AMSI SUMMER RESEARCH SCHOLARSHIPS 2023–24

SET YOUR SIGHTS ON RESEARCH THIS SUMMER

A Mutual Information-Based Metric for Assessing Partitions of Dynamical Systems

Jason Lu

Supervised by Professor Michael Small

The University of Western Australia

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Prelude

Abstract

The partitioning of dynamical systems is necessary to apply a symbolic dynamics analysis. While many partitioning schemes exist, in practice an ordinal partition is frequently used. We are interested in identifying possible alternatives to ordinal partitioning, however, there is currently no generalised method for assessing the performance of a partitioning scheme on a given system. We propose such a method with a metric that measures the mutual information between each point's symbol history under a coarse partition (one with few partitions), and its partition assignment under a fine partition (one with many partitions). Applying this metric to a set of test systems and partitions shows that it is likely suitable as it generally displays the expected behaviour. The metric shows that partitions based on trajectory history, including the ordinal partition and a newly-proposed coarse history-based partition, are the best performing. Additionally, we find that the latter, as well as a proposed weighted ordinal partition, have the potential to outperform the ordinal partition.

Acknowledgements

I would like to thank Michael for his guidance over the course of this project, and extend this thanks to the rest of the Complex Systems Group. I am also grateful to AMSI for supporting this research and for facilitating the AMSIConnect conference.

Statement of authorship

Professor Michael Small proposed the mutual information-based metric and supervised the project. The author, Jason Lu, implemented code, analysed results and wrote this report.

1 Background and motivation

Symbolic dynamics is the modelling of a discretised dynamical system by splitting its points into sets, each represented by a unique symbol, with applications in noise reduction and developing a general understanding of system behaviour [1]. Symbolic dynamics analysis requires partitioning, which is the process of assigning to each point a symbol based on some partitioning regime. Good partitions are those which produce a symbol sequence retaining a large amount of information of the original system, termed 'high-information'. A generating partition is a particular high-information partition where there is a one-to-one correspondence between each trajectory and its infinite symbolic sequence [5].

Finding generating partitions is difficult and a general method does not exist. There is an existing body of work which attempts to estimate generating partitions for the special case of two-dimensional chaotic maps, such as the Hénon and Ikeda maps [5, 7, 8, 6, 2]. However, these methods are not tested on higher-dimensional or continuous systems.

Ordinal partitions have been used to partition higher-dimensional continuous-time systems with success [1], but superior partitioning methods may exist. Furthermore, there is currently no method for assessing and comparing the performance of different partitioning methods. Thus, in this research, we have two aims: first, to develop a metric that can be used to assess the performance of a partition on a particular system, and second, to identify superior alternatives to ordinal partitioning.

In Section 2, we introduce the mutual information-based metric used to assess partitions. In Section 3 we introduce a set of test cases. In Section 4, the metric is applied to the set of test systems and partitions to verify that it offers the expected assessment of each partition's performance. After this is confirmed, in Section 5, alternatives to ordinal partitioning are explored and assessed.

2 Mutual information-based metric

Suppose we have a sequence of observations taken from a dynamical system with low precision, represented by the area within the black circles in Figure 1. The Shadowing Lemma guarantees that there will be at least one trajectory that passes through these points [4]; three are shown in the figure. Now, consider one of these observations, shown in the bottom half of the figure. With the additional information that the point must satisfy one of the trajectories, the location of the point is now known with higher precision; this higher-precision region is shown inside the red oval. In general, the precision of an observation of a dynamical system can be increased by taking more observations.

A good partitioning regime should then also preserve this property in the symbol sequence it generates. This means that a point's symbol assignment under a fine partition (equivalent to a high-precision observation) should provide information about its symbol history under a coarse partition (equivalent to a series of low-precision observations).

To quantify this property for a given partitioning regime on a dynamical system, we measure the mutual

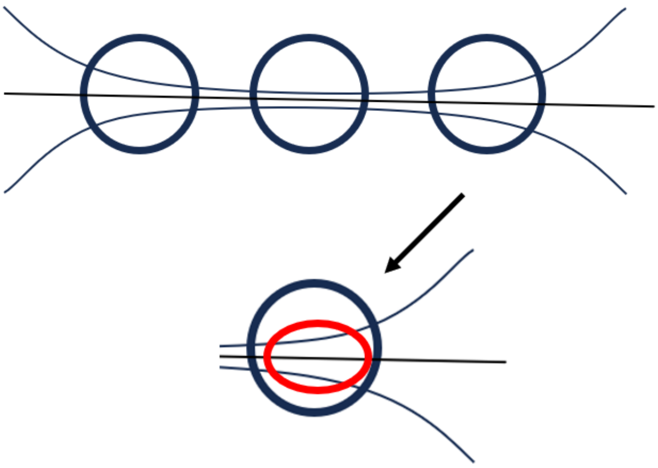


Figure 1: We can effectively increase observation precision by taking more observations.

information between the random variables representing the fine partitions of points, and the history symbol sequences of each point under a coarse partition. This tells us how much knowing the fine partition of a point reduces uncertainty about its history symbol sequence under a coarse partition, and vice versa. The mutual information between random variables X and Y of a joint probability distribution is given by:

$$MI(X,Y) = \sum_{x,y} p_{X,Y}(x,y) \log_2 \left(\frac{p_{X,Y}(x,y)}{p_X(x) p_Y(y)} \right). \tag{1}$$

Formally, suppose $\boldsymbol{P}$ is a partitioning regime that can be applied to produce a variable number of partitions. Let $\boldsymbol{P_k}$ be the partition resulting from applying $\boldsymbol{P}$ to produce k partitions. The algorithm for assessing $\boldsymbol{P}$ on orbit $\boldsymbol{X} = \{\boldsymbol{x}_n\}$ is as follows:

1. Apply $\boldsymbol{P_2}$ to $\boldsymbol{X}$.
2. Assign to each point $\boldsymbol{x}_n \in \boldsymbol{X}$ the coarse history symbol sequence $\pi_{n-d(L-1)}, \pi_{n-d(L-2)}, ..., \pi_{n-d}, \pi_n$, where π_m is the symbol for $\boldsymbol{x}_m$ under $\boldsymbol{P_2}$, and d and L are parameters called history delay and history length respectively.
3. For some $k > 2$, apply $\boldsymbol{P_k}$ to $\boldsymbol{X}$, and assign to each point $\boldsymbol{x}_n \in \boldsymbol{X}$ its corresponding fine partition symbol.
4. Produce the joint probability distribution $P_{X,Y}$, where X is the random variable representing fine partition symbols, and Y is the random variable representing coarse history symbol sequences.
5. Calculate $MI(X,Y)$ as given in Equation 1.
6. Repeat Steps 3 to 5 for various values of k.

Good partitioning regimes have high values of mutual information and result in increasing mutual information as k increases, while bad partitioning regimes have low mutual information and no strong increase in mutual information as k increases.

Parameter d is chosen to be approximately $\frac{1}{4}$ of the period of the orbit for continuous systems [11]. For discrete-time maps, we choose $d = 1$. We set $L = 5$.

3 Test cases

3.1 Test systems

3.1.1 Lorenz system

The Lorenz system is three-dimensional continuous chaotic system defined by the equations:

$$\frac{dx}{dt} = 10(y - x),$$
$$\frac{dy}{dt} = x(\frac{8}{3} - z) - y,$$
$$\frac{dz}{dt} = xy - 28z.$$

Simulations of the system are carried out using a differential equation solver, with time step $\delta t = 0.01$ and initial point $(x, y, z) = (0.5, 0.5, 0.5)$.

3.1.2 Rössler system

The Rössler system is a three-dimensional continuous chaotic system defined by the equations:

$$\frac{dx}{dt} = -y - z,$$
$$\frac{dy}{dt} = x + 0.1y,$$
$$\frac{dz}{dt} = 0.1 + z(x - 14).$$

Simulations of the system are carried out using a differential equation solver, with time step $\delta t = 0.01$ and initial point $(x, y, z) = (0, 0, 1)$.

3.1.3 Hénon map

The Hénon map is a two-dimensional discrete-time chaotic map defined by the equations:

$$x_{n+1} = 1 - 1.4x_n^2 + 0.3y_n,$$
$$y_{n+1} = x_n.$$

Initial point $(x, y) = (0, 0)$ is chosen.

3.1.4 Ikeda map

The Ikeda map is a two-dimensional discrete-time chaotic map defined by the equations:

$$x_{n+1} = 1 + 0.9(x_n \cdot \cos(t_n) - y_n \cdot \sin(t_n)),$$

$$y_{n+1} = 0.9(x_n \cdot \sin(t_n) + y_n \cdot \cos(t_n)),$$

$$t_n = 0.4 - \frac{6}{1 + x_n^2 + y_n^2}.$$

Initial point $(x, y) = (0, 0)$ is chosen.

3.1.5 Duffing oscillator

The Duffing oscillator is a continuous-time system defined by the following equation:

$$\ddot{x} + 0.3\dot{x} - x + x^3 = 0.65\cos(0.65t).$$

Taking x and $\dot{x}$ to create a two-dimensional system generates a periodic orbit following an approximately circular trajectory. Simulations of the system are carried out using a differential equation solver, with time step $\delta t = 0.1$ and initial point $(x, \dot{x}) = (0, -0.3)$. Additionally, Gaussian noise with distribution $(0, \sigma)$ is added to each point during simulation.

3.1.6 Hyperchaotic map

The hyperchaotic map is a three-dimensional discrete-time map that exhibits hyperchaotic behaviour, defined by the following equations:

$$x_{n+1} = 3.8x_n(1 - x_n) - 0.05(y_n + 0.35)(1 - 2z_n),$$

$$y_{n+1} = 0.1[(y_n + 0.35)(1 - 2z_n) - 1](1 - 1.9x_n),$$

$$z_{n+1} = 3.78z_n(1 - z_n) + 0.2y_n.$$

Initial point $(x, y, z) = (0.085, -0.121, 0.075)$ is chosen [10].

3.1.7 Hyperchaotic flow

The hyperchaotic flow is a four-dimensional continuous map that exhibits hyperchaotic behaviour, defined by the following equations:

$$\frac{dx}{dt} = -y - z,$$

$$\frac{dy}{dt} = x + 0.25y + w,$$

$$\frac{dz}{dt} = 3 + xz,$$

$$\frac{dw}{dt} = -0.5z + 0.05w.$$

Simulations of the system are carried out using a differential equation solver, with time step $\delta t = 0.01$ and initial point $(x, y, z, w) = (-20, 0, 0, 15)$ [10].

3.1.8 Mackey-Glass system

The Mackey-Glass system is a one-dimensional series defined by the following delay differential equation:

$$\frac{dx}{dt} = 0.2\frac{x_{t-17}}{1 + x_{t-17}^{10}} - 0.1x.$$

Simulations of the system are carried out using a differential equation solver, with time step $\delta t = 0.05$ and initial point $x = 0.58$.

3.1.9 Time-delay embedding

Time-delay embedding can be applied to a one-dimensional time-series to construct a W-dimensional orbit. Given the one-dimensional time-series $\{x(t)\}$, the time-delay embedded orbit is $\{\boldsymbol{x}(t)\}$, where for each point $\boldsymbol{x}(t) \in \mathbb{R}^W$, $\boldsymbol{x}(t) = (x(t-(W-1)\tau), x(t-(W-2)\tau), ..., x(t-\tau), x(t))$. The values of W and τ are embedding parameters called window size and time-delay respectively. τ is chosen to be approximately $\frac{1}{4}$ of the period of the orbit for continuous systems [11]. For discrete-time maps, we choose $\tau = 1$.

3.1.10 Surrogate data

Surrogate data is used to simulate random orbits. Surrogate data is produced from a given system by randomly drawing points with replacement from the original system in order to remove temporal correlation between observations.

3.2 Partitioning methods

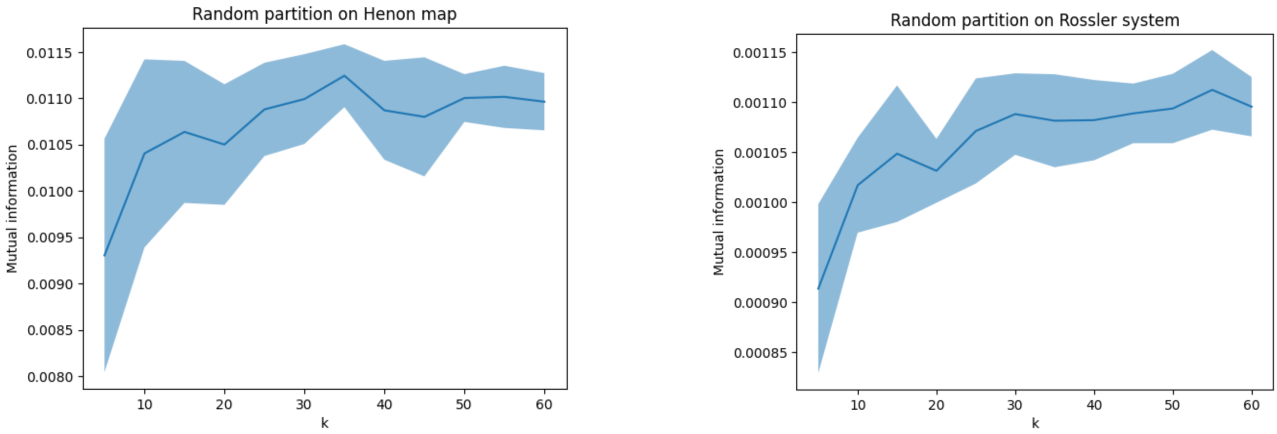
3.2.1 Ordinal partition

Consider the time-series $\{\boldsymbol{f}(t)\}$. To construct an ordinal partition, we first select a dimension of the system $\{x(t)\}$, then apply time-delay embedding to get the embedded orbit $\{\boldsymbol{x}(t)\}$. Each point $\boldsymbol{x}(t) \in \{\boldsymbol{x}(t)\}$ is assigned a symbol according to the rank order of the amplitude of each component. W is varied to vary k, the number of partitions produced. For $W = w$, there are a maximum of $w!$ possible ordinal sequences, and therefore $w!$ partitions. However, in practice, many ordinal sequences are forbidden due to the nature of a system's dynamics. Mapping back to the original state space, each point $\boldsymbol{f}(t)$ is assigned the same symbol as $\boldsymbol{x}(t)$.

Geometrically, the ordinal partition is equivalent to partitioning along the hyperplanes of equality between each pair of dimensions in embedded space, given by $x_t = x_{t-\tau}$, $x_t = x_{t-2\tau}$, $x_{t-\tau} = x_{t-2\tau}$ etc.

3.2.2 K-means clustering

K-means clustering is an unsupervised machine learning method that separates points into K clusters, and aims to minimise the sum of the squared distances between all points and their cluster means [3]. Here, K-means clustering is implemented using the Python library scikit-learn [9], and each cluster is assigned a unique symbol.



(a) Hénon map.

(b) Rössler system.

Figure 2: Mutual information vs k for random partition. The metric shows low mutual information and little increase with k, and is therefore successful in detecting that the random partition is a poor partitioning method.

3.2.3 Slice partition

To apply a slice partition, one dimension of an orbit is selected to produce a one-dimensional time-series $\{x(t)\}$. k evenly sized bins are constructed between $\min(\{x(t)\})$ and $\max(\{x(t)\})$. Each point in the orbit is assigned one of k symbols based on which bin its value in the selected dimension falls into.

3.2.4 Random partition

To apply random partitioning, points are randomly assigned a symbol from a uniform distribution.

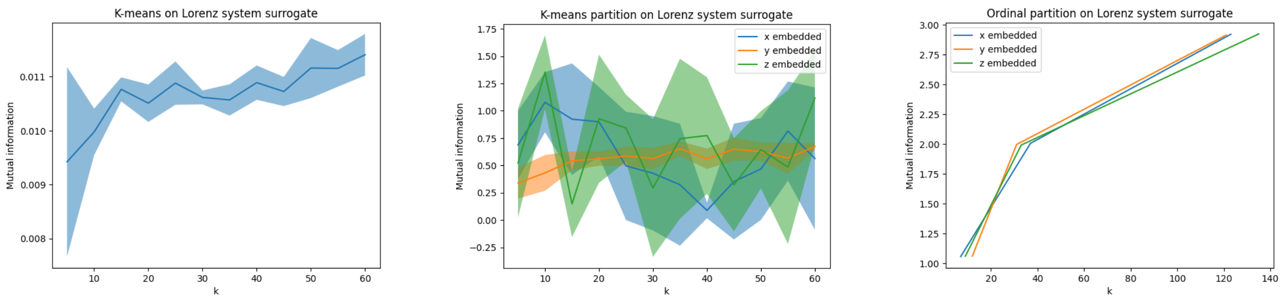
4 Results on test cases

4.1 Random partition

The random partition is known to be a poor partitioning method, as it fails to systematically encode any information. Results from the metric reflect this, with low mutual information and little growth as k increases, as shown in Figure 2 for two test systems. Therefore, here the metric behaves as expected.

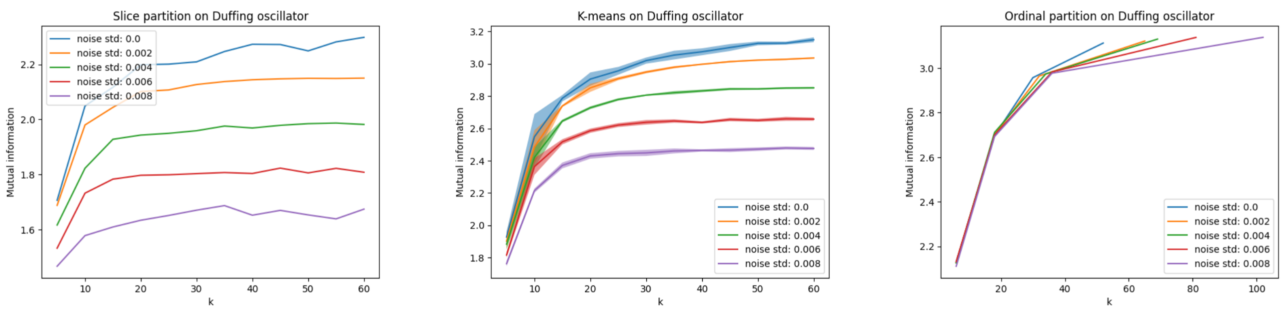
4.2 Surrogate data

As there is no temporal correlation between points in the surrogate data, all partitioning methods are expected to perform poorly. This holds true for the k-means partition in state space, shown in Figure 3a; we take the Lorenz system as an example, but results hold true for surrogate data generated from other systems as well. However, the k-means partition on embedded orbits and the ordinal partition in Figures 3b and 3c respectively show greater mutual information between fine partitions and coarse partition history. Both partitioning methods



(a) K-means in state space. (b) K-means on embedded orbit. (c) Ordinal partition.

Figure 3: Mutual information vs k for Lorenz system surrogate. K-means on the embedded orbit demonstrates higher mutual information than on the orbit in state space, while the ordinal partition demonstrates even higher mutual information as well as an increase with increasing k. These results suggest that partitioning methods utilising trajectory history perform better.



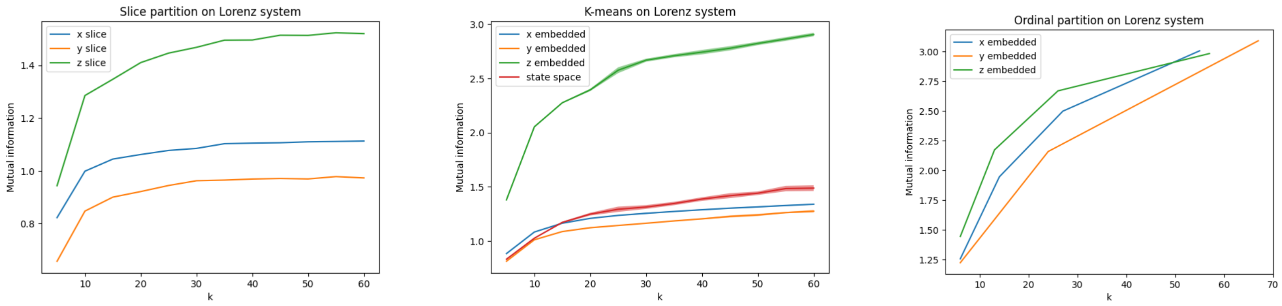
(a) Slice partition. (b) K-means partition. (c) Ordinal partition.

Figure 4: Mutual information vs k for Duffing oscillator with varying amounts of noise. The metric is able to detect that increasing noise reduces partition quality.

involve a time-delay embedding step, so partitions are applied to systems where each point contains information about trajectory history. The process of time-delay embedding then partitioning therefore results in partition assignments sharing information with trajectory history, resulting in higher mutual information, despite the original orbit containing no temporal correlation between successive points. The ordinal partition is explicitly dependent upon the relationship between successive points, while the k-means partition geometrically partitions the embedded orbit; this accounts for the ordinal partition's higher mutual information and consistent increase with k.

4.3 Duffing oscillator

As expected, the slice partition performs worse than the k-means and ordinal partitions, as shown in Figure 4. Additionally the metric detects that increasing the noise parameter σ results in decreasing partition quality, aligning with expectations.



(a) Slice partition. (b) K-means partition. (c) Ordinal partition.

Figure 5: Mutual information vs k for Lorenz system. The slice partition performs worst, followed by k-means then the ordinal partition, aligning with expectations.

4.4 Lorenz system

As expected, the slice partition performs worse than the k-means and ordinal partitions, as shown in Figure 5. The z-slice and z-embedding result in significantly higher mutual information than other dimensions. This is because the Lorenz system is symmetric under inversion through the z-axis, so the z dimension does not offer full observability of the system. The z-embedded orbit is therefore less complex than the original orbit, with only a single lobe as shown in Figure 6, resulting in greater predictability and mutual information.

The exception to this is the ordinal partition, where the z-embedding only slightly outperforms the other dimensions. This is because the partitions in the coarse z-embedded ordinal partition are not contiguous, as shown in Figure 7. This means that transitions between symbols are more difficult to associate with a specific location on the trajectory, resulting in lower mutual information between fine partitions and coarse partition history.

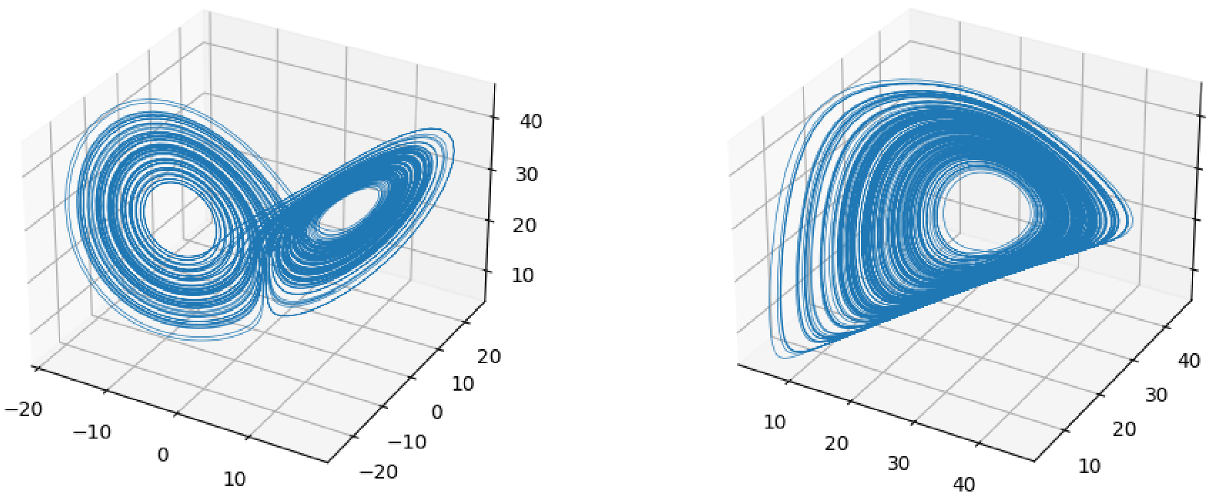
For the Lorenz system, the metric still produces reliable and understandable results. However, it demonstrates that higher mutual information alone does not guarantee a better partition, as such results can also be an artefact of factors such as embedding dimension selection.

4.5 Conclusion

The mutual information-based metric assesses partitions in accordance with expectations, and thus we conclude it is likely a suitable method for determining the quality of a partition.

5 Alternative partitioning methods

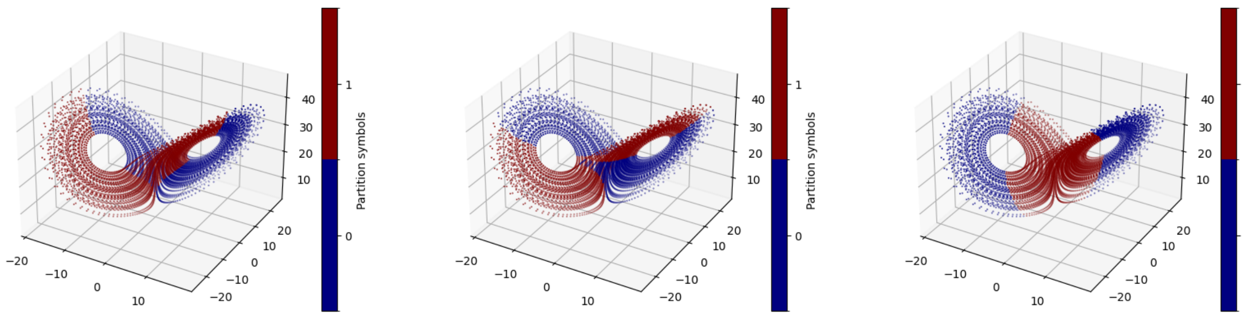
In this section, three alternatives to ordinal partitioning are investigated; all methods are assessed using the mutual information-based metric.



(a) Lorenz orbit in state space.

(b) z-embedded Lorenz orbit.

Figure 6: Lorenz orbit in state space and after z-embedding. The z-embedded orbit exhibits less complex behaviour, resulting in lower mutual information for a given partition.

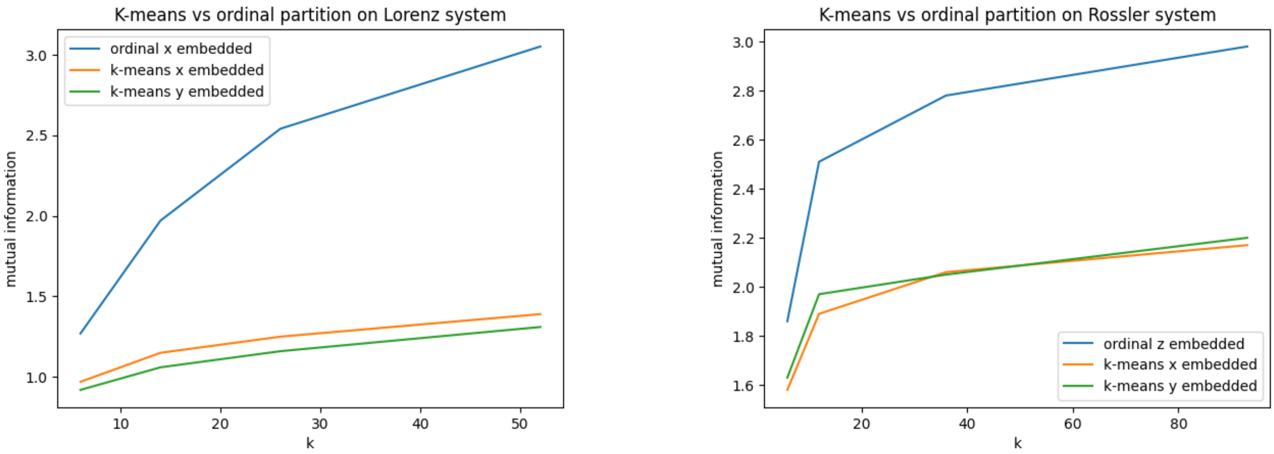


(a) x-embedded.

(b) y-embedded.

(c) z-embedded.

Figure 7: $k = 2$ ordinal partitions on Lorenz system. The z-embedded coarse partition is non-contiguous, resulting in lower mutual information for a given partition.



(a) Lorenz system. (b) Rössler system.

Figure 8: K-means vs ordinal partition on Lorenz and Rössler systems. The ordinal partition outperforms the k-means partition.

5.1 K-means vs ordinal

In this section, k-means and ordinal partitioning are compared on the Lorenz and Rössler systems. Both were tested on various systems in Section 4. However, embedding dimension W was fixed for orbits to which k-means partitioning was applied, while for ordinal partitions, W had to be increased to obtain a greater k value. This gave the ordinal partition the advantage of being able to access a greater amount of history than the k-means partition for higher values of k. To control for this, for each system tested here, the parameters values of k and W used for each trial of the ordinal partition are also applied to the k-means partition.

Results are shown in Figure 8, demonstrating that the ordinal partition still outperforms the k-means partition. Only the best performing embedding dimensions are shown (x and y for k-means for both systems, x ordinal for Lorenz and z ordinal for Rössler). Given that the mutual information-based metric measures the ability of a partition to provide information about trajectory history, it is unsurprising that the ordinal partition, which partitions based on properties of trajectory history, outperforms the k-means partition.

5.2 History-based partition

According to the mutual information-based metric, a strong partition is one in which fine partition symbols provide information about coarse partition history. Assigning fine partitions directly based on coarse partition history should therefore maximise mutual information, producing, from the metric's perspective, an optimal partition. The algorithm for this history-based partition is below:

1. Select a partitioning method (e.g. ordinal) and use it to apply coarse partition $\boldsymbol{P_2}$ to an orbit $\{\boldsymbol{x}(t)\}$.

2. Select a 'look-back dimension' l and time-delay d.

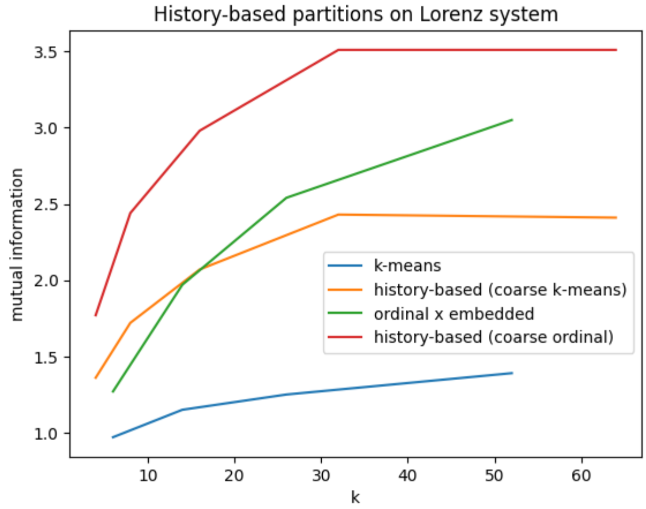


Figure 9: History-based partitions on the Lorenz system outperform conventional partitions.

3. Each point $\boldsymbol{x}(t)$ has coarse symbol history $\boldsymbol{\pi} = \pi_{n-d(l-1)}, \pi_{n-d(l-2)}, ..., \pi_{n-d}, \pi_n$. $\boldsymbol{x}(t)$ is assigned fine partition $\boldsymbol{\pi}$.

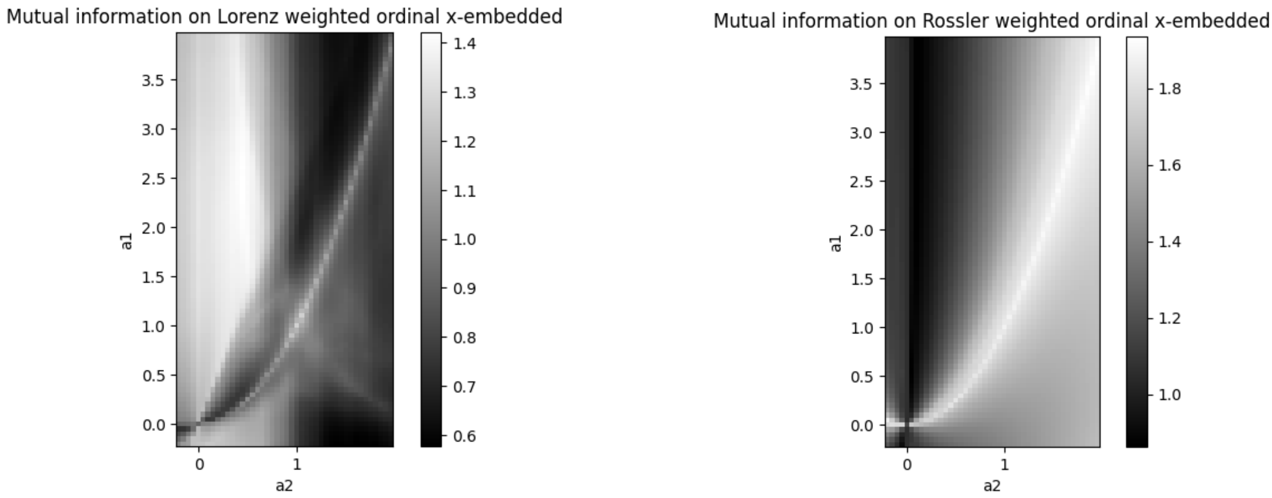
The history-based partition produces $k = 2^l$ fine partitions, and is applied to the Lorenz system with both ordinal and k-means partitions used to apply the initial coarse partition $\boldsymbol{P_2}$. Results are shown in Figure 9, along with the convnetional ordinal and k-means partitions for comparison.

The history-based partitions both outperform the conventional partitioning methods used to produce their coarse partitions. It should be noted that the parameter L, the length of coarse history considered when calculating mutual information, is set to 5. Therefore, once $l = L$ at $k = 2^5 = 32$ partitions, there is a one-to-one correspondence between the fine partition symbols and the coarse symbol history used to calculate mutual information. At this point, increasing l will subdivide existing fine partitions but will not allow them to provide more information about the length-5 coarse partition history; this accounts for the plateau for the history-based partitions at $k = 32$.

We have reason to be skeptical about this partitioning method, as it is vulnerable to the error of optimising for a metric, rather than what the metric is trying to measure. However, these results do support the hypothesis that strong partitions are those that partition points according to their trajectory history, and thus provide greater information about trajectory history under the proposed metric. The ordinal partition outperforms the k-means partition for this reason, and the history-based partition utilises trajectory history to an even greater extent.

5.3 Weighted ordinal partition

Consider a point $x(t)$ with history $x_{t-2\tau}, x_{t-\tau}, x_t$. In a typical ordinal partition with $W = 3$, we assign symbols based on the rank order of the amplitudes of these three points. In a weighted ordinal partition, we first



(a) Lorenz system.

(b) Rössler system.

Figure 10: Sweeping over values of a_1 and a_2. Lighter areas correspond to ordinal weightings with higher mutual information.

apply a weighting to the history of $x(t)$ to obtain the sequence $a_1x_{t-2\tau}, a_2x_{t-\tau}, x_t$, and then perform a regular ordinal partition on this sequence. Geometrically, this is equivalent to changing the gradients of the hyperplanes $x_t = x_{t-\tau}$, $x_t = x_{t-2\tau}$ and $x_{t-\tau} = x_{t-2\tau}$ used to partition the embedded orbit in an unweighted ordinal partition.

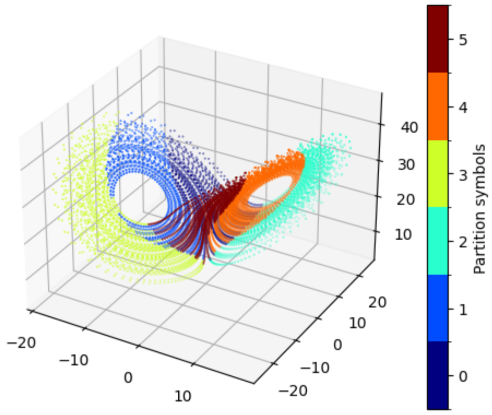
We sweep over values of a_1 and a_2 for both the Lorenz and Rössler systems, with results shown in Figure 10. In both graphs, a quadratic relationship between a_1 and a_2 results in high mutual information; the reason for this is not yet understood.

For the Lorenz system, the standard ordinal partition with $(a_2, a_1) = (1, 1)$ produces a mutual information value of $MI = 1.27$. There are two regions where higher mutual information is observed: the line $a_2 = 0$, with $(a_2, a_1) = (0, 2)$ giving $MI = 1.34$, and the region above and to the left of $(a_2, a_1) = (1, 1)$, with $(a_2, a_1) = (0.5, 2)$ giving $MI = 1.40$.

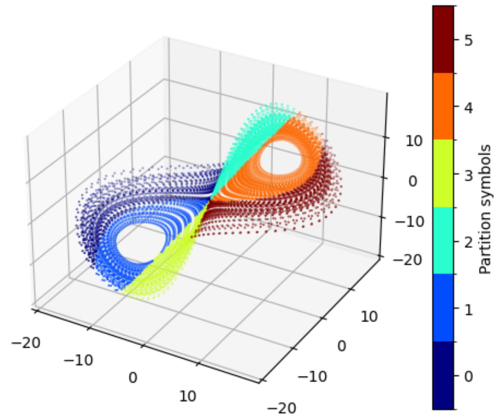
Along the line $a_2 = 0$, ordinal sequences are generated from the points $(a_1x_{t-2\tau}, 0, x_t)$. This means that ordinal sequences 123 and 321 only occur when the Lorenz system crosses the plane $x = 0$. This plane represents an important switching between the two lobes of the system; this weighting of the ordinal partition thus effectively captures an important system behaviour, resulting in high mutual information.

When compared to $(a_2, a_1) = (1, 1)$, the weighting parameters $(a_2, a_1) = (0.5, 2)$ result in a more even distribution of points among the partitions, as Figure 11 illustrates. Given similar partitioning schemes, 'fully utilising' each partition should provide more information per symbol, which can account for the higher mutual information for the weighting $(a_2, a_1) = (0.5, 2)$.

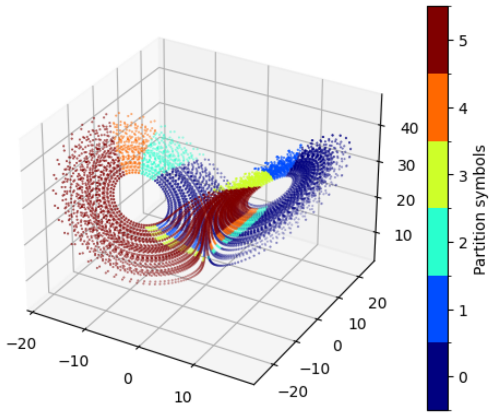
The weighted ordinal partition, combined with analysis made possible by the mutual information-based metric, presents a possible avenue for improving on the ordinal partition by maximising the amount of information preserved.



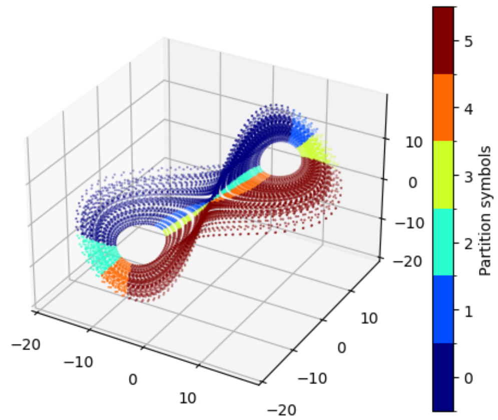
(a) $(a_2, a_1) = (0.5, 2)$, state space.



(b) $(a_2, a_1) = (0.5, 2)$, embedded.



(c) $(a_2, a_1) = (1, 1)$, state space.



(d) $(a_2, a_1) = (1, 1)$, embedded.

Figure 11: Weighted ordinal partitions on Lorenz system with $(a_2, a_1) = (0.5, 2)$ (top) and $(a_2, a_1) = (1, 1)$ (bottom). $(a_2, a_1) = (0.5, 2)$ gives a more even distribution of points among partitions.

6 Conclusions and further work

The mutual information-based metric presented is a suitable method for assessing the quality of a partition; it can continue to be used to compare the performance of other partitioning methods. We find that ordinal partitioning outperforms the k-means partition, as it assigns partition symbols based on trajectory history. The proposed history-based partition performs well under the metric for the same reason, but its usefulness in practice remains to be seen; further investigation is required here. Similarly, the ability for the weighted ordinal partition to offer improvements in certain cases has been demonstrated, but further work is needed to generalise its use.

References

[1] José María Amigó. *Permutation Complexity in Dynamical Systems.* Springer, 2010.

[2] R L Davidchack, Y C Lai, E M Bollt, and M Dhamala. Estimating generating partitions of chaotic systems by unstable periodic orbits. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics*, 61(2):1353–1356, February 2000.

[3] Abir Hadriche, Nawel Jmail, and Ridha Elleuch. Different methods of partitioning the phase space of a dynamic system. *International Journal of Computer Applications*, 93:1–5, 05 2014.

[4] Stephen M Hammel, James A Yorke, and Celso Grebogi. Do numerical orbits of chaotic dynamical processes represent true orbits? *Journal of Complexity*, 3(2):136–145, 1987.

[5] Yoshito Hirata and Kazuyuki Aihara. Estimating optimal partitions for stochastic complex systems. *The European Physical Journal Special Topics*, 222(2):303–315, Jun 2013.

[6] Yoshito Hirata, Kevin Judd, and Devin Kilminster. Estimating a generating partition from observed time series: Symbolic shadowing. *Phys. Rev. E*, 70:016215, Jul 2004.

[7] Matthew B Kennel and Michael Buhl. Estimating good discrete partitions from observed data: symbolic false nearest neighbors. *Phys. Rev. Lett.*, 91(8):084102, August 2003.

[8] David J. Miller, Najah F. Ghalyan, and Asok Ray. A locally optimal algorithm for estimating a generating partition from an observed time series. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2017.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] O.E. Rossler. An equation for hyperchaos. *Physics Letters A*, 71(2):155–157, 1979.

[11] Steven Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering.* Perseus, Reading MA, 1994.

Appendix

A Number of points in orbit simulation

When numerically simulating an orbit, we must choose how many points, n, to simulate. In all calculations, n is varied in proportion with k; this ensures that as k increases, the average number of points per partition remains constant. If n remains constant while k changes, mutual information between symbol history and the fine partition of a point increases as a result of the number of points per partition decreasing.

Practically, n is varied according to the equation $n = a \cdot k$, $k \geq 1000$. a must be chosen such that for the lowest value of k tested, n is sufficiently large to capture the orbit faithfully. Chosen values of a for each system tested are shown below:

System	a
Lorenz	2000
Rössler	20000
Hénon	2000
Ikeda	2000
Hyperchaotic map	2000
Hyperchaotic flow	20000
Mackey-Glass	10000
Duffing	60000

B τ and d values for continuous systems

For continuous systems, τ and d are chosen to be approximately $\frac{1}{4}$ of the period of their orbits. Each system has a different period; values chosen for the tested continuous orbits are shown below:

System	$\tau = d$
Lorenz	0.1
Rössler	1.5
Hyperchaotic flow	1.6
Mackey-Glass	2.5
Duffing	12.5