A New Entropy Measure Based On The Renyi Entropy Rate Using Gaussian Kernels

D Woodcock and I T Nabney
Aston University, UK

February 24, 2006

1 Abstract

The concept of entropy rate is well defined in dynamical systems theory but is impossible to apply it directly to finite real world data sets. With this in mind, Pincus developed Approximate Entropy (ApEn) [9], which uses ideas from Eckmann and Ruelle [3] to create a regularity measure based on entropy rate that can be used to determine the influence of chaotic behaviour in a real world signal. However, this measure was found not to be robust and so an improved formulation known as the Sample Entropy (SampEn) was created by Richman and Moorman [10] to address these issues. We have developed a new, related, regularity measure which is not based on the theory provided by Eckmann and Ruelle and proves a more well-behaved measure of complexity than the previous measures whilst still retaining a low computational cost.

2 Background

To understand the differences between the entropy formulations, we need to explore the theory behind them.

2.1 Entropy Rate

ApEn was originally based on the Kolmogorov-Sinai (K-S) invariant $h(\rho)$ where ρ is an ergodic probability measure. The K-S invariant (or entropy) can be seen as the *mean rate of creation of information* [3]. It is worth noting that although this invariant is often called 'entropy', it is different from the information entropy of the system [11].

If $A = \{a_1, a_2, \ldots, a_\alpha\}$ is the finite alphabet of a function f defined on sample space Ω we can consider a partition $\mathcal{Q} = \{Q_i; i = 1, 2, \ldots, \alpha\}$ defined by $Q_i = \{\omega : f(\omega) = a_i\} = f^{-1}(\{a_i\})$ [4]. We can then write the entropy as a function of the partition defined by the disjoint sets of the points that f maps to the alphabet A as

$$H_\rho(\mathcal{Q}) = -\sum_{i=1}^{\alpha} \rho(Q_i) \log \rho(Q_i), \tag{1}$$

where we also define $u \log u = 0$ if $u = 0$. So $H_\rho(\mathcal{Q})$ is the information content of the partition with respect to state ρ.

We define the notation f^τ as the function composed τ times. The inverse functions $f^{-\tau}$ can be partitioned in a similar way to above, denoted $\mathcal{Q}^\tau$, and thus can be used do describe the partition due to time evolution as

$$\mathcal{Q}^\tau = \mathcal{Q}^0 \cup \mathcal{Q}^1 \cup \ldots \cup \mathcal{Q}^{\tau-1}. \tag{2}$$

We can also write the components of this partition as

$$P_k^\tau = Q_{i_1}^0 \cap Q_{i_2}^1 \cap \ldots \cap Q_{i_\tau}^{\tau-1}, \tag{3}$$

where $i_j \in \{1, 2, \ldots, \alpha\}$ and $k \in \{1, 2, \ldots, \alpha^\tau\}$. Therefore, the information content of an interval of time period of length τ with respect to the state ρ can be written as

$$H(\mathcal{Q}^\tau) = -\sum_{k=1}^{\alpha^\tau} \rho(P_k^\tau) \log \rho(P_k^\tau). \tag{4}$$

As we observe the system iterating, we can determine the *gain* in information by observing consecutive iterates τ and $\tau + 1$ as the difference between the entropies thus

$$H(\mathcal{Q}^{\tau+1}) - H(\mathcal{Q}^\tau).$$

The *rate* of information creation [8] is the gain of information per iterate and is the the limit

$$h(\rho, A) = \lim_{\tau \to \infty} [H(\mathcal{Q}^{\tau+1}) - H(\mathcal{Q}^\tau)]. \tag{5}$$

2.2 Approximate Entropy

Equation 5 cannot be applied to real world applications as the data series is always of finite length and so the limit $\tau \to \infty$ cannot be calculated when the dynamical equations governing the system are unknown. Pincus noted that even an approximation of this may have intrinsic interest in determining the nature of a dynamical system and developed the approximate entropy (ApEn) to investigate this.

If we have a signal $\mathbf{x} = \{x(1), x(2), \ldots, x(N)\}$ then we can define a distance measure as the maximum Euclidean distance between m consecutive values starting at value x_i and the respective values starting at value x_j

$$d[\mathbf{x}_m(i), \mathbf{x}_m(j)] = \max\{\left|x_m(i+k) - x_m(j+k)\right|\},$$

for $k = 0, 1, \ldots, m-1$. If we introduce a radius r, we can find the number of $\mathbf{x}_m(j), j = \{0, 1, \ldots, N-m+1\}$ that lie within a ball of radius r centred at $\mathbf{x}_m(i)$.

$$N_i^m(r) = \text{Number of } \mathrm{d}[\mathbf{x}_\mathrm{m}(\mathrm{i}), \mathbf{x}_\mathrm{m}(\mathrm{j})] \leq \mathrm{r}, \tag{6}$$

then we can calculate the probability that a consecutive sequence repeats itself in the series (within the tolerance value) thus

$$C_i^m(r) = \frac{N_i^m(r)}{N-m+1}. \tag{7}$$

It is now possible to estimate $h(\rho)$ directly [3]. If we define

$$\phi^m(r) = \frac{1}{N-m+1} \sum_{i=1}^{N-m+1} \log C_i^m(r), \tag{8}$$

then, using Equation 5, we can say

$$h(\rho) = \lim_{r\to 0} \lim_{m\to\infty} \lim_{N\to\infty} [\phi^{m+1}(r) - \phi^m(r)]. \tag{9}$$

As this is still intractable for finite data sets, further refinements need to be made. Approximate entropy (ApEn) is essentially Equation 9 but with fixed m and r, and a fixed number (N) of data points. It is defined as

$$ApEn(m, r, N) = \phi^{m+1}(r) - \phi^m(r). \tag{10}$$

Although ApEn is derived from and resembles Equation 5, it is worth noting that it is not intended to calculate it and should be considered as a separate statistic in its own right [9].

2.3 Sample Entropy

Approximate entropy has been used to a large degree of success in a wide variety of studies. However, it has been shown that ApEn is inherently biased [10], therefore sample entropy was developed to address this bias and provide a more rigorous complexity measure.

One source of such bias is the necessity of assuring a non-zero value for Equation 6, so the logarithm taken in Equation 7 is well-defined. The method of assuring that this constraint is fulfilled in approximate entropy is by allowing $i = j$ (known as 'self-matching' [10]) in Equation 6. This means that $d[\mathbf{x}_m(i)\mathbf{x}_m(i)]$ is allowed to be 0 and therefore less than r so $N_i^m(r)$ will always be positive.

We can see how this causes bias in the statistic by considering $N_i^m(r)$ and $N_i^{m+1}(r)$. As approximate entropy can be considered as the log of the conditional probability that $N_i^m(r)$ and $N_i^{m+1}(r)$ stay the same over time we can write it as

$$ApEn(m, r, N) \approx \frac{1}{N-m} \sum_{i=1}^{N-m} \log \frac{N_i^m(r)}{N_i^{m+1}(r)}. \tag{11}$$

Now, as neither $N_i^m(r)$ or $N_i^{m+1}(r)$ can be 0 for this to be defined, the self-matching gives a positive value for the statistic which causes bias, especially in small series. SampEn removes this bias by removing all self matches.

The formulation is also slightly different. Only the first $N-m$ values of the series were used when calculating $N_i^m(r)$ and $N_i^{m+1}(r)$ ensuring an equal length of series for each value. Also, two new variables are defined, based on the *correlation integral*. The correlation integral is simply the average over i of $C_i^m(r)$ defined in Equation 7,

$$C^m(r) = \frac{1}{N-m+1} \sum_{i=1}^{N-m+1} C_i^m(r). \tag{12}$$

$N_i^{'m}(r)$ is defined to discount the self matches,

$$N_i^{'m}(r) = \text{Number of } \mathrm{d}[\mathbf{x}_\mathrm{m}(\mathrm{i}), \mathbf{x}_\mathrm{m}(\mathrm{j})] \leq \mathrm{r}, \tag{13}$$

for $j = 1, 2, \ldots, m, j \neq i$. We now define

$$U_i^m(r) = \frac{1}{N-m-1} N_i^{'m}(r), \tag{14}$$

and following from Equation 12,

$$U^m(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} U_i^m(r). \tag{15}$$

$U^{m+1}(r)$ is similarly defined for $m+1$

$$N_i^{'m+1}(r) = \text{Number of } \mathrm{d}[\mathbf{x}_\mathrm{m}(\mathrm{i}), \mathbf{x}_\mathrm{m}(\mathrm{j})] \leq \mathrm{r}, \tag{16}$$

for $j = 1, 2, \ldots, m, j \neq i$. We now proceed as before

$$U_i^{m+1}(r) = \frac{1}{N-m-1} N_i^{'m+1}(r), \tag{17}$$

$$U^{m+1}(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} U_i^{m+1}(r). \tag{18}$$

The sample entropy is the negative logarithm of the ratio of these probabilities

$$SampEn(m, r, N) = -\log \frac{U^{m+1}(r)}{U^m(r)} \tag{19}$$

We can see how this compares to the approximate entropy given in Equation 11 by noting that the $1/(N-m)$ and $1/N-m-1$ terms cancel so we can write it as

$$SampEn(m, r, N) = \log \frac{\sum_{i=1}^{N-m} N_i^{'m}}{\sum_{i=1}^{N-m} N_i^{'m+1}}. \tag{20}$$

This is precisely the log of the sum of the conditional probability that two sequences that are classified as *similar* within a tolerance of r for m points remain within r of each other at the next point [10]. For the case when the denominator is zero (i.e. each point is more than r away from every other point in the series) then the sample entropy is undefined.

3 Kernel-Based Entropy Measure

The entropy measures introduced so far can be seen as *phase space reconstruction* methods, as $\mathbf{x}_m(k)$ is a delay vector of size m.The set of these, for $k = 1, 2, \ldots, N - m + 1$, is the phase space representation of the signal for dimension m. The next step of the process is to estimate the probability that this path in phase space repeats itself. The calculation of this probability is based on a binary classification of whether two delay vectors are *similar* to each other or not, the degree of similarity allowed being within a tolerance of r. However, although this is conventional in dynamical systems theory, arguably the application of a regularity measure such as this to a time series also falls in the signal processing and pattern recognition domain where it is quite unusual; it is equivalent to estimating a probability density using a uniform noise model and does not consider distances greater than r. In probability density estimation terms, this is a square kernel Parzen window around each point $\mathbf{x}_m(i)$. The common noise model assumption is that of *additive white noise* [1]. The observed value, x_i, is a combination of the underlying latent value, y_i, plus additive Gaussian noise, ϵ thus

$$x_i = y_i + \epsilon. \tag{21}$$

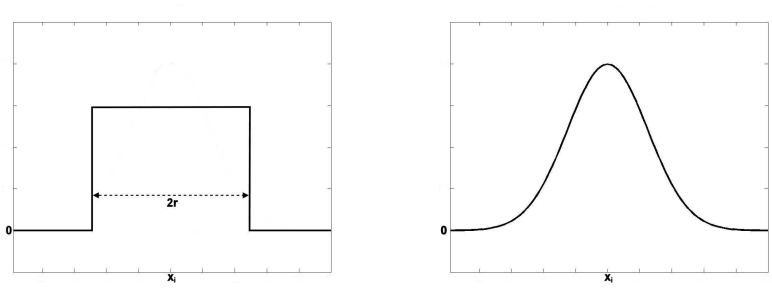
One method to use for probability density function estimation under this assumption is a Gaussian kernel Parzen window.

Using Gaussian kernels instead of square kernels would have obvious benefits, for instance, a higher probability would be assigned to points closer to the fiducial point. Also, it is easy to avoid the pitfalls associated with $\log 0$, as in most density functions, every point has a non-zero density. There is a computational issue if an outlier is so distant that the associated probability falls below computer precision but this can easily be dealt with if we are aware of it.

There are some obvious drawbacks with using Gaussian kernels too. The main one is the computational cost; one of the greatest benefits with using the square kernel method was it was very computationally efficient. However, using some mathematical properties of Gaussians, we can show how the use of Gaussian kernels in an entropy formulation can be reconciled with computational efficiency whilst still retaining a sound analytical justification.

3.1 Parzen Window

A Parzen window is a type of probability density estimation scheme that utilises kernels. A kernel is a parametric density model such as a Gaussian which is



(a) Square Kernel (b) Gaussian Kernel

Figure 1: A graphical representation of the two kernel types for Parzen window probability density estimation.

placed on top of each data point and the full density is evaluated as the sum of the kernels. In our application, we wish to determine the density function for the point $\mathbf{x}_i$. Therefore, our density estimation model can be written as

$$f_P(\mathbf{x}_i) = \frac{1}{N}\sum_{j=1}^{N} K(\mathbf{x}_i - \mathbf{x}_j, h), \tag{22}$$

where h is the window width parameter and K is the kernel function. As with any density function, it is constrained so

$$\int K(\mathbf{y}, h)\mathrm{d}\mathbf{y} = 1. \tag{23}$$

We can see parallels with the methods employed in the entropy measures outlined before. The kernel is the function $d[\mathbf{x}_m(i), \mathbf{x}_m(j)] \leq r$, which in density estimation notation would be written as

$$K(\mathbf{y}, r) = \begin{cases} 1 & \text{if } \max\{\,|y(j+k)| : 0 \leq k \leq N\} \leq r \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

with r corresponding to the window width by $h = 2r$. This can be seen in Figure 1a.

With a Gaussian kernel, the functional form is given as

$$G(\mathbf{y}, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}^T\Sigma^{-1}\mathbf{y}\right), \tag{25}$$

where Σ is the covariance matrix which controls the window width. This kernel is shown in Figure 1b.

The Gaussian kernel has some important properties; in particular, a convolution of two Gaussians yields a Gaussian thus

$$\int G(\mathbf{y}_i - \mathbf{y}_j, \Sigma_1) G(\mathbf{y}_i - \mathbf{y}_k, \Sigma_2) \, \mathrm{d}\mathbf{y} = G(\mathbf{y}_j - \mathbf{y}_k, \Sigma_1 + \Sigma_2). \tag{26}$$

We shall now show how this property can be incorporated into an entropy formulation for computational efficiency whilst retaining analytical justification.

3.2 Renyi Entropy

The family of Renyi entropies are defined as

$$H_{R_\alpha} = \frac{1}{1-\alpha} \log \int p(x)^\alpha \, \mathrm{d}x, \tag{27}$$

where α denotes the order of the entropy, $\alpha > 0$. In the limit $\alpha \to 1$, this is equivalent to the information entropy given in Equation 1.

The use of the term 'entropy' has always been rather loosely used in the approximate entropy family of complexity measures. When ϕ^m is calculated in Equation 8, the measure is simply the logarithm of the probabilities rather than the information entropy or any other standard entropy measure. However, recently it has been noted that the approximate entropy, given in Equation 10, approximates the Renyi entropy of order 1 (the information entropy) and the sample entropy, given in Equation 19, approximates the Renyi entropy of order 2 which is an unbiased estimator [2].

We use the Renyi entropy of order 2 which is termed the *quadratic entropy* as it uses on the second power of the probabilities [12]. Calculating the integral of a squared Gaussian normally would not be computationally feasible for any real world data sets. However, if we use Gaussian kernels in the quadratic entropy, we can use the property from Equation 26 to provide a much more computationally tractable result. For simplicity, we assume that the Gaussians are spherical ($\Sigma = \sigma^2 I$)

$$\begin{aligned} H_{R_2} &= -\log \int p(y)^2 \, \mathrm{d}y_i \\ &= -\log \int \frac{1}{n^2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} G(\mathbf{y}_i - \mathbf{y}_j, \sigma^2 I) G(\mathbf{y}_i - \mathbf{y}_k, \sigma^2 I) \right) \mathrm{d}y \\ &= -\log \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} G(\mathbf{y}_j - \mathbf{y}_k, 2\sigma^2 I). \end{aligned} \tag{28}$$

This means that we can precisely calculate the quadratic Renyi entropy for a probability density estimated using Gaussian kernels with a pairwise sums. This has significant computational benefits and is theoretically sound.

4 Renyi Entropy Rate

The quadratic Renyi entropy can easily be incorporated into the entropy rate framework by using these quadratic Renyi estimates in Equation 5,

$$h_{R_2}(\rho, A) = \lim_{\tau \to \infty} [H_{R_2}(\mathcal{Q}^{\tau+1}) - H_{R_2}(\mathcal{Q}^{\tau})]. \tag{29}$$

For calculating the statistic from finite data, we need to determine the time scale, m, as before, and the width of the Gaussian distribution σ. We can then define an approximation of the Renyi entropy rate as

$$(m, \sigma) = \lim_{N \to \infty} [H_{R_2}^{m+1}(r) - H_{R_2}^{m}(r)], \tag{30}$$

which, when estimated for finite data is defined as

$$(m, \sigma, N) = H_{R_2}^{m+1}(r) - H_{R_2}^{m}(r). \tag{31}$$

We term this the *Kernel Entropy* to distinguish it from other forms of entropy and to highlight the importance of the Gaussian kernels in its formulation.

The Renyi entropy rate has been discussed in a very recent paper to quantify the Gaussianity present in heart rates under various conditions [5]. The approach to estimating the probabilities is based on the method used for the sample entropy in Equation 19, rather than utilising the properties of Gaussian kernels as we have. The paper does provide an interesting insight into properties and applications of the Renyi entropy rate as opposed to the information entropy rate and independently suggests the use of Gaussian kernels would have beneficial properties.

4.1 Selection of the Parameters

Of course, for use on real data, appropriate values of m and σ need to be found. For m, the problem is no different to that in the choice of the parameter for the other entropy approaches. Therefore, for our purposes, we adopt the standard approach of using $m = 2$. However, as there may be benefits in working with different m values, the method should be applicable to as many values as possible.

The same cannot be said for the window width parameter (often referred to as the *bandwidth*). The σ value is greatly different to the r threshold and so a completely new value must be selected for this formulation to perform correctly. Fortunately, there are a number of bandwidth estimation schemes available, although most of them are inappropriate for multivariate problems such as ours as the computation becomes increasingly prohibitive, especially for higher dimensional delay vectors. Because of this, we use a Bayesian approach using Markov Chain Monte Carlo (adapted from [13]).

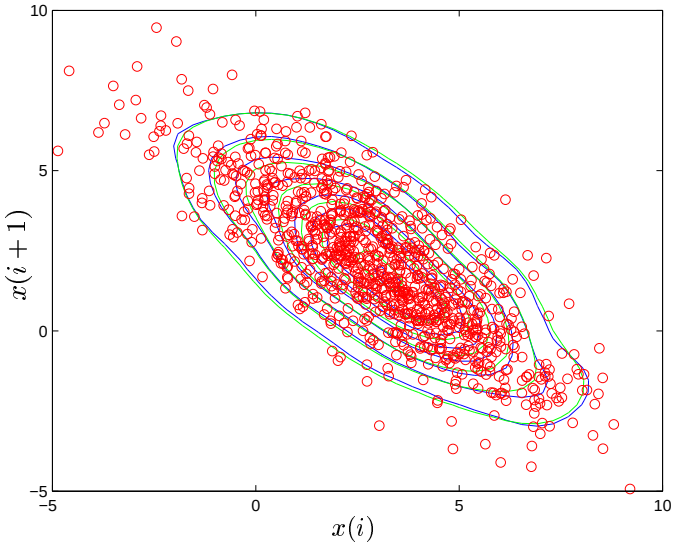


Figure 2: A plot showing the contour probabilities using a Gaussian kernel Parzen window (blue) and the normal reference rule (green) to choose the bandwidth. The 1000 data points (red) are sampled from a two dimensional Gaussian.

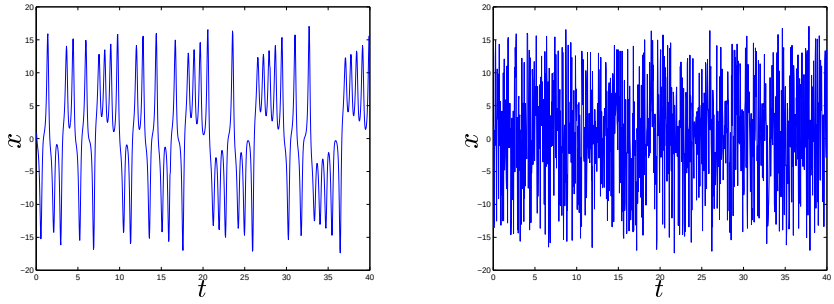
4.1.1 Bayesian Bandwidth Selection

The Bayesian approach in [13] to bandwidth selection treats the components of Σ as parameters and aims to obtain the posterior density of the components of Σ by sampling with the Markov chain Monte Carlo (MCMC) method. As our model assumes that the noise is spherical Gaussian, we can also assume that the bandwidth matrix is diagonal, so $\Sigma = \sigma I$. Using MCMC is beneficial as we want our method to be flexible and the sampling algorithm used can be applied to data of any dimension so we can determine reliable estimates for the bandwidth whatever the value of m is.

The method utilises the Kullback-Leibler (KL) information which is a non-symmetric distance measure between two densities. The aim is to minimise the distance from the target density $f(\mathbf{x})$ to the approximated density $\hat{f}(\mathbf{x})$. The KL information is defined as

$$D_{KL}(f, \hat{f}_\Sigma) = \int \log\left[\frac{f(\mathbf{x})}{\hat{f}_H(\mathbf{x})}\right] f(\mathbf{x})\mathrm{d}\mathbf{x} \tag{32}$$

$$= \int \log f(\mathbf{x}) f(\mathbf{x})\mathrm{d}\mathbf{x} - \int \log \hat{f}_\Sigma(\mathbf{x}) f(\mathbf{x})\mathrm{d}\mathbf{x}, \tag{33}$$



(a) The x value of the Lorentz series

(b) Randomly shuffled x values of the Lorentz series

Figure 3: The x value of the Lorentz series calculated for 1000 iterations and the same values in a random order to remove any time correlation.

which is nonnegative. As the first term in Equation 33 is constant and we do not know the target density, the minimisation of $D_{KL}(f, \hat{f}_\Sigma)$ is the equivalent to the maximisation of $\int \log \hat{f}_\Sigma(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x}$. Using a kernel approximation, $K_\Sigma(\mathbf{y})$ this can be written as

$$\hat{E} \log[\hat{f}_\Sigma] = \sum_{i=1}^{n} \log \hat{f}_\Sigma(\mathbf{x}_i) = \sum_{i=1}^{n} \log \left[\frac{1}{n} \sum_{j=1}^{n} K_\Sigma(\mathbf{x}_i - \mathbf{x}_j) \right]. \tag{34}$$

As the maximisation of this directly leads to a bandwidth matrix of zeros, a leave-one-out cross validation estimator $\hat{f}_{\sigma,i}(\mathbf{x}_i)$ must be used for the cost function in the MCMC method. We start by defining

$$\hat{f}_{\sigma,i}(\mathbf{x}_i) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} |\sigma I|^{-\frac{1}{2}} K\Big([\sigma I]^{-\frac{1}{2}} (\mathbf{x}_i - \mathbf{x}_j) \Big). \tag{35}$$

This forms the likelihood, $L(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n | \sigma)$. However, as we are using a Bayesian approach we need to fix a prior over σ, which in our case is

$$\pi(\sigma_k, i) \propto \frac{1}{1 + \lambda \sigma_k^2}, \tag{36}$$

for $k = 1, 2, \ldots, m$ and where λ is a hyperparameter controlling the shape of the prior density. Therefore, from Bayes theorem, the posterior (up to a normalising constant) is given as

$$\pi(\sigma | \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \propto \left[\prod_{k=1}^{m} \frac{1}{1 + \lambda \sigma_k^2} \right] \prod_{i=1}^{n} \hat{f}_{\sigma,i}(\mathbf{x}_i). \tag{37}$$

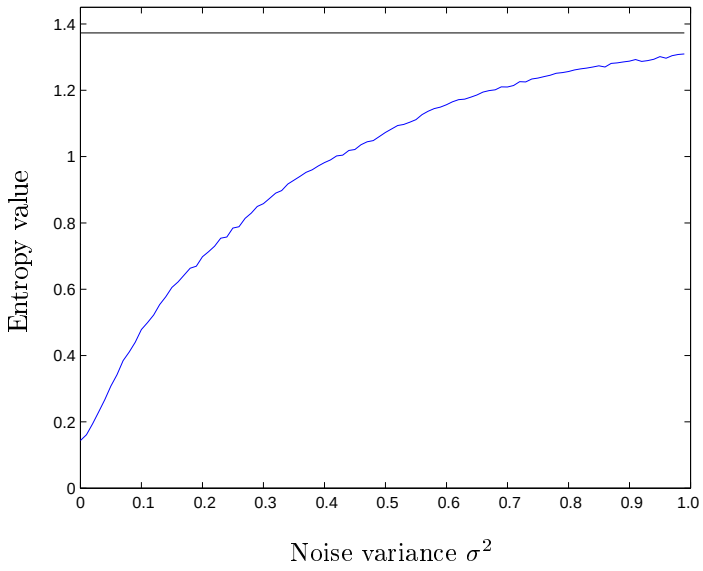


Figure 4: The new entropy measure calculated for the Lorentz series (blue) and the random ordered series (black) for increasing noise variance. The bandwidth is calculated separately for each noise value with the Bayesian MCMC approach.

We sample from this distribution using the Metropolis-Hastings algorithm implemented in NETLAB [7]. The mean of these samples gives us the estimator for the optimal bandwidth.

Figure 2 shows a comparison of the Bayesian method the normal reference rule which is a method of choosing the optimal bandwidth for Gaussian target distributions. The Bayesian method is very close to the optimal bandwidth suggested by the normal reference rule and shows its usefulness in determining the bandwidth. For distributions that are non-Gaussian, the reference rule is of no use but the Bayesian method still determines a good approximation of the optimal bandwidth.

5 Evaluation

5.1 Experiments

Both the kernel entropy and the sample entropy were calculated for the two series shown in Figure 3, the x value of the Lorentz series and the same series with the order randomly shuffled to destroy any time correlation.

Figure 4 shows the result of the new entropy measure calculated for the Lorentz series and the same values randomly reordered. As the noise level is

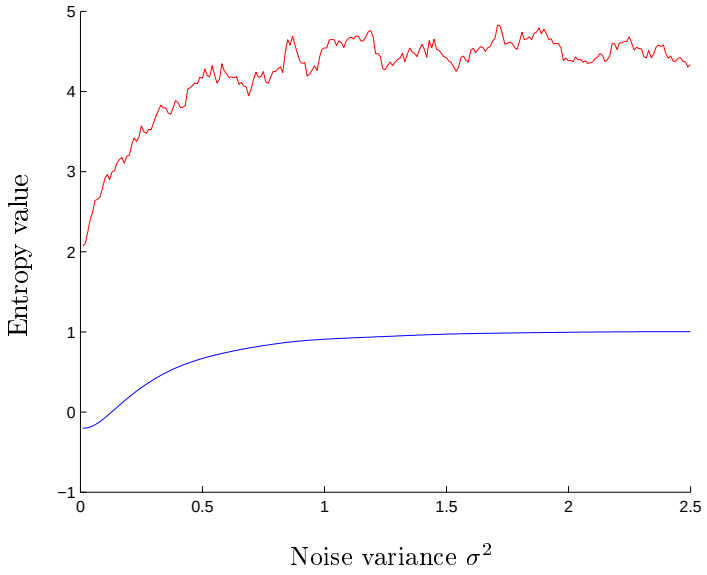


Figure 5: The entropy values for the sample entropy (red) and the kernel entropy (blue) for the Lorentz series with increasing noise.

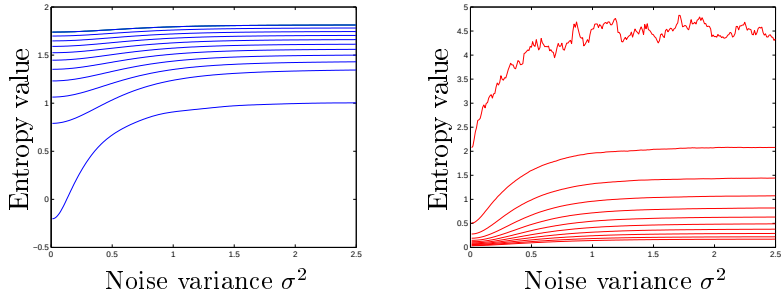
increased, the regularity of the series decreases and therefore the entropy value approaches that of the randomly ordered series. The slight fluctuations that can be seen in the curve are due to the bandwidth being chosen by the Bayesian method as that is based on a stochastic approach and such small irregularities are to be expected.

As comparison of the two entropy measures with arbitrary window width values is meaningless due to the differences inherent in the two different kernels, we compared a wide range of the values for the series with increasing additive white noise.

This is no real drawback as the statistic is still valid regardless of the scale and the scenario can be avoided with correct selection of σ such as with the Bayesian method.

5.2 Discussion

The first thing to notice is that when the kernel size is very small, both statistics behave in an unusual manner as Figure 5 shows. The sample entropy curve is very erratic, due to the small number of matches as the tolerance r is particularly low. In contrast, the new entropy curve is smooth but it does start in a negative value, something which is impossible using SampEn. This is because for a small σ, and low noise, the system is highly ordered and as points are so close to each other, a small fluctuation in their proximity (caused by sampling rate or some



(a) Kernel Entropy

(b) Sample Entropy

Figure 6: The two entropy measures calculated for ten bandwidths from 0.02 to 2 for increasing noise.

other external factor) can leave them with a greatly different value for returned by the Parzen window. Then if the $m+1$th value is close again, the Gaussian will return a higher probability, hence a lower entropy, allowing the negative value to occur. This cannot occur in SampEn as if one value is outside the threshold r it has probability zero for m and $m+1$.

Another point of interest is how increasing the bandwidth size affects the values given by the entropy measures as can be seen in Figure 6. With the sample entropy, a low choice of the bandwidth parameter yields a high value for the output, which is opposite in the the kernel entropy formulation. As the bandwidth size is increased, the sample entropy value gets smaller and the new entropy value gets larger. This is due to the differing nature of the kernels. As the square kernel grows larger it will eventually encompass all the points and so both entropy values for dimensions m and $m+1$ will be the same, giving an overall sample entropy of 0.

6 Conclusions

A new method of approximating the entropy rate from real world data was introduced. The theoretical justification behind the method was shown and how to calculate values using finite data series was suggested.

One of the potential benefits of this method is that the tolerance used in the previous methods is replaced by the bandwidth of a kernel and a suitable mathematical procedure is employed to determine the optimal value. This procedure has the advantage of being applicable to any m value and avoids many of the pitfalls associated with the classical and plug-in bandwidth estimators (which are discussed in more detail in [6]). However, it is important to stress that this method is largely untried and, as with any bandwidth estimator, it

would be inadvisable to assume that it is effective on every dataset. However, as the kernel entropy is more robust to bandwidth choice (it effectively limits the scale) than the tolerance in SampEn, the optimal choice of bandwidth is not as important.

As this report is intended to introduce and provide insight into this new entropy formulation; it is only applied to the Lorentz series, which despite the added noise, is a very ordered system. It is therefore impossible to speak of any benefits/drawbacks in the application over the previous methods with any certainty. To fully judge the effectiveness of the new method, one must apply it to a number of datasets, from real world data to fully deterministic, fully stochastic and mixtures of the two. Any gain in performance would have to be balanced with the computational cost which, especially when using the Bayesian bandwidth selection, is very high.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recgonition*. Oxford University Press, 1995.

[2] M. Costa and J. A. Healey. Multiscale entropy analysis of complex heart rate dynamics: discrimination of age and heart failure effects. In *Computers in Cardiology*, volume 30, pages 705–708, 2003.

[3] J. P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. *Reviews of Modern Physics*, 57(3):617–656, 1985.

[4] R. M. Gray. *Entropy and Information Theory*. Springer-Verlag, 1990.

[5] D. E. Lake. Renyi entropy measures of heart rate Gaussianity. *IEEE Transactions on Biomedical Engineering*, 53(1), 2006.

[6] C. R. Loader. Bandwidth selection: Classical or plug-in? *Annals of Statistics*, 27(2):415–438, 1999.

[7] I. T. Nabney. *Netlab: Algorithms for Pattern Recognition*. Springer, 1999.

[8] E. Ott. *Chaos in Dynamical Systems*. Cambridge University Press, 1993.

[9] S. M. Pincus. Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci*, 88(6):2297–2301, 1991.

[10] J. S. Richman and R Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *Am J Physiol Heart Circ Physiol*, 278(6):H2039–H2049, 2000.

[11] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, 1948.

[12] D. Xu and J. Principe. Learning from examples with quadratic mutual information. In *IEEE Signal Processing Society Workshop*, pages 155–164, 1998.

[13] X. Zhang, M. L. King, and R. J. Hyndman. Bandwidth selection for multivariate kernel density estimation using MCMC. Monash Econometrics and Business Statistics Working Papers 9/04, Monash University, Department of Econometrics and Business Statistics, Apr 2004. available at http://ideas.repec.org/p/msh/ebswps/2004-9.html.