Memory Order Decomposition of Symbolic Sequences

Unai Alvarez-Rodriguez[1, 2, 3] and Vito Latora[3, 4, 5]

[1] *Basque Centre for Climate Change (BC3), 48940, Leioa, Spain*
[2] *Data Analytics Group, University of Zurich, Zurich, Switzerland*
[3] *School of Mathematical Sciences, Queen Mary University of London, London E1 4NS, UK*
[4] *Dipartimento di Fisica ed Astronomia, Università di Catania and INFN, 95123 Catania, Italy*
[5] *The Alan Turing Institute, The British Library, London, NW1 2DB, UK*

(Dated: June 14, 2021)

We introduce a general method for the study of memory in symbolic sequences based on higher-order Markov analysis. The Markov process that best represents a sequence is expressed as a mixture of matrices of minimal orders, enabling the definition of the so-called memory profile, which unambiguously reflects the true order of correlations. The method is validated by recovering the memory profiles of tunable synthetic sequences. Finally, we scan real data and showcase with practical examples how our protocol can be used to extract relevant stochastic properties of symbolic sequences.

I. INTRODUCTION

Symbolic sequences are ubiquitous in many domains of science. For instance, we use sequences of symbols to encode the sounds that constitute our different languages, to disentangle the complexity of DNA molecules responsible of our genetic information, and also to characterize temporal evolution of physical systems. In this context, memory can provide key information about a sequence and the process generating it, as it represents the distance between causally related elements in the sequence. The Markov chain formalism [1–4], allows for an approximation of the generating process by means of a maximum likelihood estimator of a given memory or order. The problem of extracting the order of a generating process has been addressed by means of different information criteria [5–16] (belonging to the more general field of model selection [17, 18]), which provide estimates of the maximal order as a function of the likelihood and the number of parameters involved in the model.

Various higher-order models have been proposed as a formal way to analyze memory in a complex system [19–25]. These models allow to go from a time-aggregated perspective to a dynamics that respects the time-ordering of interactions. Recently, there is a growing interest for combining the statistics of different orders into a single model, see [26, 27] for multi-order models, and see [28] for a decomposition of transition probabilities in terms of generalized memory functions. However, a general and analytic framework for understanding the nested nature of memory is still missing.

In this manuscript we address this gap by introducing the novel concept of memory profile of a stochastic process, and designing an algorithm that captures the length specific correlations present in the temporal evolution of a system. Our method decomposes a Markov process into a convex sum of stochastic matrices, where the memory profile arises naturally as the set of coefficients of the subprocesses. The algorithm detects which of the correlations at a given length are spurious, by explaining them as a combination of subprocesses of lower orders. We finally validate the method on synthetic sequences and we illustrate how it works in practice to extract the memory profiles of real data coming respectively from literary texts, biology and deterministic chaotic systems.

Let us start with an empirically observed sequence $S = (s_1, s_2, \ldots, s_L)$ of length L, where the generic symbol s_i, with $i = 1, 2, \ldots, L$, is selected from an alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_A\}$ of A characters. We assume S has been created from an unknown Markov process $\mathcal{Q}$ that we call source, which can be expressed as a stochastic matrix Q. In order to study the statistical properties of $\mathcal{Q}$ through S, let us define as $x^m = (x_1, \ldots, x_m)$ an ordered string of m elements of $\mathcal{A}$. Let us also denote as s_i^m the string of length m terminating at position i in S. For simplicity we will refer to strings of length 1, single elements, with a reduced notation, using s_i instead of s_i^1. Each of the s_i^m corresponds to a unique string x^m, while the probability of finding a particular x^m in S is given by $p(x^m) = \frac{f(x^m)}{L-m+1}$, where $f(x^m)$ denotes the number of appearances of x^m in S. Assuming that sequence S is generated by Q of order m, or Q^m, the *transition probabilities* can be estimated as:

$$\pi(x_m|x_1, \ldots, x_{m-1}) = \frac{f(x^m)}{f^-(x^{m-1})} \tag{1}$$

where the reduced frequencies $f^-(x^m)$, are equal to $f(x^m)$, if $x^m \neq s_L^m$, or to $f(x^m) - 1$ if $x^m = s_L^m$.

The transition probabilities can now be organized in a $A \times A^{m-1}$ transition matrix, T^m, which, for a given order m, contains the probabilities $\pi(x_m|x_1, \ldots, x_{m-1})$ in Eq. (1) to get any of the A symbols after each of the possible ordered combinations of A^{m-1} symbols. See Appendix-A for an example of the matrix notation.

A measure of how good a model T^m of order $m > 0$ is for the observed sequence S can be obtained by the likelihood function:

$$\ell(T^m, S) = p(s_1) \prod_{i=2}^{m-1} \pi(s_i|s_{i-1}^{i-1}) \prod_{i=m}^{L} \pi(s_i|s_{i-1}^{m-1}) \tag{2}$$

where the transition probabilities are computed with the x^m associated to each s_i^m. The likelihood is a non-decreasing function of m, and estimating the maximal order M_Q of Q as the value of m at which ℓ is maximal will lead to overfitting. To cope with that we extract M_Q via the Akaike information criterion (AIC), which formalizes the intuitive idea of a trade-off between the number of parameters and the performance of a model [5].

II. MEMORY ORDER DECOMPOSITION

A. Nested models

If a sequence is perfectly described by a model of order m, the transition probabilities at $m+1$ should return the same description. Let us address this problem by defining $T^{m[+]1}$, a prediction of how T^{m+1} should be, assuming $T^m = Q$. In practice, a transition matrix T^m from an alphabet $\mathcal{A}$ is extended by taking the tensor product $T^{m[+]j} = \mathbf{1} \otimes T^m$ with a row vector $\mathbf{1}$ of length A^j with all the components equal to 1. We can now compute the distance d between T^{m+1} and $T^{m[+]1}$ to test the robustness of the $T^m = Q$ hypothesis. We will be using $d = 1 - \sigma$, where σ measures the overlap between two discrete probability vectors u and v of dimension D, and it is expressed as $\sigma(u, v) = \sum_i^D \min(u_i, v_i)$. By construction $\sigma(u, v)$ is bounded to the $[0, 1]$ interval, with $\sigma = 1$ if $u = v$. This definition of d is equivalent to a normalized vector distance, as we show in Appendix-B. In our case we deal with matrices, so we will be calculating the statistical distance for each of the columns, and afterwards weighting the contribution of columns α, with $\alpha = 0, \ldots, A^{m-1} - 1$, by the probability $p(x_\alpha^{m-1})$ of finding the string corresponding to column α in S. The final expression is:

$$\sigma(T^{m+1}, T^{m[+]1}) = \sum_{\alpha=0}^{A^m-1} p(x_\alpha^m) \sum_{\beta=0}^{A-1} \min(T_{\beta\alpha}^{m+1}, T_{\beta\alpha}^{m[+]1}) \tag{3}$$

B. Decomposition

Building on the same idea, it is possible that a matrix T^m has a non-zero overlap with its predecessors $T^{m-1}, \ldots, T^0$, implying that in the procedure of generating S not all of the new elements depend on the previous M_Q elements, some of them could have required much shorter strings, i.e. a shorter memory. In this case, the very same idea of the true order of $\mathcal{Q}$ would be misleading. We will now show that it is possible to extract the memory profile, i.e., the relevance of each order $m \leq M_T$ in T^{M_T}, by adopting a matrix decomposition procedure as it follows. In general, any column-stochastic matrix, such as the transition matrix T^{M_T}, can be decomposed as a linear combination of deterministic processes of different orders, i.e. column-stochastic Boolean matrices C_i^m of dimension $A \times A^{m-1}$ as:

$$T^M = c_0^0 C_0^{0[+]M_T-1} + \sum_{m=1}^{M_T} \sum_{i=1}^{\mathcal{C}^m} c_i^m C_i^{m[+](M_T-m)} \tag{4}$$

where $\mathcal{C}^m$ stands for the number of deterministic processes at each order, the coefficients c_i^m are real numbers weighting the different contributions, and the $m = 0$ process corresponds to a uniform model. The latter assigns an equal probability of $1/A$ to all the symbols in $\mathcal{A}$, and is considered separately from the others processes since C_0^0 is not Boolean. In order to visualize the total contribution of each order, we define the *memory profile* of the transition matrix T^{M_T} as the vector $\mathbf{t}$ whose components t_m, with $m = 0, \ldots, M_T$, are given by $t_m = \sum_i^{\mathcal{C}_m} c_i^m$. Conversely, we say that q_m represents the memory profile of the original process $\mathcal{Q}$. The particular form of the deterministic matrices C_i^m allows a one-to-one correspondence with natural numbers. In fact each of the columns of any of our C_i^m contains a single nonzero element which is equal to 1. The position of this element can be associated to a term in a power expansion base A, where the row accounts for the coefficient and the column for the power. If C_i^m has elements $e_{\alpha\beta}$, the associated number n_i^m is $n_i^m = \sum_{\alpha=0}^{A-1} \sum_{\beta=0}^{A^{m-1}-1} e_{\alpha\beta} \alpha A^\beta$, while $n_0^0 = 0$ for the uniform model. Index m in n_i^m is necessary to avoid redundancies between processes at different orders with the same associated number (See Appendix-C).

Our goal for the mixture in Eq. (4) is to have $c_i^m = 0$ for all the matrices that are the extension of a lower order matrix, i.e., reducible matrices. The standard procedure for identifying these matrices is to test whether they correspond to the tensor product of a lower order matrix. Alternatively the mapping to natural numbers introduced above allows to simplify the problem, as the natural number n_i^m associated to a process inherits its order properties. It is then enough to check whether n_i^m is divisible by a given number to prove that C_i^m has true order m. Let n_i^m be the number associated to a given process C_i^m and let $C_i^{m[+]1}$ and $n_i^{m[+]1}$ be its extension to the next order, and the number associated with it. We have:

$$n_i^{m[+]1} = n_i^m \sum_{\alpha=0}^{A-1} A^{\alpha A^{m-1}} = n_i^m \frac{A^{A^m} - 1}{A^{A^{m-1}} - 1} \tag{5}$$

This formula provides a simple reduction mechanism: a given number n_i^m has a true order m if it is not divisible by $\frac{A^{A^{m-1}}-1}{A^{A^{m-2}}-1}$, otherwise it can be reduced. This check is then repeated until the number is found to be not divisible. The order in which the reduction process terminates is the true order of the process associated with this number (see Appendix-D).

C. Algorithm

The decomposition algorithm we propose here consists of an iterative procedure that, at each step, identifies the process with the maximal coefficient and removes it from the matrix to be decomposed. The transition to a higher order is produced after ensuring that no more processes can be added to the decomposition. See Appendix-E for a fully detailed example of the algorithm.

The procedure is equivalent for each step, so let us suppose the matrix we want to decompose is T^{M_T} which can represent the original transition matrix or any of its intermediate steps of decomposition. Let us also suppose that we are currently exploring the matrices at a generic order m. The first step is to create a reduced matrix from T^{M_T}, with the dimensions of the matrices at order m, $A \times A^{m-1}$. When T^{M_T} is reduced to order m, each of the elements of the reduced matrix R^m is fed with the elements of T^{M_T} that correspond to its extension. The specific R^m we are looking for, is the one where each matrix element is the minimal of all the elements in T^{M_T} that correspond to the tensor product extension.

$$R^m_{ij} = \begin{cases} \min_{\alpha\beta} T^{M_T}_{\alpha\beta} & \text{if } m = 0 \\ \min_{\beta}\{T^{M_T}_{i\beta} | \beta \equiv j \mod A^{m-1}\} & \text{if } m \neq 0 \end{cases} \tag{6}$$

The second step is to select the matrix to be incorporated into the decomposition. This is done by finding the maximum in each of the columns of R^m. The matrix will be the one whose non-zero elements are located in the position of the maximum values. If there is more than one maximum, there is not a unique possible matrix, and we say that the process is degenerate.

The third step is to detect the coefficient, c^m_i, of the process we have just found, C^m_i. The idea is that the enlarged form of the matrix, weighted with its corresponding coefficient, $c^m_i C_i^{m[+](M_T-m)}$ is subtracted from T^{M_T}. Therefore, in order to have the maximum of the non-negative outcomes, the coefficient has to be the minimum of the set of maximum column values in R^m.

$$c^m_i = \min_{\beta} \max_{\alpha} R^m_{\alpha\beta} \tag{7}$$

As anticipated, the fourth step is to remove $c^m_i C_i^{m[+]M_T-m}$ from T^{M_T}. The result of this process is a new T^{M_T}, which is the output of the current cycle and the input of the following one. The next cycle should repeat the search in m, unless the just found coefficient is $c^m_i = 0$. This would mean that no more processes are compatible with T^{M_T} at m. If the previous condition is true, the value of m has to be updated to $m + 1$.

D. Validation

We carry out a systematic validation procedure on ensembles of synthetic sequences with different alphabets, maximal orders, and lengths. We have used two indicators (v_1,v_2), each of them a real number in $[0, 1]$, to evaluate the performance of our method: v_1 accounts for the success of the AIC in retrieving the maximal order of M_Q, and v_2 measures the overlap $\sigma(q_m, t_m)$ between t_m and q_m. Here, a sequence is generated by randomly constructing a column-stochastic matrix Q^{M_Q} for each (L, M_Q, A) triplet. The output indicators v_i are averaged over 100 realizations of different experiments with the same (L, M_Q, A) values. Therefore, v_1 is the fraction of times $M_T = M_Q$ and v_2 is the average $\sigma(\mathbf{q}, \mathbf{t})$, where $\mathbf{q}$ is the real memory profile.

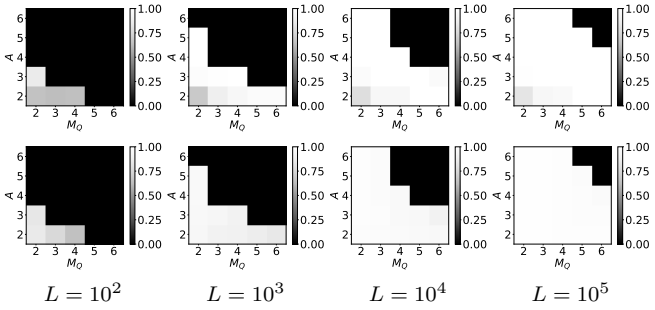

FIG. 1: Validation of the algorithm on synthetic sequences with different values of length L, maximal order M_Q and number of symbols A. The rows report respectively the computed values of v_1 and v_2 with a color code, while different columns refer to sequences with different values of L. Results are obtained as averages over 100 different realizations.

The results are shown in Fig. 1. The rows refer respectively to v_1, v_2. The color of each cell denotes the average values of v_i as a function of M_Q and A, and the columns account for four different sequence lenghts. Notice from the figure that the reliability of the order detection is affected by the length of the sequence and that of the alphabet, in the sense that we impose an upper bound in m that guarantees a minimal frequency for the strings of T^{m+1}. See Appendix-F for the specific details of our treatment of low frequency strings. Below such threshold the behaviour of the algorithm is satisfactory even for small values of L. It is noteworthy to mention that the errors in small M_Q at v_1 are compensated in v_2. In other words, even in the cases in which the AIC fails the complete algorithm succeeds in extracting the memory profile.

III. APPLICATION TO REAL SEQUENCES

We have extracted the memory profile of sequences from biology, literary texts and chaotic systems. We have selected these datasets because they correspond to alphabets that we have tested with synthetic sequences, and because each of these examples showcases a new feature of sequence analysis enabled by our protocol: the true

memory allocation across different orders, the non-trivial ranking of subprocesses and the finiteness of the number of subprocesses involved in a higher-order Markov chain decomposition. In this sense, our goal here is not to address domain specific questions. Again, we impose an upper-bound to the highest memory order m to ensure that the correlations that we find are not an effect of the finiteness of the data (See Appendix-F).

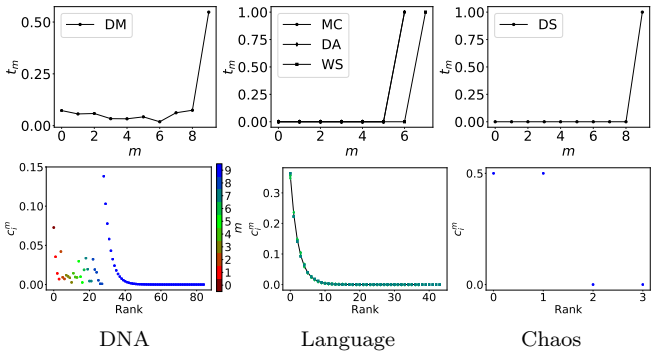

FIG. 2: Memory profile t_m (top panels) and matrix decomposition (bottom panels) of real sequences (genetic material, text from literature classics, and the Dragon sequence). In the bottom panels we plot the weights c_i^m of the decomposition in Eq. (4), sorted in decreasing order for each value of m.

A. DNA

We have studied the second chromosome of the fruit fly *Drosophila melanogaster* (DM) by selecting a DNA sequence of length 1.7×10^7 from an alphabet $\mathcal{A} = \{A, C, T, G\}$ of four letters. The first column of Fig. 2 shows that even if the estimated order is $M_T = 9$, an important fraction of the information of the higher-order markov chain is contained in subprocesses of lower orders. Therefore, roughly half of the correlations that one would associate to statistics at $m = 9$ are spurious and can be reduced. DNA is known to exhibit long-range correlations [29–33], however these appear for orders much higher than the maximal order studied here, and therefore testing whether they can be reduced would require a reformulation of the algorithm as described in Appendix-F.

B. Language

We have translated three prominent literature classics (*Don Quijote de la Mancha* MC, *La Divina Commedia* DA and *Hamlet* WS) into Morse Code using the alphabet $\mathcal{A} = \{$".","-"," "$\}$ of only 3 symbols obtaining respectively sequences of lengths 8.7×10^6, 2.4×10^6 and 7.4×10^5. In all three cases the maximal order M_T coincides with the security cut-off and the process is fully dominated by subprocesses of maximal order $m = M_T$, suggesting that the real order could be larger. Moreover, we have found that the coefficients of the subprocesses follow an exponential probability function, which uncovers a ranked organization of the building blocks of language when expressed in morse code that goes beyond the Zipf's distribution for the frequencies of words [34].

C. Deterministic Chaos

As last example we considered the Dragon Curve (DS), a deterministic process with fractal properties [35]. We have generated sequences of length $L = 5.2 \times 10^5$ (18 iterations) with an alphabet $\mathcal{A} = \{L, R\}$ of two letters representing the directions in the rotation of the dragon, either left or right. The decomposition shows that from all the possible processes at $M_T = 9$, only four are present, of which two dominate the transition matrix. This shows how despite the complexity and the number of parameters of higher-order markov chains, some processes may be decomposed with a small number of subprocesses. This implies that the model has a very low entropy, as it can be compressed in just two numbers, and is able to capture the deterministic nature of the original sequence.

IV. CONCLUSION

In conclusion, we have proposed a method to represent the mechanism generating a sequence of symbols as a mixture of processes of well defined orders. This enables to determine the memory profile of the underlying Markov process, which is an efficient way of characterizing the causal relations hidden in the sequence. We hope our method will become a standard tool in the analysis of high-order Markov chains.

ACKNOWLEDGEMENTS

U.A.-R. acknowledges support from the Spanish Government through Maria de Maeztu excellence accreditation 2018-2022 (Ref. MDM-2017-0714), from the Basque Government through the Posdoctoral Program (Ref. POS-2017-1-0022) and from the Swiss National Science Foundation (Ref. 176938). V. L. acknowledges support from the EPSRC project EP/N013492/1 and from the Leverhulme Trust Research Fellowship "CREATE: the network components of creativity and success".

[1] J.D. Hamilton, *Time Series Analysis*, (Princeton University Press, Princeton, 1994)

[2] A. A. Markov, Science in Context **19**, 591 (2006).

[3] E. Seneta, *Non-negative matrices and Markov chains* (Springer, New York, 1981).
[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley, New Jersey, 2006).
[5] H. Akaike, IEEE Trans. Autom. Cont. **19**, 716 (1974).
[6] G. Schwarz, Ann. Statist. **6**, 461 (1978).
[7] M. Buiatti, P. Grigolini, and L. Palatella, Physica A. **268**, 214 (1999).
[8] L. C. Zhao, C. C. Y. Dorea, and C. R. Gonçalves, Stat. Infer. Stoch. Process. **4**, 273 (2001).
[9] D. Dalevi and D. Dubhashi, Lect. Notes Comp. Sci. **3692**, 291 (2005).
[10] L. Pardo, *Statistical Inference Based on Divergence Measures* (Chapman and Hall, New York, 2006).
[11] R. Sinatra, D. Condorelli, and V. Latora, Phys. Rev. Lett. **105**, 178702 (2010).
[12] M. Menéndez, L. Pardo, M. Pardo, and K. Zografos, Methodol. Comput. Appl. Probab. **13**, 59 (2011).
[13] M. Papapetrou and D. Kugiumtzis, Physica A **392**, 1593 (2013).
[14] A. Baigorri, C. Gonçalves, and P. Resende, Can. J. Statist. **42**, 563 (2014).
[15] D. Pethel and W. Hahs, Physica D **269**, 42 (2014).
[16] M. Papapetrou and D. Kugiumtzis, Simul. Model. Practice Theory **61**, 1 (2016).
[17] K. P. Burnham and D. R. Anderson, Sociol. Meth. Res. **33**, 261 (2004).
[18] G. Claeskens and N. L. Hjort, *Model Selection and Model Averaging* (Cambridge University Press, Cambridge, 2008).
[19] P. Holme and J. Saramäki, Phys. Rep. **519**, 97 (2012).
[20] R. Pfitzner, I. Scholtes, A. Garas, C. J. Tessone, and F. Schweitzer, Phys. Rev. Lett. **110**, 198701 (2013).
[21] M. Rosvall, A. V. Esquivel, A. Lancichinetti, J. D. West, and R. Lambiotte, Nat. Comm. **5**, 4630 (2014).
[22] I. Scholtes, N. Wider, R. Pfitzner, A. Garas, C. J. Tessone, and F. Schweitzer, Nature Comm. **5**, 5024 (2014).
[23] L. Lacasa, I. P. Mariño, J. Miguez, V. Nicosia, E. Roldán, A. Lisica, S. W. Grill, and J. Gómez-Gardeñes, Phys. Rev. X **8**, 031038 (2018).
[24] R. Lambiotte, M. Rosvall, and I. Scholtes, Nat. Phys. **15**, 313 (2019).
[25] O. E. Williams, F. Lillo, and V. Latora, New J. Phys. **21**, 043028 (2019).
[26] I. Scholtes, in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 17, (ACM, New York, 2017). p. 1037.
[27] C. Gote, G. Casiraghi, F. Schweitzer, and I. Scholtes, Preprint at arXiv:2007.06662 (2020).
[28] S. S. Melnik and O. V. Usatenko, Phys. Rev. E **96**, 012158 (2017).
[29] C.-K. Peng, S. V. Buldyrev, A. L. Goldberger, S. Havlin, F. Sciortino, M. Simons, and H. E. Stanley, Nature **356**, 168 (1992).
[30] R. N. Mantegna, S. V. Buldyrev, A. L. Goldberger, S. Havlin, C.-K. Peng, M. Simons, and H. E. Stanley, Phys. Rev. Lett. **73**, 3169 (1994).
[31] A. Arneodo, E. Bacry, P. V. Graves, and J. F. Muzy, Phys. Rev. Lett. **74**, 3293 (1995).
[32] S. V. Buldyrev, A. L. Goldberger, S. Havlin, R. N. Mantegna, M. E. Matsa, C.-K. Peng, M. Simons, and H. E. Stanley, Phys. Rev. E **51**, 5084 (1995).
[33] P. Allegrini, M. Barbi, P. Grigolini, and B. J. West Phys. Rev. E **52**, 5281 (1995).
[34] M. E. J. Newman, Contemporary Physics **46**, 323 (2005).
[35] J.-P. Allouche and J. Shallit, *Automatic Sequences: Theory, Applications, Generalizations* (Cambridge University Press, Cambridge, 2003).

APPENDIX

A. Matrix Notation

As a simple case to illustrate our notation, let us consider a sequence S of symbols from an alphabet with $A = 2$ and with symbols $\{0, 1\}$. We first need to construct matrices T^m with $m = 0, 1, \ldots$ from the transition probabilities in Eq. (1) in the main text. Suppose the matrix for $m = 3$ reads:

$$T^3 = \begin{pmatrix} 0.1 & 0.8 & 0.3 & 0.6 \\ 0.9 & 0.2 & 0.7 & 0.4 \end{pmatrix} \tag{8}$$

This means that $\pi(0|00) = 0.1$, $\pi(1|00) = 0.9$, $\pi(0|01) = 0.8$, $\pi(1|01) = 0.2$, $\pi(0|10) = 0.3$, $\pi(1|10) = 0.7$, $\pi(0|11) = 0.6$ and $\pi(1|11) = 0.4$.

B. Statistical Distance

Let us see how σ is equivalent to the normalized norm of the difference vector.

$$\begin{aligned} d &= \frac{1}{2}\sum^{D}|x_i - y_i| = \frac{1}{2}\sum^{D}\max(x_i, y_i) - \min(x_i, y_i) \\ &= \frac{1}{2}\sum^{D} x_i + y_i - 2\min(x_i, y_i) \\ &= \frac{1}{2}\left(\sum^{D} x_i + \sum^{D} y_i - 2\sum^{D}\min(x_i, y_i)\right) \\ &= \frac{1}{2}(1 + 1 - 2\sum^{D}\min(x_i, y_i)) = 1 - \sum^{D}\min(x_i, y_i) \\ &= 1 - \sigma \end{aligned} \tag{9}$$

Up to this point, it seems unnecessary to make use of an alternative definition, if this is equivalent to the standard one. The reason supporting our decision is clarified when working with more than two distributions. If we add a new one, z, the overlap or intersection is calculated as

$$\sigma(x, y, z) = \sum_{i}^{D}\min(x_i, y_i, z_i) \tag{10}$$

The same can be done employing the normalized vector distance, but not in such simple manner.

C. Natural Label

Let us see how the mapping works in an example with $A = 2$, $m = 3$ and $n = 9$. The idea is to retrieve n from the matrix expression.

$$C_9^3 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \tag{11}$$

As introduced in the main text, the natural label formula is given by

$$n_i^m = \sum_{\alpha=0}^{A-1}\sum_{\beta=0}^{A^{m-1}-1} e_{\alpha\beta}\alpha A^{\beta} \tag{12}$$

where $e_{\alpha\beta}$ are the elements of C_i^m. Since C_i^m are stochastic boolean, there is a single non-zero element per column, and therefore, the first summation can be reduced to the rows α, such that $e_{\alpha\beta} = 1$. Following this expression we have

$$n^3 = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 9 \tag{13}$$

D. Number Reduction and Extension Mechanism

Let us first consider the extension of the matrix of the previous section in Eq. (11)

$$C_9^{3[+]1} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{14}$$

The associated number $n_9^{3[+]1} = 153$, is computed by either using Eq. (12) from the appendix or Eq. (5) from the main text. Since, the first option has already been explained in the previous section we go for the second one.

$$n_9^{3[+]1} = n_9^3 \frac{2^{2^3} - 1}{2^{2^{3-1}} - 1} = n_9^3 \times 17 = 153 \tag{15}$$

Now that we have explored the number extension, we try the opposite, the number reduction mechanism. We are interested in knowing if C_{153}^4 has $m = 4$ as its true order. In order to test that, we have to try the divisibility of 153 with 17 since $\frac{2^{2^{4-1}}-1}{2^{2^{4-2}}-1} = 17$. We get the expected result, $153 = 17 \times 9$. Our matrix does not belong to order $m = 4$, and the label of our matrix in order $m = 3$ is $n_9^3 = 9$, as we obviously knew because that has been our starting point.

We try once more and see if the same process can be expressed in order $m = 2$. In order to do so we have to test the divisibility of 9 with 5, since $\frac{2^{2^{3-1}}-1}{2^{2^{3-2}}-1} = 5$. The division doesn't retrieve a natural number, so the true order of the process is $m = 3$.

E. Decomposition Algorithm

Let us now show how to decompose matrix T^3 given in Eq. (8) as in Eq. (4) of the main text. We begin from the term corresponding to $m = 0$, using the equal probabilities $1/A$ of the uniform model and the reduced matrix in Eq. (6) of the main text. We first get $R^0 =$

$\min T^2_{\alpha\beta} = 0.1$ and $c^0_0 = A \times R^0 = 0.2$, with $C^0_0 = 0.5$ corresponding to the uniform model. Since the extension of C^0_0 is:

$$C_0^{0[+]2} = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix} \tag{16}$$

we can then substract the first term of the decomposition:

$$\begin{aligned} &T^3 - c^0_0 C_0^{0[+]2} = \\ &= \begin{pmatrix} 0.1 & 0.8 & 0.3 & 0.6 \\ 0.9 & 0.2 & 0.7 & 0.4 \end{pmatrix} - \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0.7 & 0.2 & 0.5 \\ 0.8 & 0.1 & 0.6 & 0.3 \end{pmatrix} \end{aligned} \tag{17}$$

Since C^0_0 is the only matrix at $m = 0$, there is no need to search for more compatible ones. In any case, the new reduced matrix is $R^0 = 0$, so we jump to the next level. We can move on to construct the contribution due to $m = 1$. In the first cycle of $m = 1$, R^1 is

$$R^1 = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix} \tag{18}$$

Therefore, we obtain $c^1_1 = \max\{0, 0.1\} = 0.1$, and $C^1_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Since the extension of C^1_1 is:

$$C_1^{1[+]2} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{19}$$

we can get the resultant matrix when we have removed

$$\begin{aligned} &T^3 - c^0_0 C_0^{0[+]2} - c^1_1 C_1^{1[+]2} = \\ &= \begin{pmatrix} 0 & 0.7 & 0.2 & 0.5 \\ 0.8 & 0.1 & 0.6 & 0.3 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0.7 & 0.2 & 0.5 \\ 0.7 & 0 & 0.5 & 0.2 \end{pmatrix} \end{aligned} \tag{20}$$

If we compute R^1 we will see that is null, so we can jump to the next level. We have

$$R^2 = \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix} \tag{21}$$

which means that $c^2_1 = \min\{0.5, 0.5\} = 0.5$. The matrix C^2_1 and its extension $C_1^{2[+]1}$ read

$$C^2_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad C_1^{2[+]1} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \tag{22}$$

In terms of these, we calculate the resultant total matrix.

$$\begin{aligned} &T^3 - c^0_0 C_0^{0[+]2} - c^1_1 C_1^{1[+]2} - c^2_1 C_1^{2[+]1} = \\ &= \begin{pmatrix} 0 & 0.7 & 0.2 & 0.5 \\ 0.7 & 0 & 0.5 & 0.2 \end{pmatrix} - \begin{pmatrix} 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0.2 & 0.2 & 0 \\ 0.2 & 0 & 0 & 0.2 \end{pmatrix} \end{aligned} \tag{23}$$

Again we have a null column in R^2, so we can jump to the next and last level. In this case no calculations are needed, since the remaining matrix can be expressed as boolean matrix, namely C^3_9, multiplied by a constant, $c^3_9 = 0.2$. No extension is needed in this time since the order of C^3_9, $m = 3$, is already M_T.

$$C^3_9 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \tag{24}$$

After the complete process we have

$$T^3 = 0.2 C_0^{0[+]2} + 0.1 C_1^{1[+]2} + 0.5 C_1^{2[+]1} + 0.2 C^3_9 \tag{25}$$

and more explicitly

$$\begin{aligned} &\begin{pmatrix} 0.1 & 0.8 & 0.3 & 0.6 \\ 0.9 & 0.2 & 0.7 & 0.4 \end{pmatrix} = \\ &\begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{pmatrix} + \\ &\begin{pmatrix} 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0.2 & 0.2 & 0 \\ 0.2 & 0 & 0 & 0.2 \end{pmatrix} \end{aligned} \tag{26}$$

F. Low frequency strings

We introduce an upper bound in m to make sure that the frequencies of the strings involved in the calculation of the transition probabilities are high enough. We impose a first cut-off at $m + 2 \leq \log_A L$ even before reading the values of S. This cut-off implies that the average string frequency is $f(x^{m+2}) = 1$ in a uniform model. A second threshold is introduced after reading the string frequencies in S: when a string of length $m+1$ is unique, $f(x^{m+1}) = 1$, we impose an upper bound at $m + 2$. In practice this means that one cannot extract higher-order correlations from sequences in which they are potentially present. In order to dodge this drawback, one can extract the transition probabilities from ensembles of sequences, always under the assumption that all the samples have been produced by the same Markov process. These restrictions are not needed for running the decomposition algorithm, however we still need to provide the transition probabilities for strings that are not found in S. In order to do so, we employ a simple smoothing technique for strings of null frequency: we compute their transition probabilities π by copying the ones of the previous order, which is equivalent to extend the transition matrix for the columns corresponding to those null frequency strings.