Chapter 9

Selecting Embedding Dimension for Phase Space Reconstruction

Here, we focus on one of the popular methods used for estimating the minimally necessary embedding dimension called *false nearest neighbors* (FNN) [45]. There are other methods that utilize convergence in the estimates of some nonlinear measures or metrics [5], but they are either more complex or cumbersome in the applications. The idea behind the FNN [45] method is based on the uniqueness property of a dynamical system’s phase-space trajectory as discussed at the beginning of this chapter: the *nearest neighbor* (NN) of a point in a \(d\)-dimensional embedding is false if the pair of points are close only due to the projection from a higher-dimensional (e.g., \((d + 1)\)-dimensional) phase space. Thus, the FNN will separate if the data is embedded into a \((d + 1)\)-dimensional space, while the *true* NNs will remain close. If one is able to detect all the FNN, then the minimally necessary embedding dimension can be identified as the least dimension needed to achieve zero fraction of the FNN.

There are three distinct variations of the FNN algorithm. We label them as: the *original* or *Kennel* [45], *improved* [33], and *reliable* [44] algorithms. Variations of them are frequently used in very diverse fields such as stock market price forecasting [42], astrophysics [32], engineering [75], and biology [31]. The effect of additive noise in deterministic time series on the performance of these algorithms will also be discussed.

It has become customary, in the nonlinear time series analysis, to compare the results one gets from applying an algorithm to the data to the results obtained using *surrogate data*, which possesses the same linear characteristics (i.e., power spectrum and probability density) as the original data. There are many different methods to obtain a surrogate data [66], the simplest are only preserving either probability density (by randomly reshuffling data points time sequence) or power spectrum (by randomizing the phase of the Fourier transform and then converting it back to the time domain).
CHAPTER 9. SELECTING EMBEDDING DIMENSION FOR PHASE SPACE RECONSTRUCTION

The idea is that one cannot trust the results if they cannot be differentiated from the ones obtained from the surrogate data. The estimating of minimally necessary embedding dimension requires the surrogate analysis to make we are not using algorithm that would yield low-dimensional indication for the noise. Instead of going through an exercise of frequency and time domain randomization and normalization procedures [66], we will follow [44], where they used a much simpler strategy. For each $d$-dimensional NN pair, the $(d + 1)$-dimensional coordinate distance is estimated by randomly selecting one of the coordinates from all the data. This is justifiable, if all the linear correlations are absent from the components of the reconstructed phase space (the needed decorrelation coordinate transformation is also discussed in Section 9.3.1), where the noise would appear to be effectively white. The algorithms described in the following sections will be contrasted and compared to each other using the normally and uniformly distributed white noise time series as well as the ones used in Figs. 3.2 and 8.7.

9.1 The Original False Nearest Neighbor Algorithm

The original FNN method was proposed by Kennel et al. in 1992 [45]. Working in each $d$-dimensional reconstructed phase space, for each point $x_i$ its NN $x_{j(i)}$ is identified as false if:

$$|x_{i+d\tau} - x_{j(i)+d\tau}| > r \|x_i - x_{j(i)}\|,$$

(9.1)

where $r$ is an a priori fixed threshold value (typically chosen to be 10 in Ref. [45]).

To simplify notation and streamline discussion, we define the following variables:

$$\delta_i \triangleq |x_{i+d\tau} - x_{j(i)+d\tau}|,$$

and

$$\epsilon_i \triangleq \|x_i - x_{j(i)}\|.$$

(9.2)

In what follows, we will also imply the functional dependence of index $j$ on the index $i$ (i.e., $x_j$ is always the NN to $x_i$ using the appropriate NN metric). Then, the probability that any point $x_i$ and its NN $x_j$ are FNN can be written as:

$$f_{nn}(d; r) \triangleq \text{Prob}\{\delta_i > r \epsilon_i\},$$

(9.3)

and we refer to it as the original fraction. Then, the idea is that the smallest dimension $d$ for which $f_{nn}(d; r)$ reaches zero is the minimally sufficient embedding dimension. The original fraction provides misleading results for the white random and correlated stochastic time series for which it erroneously shows low embedding dimensions and the surrogate analysis does not provide the differentiation [44]. In fact, results for uncorrelated random time series and for large amplitude autoregressive processes are virtually identical and indicate the existence of a low-dimensional attractor. This is a problem for noisy deterministic time series, since we cannot definitively attribute zero FNN to the actual existence of a low-dimensional attractor.
9.1. THE ORIGINAL FALSE NEAREST NEIGHBOR ALGORITHM

Hegger and Kantz [33] were first to clearly identify the shortcomings of this definition when noisy deterministic data were considered. For uniformly distributed white noise, the original fraction

\[ f_{nn}(d; r) = 1 - 2 \langle \epsilon_i \rangle r + \langle \epsilon_i^2 \rangle r^2, \]

where \( \epsilon_i = \epsilon_i(d) \) is the distance between the NNs in \( d \) dimensions. For uncorrelated noise time series of size \( N \), \( \langle \epsilon_i \rangle \simeq N^{-1/d} \). Therefore, this FNN fraction approaches zero when \( \langle \epsilon_i \rangle \simeq r^{-1} \) or when \( N \simeq r^d \). To always differentiate noise (which is inherently infinite-dimensional) from deterministic data (finite-dimensional), one always needs \( N \geq r^d \) to get the nonzero original fraction for uncorrelated noise time series. Unfortunately, this is not always feasible for experimental data.

To address this problem, in [45], the NNs were also counted as false if the distance in \( (d + 1) \)-dimensions was greater than the \( a \) times \( \sigma_x \) of the data (e.g., \( a = 2 \) in [45]). Thus, the FNN condition changed to:

\[ \hat{f}_{nn}(d; r) \triangleq \text{Prob}\{ (\delta_i > r \epsilon_i) \lor (\delta_i^2 + \epsilon_i^2 > a^2 \sigma_x^2) \}, \tag{9.4} \]

which we call the Kennel fraction.

Figure 9.1: Kennel fraction of FNN using Eq. (9.4) for Lorenz (a), Duffing (b), Normal (c), Uniform (d), AR(0.02) (e), AR(0.05) (f), AR(0.2) (g), and AR(0.5) (h). The gradual decrease in the FNN fraction is precipitated by incrementing \( r \) from zero to 20 by two. Lines with circles reflect the corresponding surrogate data results.
Figure 9.1 shows the results of this algorithm applied to our data sets. This solution seems to improve performance for the white random time series by increasing the FNN fraction for higher embedding dimensions. However, it exhibits a dip in the FNN fraction at lower dimensions (i.e., it underestimates the FNN fraction for large $r$ and low $d$). This dip is also present for the correlated stochastic time series especially for ones with the larger stochastic components. However, the surrogates also show a similar dip and diverge from actual data after the minimum is reached. Even with this divergence the correlated stochastic data is hard to interpret and this solution is inadequate [44]. This will also be problematic for noisy deterministic time series, since this dip cannot be uniquely attributed to the existence of a low-dimensional attractor. In addition, [45] points out that this solution may introduce false neighbors in large $d$ for an insufficient amount of deterministic data.

9.2 The Improved Algorithm

In [33], it is supposed that if the distance between the NNs is larger than $\sigma_x$, these points cannot be considered true NNs and should be discarded in computation. In this approach, the NNs for which the $d$-dimensional distances are greater or equal to $1/r$ times $\sigma_x$ are disregarded and the original fraction is used for FNN. Unfortunately, for large values of $d$ and $r$, this causes the number of available points to decrease drastically to zero for both white random and correlated stochastic data sets. The current algorithm for false NNs, as described in TISEAN package [34, 41] has the following form:

$$\tilde{f}_{nn}(d; r) \triangleq \frac{\sum_{i=1}^{N-(d-1)r} \Theta(\delta_i - r\epsilon_i) \Theta(\sigma_x - r\epsilon_i)}{\sum_{i=1}^{N-(d-1)r} \Theta(\sigma_x - r\epsilon_i)},$$  \hspace{1cm} (9.5)

where $\Theta$ is the Heavyside function and $\tilde{f}_{nn}$ is referred to as the TISEAN fraction. They claim that this solution for white random data leads to 50–60% FNN for large $N$, independent of choice of $d$ and $r$. However, in our white random and dominantly stochastic test cases, the number of points suitable for averaging drops to zero precipitously for large $d$ values. In addition, it indicates low-dimensional embedding for stochastic or random data and does not provide any differentiation with the surrogate analysis.

9.3 The Reliable Algorithm

In [44], some of the ideas in [33] were further expanded and a new FNN algorithm was proposed in conjunction with new and interesting false nearest strands (FNS) idea. These new FNN and FNS methods were shown to be usable for noisy or stochastic time series. In [44] several important improvements to the FNN algorithm were proposed: (1) only NNs that are temporarily uncorrelated are considered to focus exclusively on spatial proximity; (2) instead of FNN, FNS were advocated to
make sure only true NNs were considered in the calculations even for noisy data; and (3) a spatial decorrelation transform was introduced to deal with linear correlation bias due to small delays.

### 9.3.1 Spatial Decorrelation

The spatial decorrelation step is aimed at negating the effect of the too-small delay used in the reconstruction, which makes the FNN fraction small irrespective of the actual dynamics. In addition, this step makes the method insensitive to the appropriate choice of the time delay, unless it is chosen to be unreasonably large.

The basic idea of the decorrelation is to counteract the bunching of all reconstructed points near a thin tube along the hyperdiagonal of a phase space. Given the original \((d + 1)\)-dimensional reconstructed phase-space trajectory \(Y \in \mathbb{R}^{(d+1) \times [N-(d+1)]}\), where the \(i\)-th column is given by the \(i\)-th delay vector \(x_i \in \mathbb{R}^{d+1}\), \(Y\) is first transformed into its proper orthogonal space \(\tilde{Y}\) using singular value decomposition [25]:

\[ Y = UCV^T, \quad \Rightarrow \quad \tilde{Y} = U^TY, \quad (9.6) \]

Then \(\tilde{Y}\) is rotated in the phase space so that the first \(d\)-coordinates are only a function of the first \(d\) coordinates of the original phase space \(Y\). To accomplish this we take the last column of \(\tilde{Y}\), \(u \in \mathbb{R}^{d+1}\), and then the needed orthogonal transformation is given by the following Householder matrix [25]:

\[ Q = H (u - |u| e_{d+1}), \quad (9.7) \]

where \(e_{d+1}\) is the unit vector in the \((d + 1)\) direction. Then the decorrelated phase space is given by:

\[ \bar{Y} = Q\tilde{Y}. \quad (9.8) \]

In our test samples, the choice of delay time is close to optimal and the use of decorrelation does not provide any significant benefits, but in practice it can be a differentiating factor. Thus, we have included this step in the results presented from now on.

### 9.3.2 Temporal Decorrelation

As opposed to the TISEAN fraction, Ref. [44] advocates to look at just the distance between the \((d + 1)\)-th coordinates of the NNs. If we only consider the FNN- and not the FNS-based metric we will get:

\[ \bar{f}_{nn}(d; s) \triangleq \text{Prob}\{\delta_i > s \sigma_x\}, \quad (9.9) \]

where, to identify true NNs for each point \(x_i\), its NN \(x_j\) is determined such that:

\[ j = \arg\min_j \|x_i - x_j\|, \quad \text{subject to } |i - j| > w, \quad (9.10) \]
where \( w \) is a Theiler window [72, 73] used to remove temporal correlations. We call the metric defined by Eq. (9.9) the reliable fraction. Ref. [44] discusses the appropriate choice of the Theiler window and advocates to use two or three times the first minimum of average mutual information as a good rule of thumb. Another alternative is to match the \( w \) time scale to the point at which the average mutual information reaches its asymptotic zero value.

The results for Eqs. (9.9), and (9.10) using a Theiler window of four times the first minimum of the average mutual information are shown in Fig. 9.2 (left plots). For the deterministic time series, we observe an initial underestimation of the FNN fraction with the increase in \( s \) value. The fractions reach zero for an appropriate embedding dimension and then show gradual increase for small \( s \) values with the increase in \( d \) values. This gradual increase is more pronounced for the Duffing data and is absent for larger \( s \) values. For the white random data sets the FNN fraction behaves as expected, showing linear decrease as \( s \) is increased, while remaining constant for all values of \( d \). For the correlated stochastic data this metric is more interesting and shows trends intermediate between the trends for the purely white random and deterministic data. Specifically, it can clearly be used with large \( s \) values to differentiate small stochastic component from the larger deterministic data—plots (e,f). The results for the surrogates are clearly separated from the actual data, especially for larger \( s \) values, and remain constant for all \( d \).

Figure 9.2: Reliable FNN (left plots) and reliable FNS (right plots) fraction of Eq. (9.9) applied to Lorenz (a), Duffing (b), Normal (c), Uniform (d), AR(0.02) (e), AR(0.05) (f), AR(0.2) (g), and AR(0.5) (h). The gradual decrease in the FNN fraction is precipitated by incrementing \( s \) from zero to two by 0.1. Lines with circles reflect the corresponding surrogate data results.
9.3. THE RELIABLE ALGORITHM

9.3.3 False Nearest Strands

In the same paper [44], the shortcomings of the previous FNN methods for the deterministic data contaminated by small random noise are also identified. In this case, the NNs identified through the analysis may be proximal not due to the deterministic dynamics or the projections, but due to noise. If the proportion of the misidentified NNs is large, then the algorithm will overestimate the corresponding FNN fraction. To identify only true (in terms of deterministic dynamics and projection only) NNs, they propose to consider NN strands (NNS). The following procedure is used to identify NNS:

1. Identify NN $x_{j(i)}$ for each point $x_i$ given by Eq. (9.10).

2. If there is any pair of points (for a minimal $k = k^*$) from a set of preceding NNs $\{x_{i-k}, x_{j(i-k)}\}_{k=1}^w$ such that $j(i) = k^* + j(i-k^*)$, then $[x_i, x_{j(i)}]$ are assigned to the strand containing $[x_{i-k}, x_{j(i-k^*)}]$. Otherwise, $[x_i, x_{j(i)}]$ are assigned to a new pair of NN strands.

3. After examining all the data, $N_s$ number of NNS are obtained, each containing a set of NN points $S_k$ ($k = 1, \ldots, N_s$). Then the $k$-th strand is designated false if $\langle \delta_i \rangle_k > s \sigma_x$, where

$$\langle \delta_i \rangle_k \triangleq \langle \delta_i \rangle_{i \in S_k} = \frac{1}{\|S_k\|} \sum_{i \in S_k} \delta_i.$$  \hspace{1cm} (9.11)

Alternatively we can define Reliable FNS Fraction as:

$$\bar{f}_{ns}(d; s) \triangleq \text{Prob}\{\langle \delta_i \rangle_k > s \sigma_x\}.$$  \hspace{1cm} (9.12)

The results of the FNS calculations for our test time series are shown in Fig. 9.2 (right plots). The decrease in the FNS fraction with the increase in $s$ is still present as reliable FNN. For the deterministic data, some gradual increase in this fraction with the increase in $d$ for small values of $s$ is also present. In addition, and in contrast with the reliable FNN fraction, this metric does not converge to zero as fast and indicates larger minimal embedding dimension for the deterministic data. The results for the white random data are still indistinguishable from the ones for the surrogate data. However, the white random data trends show a gradual increase in the FNS fraction not observed reliable FNN. For the correlated stochastic data sets, FNS trends are clearly delineated from the surrogates for the low level of stochasticity, and show results similar to the white random data for the larger stochastic components. To accurately estimate embedding dimension in deterministic cases, one needs to set some threshold value for the FNS fraction since it does not reach zero as fast as in the reliable FNN fraction. This task is made easier by the clear separation from the surrogates FNS fraction, which stays relatively level with the increase in $d$. For the larger $s$ values, the Duffing FNS fraction also indicates lower than expected embedding dimension.

To see the advantage of these new metrics when dealing with noisy data, we apply them to Duffing data with additive Gaussian noise as shown in Fig. 9.3. For the reliable FNN algorithm, the
new fraction floor rises with the noise level, but it is still usable for small values of \( s \) and low 5% noise level Fig. 9.3(a). At the same time, the surrogate’s FNN fraction stays constant with respect to \( d \) and asymptotes down with the increase in \( s \) to approximately 50% fraction. The FNS data looks better for the low noise levels Fig. 9.3(b,d) and exhibits similar surrogate behavior. Therefore, it provides better separation between the surrogate and actual data FNS fractions compared to the FNN fractions. One might be able to estimate embedding dimension for the low noise levels in Fig. 9.3 if an appropriate threshold value is set. However, both reliable fractions cannot identify the embedding dimension for the larger noise levels.

In what follows, we want to further expand on some ideas in [44] and propose an FNN algorithm that is usable for noisy or stochastic time series with deterministic components. We begin with exploring the nearest-neighbor statistics for deterministic, white random, and correlated stochastic time series.

Figure 9.3: Reliable FNN (first column: a,c,e,g) and FNS (second column: b,d,f,h) algorithm applied to Duffing data with 5% (a,b), 10% (c,d), 20% (e,f), and 40% (g,h) noise. The gradual decrease in the fractions is precipitated by incrementing \( s \) from zero to two by 0.1. Lines with circles reflect the corresponding surrogate data results.
9.4 Nearest-Neighbor Statistics

To investigate the problems outlined in the previous sections, we investigate NN statistics for our test time series. First, consider the expected average nearest-neighbor distance $\langle \epsilon_i(d) \rangle$ for white random time series. Given a $d$-dimensional reconstruction of normally distributed random time series, the distance between any two points will be distributed according to a $\chi_d$ ($d$-dimensional chi distribution [30]). We also proposition that the expected value of the distance between the NN points will be proportional to the mean of the corresponding $\chi_d$ distribution and to the average density of points in $d$ dimensions $(\sqrt{d}/N)^{1/d}$. This will result in the following approximation:

$$\langle \epsilon_i(d) \rangle \simeq \sqrt{2} \frac{\Gamma((d+1)/2)}{\Gamma(d/2)} \left( \frac{\sqrt{d}}{N} \right)^{1/d}, \quad (9.13)$$

where $\Gamma$ is the gamma function. Figure 9.4(a) shows the expected values of nearest-neighbor distances versus the embedding dimension for all eight test time series, without accounting for temporarily correlated NNs. The increase between the mean nearest-neighbor distance is more drastic for the white random data than the deterministic time series, and Eq. (9.13) follows this increase for the white random time series very closely.

The deterministic time series show more power-law type increase after the needed embedding dimension is reached ($d \sim 3$–4, in both case), and it has a considerably lower rate than for the random data. The nearest-neighbor distance for deterministic data is expected to grow according to local divergence rate $e^{\lambda_{\text{max}} dt s}$, where $\lambda_{\text{max}}$ is maximal Lyapunov exponent, after the minimum embedding dimension is reached. This can be expressed as:

$$\langle \epsilon_i \rangle \approx \varepsilon \left( e^{\lambda_{\text{max}} dt s} - 1 \right), \quad \forall d \geq m, \quad (9.14)$$

where $\varepsilon$ is a constant reflecting the point density in the phase space. The parameters that follow the observed curves closely are: $\varepsilon = 0.0404 \pm 0.0008$ and $\lambda_{\text{max}} t s = 0.0689 \pm 0.0008$ for the Lorenz

![Figure 9.4: Expected value of nearest-neighbor distances in $d$ dimensions without (a) and with (b) temporarily decorrelated NNs](image-url)
data, and \( \varepsilon = 0.5110 \pm 0.0742 \) and \( \lambda_{\text{max}} \tau t_s = 0.0167 \pm 0.0020 \) for the Duffing data. The exponential growth in Eq. (9.14) is only expected until it reaches the attractor size at which point it will saturate. Even if it is allowed to grow unimpeded it will only cross Eq. (9.13) somewhere \( d \sim 60 \). Thus, when dealing with low-dimensional deterministic systems (e.g., \( d < 40 \)) their expected nearest-neighbor distance will stay well below the distance expected for the white random data in the same dimension.

The correlated stochastic time series show behavior somewhere in the middle of the white random and deterministic nearest-neighbor distance trends. They start close to the white random at low dimensions and then diverge to smaller distances for the larger embedding dimensions.

The same plot was regenerated for the distance between temporarily decorrelated NNs as shown in Fig. 9.4(b). The main observed differences are that the large stochastic component data now closely follows the white random distances, while the small stochastic component data shows the intermediate behavior. In addition, the deterministic data have more pronounced trends clearly showing differences in the divergence rates. Fitting the Eq. (9.14) to these curves gives the following parameters: \( \varepsilon = 0.0333 \pm 0.0007 \) and \( \lambda_{\text{max}} \tau t_s = 0.0790 \pm 0.0008 \) for the Lorenz data, and \( \varepsilon = 0.0425 \pm 0.0020 \) and \( \lambda_{\text{max}} \tau t_s = 0.1212 \pm 0.0021 \) for the Duffing data. Thus, we observe: (1) temporal correlations considerably stint the divergence rates, especially for the Duffing data; and (2) divergence exponents after removing temporal correlations for Lorenz (\( \lambda_{\text{max}} \approx 0.5 \)) and Duffing (\( \lambda_{\text{max}} \approx 0.1 \)), while not being suitable for estimating maximal Lyapunov exponents, are close to their expected values [70] (experimentally estimated largest Lyapunov exponents from data are \( \lambda_1 \approx 0.825 \) for Lorenz and \( \lambda_1 \approx 0.134 \) for Duffing).

Now, let us consider the \( (d+1) \)-th coordinates of the NNs in \( d \) dimensions, which are still uncorrelated random numbers for the white random time series. Thus, the distance between these \( (d+1) \)-th coordinates for normally distributed variables \( (x_i \sim N[0, \sigma_x]) \) follows a chi distribution [30], and the corresponding expected value can be expressed as:

\[
\langle \delta_i \rangle = E[\delta_i] = \frac{2 \Gamma(1) \Gamma(1/2)}{\Gamma(1/2)} \sigma_x \approx 1.1284 \sigma_x. \tag{9.15}
\]

Now, it is easy to show that for the uniformly distributed variables (i.e., \( x_i \sim U[-a, b] \)), this expected value becomes:

\[
\langle \delta_i \rangle = \frac{2}{\sqrt{3}} \sigma_x \approx 1.1547 \sigma_x. \tag{9.16}
\]

We conclude that for some general uncorrelated random time series the average distance between the \( (d+1) \)-th coordinates is expected to be slightly greater than the corresponding \( \sigma_x \) (\( \sim 1.12-1.16 \sigma_x \)).

A similar relationship is expected for the colored-noise time series if the delay used in the reconstruction is greater than the correlation time. However, for a deterministic time series we expect this \( (d+1) \) component’s distance to be small for true NNs. The estimation results using our synthetic

\[1\] If \( x_i \sim N[0, \sigma_x] \), then \( x_i - x_j \sim N[0, 2 \sigma_x] \), and \( \frac{|x_i - x_j|}{2 \sigma_x} \sim \chi_1 \).
9.4. NEAREST-NEIGHBOR STATISTICS

The statistical analysis described in this section can be summarized as:

1. Temporal correlations significantly alter the observed results and need to be accounted for...
2. Of the considered temporarily decorrelated metrics, \((d + 1)\)-th coordinate distance \(\langle \delta_i \rangle\)—as shown in Fig. 9.5(b)—provides the best indication of the minimal embedding dimension and superior separation between the deterministic and random/stochastic trends, which remain constant with the increase in \(d\). Thus, this metric can be used directly for estimating the embedding dimension for the noise free deterministic data. However, for noise contaminated deterministic data, it loses its ability to clearly identify the minimal embedding dimension as shown in Fig. 9.6.

3. Both \(\langle \epsilon_i \rangle\) and \(\langle \delta_i \rangle\) for deterministic data show exponential growth after minimal embedding dimension is reached. Thus, the original FNN fraction shown in Fig. 9.7 for both without (a) and with (b) temporarily decorrelated NNs is a good differentiator of deterministic data for low embedding dimensions. Setting a threshold value near 5 \(\sim\) 10, this ratio with tem-

---

**Figure 9.6:** Expected NN distance \(\langle \delta_i \rangle\) versus the embedding dimension \(d\) for noisy Lorenz (a) and Duffing (b) time-series in accurate FNN analysis (removal of temporarily correlated points from the FNN analysis increases the differentiating power of the NN statistics).

---

**Figure 9.7:** Expected value of the Kennel ratio in \(d\) dimensions without (a) and with (b) temporarily decorrelated NNs
9.4. NEAREST-NEIGHBOR STATISTICS

Poral decorrelation provides accurate estimates of FNN for \( d < 10 \), and clear separation of deterministic trends.

9.4.1 New Composite-Fraction-Based Metrics

Using the results of the statistical analysis described in Section 4, and combining them with the original definition of FNN, the following FNN composite fraction can be advocated:

\[
R_{nn}(d; r, s) \triangleq \text{Prob} \left\{ \left( \delta_i > r \epsilon_i \right) \lor \left( \delta_i > s \sigma_x \right) \right\} .
\] (9.17)

This new ratio has two parameters: \( r \) is the same as for the original fraction, while \( s \) is the same as for the reliable fraction. The first part is expected to serve as good indication of FNN for low \( d \), while the second provides good indication of minimal embedding dimension and best separation of deterministic data from random/stochastic for larger \( d \).

This composite metric still does not address the problem of NNs being close to each other due to the presence of noise instead of either dynamics or projection. Therefore, we propose a procedure similar to the FNS, where instead of looking up just one NN (NN) for each point \( x_i \), we look up \( k \) NNs. Then, the true, temporarily uncorrelated NN (i.e. the one close only due to dynamics or projection) is identified by evaluating the preceding and future \( l \) states of all \( k \) NNs. In particular, we evaluate the average distances between the base strand and all \( k \) NN strands of length \( 2l + 1 \).

The strand with the smallest average distance from the base strand will contain the needed true NN.

9.4.2 New Composite-Fraction-FNN Algorithm

The composite FNN procedure can be summarized as follows:

1. For each point \( x_i \), identify its temporarily uncorrelated \( k \) NNs

\[
\{ x_{j(m)} \}_{m=1}^{k} = \{ x_{j(m)} : |i - j(m)| > w, \; m = 1, \ldots, k \}
\]

just like in Eq. (9.10) to eliminate temporarily correlated NNs.

2. Then the true NN \( x_j \) is identified as:

\[
j(m) = \arg \min_m \frac{\sum_{q=-l}^{l} \| x_{j(m)+q} - x_{i+q} \|}{(2l + 1) \| x_{j(m)} - x_i \|} ,
\] (9.18)

and used in Eq. (9.17) to determine FNN fraction.

Equation (9.18) indicates that the composite fraction will also be a function of \( l \) and \( k \) parameters:

\[
R_{nn} = R_{nn}(d; r, s, l, k) .
\] (9.19)
It is reasonable to set the value of $l \leq w/2$ in Eq. (9.18) since we are focusing on temporal correlations. The choice of $k$ can be influenced by many factors, such as noise level, density of points in the embedding, and sampling time. Here we only present results obtained with just one NN point ($k = 1$ and $l = 0$) for noise free data. In addition, we use $k = 10$ and $l = \tau$ for data with additive noise. However, we explicitly set these values in what follows and do not study their influence on the quality of the results, which is left for future work. Therefore, $R_{nn} = R_{nn}(d; r, s, \tau, k) \triangleq R_{nn}(d; r, s)$ in what follows.

Figure 9.8: Eq. (9.17) based composite fraction FNN algorithm with $s = 0.5$ (left plots) and with $r = 4$ (right plots) for Lorenz (a), Duffing (b), Normal (c), Uniform (d), AR(0.02) (e), AR(0.05) (f), AR(0.2) (g), and AR(0.5) (h). The gradual decrease in the fractions is precipitated by incrementing $r$ from zero to 20 by two and $s$ from zero to two by 0.1. Lines with circles reflect the corresponding surrogate data results.

The results using Eq. (9.17) are shown in Fig. 9.8 for fixed $s = 0.5$ (left plots) and for fixed $r = 4$ (right plots) for all the test data. For the deterministic data, the algorithm works well for $r \geq 4$ and for all $s > 0.2$ values, while for the same $r$ values it provides correct indication of high-dimensional data for the white and correlated stochastic time series. In particular, the actual results are indistinguishable from the surrogate analysis for the random data and the composite fraction never drops below 50%. For correlated stochastic data with a smaller stochastic component, we can clearly see that there is some deterministic component to the data, while for a large stochastic
component, the results are very similar to the ones for the white noise. However, there still is the small but clearly identifiable separation between the surrogate and correlated stochastic data. This is especially evident for $s = 0.5$ plots, where for larger values of $d$ there is a clear convergence in the FNN composite fraction to the same constant value for the surrogates, and to different values for all the correlated stochastic data sets. The separation of this surrogate FNN fraction from the actual FNN fraction seems to scale with $d$ and the relative magnitude of the deterministic component. Therefore, we can use this separation between the actual and surrogate data composite fractions as an indication of some deterministic trends in the stochastic data. However, this needs further investigation for practical applicability in the analysis, with the purpose of relating the observed differences with the surrogates to the relative strength of the deterministic components.

One may expect the removal of temporal correlations and the new definition of NN points to positively alter the Kennel fraction results. However, the observed improvements were marginal: (1) marginally better indication of minimal embedding dimension for the deterministic data; (2) removal of artificial separation between the surrogates and actual data for the stochastic time series; and (3) accentuation exponential divergence for the deterministic data for larger embedding dimensions. Thus, the main deficiencies that were identified previously still remain.

9.4.3 New-Composite-Fraction-FNS Algorithm

We have also modified the FNS algorithm to use a new metric specified by Eq. (9.17). We use exactly the same procedure as before, except instead of Eq. (9.12) we use:

$$ R_{ns}(d; r, s) \triangleq \text{Prob} \left\{ \langle \delta_i \rangle_k > r \langle \epsilon_i \rangle_k \vee \langle \delta_i \rangle_k > s \sigma_x \right\}, \quad (9.20) $$

where

$$ \langle \epsilon_i \rangle_k \triangleq \langle \epsilon_i \rangle_{i \in S_k} = \frac{1}{\|S_k\|} \sum_{i \in S_k} \epsilon_i. \quad (9.21) $$

The results of calculation using Eq. (9.20) are shown in Fig. 9.9 for fixed $s = 0.5$ (left plots) and for fixed $r = 10$ (right plots) for all the test data.

For all the simulations we have used $w = \tau$. The differences with the new FNN are similar to what was observed previously in the reliable algorithm. The decrease of the FNS composite fraction to zero is slower and more gradual than for the FNN composite fraction for the deterministic data. However, the FNS composite fraction shows a much clearer indication for $d = 3$ for the Lorenz and $d = 4$ for the Duffing data as opposed to FNN, where we have to plot data on a log scale to clearly see the minima at the same values. This algorithm can be clearly used to differentiate the deterministic versus stochastic or random data using surrogate analysis. In addition, for the smaller stochastic component data, the surrogate analysis also provides the needed indication of the deterministic component. However, this indication is absent for the larger stochastic components for which they are indistinguishable from the random data sets.
Figure 9.9: Eq. (9.20) based composite fraction FNS algorithm with \( s = 0.5 \) (left plots) and with \( r = 10 \) (right plots) applied to Lorenz (a), Duffing (b), Normal (c), Uniform (d), AR(0.02) (e), AR(0.05) (f), AR(0.2) (g), and AR(0.5) (h). The gradual decrease in the fractions is precipitated by incrementing \( s \) from zero to two by 0.1 and \( r \) from zero to 20 by two. Lines with circles reflect the corresponding surrogate data results.

9.4.4 Noise Robustness of Composite Fraction Algorithms

To test the applicability of these new algorithms for the noisy deterministic time series we have tested them on both Lorenz and Duffing data contaminated by additive noise as described before. The composite FNN fraction results for noisy Lorenz and Duffing are shown in Fig. 9.10 for \( k = 1 \) NN (left plots) and for \( k = 10 \) NNs (right plots), respectively. For both cases, we set \( r = 10 \) and \( s = 1 \). Plots demonstrate that using ten NNs in the algorithm considerably improves on results using just one NN. Composite FNN fractions with ten NNs have a clear inflection point, or elbow, in their trends at the appropriate minimal embedding dimension for low noise levels. Even for high noise levels the estimates for Duffing are good and for Lorenz only overestimate the embedding dimension by one.

New composite FNS plots for noisy Lorenz and Duffing data are shown in Fig. 9.11 left and right plots, respectively. Compared to the composite FNN fraction plots, these do not show as clear of an indication when the FNS composite fraction levels out or has a clear inflection point at the
appropriate minimal embedding dimensions. Instead, it has a more gradual transition requiring some careful thresholding to identify the appropriate minimal embedding dimensions. It should be noted that these results are similar to the ones obtained for the stochastic time series with small stochastic components. Therefore, we have to be careful when making any judgments about the nature of additive noise versus systematic stochasticity. However, we can still clearly use these new methods in both cases to identify the deterministic components in the data and their dimensionality.

Figure 9.10: Eq. (9.19) based FNN algorithms with \( k = 1 \) (left plots) and \( k = 10 \) (right plots) for Lorenz (first and third columns) and Duffing (second and fourth columns) and the surrogate data with 5\% (a,b), 10\% (c,d), 20\% (e,f), 40\% (g,h) additive noise levels. Lines with circles reflect the corresponding surrogate data results.

9.4.5 Concluding Remarks on False Nearest Neighbors

1. The expected \((d + 1)\)-dimensional distance between the \(d\)-dimensional nearest neighbors is a good indicator of minimal embedding dimension for the clean deterministic time series, but it loses its differentiation when even small additive noise is introduced into the deterministic data.

2. New algorithms based on the composite fraction overcome the deficiencies of the earlier meth-
ods, and are applicable even for noisy deterministic data.

3. While both composite FNN and FNS concepts provide results that can be used for estimating the minimal embedding dimension even in a noisy data set, the composite FNS algorithm provides a clearer indication of the minimal embedding dimension for the noise free data compared to the composite FNN.

4. However, the composite FNN algorithm has a more differentiating power and provides less ambiguous results for the deterministic data contaminated with additive noise.

5. These FNN/FNS composite fractions are also suitable in evaluating the presence and relative magnitude of a deterministic component in a stochastic or noisy data set.

Figure 9.11: Eq. (9.20) based FNS algorithms for Lorenz (left column) and Duffing (right column) and the surrogate data with 5% (a,b), 10% (c,d), 20% (e,f), 40% (g,h) additive noise levels. Lines with circles reflect the corresponding surrogate data results.
Problems

Problem 9.1

Create an artificial time series by integrating the Chua’s circuit [53, 54] by Matsumoto et al. (1985):

\[
\begin{align*}
\dot{x} &= \alpha \left(y - x + bx + \frac{1}{2}(a - b)(|x + 1| - |x - 1|)\right) \\
\dot{y} &= x - y + z \\
\dot{z} &= -\beta y
\end{align*}
\]

(9.22) (9.23) (9.24)

using these parameters: \(\alpha = 9\), \(\beta = 100/7\), \(a = 8/7\), and \(b = 5/7\). Start the simulation using the following initial conditions: \(x(0) = 0\), \(y(0) = 0\), and \(z(0) = 0.6\). In particular, using the \(x\) and \(y\) time series:

1. Make sure you are using an appropriate sampling frequency or sampling time period to generate the trajectory (check using power spectrum and auto-correlation), and record at least 60,000 evenly sampled points.

2. Determine the appropriate delay for the delay coordinate reconstruction and verify it visually.

3. Use false nearest neighbors method to determine the minimally necessary embedding dimension using composite FNN fraction algorithms.

4. Now add some Gaussian noise to the data and do the same calculations for different signal-to-noise ratios.

5. Interpret your results.
Some comments on using the supplied programs to plot the FNN results.

1. `false_neighbors_kd_n.m` outputs only the statistics on the nearest neighbors in the requested dimensions in `nn` structure.

2. `plot_false_neighbors.m` processes the output of the `false_neighbors_kd_n.m` and provides the actual false nearest neighbor metrics inside the `nn` structure.

3. You can ask for ranges for the input thresholds `r` (usually between 0 to 20, say `r = 1:2:20;`), and `s` (usually between 0 to 1, say `s = 0.1:0.1:1;`)

4. Most of the FNN metrics inside `nn` are matrices (which can be plotted using `mesh` or `surf` commands), however two most important ones `nn.P` and `nn.S` are three-dimensional arrays. They need special care in plotting.

5. The first coordinate in `nn.P` and `nn.S` corresponds to the embedding dimensions considered, the second coordinate is threshold for the Kernel ratio (`r`), and the third threshold for the `(d + 1)`-th distance (`s`).

6. To do a 3D plot of FNNs versus `d` embedding dimension and `r` you can just use `mesh(nn.P(:, :, k));`, where `k` is the index of the `s` for which you want to make a plot.

7. To do a 3D plot of FNNs versus `d` embedding dimension and `s`, first you need to permute the three-dimensional array as `nnP = permute(nn.P, [1, 3, 2]);` and then you can use `mesh(nnP(:, :, k));` to plot, where `k` now is the index of the `r` for which you want to make a plot.