Efficient implementation of the Gaussian kernel algorithm in estimating invariants and noise level from noisy time series data

Dejin Yu,[1,*] Michael Small,[1] Robert G. Harrison,[1] and Cees Diks[2]
[1]*Department of Physics, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, United Kingdom*[†]
[2]*CeNDEF, Department of Economics, University of Amsterdam, Roetersstraat 11, NL-1018 WB Amsterdam, The Netherlands*
(Received 2 June 1999)

We describe an efficient algorithm which computes the Gaussian kernel correlation integral from noisy time series; this is subsequently used to estimate the underlying correlation dimension and noise level in the noisy data. The algorithm first decomposes the integral core into two separate calculations, reducing computing time from $O(N^2\times N_b)$ to $O(N^2+N_b^2)$. With other further improvements, this algorithm can speed up the calculation of the Gaussian kernel correlation integral by a factor of $\gamma\sim(2-10)N_b$. We use typical examples to demonstrate the use of the improved Gaussian kernel algorithm.

PACS number(s): 05.45.Tp, 02.50.−r, 05.45.Ac, 05.45.Jn

I. INTRODUCTION

Estimation of dimensions, entropies, and Lyapunov exponents has become a standard method of characterizing the complex temporal behavior of a chaotic trajectory from measured time series data [1]. Of these, the most widely used is the correlation dimension. This is largely due to the fact that Grassberger and Procaccia found a simple algorithm to compute the correlation integral and the correlation dimension is believed to be a more relevant measure of the attractor than the other dimension quantities [2]. Many fast and efficient algorithms have since been developed for calculation of the correlation dimension and found to be successful in characterizing dynamics from *clean* time series data [3,4].

When applied to experimental data, the dimension algorithms have limitations since all recorded data are to some extent corrupted by *noise*, which masks the scaling region at small scales. It has been shown that a 2% noise is serious enough to prevent accurate estimation [5]. From an applied point of view, the problem of characterizing nonlinear time series in the presence of noise is therefore nontrivial and of practical significance. Considerable effort has been given to understanding the influence of noise on dimension measurements and exploring new scaling laws; see [6–15]. A comparison of methods dealing with the influence of Gaussian noise on the correlation dimension can be found in [16]. In addition, a Grassberger-Procaccia-type algorithm for estimating entropy has been developed to characterize experimental data [17].

In this paper, we address the estimation of correlation exponents and noise level in the presence of Gaussian noise in time series data. In the presence of noise, the dimension and entropy are defined as those invariants of the underlying clean time series [12,14,16]. Particular attention is paid to efficient implementation of the Gaussian kernel algorithm (GKA) developed by Diks [12]. We first decompose the integral core into two separate calculations, which reduces computing time from $O(N^2\times N_b)$ to $O(N^2+N_b^2)$, where N and N_b are the length of reconstructed state vectors and the number of computed bandwidth values, respectively. With other further improvements, this algorithm can speed up the calculation of the Gaussian kernel correlation integral by a factor of $\gamma\sim(2-10)N_b$. We also present robust methods used for nonlinear least squares fitting in extracting correlation exponents and noise level. Armed with the improved algorithms, we find that the GKA provides a reasonable estimate of the correlation dimension and noise level even when the contaminated noise is as high as 50% of the signal content.

Our problem is formulated as follows. Suppose that we have a scalar time series $\{s_i:i=1,2,...,N_s\}$ sampled at equally spaced times $t_i=i\Delta t$, where Δt is the sampling time interval. The data is assumed to be corrupted by Gaussian noise. Our aim is to measure two dynamical invariants—correlation dimension and entropy—and estimate the noise level in the time series.

Following Takens [18], the underlying attractor can be reconstructed using delay co-ordinates. The reconstruction embeds the measured time series $\{s_i\}$ in an m-dimensional Euclidean space to create $N=N_s-(m-1)\tau$ delay state vectors $\{\mathbf{x}_i:i=1,2,...,N\}$ in terms of

$$\mathbf{x}_i=[s_i,s_{i+\tau},s_{i+2\tau},\ldots,s_{i+(m-1)\tau}]^T, \tag{1}$$

where τ is an integer, referred to as a time lag (the delay time is $\delta t=\tau\Delta t$), and m is called the embedding dimension.

This paper is organized as follows. After this introduction, we describe the Gaussian kernel algorithm and its direct implementation in Sec. II. We explore the efficient implementation and simplified calculation of the Gaussian kernel correlation integral in Sec. III. Section IV is devoted to technical considerations and further improvements of the GKA. Numerical tests are presented by way of examples in Sec. V. Finally, we conclude this work in Sec. VI.

II. GAUSSIAN KERNEL ALGORITHM

A. Theoretical background

The essence of the Gaussian kernel algorithm in estimating correlation exponents and noise level is summarized as

*Author to whom correspondence should be addressed. Electronic address: phydjy@phy.hw.ac.uk
[†]URL: http://www.phy.hw.ac.uk/resrev/ndos/index.html

follows [12]. (1) The Gaussian kernel correlation integral $T_m(h)$ for the noise-free case scales as

$$T_m(h)=\int d\mathbf{x}\,\rho_m(\mathbf{x})\int d\mathbf{y}\,\rho_m(\mathbf{y})e^{-\|\mathbf{x}-\mathbf{y}\|^2/4h^2}$$
$$\sim e^{-mK\delta t}\left(\frac{h}{\sqrt{m}}\right)^D \quad \text{for} \quad h\to 0,\ m\to\infty, \tag{2}$$

where D and K are the correlation dimension and entropy, h is referred to as the bandwidth, and $\rho_m(\mathbf{x})$ is the distribution function. The scaling behavior $T_m(h)\sim e^{-mK\delta t}h^D$ in Eq. (2) was first justified by Ghez *et al.* [19] and latter by Diks [12] with inclusion of the factor $(1/\sqrt{m})^D$, in which the m dependence was originally introduced by Frank *et al.* [20] to improve convergence of K. (2) In the presence of Gaussian noise, the distribution function $\tilde{\rho}_m(\mathbf{y})$ can be expressed in terms of a convolution between the underlying noise-free distribution function $\rho_m(\mathbf{x})$ and a normalized Gaussian distribution function with standard deviation σ [12,21], i.e.,

$$\tilde{\rho}_m(\mathbf{y})=\int d\mathbf{x}\,\rho_m(\mathbf{x})\rho_m^g(\|\mathbf{y}-\mathbf{x}\|)$$
$$=\frac{1}{(\sigma\sqrt{2\pi})^m}\int d\mathbf{x}\,\rho_m(\mathbf{x})e^{-\|\mathbf{y}-\mathbf{x}\|^2/2\sigma^2}, \tag{3}$$

where

$$\rho_m^g(\|\mathbf{y}-\mathbf{x}\|)=\frac{1}{(\sigma\sqrt{2\pi})^m}e^{-\|\mathbf{y}-\mathbf{x}\|^2/2\sigma^2}, \tag{4}$$

accounts for noise effects in the m-dimensional space. $\|\cdot\|$ is the Euclidean (L_2) norm.

Under conditions (2) and (3), the Gaussian kernel correlation integral $\tilde{T}_m(h)$ in the presence of Gaussian noise and the corresponding scaling law become [12]

$$\tilde{T}_m(h)=\int d\mathbf{x}\,\tilde{\rho}_m(\mathbf{x})\int d\mathbf{y}\,\tilde{\rho}_m(\mathbf{y})e^{-\|\mathbf{x}-\mathbf{y}\|^2/4h^2} \tag{5}$$
$$=\left(\frac{h^2}{h^2+\sigma^2}\right)^{m/2}\int d\mathbf{x}\,\rho_m(\mathbf{x})\int d\mathbf{y}\,\rho_m(\mathbf{y})$$
$$\times e^{-\|\mathbf{x}-\mathbf{y}\|^2/4(h^2+\sigma^2)}$$
$$=\phi\left(\frac{h^2}{h^2+\sigma^2}\right)^{m/2}e^{-mK\,\delta t}\left(\frac{h^2+\sigma^2}{m}\right)^{D/2} \tag{6}$$
$$\text{for}\ \ \sqrt{h^2+\sigma^2}\to 0,\ \ m\to\infty,$$

where ϕ is a normalized constant.

In Eq. (6), D and K are the two invariants to be estimated, and the parameter σ is referred to as the noise level, defined as

$$\sigma=\frac{\sigma_n}{\sigma_s}=\frac{\sigma_n}{\sqrt{\sigma_c^2+\sigma_n^2}}, \tag{7}$$

where σ_s, σ_c, and σ_n are standard deviations of the input noisy signal $\{s_i\}$, underlying clean component $\{c_i\}$, and Gaussian noise part $\{n_i\}$. In the total signal $s_i=c_i+n_i$, we have assumed $\{c_i\}$ and $\{n_i\}$ to be statistically independent, i.e., $\overline{s}=\overline{c}+\overline{n}$ and $\sigma_s^2=\sigma_c^2+\sigma_n^2$, where $\overline{s}$, $\overline{c}$, and $\overline{n}$ are the means of $\{s_i\}$, $\{c_i\}$, and $\{n_i\}$, respectively.

B. Direct implementation of the GKA

The numerical implementation of the Gaussian kernel algorithm requires the transformation of the input time series data $\{s_i : i=1,2,\ldots,N_s\}$ into a new time series $\{v_i : i=1,2,\ldots,N_s\}$ according to

$$v_i=\frac{s_i-\overline{s}}{\sigma_s}. \tag{8}$$

Under the transformation (8), the noise effect is described by the distribution function (4) and the standard deviation of the noise part is σ in Eq. (7). Accordingly, the delay state vectors are reconstructed by replacing $\{s_i\}$ with $\{v_i\}$ in Eq. (1).

In the case of discrete sampling, we assume vector points on the attractor to be dynamically independently distributed according to $\tilde{\rho}_m(\mathbf{x})$ and use an average over delay vectors to replace the integrals over the vector distributions in Eq. (5). Consequently, $\tilde{T}_m(h)$ can be computed by [12]

$$\hat{T}_m(h)=\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j\neq i}^{N}e^{-\|\mathbf{x}_i-\mathbf{x}_j\|^2/4h^2}. \tag{9}$$

In practice, $\hat{T}_m(h)$ is calculated at a series of discrete bandwidth values h_k $(k=0,1,2,\ldots,N_b)$. The parameters D, K, and σ are then extracted through fitting the scaling relation (6) to the Gaussian kernel correlation sum $\hat{T}_m(h)$ computed from Eq. (9). One can see that such a direct implementation of the GKA has a computational complexity $O(N^2\times N_b)$.

C. Nonlinear fitting

After computing the Gaussian kernel correlation sum, the parameters D, K, and σ in Eq. (6) can be extracted using nonlinear least squares fitting [22]. Here we exemplify the GKA described above; a robust fitting procedure will be presented in Sec. IV.

Figure 1 illustrates the fitted D, K, and σ as a function of embedding dimension. The clean data are generated by the standard Hénon map [23]. The noisy data are produced by adding 5% Gaussian noise to the output of the first variable. In calculating $\hat{T}_m(h)$ from Eq. (9), we use the following parameters: $N=5000$, $N_{\text{ref}}=500$, $N_b=100$, $\log_2(\epsilon_l)=-10$, and $m=1,2,\ldots,10$. We choose ϵ_u to be equal to the attractor diameter. A definition and discussion of these parameters will be detailed below. As seen in Fig. 1, the fitted correlation exponents D and K and noise level σ converge to their true values when the embedding dimension is increased beyond $m=3$. We note that a convergence in D and σ is readily achieved while K exhibits fluctuations, since saturation should only appear in principle as $m\to\infty$. In this example, the average values of the correlation exponents and noise level, taken over embedding dimensions $m=3-10$, are ob-

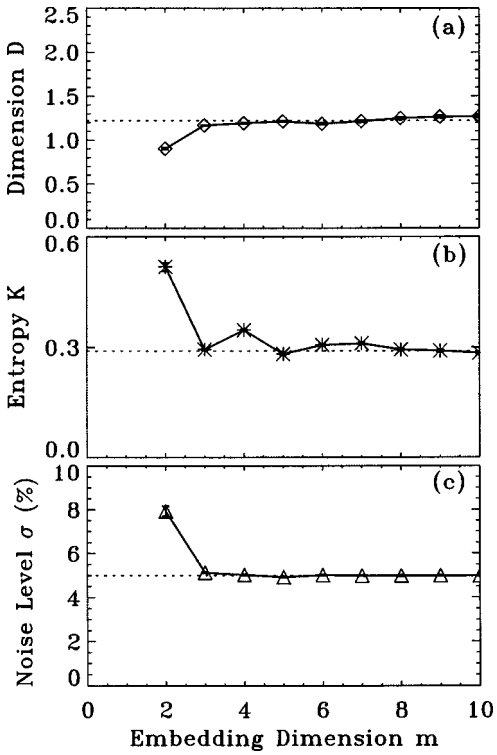

FIG. 1. Estimation of (a) correlation dimension D, (b) correlation entropy K, and (c) noise level σ using Hénon map data. The dotted lines give their ''true'' values $D_{\text{true}}=1.22$, $K_{\text{true}}=0.29$ [18], and $\sigma_{\text{in}}=5\%$. $h_c=0.2$ is used; see below for its definition.

tained as $\overline{D}=1.227\pm0.011$, $\overline{K}=0.301\pm0.003$, and $\overline{\sigma}=4.984\pm0.028$.

III. SIMPLIFIED ALGORITHM

We notice that $\hat{T}_m(h)$ in Eq. (9) is essentially an average of the function $e^{-\|\mathbf{x}_i-\mathbf{x}_j\|^2/4h^2}$ over all pairs (i,j) with an equal unitary weight. For a given bandwidth h_k, this algorithm must perform $(N-1)N$ operations to scan all distances. For another bandwidth $h_{k'}$ $(k'\neq k)$, all the same distances are revisited again. Such a distance scanning process is repeated N_b+1 times, leading to $O(N^2\times N_b)$ operations, which wastes a large amount of computing time. The algorithm can be simplified by eliminating repetition in distance scanning. Since the same distances will be used for all bandwidth values we can calculate a binned interpoint distance distribution $\mathcal{C}_m(\epsilon_k)$ once and then take the average by summing over indices $(k=0,1,2,...,N_b)$ of all binned interpoint distances, instead of pairs (i,j). As a result, the Gaussian kernel correlation sum is approximated by averaging $e^{-\epsilon_k^2/4h^2}$ over all interpoint distances with the weight function $\mathcal{C}_m(\epsilon_k)$. Under these considerations, Eq. (9) can be rewritten as

$$\hat{T}_m(h)=\frac{1}{N(N-1)}\sum_{k=0}^{N_b}\left(\sum_{i=1}^{N}\sum_{j\neq i}^{N}\theta_2(\epsilon_{ij},\epsilon_k,\epsilon_{k+1})\right)e^{-\epsilon_k^2/4h^2}, \tag{10}$$

where $\epsilon_{ij}=\|\mathbf{x}_i-\mathbf{x}_j\|$, and $k=0$ and N_b correspond to minimum (ϵ_l) and maximum (ϵ_u) distances, respectively. Here we assume that $\epsilon_l=0$ and ϵ_u equals the diameter of the attractor. The summand $\theta_2(\epsilon_{ij},\epsilon_k,\epsilon_{k+1})$ is a double step function with $\epsilon_k<\epsilon_{k+1}$, defined through the Heaviside step function $\theta(\cdot)$ by $\theta_2(\epsilon,\epsilon_1,\epsilon_2)=\theta(\epsilon-\epsilon_1)\theta(\epsilon_2-\epsilon)$, that is,

$$\theta_2(\epsilon,\epsilon_1,\epsilon_2)=\begin{cases}1 & \text{if } \epsilon_1\leqslant\epsilon<\epsilon_2\\ 0 & \text{otherwise.}\end{cases} \tag{11}$$

Noticing that the term $\Sigma_{i=1}^{N}\Sigma_{j\neq i}^{N}\theta_2(\epsilon_{ij},\epsilon_k,\epsilon_{k+1})$ in Eq. (10) depends on the index k only, this allows us to decompose Eq. (10) into two separate steps: (1) Calculation of the binned interpoint distance distribution $\mathcal{C}_m(\epsilon_k)$, given by

$$\begin{aligned}\mathcal{C}_m(\epsilon_k)&=\{\text{Number of pairs } (i,j) \text{ whose distances}\\ &\quad \text{satisfy } \epsilon_{ij}=\|\mathbf{x}_i-\mathbf{x}_j\|\in[\epsilon_k,\epsilon_{k+1})\}\\ &=\sum_{i=1}^{N}\sum_{j\neq i}^{N}\theta_2(\|\mathbf{x}_i-\mathbf{x}_j\|,\epsilon_k,\epsilon_{k+1});\end{aligned} \tag{12}$$

(2) calculation of the Gaussian kernel correlation sum in terms of

$$\hat{T}_m(h)=\frac{1}{N(N-1)}\sum_{k=0}^{N_b}\mathcal{C}_m(\epsilon_k)e^{-\epsilon_k^2/4h^2}. \tag{13}$$

In Eq. (12), we choose the binned interpoint distances ϵ_k in a way such that they are equidistant in a logarithmic scale. This choice has its numerical advantage since it gives a high resolution at the small scale where the interpoint distance distributions are more relevant to our calculations. In addition, the bandwidth values h_k are determined in the same way as ϵ_k.

Since $\mathcal{C}_m(\epsilon_k)$ can be readily computed, the calculation of $\hat{T}_m(h)$ based on Eq. (13) becomes simple. Equations (12) and (13) jointly lead to a computational complexity $O(N^2+N_b^2)$. By comparison with the direct calculation based on Eq. (9), this reduces the computing time by a factor of $\gamma\simeq N_b[1-(N_b/N)^2]\sim N_b$.

The relation between $\mathcal{C}_m(\epsilon_k)$ and the Grassberger-Procaccia (GP) correlation integral $\mathcal{C}_m(\epsilon)$ in the Euclidean norm is

$$\hat{\mathcal{C}}_m(\epsilon)=\frac{1}{N(N-1)}\sum_{k=0}^{N_\epsilon}\mathcal{C}_m(\epsilon_k), \tag{14}$$

where the integer N_ϵ corresponds to $\epsilon=\epsilon_{N_\epsilon}$. A further understanding of Eq. (13) can be gained from an alternative expression of the Gaussian kernel correlation integral. To do so, we define the Gaussian kernel correlation integral $T_m(h)$ in terms of interpoint distance distribution density $\eta_m(\epsilon)$ and Gaussian kernel function $w(\epsilon/h)=e^{-\epsilon^2/4h^2}$ as [12]

$$T_m(h)=\int_0^\infty\eta_m(\epsilon)w\left(\frac{\epsilon}{h}\right)d\epsilon, \tag{15}$$

where $\eta_m(\epsilon)$ is given by

$$\eta_m(\epsilon)=\frac{dC_m(\epsilon)}{d\epsilon}, \tag{16}$$

where $C_m(\epsilon)$ is the GP correlation integral with the hard kernel function $w(r/\epsilon)=\theta(\epsilon-r)$. Substituting Eq. (16) into Eq. (15), the Gaussian kernel correlation integral $T_m(h)$ is expressed as

$$T_m(h)=\int_0^\infty e^{-\epsilon^2/4h^2}dC_m(\epsilon). \tag{17}$$

This is the integral form of Eq. (13).

If one performs a partial integration in Eq. (17), $T_m(h)$ can be expressed in the form of

$$T_m(h)=\frac{1}{2h^2}\int_0^\infty e^{-\epsilon^2/4h^2}C_m(\epsilon)\epsilon\, d\epsilon. \tag{18}$$

In previous work, this formula was used to estimate $\hat{T}_m(h)$ [24,25].

IV. COMPUTATIONAL CONSIDERATIONS

A. Efficient binning

The use of the simplified algorithm given by Eqs. (12) and (13) can speed up the calculation of $\hat{T}_m(h)$. For instance, for $N=10\,000$ and $N_b=200$, this gives rise to a gain factor $\gamma\sim 200$. We note that most of the time is consumed in computing the binned interpoint distance distribution $\mathcal{C}_m(\epsilon_k)$ in Eq. (12), which takes $N(N-1)$ operations. Apart from the remarkable advance by using Eqs. (12) and (13), a further reduction of computing time is achievable. In this algorithm, we adopt three improvements.

(1) A set of $N_{\rm ref}$ representative points randomly chosen on the attractor are used for the sum over i to replace the original N points in Eq. (12). This reduces $O[N(N-1)]$ to $O[(N-1)\times N_{\rm ref}]$. Usually, we use $N_{\rm ref}\sim N/10$.

(2) We observe that large distances have a negligible contribution to the Gaussian kernel correlation sum $\hat{T}_m(h)$, as in the GP algorithm [26]. Thus we set an upper limit of the correlation distance ϵ_u to be smaller than the attractor diameter, above which the binning is not done. This can save computing time by a factor of 10^0-10^1, depending on the value of ϵ_u.

(3) A recursive version is used. Since we compute $\mathcal{C}_m(\epsilon_k)$ for a series of consecutive embedding dimensions $m=1,2,...,M$ and in the L_2 norm $\epsilon_k^2(m+1)=\epsilon_k^2(m)+(v_{i+m\tau}-v_{j+m\tau})^2$, the distance in the m dimension is successively used in the $(m+1)$ dimension.

These three binning methods are found to be very efficient and speed up the calculation by a factor $\sim 10^1-10^2$ at least.

B. Robust fitting procedure

Direct fitting based on Eq. (6) does not make sense. We find that though D and σ can be extracted with a high precision, there is an uncertainty between ϕ and K. This is because ϕ and $e^{-mK\delta t}$ are not independent for a fixed m but their product $\beta=\phi e^{-mK\delta t}$ is a real independent parameter. In the nonlinear fitting process, as long as a stable β is obtained, the program will return values of ϕ and K. But such ϕ and K are in general arbitrary.

In the following, we introduce a three-step fitting method, which is found to be quite robust. The described method can extract D and σ to a high precision and obtain a fair estimate of K. The procedure is detailed as follows.

(1) Fitting D, σ, and β. We use the relation (6) between $\tilde{T}_m(h)$ and h to fit D, σ, and β. The values of D and σ obtained will be used in the next step. Letting $\beta=\phi e^{-mK\delta t}$ yields a model equation

$$y(h)=\tilde{T}_m(h)=\beta h^m m^{-D/2}(h^2+\sigma^2)^{(D-m)/2}. \tag{19}$$

(2) Fitting K and deriving ϕ. We use $y(h)=\tilde{T}_{m+1}(h)/\tilde{T}_m(h)$ to eliminate ϕ, leading to a model

$$y(h)=he^{-K\delta t}\left(\frac{m}{m+1}\right)^{D/2}(h^2+\sigma^2)^{-1/2}. \tag{20}$$

We fit K with D and σ fixed. Then ϕ can be derived from $\phi=\beta e^{mK\delta t}$, where β is given in the last step.

(3) Refitting D, K, and σ. We use the original relation (6) to fit D, K, and σ with the fixed ϕ derived in the last step. D, K, and σ obtained in the last two steps are used as trial values.

Note that the use of Eq. (20) requires computing the Gaussian kernel correlation integrals for two consecutive embedding dimensions. In practice, we compute $\hat{T}_m(h)$ for $m=1,2,...,M$ and extract D, K, and σ as functions of m. Thus the fitting procedure described above is performed for consecutive embedding dimensions. Furthermore, the standard deviations of the correlation integral $\hat{T}_m(h)$ are used as weights in the fitting procedure [12]. This can greatly reduce fitting errors by comparison with a calculation using equal weights. The latter results in large deviations at higher embedding dimensions.

C. Choice of cutoff bandwidth h_c

There exists an unsolved technical problem in the nonlinear fitting procedure described above, that is, determination of the largest bandwidth h_c to be used within the scaling region. This is a difficult task in the presence of noise. The reason is simple: noise masks the scaling region at the small scale. Intuitively, h_c should be small enough to keep fitting in the scaling region presented by the underlying noise-free dynamics. But h_c cannot be so small as to prevent extracting D. On the other hand, h_c should not be so large so as to exceed the scaling region. In the general case, we suggest a choice of $h_c\geqslant 3\sigma$.

Numerical simulations show that the fitted σ is insensitive to the choice of cutoff bandwidth h_c. Thus, an iteration scheme can be adopted. In the first run, a trial h_c, for example, $h_c=0.5$, is used so as to obtain a fitted σ. The value of 3σ is in turn adopted as the cutoff bandwidth h_c.

Caution should be used when the noise level is either high or low. For example, a 30% noise level will give rise to $h_c=0.9$. This value is close to the upper limit of the linear scaling region for most chaotic attractors (tested). On the other hand, when the noise level is low, say, below 3%, the value of 3σ is to small to be used for h_c. In practice, we recommend fitting D, K, and σ for a series of cutoff bandwidth values starting from a small value and from this iden-

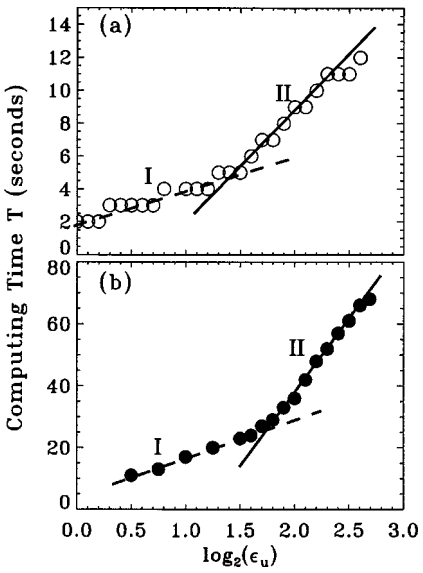

FIG. 2. Computing time as a function of the upper limit of scale ϵ_u, given by $\log_2(\epsilon_u)$, for $\sigma_{\text{in}}=10\%$. (a) Hénon map, $N=5000$ and $m=1,2,\ldots,10$ and (b) Lorenz chaos, $N=10\,000$ and $m=1,2,\ldots,15$. Two linear regions I and II are indicated by the dashed and solid lines, respectively. Note that the appearance of steps in (a) is because we use one second as the time unit.

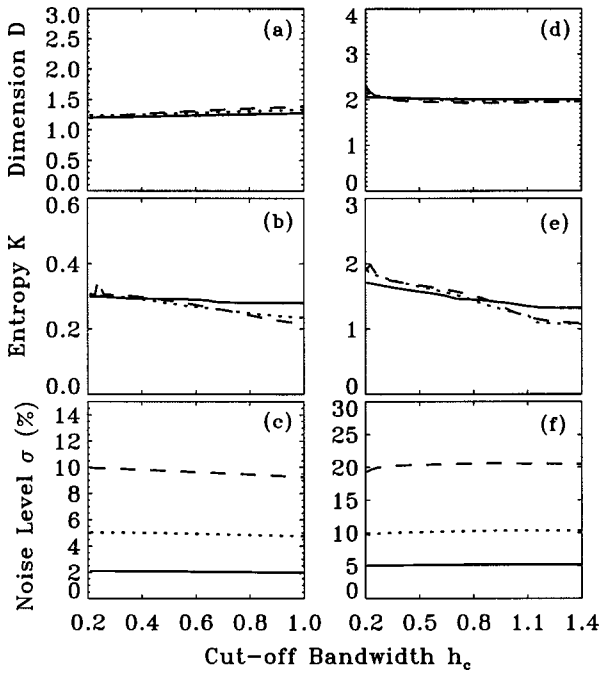

FIG. 3. Fitted parameters D, K, and σ as a function of the cutoff bandwidth h_c in the Hénon map (left column) and Lorenz model (right column). The solid, dotted, and dashed lines correspond to three input noise levels, $\sigma_{\text{in}}=2\%$, 5%, and 10% for the Hénon map and $\sigma_{\text{in}}=5\%$, 10%, and 20% for the Lorenz model.

tifying the saturation region of the scaling parameters as a function of h_c. A good cutoff bandwidth is thus chosen as a value below the deformation region, but as large as possible to reduce the fitting errors.

V. EXAMPLES AND TESTS

In this section, we demonstrate the performance of the simplified Gaussian kernel algorithm presented in Secs. III and IV by applying it to some well-known chaotic systems, namely, the Hénon map and Lorenz chaos. First the clean data are generated from the standard models [23]. The noisy time series are then prepared by adding a Gaussian noise component with a standard deviation σ_{in} to the noise-free data. In all cases, we fix $N_b=200$, $\log_2(\epsilon_l)=-10$. The number of delay vectors N, the embedding dimension m, and the upper limit of scale ϵ_u are left to be adjustable. The computing time to be used below is the full CPU time of the program and all tests are done on an ULTRA 10 SUN Workstation. In addition, the recursive algorithm is adopted for both direct and improved Gaussian kernel algorithms and the number of reference points is set to be $N_{\text{ref}}=N/10$.

A. Speed with respect to direct implementation

In order to make a comparison with the direct implementation of the GKA in Secs. II B and II C we first set the upper limit of the correlation distance ϵ_u to be equal to the diameter of the attractor. In this case, all interpoint distances are binned in order to test the improvement of the simplified algorithm Eqs. (12) and (13) with respect to the direct calculation of Eq. (9). Two groups of controlled tests have been conducted using $N=10\,000$, $m=1,2,\ldots,15$ for the Lorenz chaos and $N=5000$, $m=1,2,\ldots,10$ for the Hénon map. Not surprisingly, the direct calculation of $\hat{T}_m(h)$ takes quite a long time; for the former 29 569 seconds (>8 hours) are required while for the latter 4679 seconds are needed. By contrast, the simplified algorithm is much faster than the direct implementation of the GKA and, as expected, gives a speeding-up factor $\gamma=T_{\text{direct}}/T_{\text{simplified}}\simeq 400$. This value is twice N_b since the exponential operation is time consuming in Eq. (9). It follows that the improvement by using Eqs. (12) and (13) is indeed significant.

We next examine the dependence of the computing time on the upper limit of the correlation distance ϵ_u. Figure 2 depicts the computing time for the Hénon map and Lorenz system. Our tests are terminated at the smallest scale ϵ_0, below which the measured D and σ deviate from their corresponding true values by 10%, in Fig. 2 $\log_2(\epsilon_0)=0$ for the Hénon map and $\log_2(\epsilon_0)=0.5$ for the Lorenz system. By comparison with the direct implementation, a total speeding-up factor $\gamma\simeq 2500$ has been achieved at ϵ_0. Moreover, we see that the use of ϵ_u alone leads to an improvement of speed by 1–6 times. A further feature is observed in Fig. 2: that there is a turning point which separates two linear regions. Below this point, the computing time shows a relatively weak dependence on $\log_2(\epsilon_u)$, while in the second region decreasing ϵ_u gives rise to a significant reduction of computing time. This leads us to suggest that ϵ_u be set within the linear region I.

B. Cutoff bandwidth h_c

Figure 3 shows the average values of D, K, and σ on increasing the cutoff bandwidth h_c in the Hénon map and Lorenz system. Plotted are average values taken over $m=3-6$ for the Hénon map ($N=5000$) and over $m=6-10$ for the Lorenz model ($N=10\,000$). These represent mea-

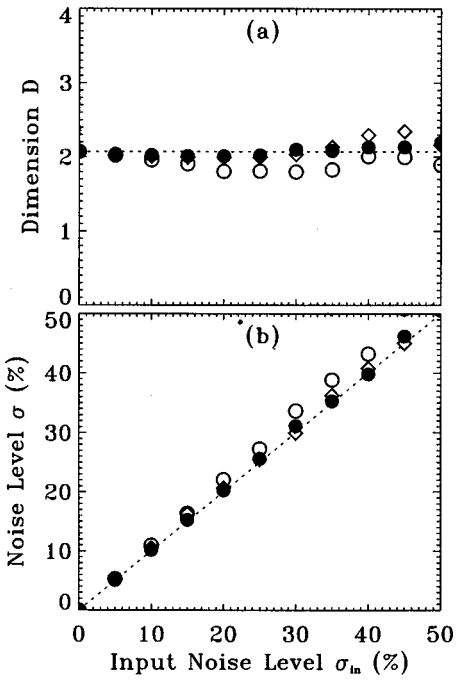

FIG. 4. Measurements of (a) correlation dimension D and (b) noise level σ using the Gaussian kernel algorithm. ●, ○, and ◇ correspond to Gaussian, uniform, and a combination of uniform with Gaussian noise, respectively. The dotted lines give their true values. Calculations are performed based on averages over embedding dimensions $m=8-15$.

surements in the discrete mapping and continuous flow systems. For both types of dynamics, it is obvious that there is a broad scaling region where both the correlation dimension D and noise level σ fitted are saturated around their true values and from this h_c can be chosen. Notice that the correlation entropy K exhibits a sensitive dependence on the cutoff bandwidth h_c for $\sigma_{\text{in}} \geqslant 5\%$, decreasing with h_c, and shows considerable difference for different noise levels at the large bandwidth. The latter indicates that it is more difficult to measure K when the noise level is high. Further, K should be estimated when m is sufficiently large [20]. The tests given in Fig. 3 do not meet this limiting condition.

C. Precision against noise level

We examine the estimated D and σ as a function of σ_{in} for different types of independent and identically distributed (IID) noise: Gaussian, uniform, and a combination of the Gaussian with uniform IID noise with each being 50%. The Lorenz system is used as a representative example. The input noise level σ_{in} is set from 0% to 50%. Numerical results are shown in Fig. 4. As can be seen, the measurements of correlation dimension D and noise level σ are in good agreement with their true values for pure Gaussian IID noise up to $\sigma_{\text{in}}=50\%$. The results also show good consistency for the combined noise when the noise level is $\sigma_{\text{in}}<40\%$. This is what is expected since the GKA is established under an assumption of Gaussian noise. By contrast, the correlation dimension is underestimated and exhibits a large deviation in the case of the pure uniform IID noise for $\sigma_{\text{in}} \geqslant 20\%$, but the Gaussian kernel algorithm still provides a fair estimate of D and σ, as indicated by circles in Fig. 4, in particular when $\sigma_{\text{in}}<20\%$. These results show that the Gaussian kernel algorithm is a reliable tool for measuring the correlation dimension and noise level from noisy time series and works well for different types of noise sources when the noise level is below 20%. This nice property provides a basis for characterizing experimental data that are believed to contain both types of noise.

VI. CONCLUDING REMARKS

The main point in this paper is to develop an efficient algorithm to simplify the calculation of the Gaussian kernel correlation integral. Numerical simulations show that our improved algorithm is computationally efficient and speeds up the calculation by a factor $\gamma \sim (2-10)N_b$ by comparison with direct implementation. We hope that this improved algorithm meets broad computational needs and can find widespread applications in characterizing experimental data.

We find that the GKA not only works for pure Gaussian noise, but is also applicable to other types of noise provided that the underlying noise level is relatively low, say, below 20%, such as a combination of Gaussian with uniform IID noise and even uniformly distributed noise. This property is of practical importance since the noise type is usually unknown *a priori* in experimental time series data. More generally, any filtering and the presence of multiple noise sources will turn the noise into an approximately Gaussian distribution according to the central limit theorem. Recently, the GKA has been successfully used to characterize electrocardiograph data of ventricular fibrillation [27].

The simplified GKA provides a reliable estimation of the correlation dimension and noise level. However, it seems difficult to extract exactly the correlation entropy K from the non-linear fitting procedure described in Sec. IV B. This is understandable because we have assumed ϕ to be a constant in Eq. (6), which in turn leads to Eq. (20). This is true only when $\sqrt{h^2+\sigma^2}$ is small. Therefore, for fixed σ, the deviation of K will increase with the bandwidth h_c. On the other hand, the correlation entropy is asymptotically obtained only as $m \to \infty$ according to its definition.

Note that, in writing Eq. (9), we assume that the delay vectors are independently distributed on the attractor according to the distribution $\tilde{\rho}_m(\mathbf{x})$. This is not always true. For data generated from continuous dynamical systems, serial temporal correlation must be ruled out [28]. Finally, though the simplified algorithm is much faster than the direct implementation of the GKA, it is not suitable for too long time series since the algorithm for $\mathcal{C}_m(\epsilon_k)$ is still $\mathcal{O}(N \times N_{\text{ref}})$. Nevertheless, the improved GKA is fast enough for $N \sim 10^4-10^5$ long on most current workstations and personal computers.

Source codes (FORTRAN 77) can be obtained on request from the first author.

ACKNOWLEDGMENT

This work was funded by a Research Development Grant by the Scottish Higher Education Funding Council (SHEFC), Grant No. RDG/078.

[1] *Measures of Complexity and Chaos*, edited by N. B. Abraham, A. M. Albano, A. Passamante, and P. E. Rapp (Plenum, New York, 1989).
[2] P. Grassberger and I. Procaccia, Phys. Rev. Lett. **50**, 346 (1983); Physica D **9**, 189 (1983).
[3] *Dimensions and Entropies in Chaotic Systems*, edited by G. Mayer-Kress (Springer-Verlag, Berlin, 1986).
[4] J. Theiler, J. Opt. Soc. Am. A **7**, 1055 (1990).
[5] T. Schreiber and H. Kantz, in *Predictability of Complex Dynamical Systems*, edited by Y. A. Kravtsov and J. B. Kadtke (Springer, New York, 1996).
[6] E. Ott, E. D. Yorke, and J. A. Yorke, Physica D **16**, 62 (1985).
[7] M. Möller, W. Lange, F. Mitschke, N. B. Abraham, and U. Hübner, Phys. Lett. A **138**, 176 (1989).
[8] R. L. Smith, J. R. Stat. Soc., Ser. B Methodol. **54**, 329 (1992).
[9] G. G. Szpiro, Physica D **65**, 289 (1993).
[10] T. Schreiber, Phys. Rev. E **48**, R13 (1993).
[11] J. C. Schouten, F. Takens, and C. M. van den Bleek, Phys. Rev. E **50**, 1851 (1994).
[12] C. Diks, Phys. Rev. E **53**, R4263 (1996).
[13] H. Oltmans and P. J. T. Verheijen, Phys. Rev. E **56**, 1160 (1997).
[14] D. Kugiumtzis, Int. J. Bifurcation Chaos Appl. Sci. Eng. **7**, 1283 (1997).
[15] J. Argyris, I. Anderadis, G. Pavlos, and M. Athanasiou, Chaos, Solitons and Fractals **9**, 343 (1998).
[16] T. Schreiber, Phys. Rev. E **56**, 274 (1997).
[17] J. G. Caputo and P. Atten, Phys. Rev. A **35**, 1311 (1987).
[18] F. Takens, in *Dynamical Systems and Turbulence, Warwick, 1980*, edited by D. A. Rand and L. S. Young (Springer-Verlag, New York, 1981), Vol. 898, pp. 366–381.
[19] J. M. Ghez and S. Vaienti, Nonlinearity **5**, 777 (1992); J. M. Ghez, E. Orlandini, M. C. Tesi, and S. Vaienti, Physica D **63**, 282 (1993).
[20] M. Frank, H.-R. Blank, J. Heindl, M. Kaltenhäuser, H. Köchner, W. Kreische, N. Müller, S. Poscher, R. Sporer, and T. Wagner, Physica D **65**, 359 (1993).
[21] M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, Physica D **51**, 52 (1991).
[22] W. H. Press, S. A. Teukolski, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN*, 2nd ed. (Cambridge University Press, Cambridge, 1992).
[23] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, Physica D **16**, 285 (1985).
[24] T. Schreiber, Phys. Rep. **308**, 1 (1999).
[25] A large error in estimating D, K, and σ may result. This is because Eq. (17) is derived under the condition that $\epsilon_l=0$ and $\epsilon_u=\infty$. This is not the case when $\hat{T}_m(h)$ is numerically computed.
[26] J. Theiler, Phys. Rev. A **36**, 4456 (1987).
[27] Dejin Yu, M. Small, R. G. Harrison, C. Robertson, G. Clegg, M. Holzer, and F. Sterz, Phys. Lett. A **265**, 68 (2000).
[28] J. Theiler, Phys. Rev. A **34**, 2427 (1986).