.}$$

Tab. 1. Execution time of algorithm $\mathcal{A}$ for data sequences of different lengths.

Number of data	Truncation point	Time in Sec.
10000	4900	180.43
	4000	154.28
	3000	124.79
	2000	93.1
	1000	61.79
5000	2400	57.45
	2000	50.25
	1000	32.51
2000	900	14.83
	500	10.55
	100	6.37
1000	400	5.72
	200	4.12
	100	3.35
512	240	2.52
	200	2.2
	100	1.54
128	50	0.55
	30	0.39
	10	0.28

The execution time has been measured using IBM PC/386/33 MHz computer with the math coprocessor. Analysis of Table 1 shows that even for relatively long data sequences $n = 10^4$, the execution time is acceptable, being negligible for short data sequences. This makes it possible to repeat calculations many times in order to find the best truncation point.

Performance of algorithms based on trigonometric series expansion is well documented in the literature (see, e.g. Eubank, 1988). For this reason we provide only one

example of behavior of algorithm $\mathcal{A}$ for a finite sample of length 300. The data in Figure 1 were generated equidistantly from the regression function of the form:

$$f(x_i) = 5 \cdot \sin^2(x_i) + \epsilon_i\,, \quad i = 1, 2, \ldots 300$$

where ϵ_i are pseudorandom numbers uniformly distributed in $[-0.75, 0.75]$. Discrete data points are joined by thin lines in Figure 1 in order to underline their variability. The resulting estimate (thick line in Fig. 1) was not optimized with respect to N and the value $N = 10$ (Truncation Point = 140) was assumed.

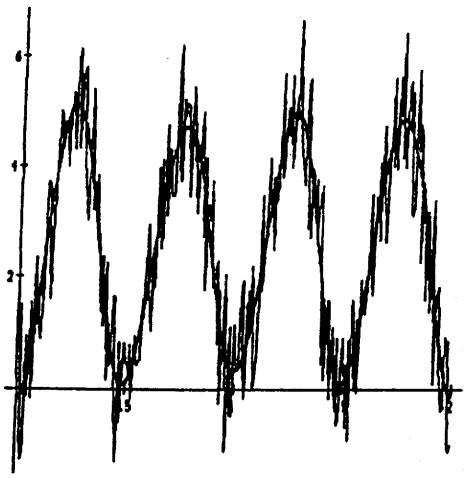

Fig. 1. Example of algorithm's performance.

As one can notice from Figure 1, even without a carefull optimization with respect to N the estimation method gives a satisfactory result.

References

Agarwal G.G. and Studden W.J. (1980): *Asymptotic integrated mean square error using least squares and bias minimizing splines.* — Ann. Statist., v.8, pp.1307–1325.

Benedetti J. (1977): *On the nonparametric estimation of regression function.* — J. Roy. Statist. Soc., Ser. B, v.39, pp.248–253.

Cheng K.F. and Lin P.E. (1981): *Nonparametric estimation of regression function.* — Z. Wahrscheinlichkeitstheorie Verw. Gebiete, v.57, pp.223–233.

Collomb G. (1985): *Nonparametric regression: An up-to-date bibliography.* — Statistics, v.16, pp.309–324.

Cooley J.W., Lewis P.A. and Welch P.D. (1977): *Fast Fourier Transform and its application in time series analysis.* — In: Enslein K., Ralston A., Wilf H.S. (Eds.): Statistical Methods for Digital Computers, New York: Wiley and Sons.

Cox D.D. (1984): *Multivariate smoothing spline functions.* — SIAM J. Numer. Anal., v.21, pp.789–813.

Dudgeon D.E. and Mersereau R.M. (1984): *Multidimensional Digital Signal Processing.* — Prentice Hall: Englewood Cliffs.

Eubank R.L. (1988): *Splines and Nonparametric Regression.* — New York: Dekker.

Eubank R.L. and Speckman A.P. (1991): *Convergence rates for trigonometric and polynomial–trigonometric regression estimators.* — J. Statis. and Prob. Lett., v.11, pp.119–124.

Gasser Th. and Muller H.G. (1979): *Kernel estimation of regression functions.* — In: Th. Gasser, M. Rosenblatt (Eds.): Smoothing Techniques for Curve Estimation, Lecture Notes in Mathematics, v.757, Berlin: Springer, New York: Heidelberg.

Georgiev A.A. (1984): *Kernel estimates of functions and their derivatives with applications.* — Stat. Prob. Lett., v.2, pp.45–50.

Georgiev A.A. (1986): *A fast algorithm for curve fitting.* — Proc. 7–th Symp. *Computational Statistics*, COMPSAT 86, Roma, Sept. 1986, Vien: Physica Verlag.

Georgiev A.A. and Greblicki W. (1986): *Nonparametric function recovering from noisy observations.* — J. Statist. Planing and Inference, v.13, pp.1–14.

Greblicki W. and Pawlak M. (1985): *Fourier and Hermite series estimators of regression function.* — Ann. Inst. Stat. Math., v.37, No.3, pp.443–454.

Gyorfi L. (1981): *Recent results on nonparametric regression estimate and multiple classification.* — Problems of Control and Information Theory, v.10, No.1, pp.43–52.

Mehalik M.A., Rustan P.L. and Route G.P. (1985): *Effects of architecture implementation on DFT algorithm performance.* — Trans. IEEE, v.ASSP–33, pp.684–693.

Nussbaum M. (1985): *Spline smoothing in regression models and asymptotic efficiency in* L_2. — Ann. Statist., v.13, No.3, pp.984–997.

Pawlak M. and Greblicki W. (1985): *Pointwise consistency of the Hermite series density estimate.* — Stat. and Prob. Lett., v.3, pp.65–69.

Priestley M.B. and Chao M.T. (1972): *Non–parametric function fitting.* — J. Roy. Statist. Soc., v.B34, pp.385–392.

Rafajłowicz E. (1984a): *Nonparametric algorithm for input signals identification in static distributed parameter systems.* — IEEE Trans. Automatic Control, v.AC–29, pp.631–633.

Rafajłowicz E. (1984b): *Nonparametric algorithm for identification of weakly nonlinear static distributed–parameter systems.* — Syst. and Control Lett., v.4, pp.105–109.

Rafajłowicz E. (1987): *Nonparametric orthogonal series estimators of regression: A class attaining the optimal convergence rate in* L_2. — Stat. and Prob. Lett., v.5, pp.219–224.

Rafajłowicz E. (1992): *Nonparametric identification when errors–in–variables are present.* — Prep. 1992. Instit. of Engn. Cybernetics, Technical University of Wrocław, Poland, (Int. Systems Sci. – in press).

Rutkowski L. (1982): *On system identification by nonparametric function fitting.* — IEEE Trans. Automatic Control, v.AC–27, pp.225–227.

Schuster E. and Yakowitz S. (1979): *Contributions to the theory of nonparametric regression with application to system identification.* — Ann. Statist., v.10, pp.1040–1053.

Silverman B.W. (1982): *Kernel density estimation using the Fast Fourier Transform.* — Applied Statistics, v.31, pp.93–99.

Storn R. (1986): *Fast algorithms for the discrete Hartley transform.* — Archiv für Elektronik und Ubertragungstechnik, v.40, pp.233–240.

Received January 4, 1993
Revised December 15, 1993