Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Entropic Measures in Human Mobility: the BellaMossa database in Bologna.

Relatore:
Prof. Armando Bazzani

Presentata da:
Giulio Colombini

Anno Accademico 2017/2018

Perché il diletto di sentir la voce
Delle Sirene tu non perda. E dove
Pregassi, o comandassi a' tuoi di sciorti,
Le ritorte raddoppino, ed i lacci.

Odissea, Libro XII, 70-73
(trad. I. Pindemonte)

Sommario

Uno dei campi d'interesse della Fisica dei Sistemi Complessi è lo studio quantitativo della mobilità umana. Lo scopo di questa tesi è lo sviluppo e l'applicazione di un algoritmo per il calcolo di entropie a partire da traiettorie sperimentali, per permettere un approccio al problema tramite la Fisica Statistica. É stato scelto un approccio basato su un algoritmo di compressione lossless di tipo Lempel-Ziv 78, adattato alla stima di entropie. Utilizzando dati dal database BellaMossa 2017, si calcolano le distribuzioni di entropia di traiettorie ciclistiche e pedonali raccolte a Bologna durante l'estate 2017. Il calcolo delle distribuzioni d'entropia viene svolto al variare della durata temporale dei viaggi, e si discute una possibile correlazione con la domanda di mobilità realizzata dai dati registrati.

Abstract

One of the main topics in Physics of Complex Systems is the quantitative study of human mobility. The aim of this thesis is to develop and apply an algorithm to compute entropies from experimental trajectories, to allow a Statistical Physics approach. An approach based on a lossless Lempel-Ziv 78 compression algorithm has been chosen, adapting it to the estimation of entropy. Using data from the BellaMossa 2017 database, the entropy distributions were computed, for the trajectories of cyclists and pedestrians recorded in Bologna during the spring and summer 2017. The entropy distributions were computed with respect to the travels' time duration, and a possible correlation with the mobility demand realised by the recorded data is discussed.

Contents

Introduction

The aim of the present thesis, is to analyse the entropic properties of human mobility, using geolocalised mobility data and a lossless compression algorithm. The idea is to follow some of the procedures described in [9], substituting the activity encoding used in said article with a coordinates encoding. The applications of a Statistical Physics approach to study human mobility has been considered in the framework of Complex Systems Physics to highlight statistical laws underlying a cognitive behaviour of particles representing subjects. We're interested in the pedestrians' and cyclists' mobility, in the area of the city of Bologna. The data used are from the BellaMossa 2017 initiative, and they were obtained through a collaboration with SRM Bologna [2]. The area under exam contains the whole city centre, the periphery and portions of the nearby towns Casalecchio di Reno, San Lazzaro di Savena and Calderara di Reno. In the first section of Chapter 1, I briefly explain the database origin, and some of its technical properties. The first operation performed on the data has been to remove any activity different from pedestrians and cyclists, or trespassing the region of interest boundaries. Since the data acquisition instrument consisted mainly in handheld devices such as telephones, it has been deemed necessary to filter the data to remove acquisition errors. So a further filtering mechanism has been devised, using as a discriminant the inter-record velocities. The details on this are presented in Section 1.3. In the same section I conduct an analysis on the distribution of the travels' duration and spatial length. The encoding procedure that has been used, consisted in a subdivision of the region of interest in 200m sided squares, each assigned a numeric code. The choice of 200m as cell width is a compromise between the necessity of analysing typical walking trips and without introducing too many details in the bicycle trips. This tessellation allows to assign to any point in the region of interest the code of the cell to which it belongs. Two types of strings are built, the *timed patterns* and the *jump patterns*. The first kind encodes the user location for each 10 seconds passed. The second only adds a character whenever the user moves to another cell. The encoding procedures are described in Section 1.5. The theoretical background for the definition of entropy, and for its computation

through compression algorithms, is laid out in Chapter 2. The idea of using entropy as a characterising magnitude for the motion of a system is introduced in the theory of dynamical systems. In Section 2.1 I give the definitions of a dynamical system, of ergodicity for a dynamical system, of a phase space partition and of dynamical entropy. Phase space partition is a procedure that allows the so-called *coding* of the trajectories. In this context the dynamical entropy of a phase flow quantifies the amount of refinement induced on the partition by the flow's action. I also present the concept of Markov Partitions. These are the partitions that allow a 1-1 correspondence between the encoded strings and the trajectories. I finish by highlighting that phase space partition allows to view a dynamical system as a stochastic process. In Section 2.2 I present the theory of stochastic processes, and the definition of entropy in the field of random variables. In particular I show the definition of ergodicity for Markov Chains, to show the similarity with dynamical systems. Then I compare the definitions of entropy and entropy rate for a stochastic process, and prove that for stationary processes they coincide. In Section 2.3 I show that the entropy rate of a process can be given the fundamental meaning of *average description length*, for an optimally encoded information source. I conclude the section, and the chapter, by describing the theory justifying the use of the LZ78 lossless compression algorithm to estimate the entropy rate of a symbolic stochastic process, and the algorithm itself. In Chapter 3, I show the results obtained from the data. I begin by building a connectivity graph of the cells, connecting cells that are the ends of a travel. I did this to check whether the mobility in Bologna is organised in a single, highly interconnected ,nucleus or in several of them which are weakly connected among themselves. Finally I show the results obtained from the entropy computation and I discuss the relevance of the entropy measures with respect to the duration of the considered trips.

Chapter 1

The experimental data and their processing

In this chapter I present the experimental data that has been used in the thesis and the data analysis performed to filter the data, and to produce the symbolic sequences used for entropy calculation. I start by briefly describing the database and the means by which it has been created. I then give some generalities on the selection of the types of mobility and of the Region of Interest. Finally I explain the strategies that were devised to clean the database of some errors such as duplicate records.

1.1 The BellaMossa 2017 initiative and database

The data used in this work were gathered during the BellaMossa intiative, active in the Bologna city area between April and September of 2017. To take part in the initiative people were required to download an application on their mobile phones, which allowed the user to declare the performance of mobility activities such as movements on foot, by bicycle, transfers by means of public transportation or car sharing. Being given previous authorisation by the user, during the carrying out of the activities, the device's latitude and longitude* measured by its GPS system were periodically sent to a remote server. Along with said data each record was provided with a numerical activity-unique ID code, a timestamp and the declared activity type. These data were then partially elaborated by BellaMossa. The data were provided to the Physics of Complex Systems group (PhySyCom) of the University of Bologna Physics and Astronomy Department (DIFA), thanks to a collaboration with SRM Bologna.

*As in common practice, latitude is expressed in degrees above the earth's equator and longitude is expressed in degrees to the east of the Greenwhich meridian.

The data were aggregated in twelve two-week long CSV files(Comma Separated Values). In Figure 1.1 we display the first six lines of one of the files: the first is a header containing the names of each record's fields, the remaining five are examples of records.

```
ActivityId,ActivityType,Time,Latitude,Longitude,Accuracy,Speed,IdentifiedType,IdentifiedConfidence
1145763,Car_Share,2017-04-01 00:01:33,44.60015,10.94717,50,21.8,Unknown,100
1145783,Cycle,2017-04-01 00:19:36,44.48921,11.3405,32,0,InVehicle,42
1145783,Cycle,2017-04-01 00:19:45,44.48908,11.34053,12,1.69,InVehicle,62
1145783,Cycle,2017-04-01 00:19:47,44.48898,11.34043,12,1.96,InVehicle,62
1145783,Cycle,2017-04-01 00:19:49,44.4889,11.34037,8,2.4,InVehicle,62
```

Figure 1.1: The first six lines from one of the files supplied by BellaMossa.

There are some fields other than the ones containing the geolocalisation data, time and activity information mentioned before. Two of them (`Accuracy`, `Speed`) were probably attributed depending respectively on the quality of the signal and some estimate on the person's velocity. The remaining two (`IdentifiedType`, `IdentifiedConfidence`) were likely obtained by some kind of inference system. We kindly thank BellaMossa for the opportunity of studying the Bologna city area mobility through their data.

1.2 Region of interest and activity type filtering

The first filtering actions performed on the data were to remove all activities but pedestrians and cyclists, and to remove all users who were, at any time in their activity, out of a given region of interest. Moreover only trajectories spanning more than 15 minutes in time were kept, as this has been considered the minimal time a mobility activity has to last in order to be significant. While going through this process, moreover, fields deemed not relevant were removed. These were `Accuracy`, `Speed`, `IdentifiedActivity`, `IdentifiedConfidence`. This choice has been made on grounds of ease of use, as an approximate, yet coherent and more informative, value of speed could be obtained also from space and time data, and of precaution, as the system used for the ranking of data accuracy, and the inference of the activity type was unknown.
The discriminating field used for activity type assignment has been `ActivityType`, that is the activity type declared by users prior to the commencement of such activity.
Every `ActivityType` value different from `Walk` or `Cycle` has been ignored, and the whole activity related to it discarded, as buses, trains and car sharing platforms weren't the object of the present study. For ease of subsequent use the values `Walk` and `Cycle` have also been mapped respectively to 0 and 1. In order to restrict the analysis to the sole Bologna city centre and suburbs area,

a rectangular region of interest has been defined by the boundaries in Table 1.1.

The North-Western corner is a point roughly 2 km to the West of Airport Guglielmo Marconi, whereas the South-Eastern one is about 1 km east of Bellaria Hospital, as it is visible in the map in Figure 1.2. The rectangle delimited by such corners is called region of interest as any activity which exceeds these limits at any time is excluded from the later stages of the study.

From a practical point of view this first filtering stage has been implemented in a Python program making use of data processing functions from the `pandas` data analysis module. For each of the files the data were first loaded on Random Access Memory as a `pandas` DataFrame. Records were then grouped by the travel-unique `ActivityId` so that each group was identified by its Id. Groups were then re-concatenated on condition that both latitude's and longitude's minima and maxima were inside the region of interest borders, that the `ActivityType` field evaluated either to `Walk` or `Cycle` and that the difference of the final and initial timestamps was equal or greater than 15 minutes, otherwise discarded.

This first selection has reduced noticeably the database size, which from 12.4 GB has been cut down to 3.9 GB on hard drive.

Rectangle corner	Latitude (°)	Longitude (°)
North-Western	44.53518	11.26029
South-Eastern	44.46417	11.40243

Table 1.1: The region of interest corners' latitude and longitude.

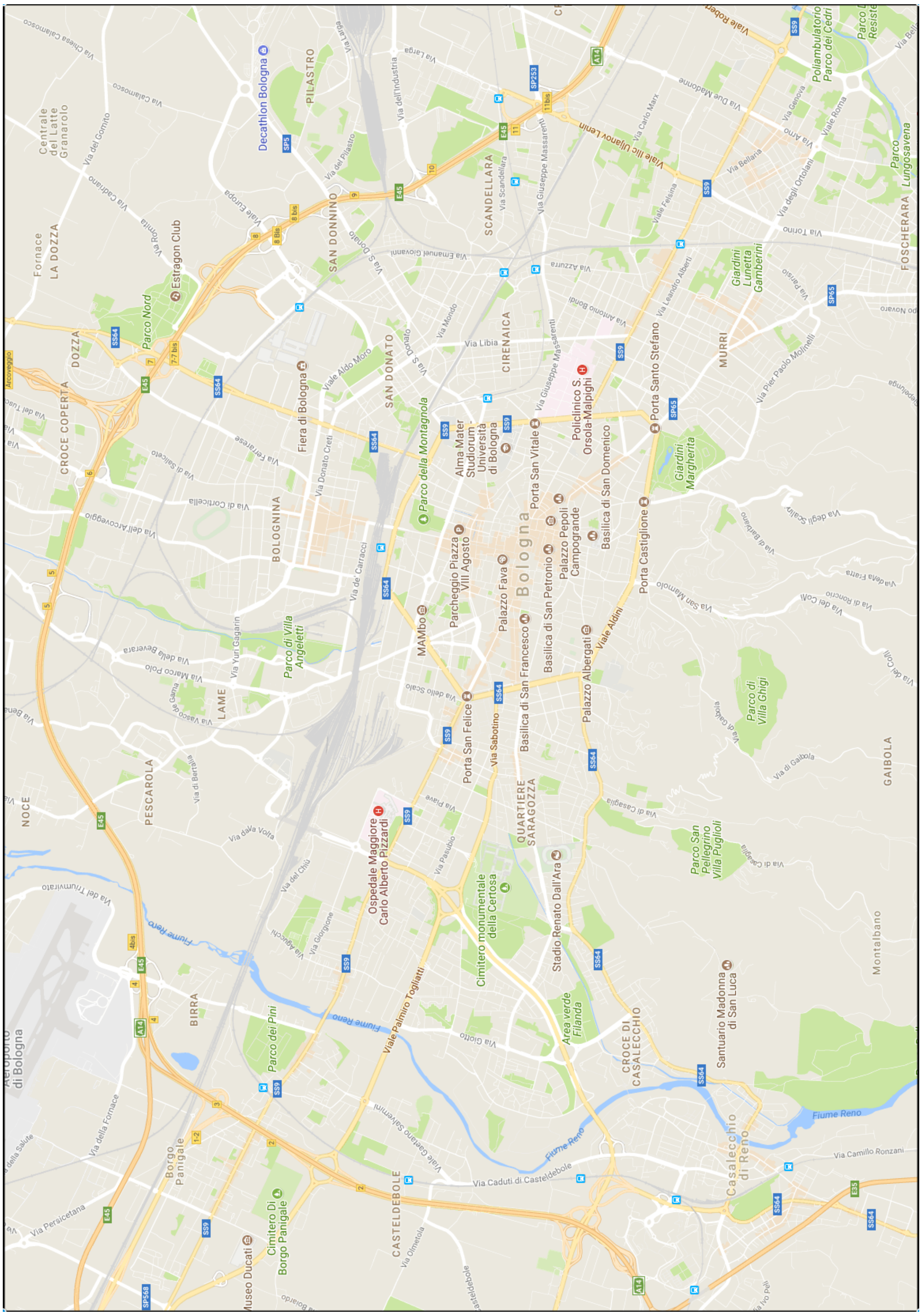

Figure 1.2: A map of the designated region of interest: to the top-left of the figure the airport is partially visible, and similarly is the Bellaria hospital to the bottom-right. Image courtesy of Google Maps.

1.3 Speed filtering and GPS errors detection

The most common source of errors in geolocalised data comes from losses of signals by the utilised GPS devices. Each travel comes in the CSV files as a series of time-stamped coordinates. By taking a look at the inter-record distance distribution shown in Figure 1.3, I found out that the tracking system probably attempted to record a point for each 10m of distance travelled, thus making the single series of data ideally evenly sampled with respect to space. For information concerning the approximations made in distance calculation the reader is directed to Appendix A. Assuming different average speeds for cyclists and pedestrians, the 10m sampling hypothesis is consistent also with the inter-record time distribution in Figure 1.4.

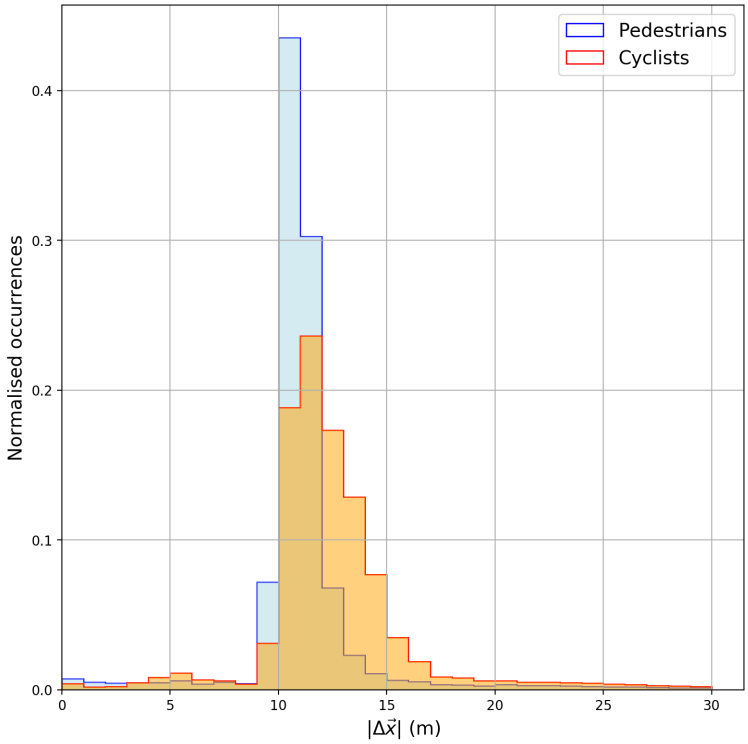

Figure 1.3: Inter-record distance distribution, showing a pronounced peak slightly above 10m. The histogram was made with 30, 1 m wide bins on the range 0-30 m.

This however doesn't prove to be realistic in practice. Due to minor GPS or internet signal issues or sudden, but reasonable, velocity changes, points may be further or closer to each other than the said 10m, even in a generally

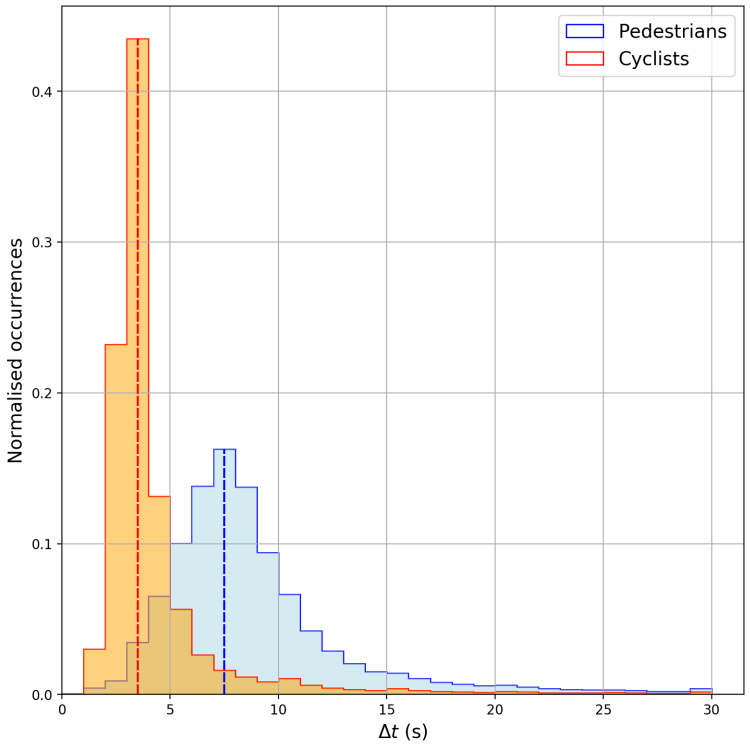

Figure 1.4: Inter-record time interval distribution: two peaks are visible, the most prominent around 3s (marked in red) is mostly due to cyclists, while the less evident one around 7s (marked in blue) mostly contains pedestrians. The histogram was made with 30, 1 s wide bins on the range 0-30 s.

good trajectory. Thus there really isn't a spatial inter-record distance threshold to distinguish correctly measured points from errors. The same can be said for inter-record time intervals. These considerations, therefore, point out the need to devise a different error detection strategy. As points could be further apart than expected in space or in time, I computed for each pair of consecutive records the average velocity, and rejected a trajectory if it presented at least one velocity spike incompatible with the type of activity that generated it (pedestrian or cyclist). The selection of the speed thresholds to use in a filter of this kind could be done using the *preferred walking speed* and *preferred cycling speed*, that is the speeds kept on average by pedestrians and cyclists during their movement. There are papers such as [5] presenting measures of these quantities for several environments with different degrees of urbanisation. This, although, proves to be too strict a criterion, even conceding tolerances of up to 5σ on the mean values reported in literature. Users in fact can exceed

also largely, though for short periods of time, the preferred speed for their kind of activity. This becomes evident as one takes into account, for example, the quick accelerations needed at times in the act of crossing a road.
Indeed, even allowing through the filter values as high as 5σ over the preferred speed limits, only about 5% of the database made it through the filter. Upon closer inspection of the problematic trajectories, I found out that presumed errors emerged near the points where one would naturally expect quick accelerations and decelerations. Another cause of such spikes can be a minor loss of GPS signal, not significant enough to compromise the overall quality of the whole trajectory. Therefore it is reasonable to raise the limit even a little further physically sensible speeds, in order to take such effects into account. The values used as speed acceptability limits were then set to those specified in Table 1.2.

Users	Preferred speed (m/s)	Filter limit (m/s)
Pedestrians	1.34 ± 0.37	7
Cyclists	4.875 ± 1.085	21

Table 1.2: The preferred speed values, as reported in experimental literature on the subject, along with the limits used for filtering in this study.

There were two other kinds of issue with the records. More specifically: there were both activities with duplicate points, that is with two consecutive records with equal coordinates and timestamps, and ubiquitous records, that is two consecutive records with same timestamp but different coordinates. The former problem has simply been addressed by removing one of the two duplicates. On the other hand ubiquitous records, which were present in a very small amount of cases, were considered indicators of poor signal quality and any activities exhibiting such features weren't let into the filtered database. The inter-record speed histogram in Figure 1.5 shows two peaks. The histogram is peaked around values compatible with the ones presented in Table 1.2, as one in fact should expect, as the preferred speed is kept for most of the activity, and therefore represents the foremost contribution to the distribution. Nevertheless, higher speeds are reasonable on short times as discussed before, thus creating, probably along with a fraction of faster-than-average people, a decreasing tail. This tail decreases to zero quite quickly for higher speeds, a fact that is in accord with the fact that the great deviations grow less likely as the deviation grows. After the application of these filters, the database contained 122175 pedestrian activities and 131882 cyclist activities.

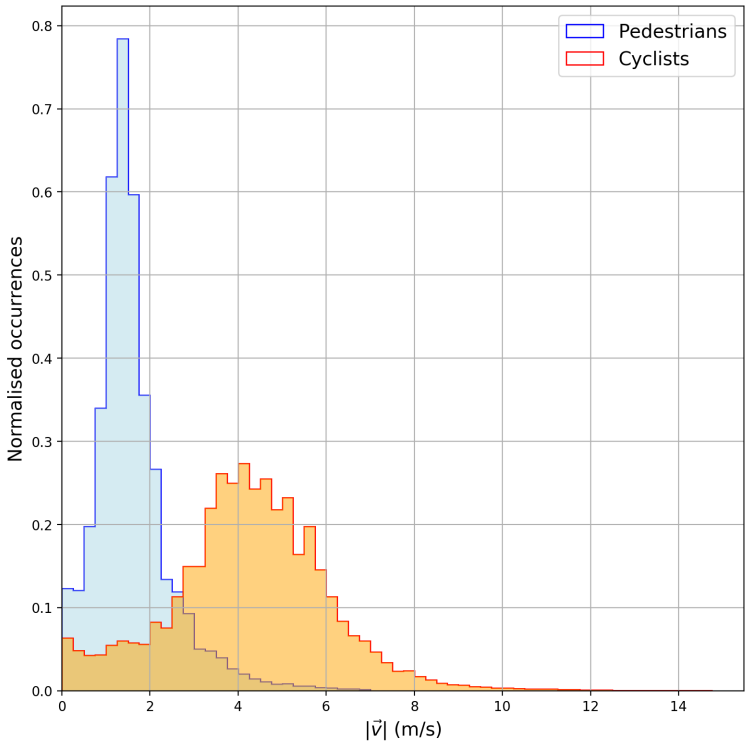

Figure 1.5: The speed histogram shows two peaks in correspondence of the preferred speeds for both kinds of mobility taken into consideration. The histogram has been made with 28, 0.5 m/s wide bins, on the range 0-14 m/s.

1.4 Travel times and distances statistics

In Figure 1.6 I present the travelling time histogram, intended as the time between the last and first record in an activity. The data distribution goes to zero very quickly, faster than an exponential. There could be periods of stationarity linking a number of actual travels. In the following section I'll devise a way to spot them and break down the travels in actual mobility segments.

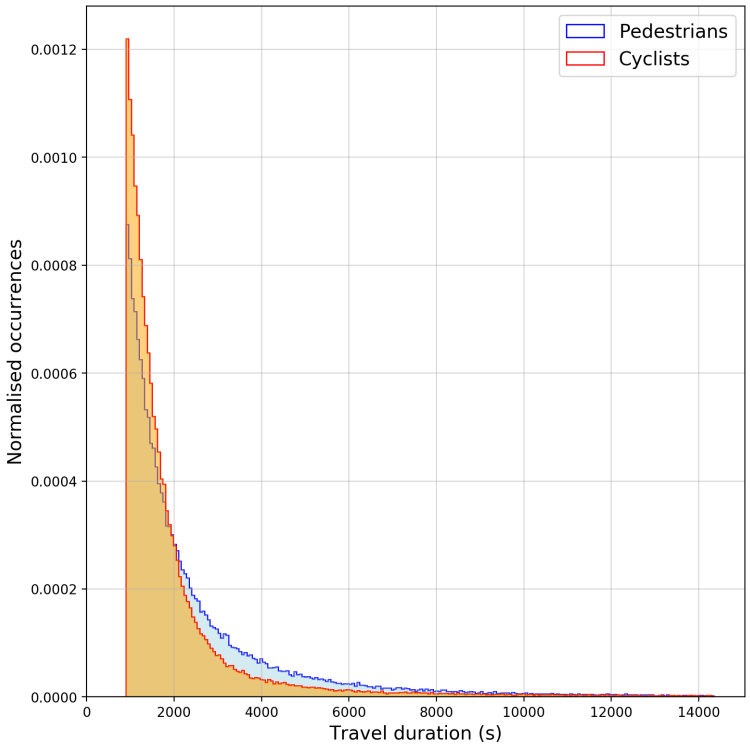

Figure 1.6: The histogram of travel times, it shows a faster-than-exponential decay, in function of the travel time.

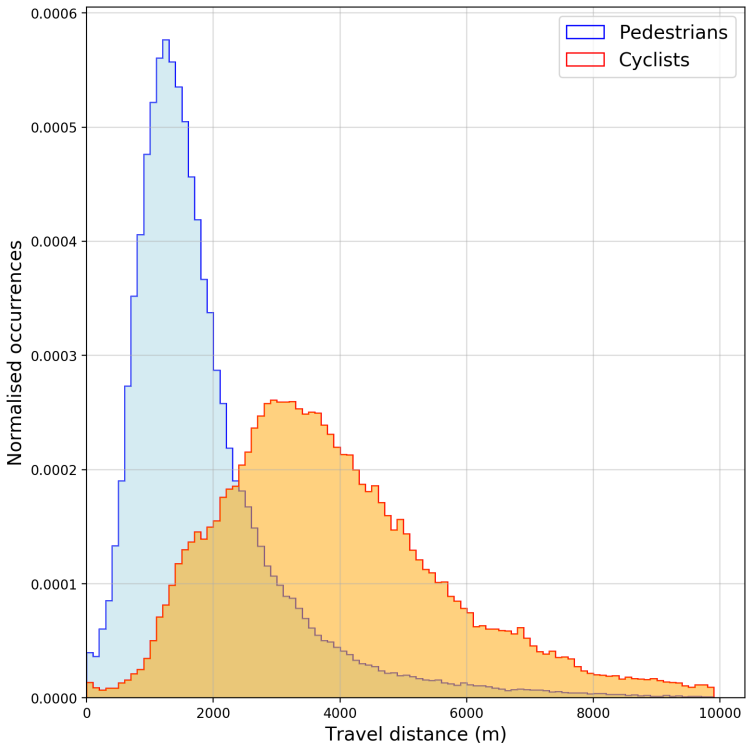

Figure 1.7: The histogram of travel distances appears as a superposition of two asymmetrical bell-shaped distributions. The histogram has 100 bin, each of 100m, on the range 0-10000 m.

In Figure 1.7, the travel distance distribution can be seen. It appears as a superposition of two asymmetrical bell-shaped distributions. The first peak corresponds to pedestrians, the second to cyclists. Such behaviour is expected, since cyclists are quicker than pedestrians, on average. Thus, when given the same amount of time, they can go farther.

1.5 Encoding of the experimental trajectories

In order to compute the entropy rate for the mobility in Bologna, it is necessary to encode the trajectories as sequences of symbols, each representing a partition of the space involved in the present study. A grid of squares with a side of 200m has been chosen, covering the region of interest. These squares were first numbered each with a pair $(i,\ j)$, with the first index indicating the row and the second the column, as one would do for the entries of a matrix. This yields a grid of 40 rows by 57 columns, leading to a total of 2280 cells. Using 2280 proper characters would require the usage of an extended character set, thus potentially creating endianity issues. Consequently I decided to use unsigned integers, simply obtained by counting the cells from the top left corner of the region of interest, rows first. The expression used to map a point to its code is in Equation 1.1.

$$\begin{aligned}(\mathrm{lat}, \mathrm{lon}) &\mapsto \left(\left\lfloor \frac{r_{Earth}}{l} \frac{2\pi(\mathrm{lat} - \mathrm{lat}_0)}{360^\circ} \right\rfloor, \left\lfloor \frac{r_{Earth}}{l} \frac{2\pi(\mathrm{lon} - \mathrm{lon}_0)\cos\left(\frac{2\pi \mathrm{lat}}{360^\circ}\right)}{360^\circ} \right\rfloor \right) \\ &\equiv (i, j) \\ &\mapsto \left\lfloor \frac{r_{Earth}}{l} \frac{2\pi(\mathrm{lat} - \mathrm{lat}_0)}{360^\circ} \right\rfloor \cdot \max(j) + \left\lfloor \frac{r_{Earth}}{l} \frac{2\pi(\mathrm{lon} - \mathrm{lon}_0)\cos\left(\frac{2\pi \mathrm{lat}}{360^\circ}\right)}{360^\circ} \right\rfloor \\ &\equiv i \cdot \max(j) + j \end{aligned} \tag{1.1}$$

Where $(\mathrm{lat}_0,\ \mathrm{lon}_0)$ are the coordinates of the north-western corner, d is the cell side length and r_{Earth} is the Earth's radius. Through Formula 1.1, it is possible to attribute to any geographical point in the region of interest a pair of non-negative integers (i, j) and then a code $i \cdot \max(j) + j$, thus effectively transforming any trajectory into a string of symbols.

The actual process of building the strings opens two options:

- Setting a time resolution, and writing the symbol identifying the cell a person is in, for each passing time period. In this case we're building a *timed pattern*.

- Writing a symbol only when a person changes cell location, disregarding time dependence. These strings are called the *jump patterns*.

In this thesis I will only work with *timed patterns*. The first option, although, poses the problem of the sample's non-homogeneity in the time domain. The approach used in this work has been the following. The time resolution has been set to periods of ten seconds starting from the 1st of January 1970, that is to the integer division of UNIX time by ten. Then with each of the inhomogeneous trajectories a polygonal chain has been built, and on it points were taken at each time interval, interpolating across the experimental points, the procedure is made clear by the interpolation formula 1.2, where $(\text{lat}, \text{lon})_i$ and $(\text{lat}, \text{lon})_{i+1}$ are two consecutive points, respectively at times t_i and t_{i+1}, and we're willing to interpolate at time t_{int}.

$$(\text{lat},\ \text{lon})_{int} \equiv \frac{t_{int} - t_i}{t_{i+1} - t_i} \left((\text{lat}, \text{lon})_{i+1} - (\text{lat}, \text{lon})_i\right) + (\text{lat}, \text{lon})_i \tag{1.2}$$

As it is clear from the expression, the interpolation is appropriately performed as long as $t_i \leq t_{int} \leq t_{i+1}$. Otherwise, indeed, it wouldn't even be an interpolation.

All the operations described so far, were implemented by me in a number of Python programs. I mostly used the modules `numpy` and `pandas` for calculations and the management of data. All of the programs are available on my repository on GitHub: `http://www.github.com/GColom/Bolo_BM`[1].

Chapter 2

Theoretical apparatus

In this chapter I outline the theoretical framework used in the present work. I start by pointing out how a connection between a dynamical system and a discrete deterministic dynamics can be made. Provided that some measure-theoretic requirements are met, a probabilistic setting is built, and a symbolic dynamics can be approximated by a stochastic process. Such probabilistic setting allows for the definition of dynamical entropy, which can be used to characterise the system. Eventually, Information Theory will provide the tools necessary to evaluate entropy rates experimentally, without explicit knowledge of the phase flow.

2.1 Dynamical systems and phase space coding

We start by defining the concept of dynamical system. A natural definition is the following:

Definition 2.1.1: Classical Dynamical System
Let $\mathcal{M}$ be a smooth manifold, μ a measure on $\mathcal{M}$ defined by a continuous density, let $\Phi^t : \mathcal{M} \to \mathcal{M}$ be a one-parameter group of measure-preserving diffeomorphisms. The triplet $(\mathcal{M}, \mu, \Phi^t)$ is called a Classical Dynamical System.

Although this definition is enough to encompass many of the features of classical mechanics, a more general one can be given.

Definition 2.1.2: Abstract Dynamical System
Let $(\mathcal{M}, \mu)$ be a measure space, and let Φ^t be a one parameter group of measure-preserving automorphisms (mod 0) of $(\mathcal{M}, \mu)$. Then $(\mathcal{M}, \mu, \Phi^t)$ is called an Abstract Dynamical System.

For a deeper explanation of the concepts involved in these definitions, the reader is invited to look at Appendix 6 in [3].

If $t \in \mathbb{R}$ the system is called a *Continuous Time Dynamical System* and Φ^t is a continuous group of transformations, that in many physical situations is defined as the solution to a set of differential equations. If, instead, $t \in \mathbb{Z}$ one is considering a *Discrete Time Dynamical System*, and the group describing the time evolution is discrete. In this case it is custom to call Φ the single time-step evolution operator, since any Φ^n can be obtained by iterated composition. It is clear that the second definition also retains in itself the first one. I will now present two interesting examples of both kinds of systems.

Example 2.1.1: Arnol'd's Cat Map
Let's take $\mathcal{M} = \{(x, y) \mod 1\}$ as our configuration space with the measure $dxdy$, and let the discrete-time flow be defined by

$$\Phi(x, y) = (x + y, x + 2y) \pmod 1 \tag{2.1}$$

By extending the flow to the whole $\mathbb{R}^2$ plane one obtains the following linear application:

$$\tilde{\Phi}(x, y) = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.2}$$

As $\det \tilde{\Phi} = 1$ the flow clearly preserves the measure, thus we are confronting a dynamical system. $\tilde{\Phi}$ has two real eigenvalues $0 < \lambda_2 < 1 < \lambda_1$. It can be shown that for time-step $n \to \infty$ the orbits of Φ are close to those of the continuous-time map defined by:

$$\dot{x} = 1, \qquad \dot{y} = 1 - \lambda_1 \tag{2.3}$$

For maps of this kind, on spaces analogous to the torus $\mathcal{M}$ there exists a Theorem due to Jacobi (ref. Appendix 1 of [3]), that states that if $\frac{\dot{y}}{\dot{x}} = 1 - \lambda_1 \notin \mathbb{Q}$ then for each set $A \subset \mathcal{M}$, $\Phi^n A$ converges to a dense helix on the torus. In figure 2.1 is a representation of the map's action. This system is an example of Classical Dynamical System. The Cat Map has also a chaotic behaviour, in the sense that, for any two points $\vec{x}_1$, $\vec{x}_2 \in \mathcal{M}$, the distance $\|\Phi^t \vec{x}_1 - \Phi^t \vec{x}_2\|$ diverges as $e^{\lambda t}$. This system also has a known Markov Partition (see Def.).

Example 2.1.2: Bernoulli Schemes
Let's define $\mathcal{M}$ to be the set of bi-infinite sequences of a given alphabet of symbols. In order to do this let's first define $\mathbb{Z}_n = 0, 1, ..., n - 1$ the set of the first $n - 1$ non-negative integers as our alphabet. It is natural to define $\mathcal{M} = \mathbb{Z}_n^{\mathbb{Z}}$, that is to view bi-infinite strings of symbols as elements in the cartesian product of countable copies of the alphabet $\mathbb{Z}_n$. The second element

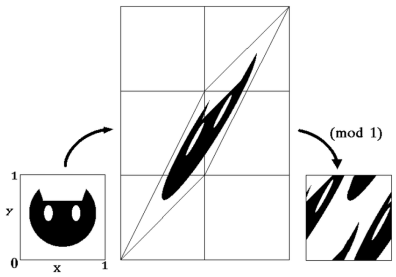

Figure 2.1: A representation of the action of Φ and $\tilde{\Phi}$, each on its domain. Image reference is [4]

.

needed to define an Abstract Dynamical System is the σ-algebra of measurable sets. The generating sets for the σ-algebra are the strings A_i^j defined as follows:

$$A_i^j = \{m | a_i = j\} \quad i \in \mathbb{Z}, j \in \mathbb{Z}_n \tag{2.4}$$

In fact A_i^j is a string that as its only defining requirement has character j in its i-th position. Clearly, a countable intersection of such strings can generate any element of $\mathcal{M}$. The third structure needed to define an Abstract Dynamical System is a measure. It is possible to define a normalised measure on the alphabet, in other words, a discrete probability function (PF):

$$\mu(0) = p_0, \quad \mu(1) = p_1, \quad ... \quad \mu(n-1) = p_{n-1} : \quad p_i \text{ such that } \sum_{i=0}^{n-1} p_i = 1 \tag{2.5}$$

The next step would be to assign to each A_i^j the measure $\mu(A_i^j) = p_j$, this is sufficient to fully define a measure on $\mathcal{M}$ since each A_i^j can be viewed as a generator of the σ-algebra on $\mathcal{M}$, distinct from the others, therefore the measure of an intersection of such terms is the product of the measures of the terms themselves. The measure on $\mathcal{M}$ has then the expression below:

$$\mu(m) = \mu(A_{i_1}^{j_1} \cap A_{i_2}^{j_2} \cap ... \cap A_{i_k}^{j_k}) = \prod_{l=1}^{k} \mu(A_{i_l}^{j_l}) \tag{2.6}$$

Where the first equality is possible thanks to the way we defined the A_i^j sets:

$$m = (a_{i_1} = j_1, a_{i_2} = j_2, ..., a_{i_k} = j_k) = \bigcap_{l=1}^{k} A_{i_l}^{j_l} \tag{2.7}$$

The last necessary item is the automorphism, we'll use a discrete time shift $\Phi : \mathcal{M} \to \mathcal{M}$ defined as pointed out in

$$\Phi : m = (\dots, a_i, \dots) \mapsto m' = (\dots, a'_i, \dots) \quad \text{with } a'_i = a_{i-1} \tag{2.8}$$

This kind of automorphism is also called a shift. Let's verify that this shift is measure-preserving. It is sufficient to verify the action of Φ on the σ-algebra generators. Indeed one has:

$$\mu(\Phi A_i^j) = \mu(A_{i-1}^j) = p_j = \mu(A_i^j) \tag{2.9}$$

Thus a Bernoulli Scheme has the structure of a dynamical system. This scheme is relevant because it is a Symbolic Dynamical System, that is an important link between ergodic theory and stochastic processes. Moreover, this system already has a normalised measure to start with, providing a natural link with stochastic processes.

A very important class of dynamical systems is that of Ergodic Systems. These are the systems for which the mean values of observables, on the long run, do not depend on the initial conditions. It is necessary to set these condition formally.

Definition 2.1.3: Time and space means of a function
Let $(\mathcal{M}, \mu, \Phi^t)$ be a dynamical system, and $f : \mathcal{M} \to \mathbb{C}$, $f \in L^1(\mathcal{M})$. The time mean f^* of f is defined by:

$$f^*(x) \equiv \lim_{T \to +\infty} \frac{1}{T} \int_0^T f(\Phi^t x) dt \tag{2.10}$$

Where the integral is turned into a sum for discrete maps.
The space mean $\bar{f}$ of f is defined by

$$\bar{f} \equiv \frac{1}{\mu(\mathcal{M})} \int_{\mathcal{M}} f(x) d\mu \tag{2.11}$$

Where both definitions can be given only if the defining expressions exist.

Note that eq. 2.10 defines a function on $\mathcal{M}$, potentially varying from point to point, conversely, eq. 2.11 defines a constant. The definition of ergodicity can now be posed.

Definition 2.1.4: Ergodic System
A system $(\mathcal{M}, \mu, \Phi^t)$ is said to be ergodic if and only if:

$$\forall f : \mathcal{M} \to \mathbb{C}, f \in L^1(\mathcal{M}, \mu) \qquad f^*(x) = \bar{f} \quad a.e. \tag{2.12}$$

Inspecting closely this definition one can realise that in order for it to be satisfied the time mean f^* must not depend on the initial conditions x. An important property that is satisfied by ergodic systems is indecomposability, that means they cannot be divided into phase-flow-invariant subsets. This indeed proves to be a theorem.

Theorem 2.1.1 (Ergodicity $\iff$ Indecomposability)**.** *A dynamical system is ergodic if, and only if, it is indecomposable, that is if every measurable set invariant under the phase flow has measure 0 or 1.*

This result implies that either the whole system or none of it is invariant under the phase flow, up to null-measure sets. A very important kind of operation that can be performed on dynamical systems is **Phase Space Coding**. It allows to transform any regular dynamical system into a symbolic one and to make foundations for interesting links with stochastic processes. The first step is to define a partition on the phase space.

Definition 2.1.5: Phase Space Partition
Let $\mathcal{M}$ be the phase space of a dynamical system. A phase space partition on $\mathcal{M}$ is any (at most numerable) family of subsets $\mathcal{P} = \{P_i | P_i \subset \mathcal{M}\}$ such that:

1. $\mu(P_i \cap P_j) = 0$ *if* $i \neq j$
2. $\mathcal{M} = \bigcup_i P_i$

In this context, P_is are called atoms. A partition is said to be measurable if $\mu(\bigcup_i P_i) < +\infty$.

If one associates a symbol to each of the atoms in a partition one effectively builds a coding for the system: $P_j \mapsto j$. In fact, a partition naturally defines a map C, that sends each point in the phase space to the string containing the symbols of the atoms visited by the system when starting from there, at each evolution step, as determined by the phase flow.

$$C : x \mapsto (\ldots, j_{n-1}, j_{n-1}, j_{n-1}, \ldots)$$
$$\iff$$
$$\ldots, \Phi^{n-1}x \in P_{n-1}, \Phi^n x \in P_n, \Phi^{n+1}x \in P_{n+1}, \ldots \quad (2.13)$$

Any partition generates a coding, but we want to know on what conditions the strings can be in a 1-1 relation with the trajectories. This leads to the concept of Markov partitions. In order to introduce this concept it is necessary to know what a stable and unstable manifolds are, see Appendix B for the definitions.

Definition 2.1.6: Markov partition
Let P *be a phase space partition for a given system* $(\mathcal{M}, \mu, \Phi)$. P *is a Markov partition with respect to* Φ *if and only if for any* $x \in P_i$ *such that* $\Phi x \in P_j$ *with* $P_i, P_j \in \mathsf{P}$*:*

$$\begin{aligned} \Phi(W^u(x) \cap P_j) &\supset W^u(\Phi x) \cap P_j \\ \Phi(W^s(x) \cap P_j) &\subset W^s(\Phi x) \cap P_j \end{aligned} \tag{2.14}$$

In other words: a partition is a Markov partition for a phase flow if and only if each atom's image crosses completely each other atom it intersects along the expanding direction, and falls completely inside it along the contracting one. If one system is coded with a Markov partition there is a one to one correspondence between orbits and strings. It is possible to give a definition of dynamical entropy for a phase flow, with respect to a partition of the phase space. Given that one is working with a finite or σ-finite measure (both cases are reunited by the *measurable partition* request)*, it is possible to normalise such measure and set $\mu(\mathcal{M}) = 1$, thus building a probability measure. In this context one can define the entropy of a partition similarly to that of a random variable.

Definition 2.1.7: Entropy of a partition
Let $\mathsf{P} = \{P_i\}_{i \in I}$ *with* I *at most numerable, be a partition of phase space. The entropy of partition* P *is defined by:*

$$H(\mathsf{P}) \equiv \sum_{i \in I} \mu(P_i) \log_2 \left(\frac{1}{\mu(P_i)} \right) \tag{2.15}$$

The base of the logarithm in the definition fixes the unit of measurement: bits *for base 2,* nats *for base e.*

The maximal entropy of a partition is obtained when all the atoms have equal measure. The result can be proved by maximising the right hand side of eq. 2.1.7 on the constraint $\sum_i \mu(P_i) = 1$ using Lagrange's multipliers. This definition of entropy of a partition matches the one that we'll give for a discrete probability function. The probability function is defined by the measures of the partition's atoms and the associated alphabet is made up of the symbols given to each of them. Let's now take into account the effects of time evolution. If one interprets the atoms of a partition as the states of a stochastic process, it is possible to select one of them, A, for the system to be in at $t = 0$,and let it

*A σ-finite measure is one where the whole measure space can be expressed as union of countable elements of finite measure. This allows, by introduction of adequate weights, having a finite measure value for the whole space.

evolve one step forward in time. In general, then, one can intepret $\mu(\Phi A \cap P_i)$ as the probability of the system falling in state i at the time instant immediately successive to when the system was in A, one can denote this with $P({}_0A, {}_1P_i)$, by intending implicitly that time occurrence of events is ordered from left to right. Similarly to what we just did one can ask for $P({}_0A, {}_2P_i)$, and obtain:

$$P({}_0A, {}_2P_i) = \mu(\Phi^2 A \cap P_i) \tag{2.16}$$

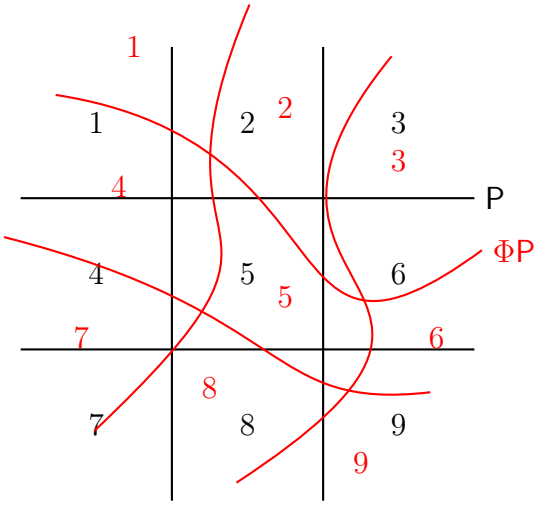

Figure 2.2: A representation of the action of Φ on partition P, the initial partition has been drawn in black, the evolved copy in red. The probabilities such as that in 2.16 are the measures of the overlapping regions.

By looking at Figure 2.2 it is easy to picture the geometrical meaning of 2.16: the probability of being in P_5 at time 0 and in P_6 at time 1 is given by the measure of the intersection of the black region marked by 6 and the red one labelled 5.

Moreover, one can extend the analysis to more than to two instants:

$$P({}_0A, {}_1P_i, {}_2P_j) = \mu((\Phi^2 A \cap P_j) \cap (\Phi A \cap P_i)) \tag{2.17}$$

In general then, the probability distribution of the sequences of N symbols has alphabet defined by all the intersections of N copies of the initial partitioning, each evolved one step further in time with respect to the preceding one, and PMF defined by the measure of such intersections. That is:

$$P(i_0\,, i_1\,, \ \ldots\ , \ i_{n-1}, \ i_n) = \mu(\Phi^n P_{i_0} \cap \Phi^{n-1} P_{i_1} \cap \ \ldots\ \cap \Phi P_{i_{n-1}} \cap P_{i_n}) \tag{2.18}$$

In order to make writing clearer one can define an n-ary operation on partitions.

Definition 2.1.8: $\vee$-intersection of a collection of partitions
Let $\mathfrak{G}$ be a collection of partitions of a given phase space $\mathcal{M}$, let $N \equiv \#\mathfrak{G}$ be the cardinality, at most numerable, of $\mathfrak{G}$. We define on this structure the operation called $\vee$-intersection as follows:

$$\bigvee_{\mathsf{P}_i \in \mathfrak{G}} \mathsf{P}_i = \{P_1 \cap P_2 \cap P_3 \cap \ldots \cap P_n | P_1 \in \mathsf{P}_1,\ P_2 \in \mathsf{P}_2,\ P_3 \in \mathsf{P}_3,\ \ldots,\ P_N \in \mathsf{P}_N\} \tag{2.19}$$

In measure spaces language, we are asking for the partition made up by the generators of the join σ-algebra of all the σ-algebras generated by the partitions in $\mathfrak{G}$. With the setting of this definition, the sets whose measure is taken in the right hand sides of eq.s 2.16, 2.17 and 2.18 are all elements of the $\vee$-intersection of appropriately evolved copies of the initial partition. Given a partition we have defined an entropy over it, this enables us to define the entropy of a partition with respect to an automorphism.

Definition 2.1.9: Entropy of a partition with respect to an automorphism
Let $(\mathcal{M}, \mu, \Phi)$ be a dynamical system, let P be a measurable partition on $\mathcal{M}$. We define the entropy of P relative to Φ by:

$$H(\mathsf{P}, \Phi) \equiv \lim_{n \to \infty} \frac{H(\Phi^{n-1}\mathsf{P} \vee \Phi^{n-2}\mathsf{P} \vee \Phi^{n-3}\mathsf{P} \vee \ldots \vee \mathsf{P})}{n} \tag{2.20}$$

Theorem 2.1.2. *The limit in Def. 2.1.9 always exists.*

The proof relies on the fact that at each time step the entropy increases, but of a smaller amount each time. Consequently, the entropy difference between a time step and the preceding one forms a positive, non-increasing sequence. Thus it has a limit, and therefore the sequence of entropies converges by the Cauchy criterion.

The n denominator is present to keep entropy from diverging. Indeed, in the worst case scenario, with all partitions intersecting all their evolved copies, the number of probability atoms N grows as N^n, and thus entropy grows as n, as n, the time index, goes to infinity.

The entropy of a dynamical system, therefore, is an indicator of how finer the partition gets as the system evolves. Secondly: by comparing this definition of entropy and the one for a stochastic process, yet to be presented, we will see that there are strong similarities, and that such similarities are symptom of a strong conceptual link between stochastic processes and dynamical systems introduced by the partitioning. For a deeper view on these topics the reader is invited to look at [3] and [6].

2.2 Stochastic Processes

Informally speaking, a stochastic process is a collection of random variables indexed by an integer, usually given the meaning of time, where probability is defined on collections of variables. We shall first define some basic probability theory concepts such as that of probability space for a given experiment.

Definition 2.2.1: Probability space
Let Ω be the set of possible outcomes *of a given probabilistic experiment, also called the* sample space*. Let $\mathcal{F}$ be the σ-algebra on Ω of* probabilistic events*. Finally let a measure $p : \mathcal{F} \to [0, 1]$ such that $p(\Omega) = 1$ be the* probability measure*. The triplet $(\Omega, \mathcal{F}, p)$ is called a* probability space*.*

The next concept needed is that of random variable, as a function of admitted probabilistic events.

Definition 2.2.2: Random variable
Let $(\Omega, \mathcal{F}, p)$ be a probability space. Any real and measurable function $\mathbf{X}$ defined on Ω is called a random variable*. The set of values a certain variable can assume is called its* alphabet *and is denoted by χ if it is finite or countable.*

From now on we'll only work with discrete random variables. Note that in this setting there is no need to define a probability distribution for a given random variable. In fact such distribution is naturally defined by the measure on Ω.

Definition 2.2.3: Probability distribution of a random variable
Let $\mathbf{X}$ be a random variable defined on a given probability space $(\Omega, \mathcal{F}, p)$. The probability function *P is the function defined by:*

$$P(\mathbf{X} = x) \equiv p(f_x) \tag{2.21}$$

Where $a_x \in \mathcal{F}$ is defined as:

$$a_x \equiv \{\omega | \omega \in \Omega, \mathbf{X}(\omega) = x\} \tag{2.22}$$

In the following, whenever we'll want to refer to $P(\mathbf{X} = x)$ we'll just write $P(x)$, in order to unburden the notation.

One can study the probability of outcomes from more than one variable.

Definition 2.2.4: Joint probability distribution
Let $\mathbf{X}$ and $\mathbf{Y}$ be two random variables on the same sample space. Their joint

PMF can be defined on the same lines of what was done for a single variable by setting:

$$P(\mathbf{X} = x, \mathbf{Y} = y) = p(a_x \cup a_y) \tag{2.23}$$

with:

$$a_x \equiv \{\omega | \omega \in \Omega, \mathbf{X}(\omega) = x\} \tag{2.24}$$

$$a_y \equiv \{\omega | \omega \in \Omega, \mathbf{Y}(\omega) = y\} \tag{2.25}$$

Another important concept is the conditional probability distribution: this quantifies the confidence that one variable has a certain value, given that we know the one of another value. In particular, measure theory allows us to set a precise geometric definition of this apparently causal link.

Definition 2.2.5: Conditional probability distribution
Let $\mathbf{X}$ *and* $\mathbf{Y}$ *be two random variables. We define the probability of "*$\mathbf{X} = x$ *given that* $\mathbf{Y} = y$*" as*

$$P(\mathbf{X} = x | \mathbf{Y} = y) = \frac{p(a_x \cap a_y)}{p(a_y)} \tag{2.26}$$

Where a_x *and* a_y *have the same meaning as in defs. 2.2.3 and 2.2.4. In the following the conditional distribution will be indicated by* $P(\mathbf{X}|\mathbf{Y} = y)$*, and its values by* $P(x|\mathbf{Y} = y)$*, possibly omitting even the* $\mathbf{Y}$ *if there's no chance of misunderstanding.*

Given a certain random variable it can be useful to define some informative quantities on it. The first is the expected value of a function with respect to the variable.

Definition 2.2.6: Expected value of a function of a random variable
Let $\mathbf{X}$ *be a random variable and let* f *be a function defined on* χ*, the alphabet of* $\mathbf{X}$*. We define* $\langle f \rangle_{\mathbf{X}}$*,the* expected value *of* f *with respect to* $\mathbf{X}$*, as:*

$$\langle f \rangle_{\mathbf{X}} \equiv \sum_{x \in \chi} P(x) f(x) \tag{2.27}$$

By averaging the values of appropriate functions one can generate several quantities, for example if one sets $f(x) = x$ the mean value is called the Expected Value of the distribution. Another important quantity which can be viewed as the mean value of a function is Entropy.

Definition 2.2.7: Entropy of a random variable
Let $\mathbf{X}$ *be a random variable with alphabet* χ*, the* entropy *of* $\mathbf{X}$ *is defined by:*

$$H(\mathbf{X}) \equiv \sum_{x \in \chi} P(x) \log_2 \left(\frac{1}{P(x)} \right) = -\langle \log_2 P(x) \rangle_{\mathbf{X}} \tag{2.28}$$

Where, on grounds of continuity, we set the convention $0\log_2(0) = 0$ *in order to exclude automatically any character with probability 0 that could be present in the alphabet.*

We'll see that the entropy of a random variable carries important information on the informative content of the variable. We define entropy also for a conditioned variable.

Definition 2.2.8: Entropy of a conditioned random variable
Let $\mathbf{X}$ *and* $\mathbf{Y}$ *be random variables with alphabets* $\chi_{\mathbf{X}}$ *and* $\chi_{\mathbf{Y}}$*, the entropy of* $\mathbf{X}|y$ *with* $y \in \mathbf{Y}$ *is defined by:*

$$H(\mathbf{X}|y) \equiv \sum_{x\in\chi} P(x|y)\log_2\left(\frac{1}{P(x|y)}\right) = -\langle \log_2 P(x|y)\rangle_{\mathbf{X}} \tag{2.29}$$

The generalisation to the joint distribution of a collection of variables is natural: it simply is necessary to sum over the cartesian of all alphabets. On the other hand it is possible to define the conditional entropy of a variable in the following way.

Definition 2.2.9: Conditional entropy of a random variable
Let $\mathbf{X}$ *be a random variable with alphabet* $\chi_{\mathbf{X}}$*, and let* $\mathbf{Y}$ *be another random variable, with alphabet* $\chi_{\mathbf{Y}}$*. We define the* entropy of $\mathbf{X}$ given $\mathbf{Y}$ *as follows.*

$$H(\mathbf{X}|\mathbf{Y}) \equiv \sum_{y\in\chi_{\mathbf{Y}}} P(y)H(\mathbf{X}|y) = \langle P(y)H(\mathbf{X}|y)\rangle_{\mathbf{Y}} \tag{2.30}$$

Note the difference between def. 2.2.8 and 2.2.9: in the first we are fixing the value of y, in the second one we are averaging over all possible y.

There are a few theorems on entropies which will be needed in the future.

Theorem 2.2.1 (Chain rule for entropy)**.** *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be two random variables, the following equality holds:*

$$H(\mathbf{X},\,\mathbf{Y}) = H(\mathbf{X}|\mathbf{Y}) + H(\mathbf{Y}) = H(\mathbf{Y}|\mathbf{X}) + H(\mathbf{X}) \tag{2.31}$$

Let $\{\mathbf{X}_i\}_{1\leq i\leq n}$ *be a collection of random variables, the above formula generalises to* n *variables as:*

$$H(\mathbf{X}_1,\,\mathbf{X}_2,\,\ldots,\,\mathbf{X}_n) = \sum_{i=1}^{n} H(\mathbf{X}_i|\mathbf{X}_{i-1},\,\ldots,\mathbf{X}_1) \tag{2.32}$$

Proof. We shall prove 2.31 explicitly and use it recursively to prove 2.32.

$$\begin{aligned}
H(\mathbf{X},\mathbf{Y}) &= -\sum_{x\in\chi_{\mathbf{X}}}\sum_{y\in\chi_{\mathbf{Y}}} P(x,y)\log_2(P(x,y)) \\
&= -\sum_{x\in\chi_{\mathbf{X}}}\sum_{y\in\chi_{\mathbf{Y}}} P(x,y)\log_2(P(y|x)P(x)) \\
&= -\sum_{x\in\chi_{\mathbf{X}}}\sum_{y\in\chi_{\mathbf{Y}}} P(x,y)\log_2(P(y|x)) - \sum_{x\in\chi_{\mathbf{X}}}\sum_{y\in\chi_{\mathbf{Y}}} P(x,y)\log_2(P(x)) \\
&= -\sum_{x\in\chi_{\mathbf{X}}} P(x)\sum_{y\in\chi_{\mathbf{Y}}} P(y|x)\log_2(P(y|x)) - \sum_{x\in\chi_{\mathbf{X}}} P(x)\log_2(P(x)) \\
&= -\sum_{x\in\chi_{\mathbf{X}}} P(x)H(\mathbf{Y}|x) + H(\mathbf{X}) \\
&= H(\mathbf{Y}|\mathbf{X}) + H(\mathbf{X})
\end{aligned} \tag{2.33}$$

Thus proving 2.31. We can use this result iteratively to prove 2.32.

$$\begin{aligned}
H(\mathbf{X}_1,\mathbf{X}_2) &= H(\mathbf{X}_1) + H(\mathbf{X}_2|\mathbf{X}_1) \\
\\
H(\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3) &= H(\mathbf{X}_1) + H(\mathbf{X}_2,\mathbf{X}_3|\mathbf{X}_1) \\
H(\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3) &= H(\mathbf{X}_1) + H(\mathbf{X}_2|\mathbf{X}_1) + H(\mathbf{X}_3|\mathbf{X}_1,\mathbf{X}_2) \\
&\vdots \\
H(\mathbf{X}_1,\mathbf{X}_2,\,\ldots,\mathbf{X}_n) &= H(\mathbf{X}_1) + H(\mathbf{X}_2|\mathbf{X}_1) + \ldots + H(\mathbf{X}_n|\mathbf{X}_{n-1},\,\ldots,\mathbf{X}_1) \\
&= \sum_{i=1}^{n} H(\mathbf{X}_i|\mathbf{X}_{i-1},\,\ldots,\mathbf{X}_1)
\end{aligned} \tag{2.34}$$

□

Theorem 2.2.2 (Entropy is never negative)**.** *For any random variable* $\mathbf{X}$*, its entropy* $H(\mathbf{X})$ *as defined in def. 2.2.7 is a non-negative quantity.*

Proof.

$$\forall x \in \chi_{\mathbf{X}} : 0 \leq P(x) \leq 1 \implies -P(x)\log_2(P(x)) \geq 0 \quad \forall x \tag{2.35}$$

□

Theorem 2.2.3 (Conditioning reduces entropy)**.** *For any two random variables* $\mathbf{X}$ *and* $\mathbf{Y}$*, the following holds:*

$$H(\mathbf{X}|\mathbf{Y}) \leq H(\mathbf{X}) \tag{2.36}$$

The proof of this Theorem is a mere property of a quantity called Mutual Information, but its introduction would be probably out of context here, so I redirect the reader to [7] for an ampler disquisition on the foundations of Information Theory.

We would like to introduce some kind of dynamics in this probabilistic setting. This leads to the definition of stochastic process.

Definition 2.2.10: Stochastic Process
Let $\{\mathbf{X}_i\}$ *be an indexed collection of random variables on the same probability space and with the same alphabet* χ*. Let the probability distribution be defined jointly on this collection.*

$$P(\ldots,\, \mathbf{X}_i = x_1,\, \mathbf{X}_{i+1} = x_2,\, \mathbf{X}_{i+2} = x_3,\, \ldots) = P(\ldots,\, x_1,\, x_2,\, x_3,\, \ldots) \quad (2.37)$$

with $\ldots,\, x_1,\, x_2,\, x_3,\, \ldots \in \bigtimes \chi$ *a suitable (at most numerable) quantity of times. In particular if in the definition of the collection one sets* $n \leq i \leq m$ *with* $n \leq m \in \mathbb{Z}$ *the PMF is defined on finite sequences*[†]*. This kind of structure defines a* Stochastic Process*.*

Usually the index i is given the meaning of a discretisation of time. A particular class of stochastic processes is that of Markov Processes. A definition of them can be given as a case of n-th order process.

Definition 2.2.11: Stochastic Process of m-th order
Let $\{\mathbf{X}_i\}$ *be a stochastic process. It is said to be a* process of m-th order *if and only if the following property on conditional probability holds:*

$$P(\mathbf{X}_{n+1}|\mathbf{X}_n,\, \mathbf{X}_{n-1},\, \ldots) = P(\mathbf{X}_{n+1}|\mathbf{X}_n,\, \mathbf{X}_{n-1},\, \ldots,\, \mathbf{X}_{n-m+1}) \quad (2.38)$$

That means asking that conditioning affects the transition probability up to m states before and including the current one. A stochastic process of 1st order is generally called a Markov Chain *or* Markov Process*.*

We shall now give some definitions of different kinds of stochastic processes.

Definition 2.2.12: Stationary stochastic process
Let $\{\mathbf{X}_i\}$ *be a stochastic process.* $\{\mathbf{X}_i\}$ *is said to be* stationary *if the PMF is invariant under shift of the indices:*

$$P(\mathbf{X}_1,\, \mathbf{X}_2,\, \ldots,\, \mathbf{X}_n) = P(\mathbf{X}_{1+m},\, \mathbf{X}_{2+m},\, \ldots,\, \mathbf{X}_{n+m}) \quad (2.39)$$

with $m \in \mathbb{Z}$*.*

[†]This is the case we'll be concerned with.

In order to make some links with the characteristics we defined on dynamical systems we need to enter the classification of states for a stochastic process. The main tool in states classification is the higher-order transition probability.

Theorem 2.2.4 (Chapman-Kolmogorov Equation)**.** *Let* $\{\mathbf{X}_i\}$ *be a stochastic process. The following holds:*

$$P(\mathbf{X}_1,\, \mathbf{X}_2,\, \ldots,\, \mathbf{X}_{n-1}) = \sum_{x \in \chi_{\mathbf{X}_n}} P(\mathbf{X}_1,\, \mathbf{X}_2,\, \ldots,\, \mathbf{X}_{n-1},\, \mathbf{X}_n = x) \tag{2.40}$$

Where the marginalisation has been effected on the n-th variable for ease of notation, but one could have chosen any in the available range.

Proof. The proof consists of a marginalisation with respect to the variable one wants to remove. □

The Chapman-Kolmogorov Equation allows us to evaluate transition probabilities for given initial and final states and number of steps. In the following we'll be dealing with Markov Chains only.

Definition 2.2.13: n-steps transition probability
Let $\{\mathbf{X}_i\}$ *be a Markov Chain. We denote the probability of passing from state* i *to state* j *in* n *steps as* p_{ij}^n*.*

Theorem 2.2.5 (Expression for transition probabilities in n steps)**.** *Let* $\{\mathbf{X}_i\}$ *be a Markov Chain. The following expression holds for* p_{ij}^n*:*

$$p_{i_1 i_2}^n = \sum_{i_2 \in \chi_{\mathbf{X}_2}} \cdots \sum_{i_{n-1} \in \chi_{\mathbf{X}_{n-1}}} P(i_1, i_2,\, \ldots,\, i_{n-1},\, i_n) \tag{2.41}$$

By convention it is set that $p_{ii}^0 = 1$*.*

Proof. The proof consists in a repeated application of the Chapman-Kolmogorov Equation, to marginalise the joint PMF with respect to the intermediate transitions. □

There are some ulterior auxiliary quantities useful in dealing with the classification of states. A very important one is the *first passage probability after* n *steps* f_{ij}^n, the probability of arriving for the first time on j after having started from i, n time steps ago. By summing over all possible ns one can define $f_{ij} \equiv \sum_{\nu=1}^{\infty} f_{ij}^{\nu}$ the probability of ever reaching state j from i. If such quantity converges to 1 it is possible to set the following definition.

Definition 2.2.14: First passage distribution
Let $\{\mathbf{X}_i\}$ *be a Markov Chain. We call* First passage distribution *through* j *from* i*, the probability distribution of* n *defined by* f_{ij}^n *for given* i, j *and denote it as* $\{f_{ij}^n\}$*, if the sum of such terms* $\sum_{\nu=1}^{\infty} f_{ij}^{\nu}$ *converges to 1.*

In particular by setting $i = j$, and if the sum of the probabilities is normalised, one obtains the *recurrence times distribution*, $\{f_{ii}^n\}$ for state i, that is the probability of returning to state i for the first time after n time steps since having started from there. I will present, without proving it, an equation that implicitly defines the $\{f_{ij}^n\}$ as its solutions.

Theorem 2.2.6 (Equation for the first passage probabilities)**.** *Let* $\{\mathbf{X}_i\}$ *be a Markov Process. The first passage probabilities* $\{f_{ij}^n\}$ *are a solution to the following equation:*

$$p_{ij}^n = \sum_{\nu=1}^{n} f_{ij}^{\nu} p_{jj}^{(n-\nu)} \tag{2.42}$$

Interpreting, if it is possible, $\{f_{ii}^n\}$ as the distribution of the recurrence times, it is feasible to take the mean value of n over it, thus defining the *mean recurrence time.*

Definition 2.2.15: Mean recurrence time
Let $\{\mathbf{X}_i\}$ *be a Markov Process. Let* $\{f_{jj}^n\}$ *be the associated recurrence time distribution, definable if the recurrence time probabilities are normalised. The mean recurrence time in this case is defined by:*

$$\mu_j \equiv \langle n \rangle_{\{f_{jj}^n\}} \tag{2.43}$$

With the aid of these definitions we can give a general classification of the states in a process's alphabet.

Definition 2.2.16: Period of a state
Let $\{\mathbf{X}_i\}$ *be a Markov Chain.* $t > 1$ *is called the period of state* j *if* $p_{jj}^n = 0$ *unless* $n = \nu t$ *with* ν *any non-zero integer, and* t *is the largest integer with this property. If no such integer* $t > 1$ *exists then* j *is called an* aperiodic *state.*

Taking periodicity out of the picture, another legitimate question is whether a return to a given state is certain. In this view, we give a definition of persistence.

Definition 2.2.17: Persistent state
Let $\{\mathbf{X}_i\}$ *be a Markov Process. A state* i *is said to be* persistent *if* $f_{ii} = 1$*, that is if a return to it is asymptotically certain. A persistent state is called a* null state *if its mean recurrence time diverges* $\mu_i = \infty$*.*

In this context it is possible to define ergodicity as a "local" property.

Definition 2.2.18: Ergodic state
Let $\{\mathbf{X}_i\}$ *be a Markov Process. A persistent and aperiodic state* i *with* $\mu_i < \infty$ *is called* ergodic.

An important property in stochastic processes is irreducibility.

Definition 2.2.19: Irreducible process
Let $\{\mathbf{X}_i\}$ *be a Markov Chain. It is called* irreducible *if each state can be reached from every other state.*

The irreducibility of a chain implies that all states are of the same type, that is, they do or do not satisfy all the same definitions in matters of ergodicity, periodicity, etc.

Theorem 2.2.7. *All states of an irreducible Markov Chain are of the same type.*

This result is instrumental to another, more important, one.

Theorem 2.2.8. *If a Markov Chain is irreducible and ergodic, the limits defined by:*

$$u_j = \lim_{n\to\infty} p_{ij}^n \tag{2.44}$$

exist and are independent of i*. The* u_j *define a probability distribution since:*

$$\begin{aligned} u_j &> 0 \\ \sum_j u_j &= 1 \end{aligned} \tag{2.45}$$

Moreover, the u_j *define a stationary state of the chain:*

$$u_j = \sum_i u_i p_{ij} \tag{2.46}$$

In the opposite direction: if for an irreducible and aperiodic chain there exist some real numbers u_j *satisfying 2.45 and 2.46 then all states are ergodic (i.e. the chain is ergodic), the* u_j *satisfy 2.44 and for each of them:*

$$u_j = \frac{1}{\mu_j} \tag{2.47}$$

With μ_j *the mean recurrence time for state* k*.*

For a proof of this theorem and the preceding one, the reader is invited to look at Chapter 15, Paragraphs 6 and 7, of [8]. This last Theorem is particularly interesting if we remember the results on ergodic systems recalled in Section 2.1. Indeed it shows that, similarly to ergodic systems, on the long run the process loses information on his starting state, and the chance to find it in a state rather than another depends only on the state itself. The difference, although, is in the fact that since no measure is given, *a priori*, on the alphabet, the probability will be determined, in this case by the u_js of the stationary distribution.

We'd like to introduce Entropy for stochastic processes.

Definition 2.2.20: Entropy of a stochastic process
Let $\{\mathbf{X}_i\}$ be a stochastic process, we define its entropy $H(\chi)$ as follows:

$$H(\chi) = \lim_{n\to\infty} \frac{1}{n} H(\mathbf{X}_1,\, \mathbf{X}_2,\, \mathbf{X}_3,\, \ldots,\, \mathbf{X}_n) \tag{2.48}$$

When such quantity is defined.

The entropy of a stochastic process coincides with that of a partitioned dynamical system. There is another quantity, related to entropy, in a stochastic process.

Definition 2.2.21: Entropy rate of a stochastic process
Let $\{\mathbf{X}_i\}$ be a stochastic process, we define its entropy rate $H'(\chi)$ as follows:

$$H'(\chi) = \lim_{n\to\infty} H(\mathbf{X}_n|\mathbf{X}_{n-1},\, \mathbf{X}_{n-2},\, \mathbf{X}_{n-3},\, \ldots,\, \mathbf{X}_1) \tag{2.49}$$

When such quantity is defined.

We now prove that for stationary processes, both limits exist, and they coincide.

Theorem 2.2.9. *For a stationary stochastic process the limit in eq. 2.49 exists.*

Proof. Let $\{\mathbf{X}_i\}$ be a stochastic process, its entropy rate reads:

$$H'(\chi) = \lim_{n\to\infty} H(\mathbf{X}_n|\mathbf{X}_{n-1},\, \mathbf{X}_{n-2},\, \mathbf{X}_{n-3},\, \ldots,\, \mathbf{X}_1) \tag{2.50}$$

We perform the following operations:

$$\begin{aligned} H(\mathbf{X}_n|\mathbf{X}_{n-1},\, \ldots,\, \mathbf{X}_1) \leq & H(\mathbf{X}_n|\mathbf{X}_{n-1},\, \ldots,\, \mathbf{X}_2) \\ = & H(\mathbf{X}_{n-1}|\mathbf{X}_{n-2},\, \ldots,\, \mathbf{X}_1) \end{aligned} \tag{2.51}$$

Where the inequality holds because conditioning reduces entropy, and the equality is a consequence of the shift invariance of probability for stationary processes. Therefore the entropy rates form a non-negative, non-increasing sequence, and thus have a limit, $H'(\chi)$. □

Theorem 2.2.10. *For a stationary stochastic process the limit in eq. 2.48 exists, and is equal to* $H(\chi)$.

Proof. By the chain rule one can decompose the right hand side of def. 2.2.20, before taking the limit, into a sum of entropy rate terms:

$$\frac{H(\mathbf{X}_1,\, \mathbf{X}_2,\, \mathbf{X}_3,\, \ldots,\, \mathbf{X}_n)}{n} = \frac{1}{n}\sum_{i=1}^{n} H(\mathbf{X}_i|\mathbf{X}_{i-1},\, \ldots,\, \mathbf{X}_1) \tag{2.52}$$

The Cesàro mean theorem states that, if a sequences converges to a limit, then the sequence of its running averages converges to the same limit. Taking the limit on both sides in eq. 2.52, the right hand side is the running average of a sequence of terms of the form $H(\mathbf{X}_i|\mathbf{X}_{i-1},\, \ldots,\, \mathbf{X}_1)$, but those converge to $H'(\chi)$, therefore by Cesàro's mean theorem, the RHS converges to the same value, thus yielding:

$$H(\chi) = H'(\chi) \tag{2.53}$$

□

An interesting comparison can be made by taking a look simultaneously at def. 2.1.9, given for a dynamical system, and 2.2.20. In fact, the (normalised) measure of the elements of the $\vee$-intersection of the n time evolutions of the partition has the very same meaning of the joint probability mass function of n variables, defining a stochastic process: the probability for the system to have visited a given ordered collection of n states during its evolution. In this sense the states of the stochastic process and the atoms of the partition are identified. Moreover: if the phase flow is discrete and it is a function of the sole coordinates, the evolved of a set depends only on the starting set itself, therefore the kind of process generated upon partitioning is Markovian, with the transition probabilities from a state to another given by the measure of the overlapping region of the starting one and the evolved one, normalised by the measure of the starting one. This parallel is also suggested by the fact that some characteristics (e.g. ergodicity) can be defined for both kinds of systems, presenting similar interconnections with each other in both settings (e.g. the long-term memory loss). Also, some quantities' definition coincide (e.g. entropy), upon correct interpretation of the playing terms. This allows us to introduce a general rule for the discretisation of a dynamical system:

Given a generic, discrete-time, dynamical system $(\mathcal{M}, \mu, \Phi)$, if it is possible to normalise μ into a probability measure, then it is possible to generate a stochastic process upon introduction of a partition P defining the states. In this context the entropy rate of the process coincides with the entropy of the system with respect to P, and the former can be studied in lieu of the latter. If Φ is solely a function of the current coordinates, the process is a Markov Chain, with transition probabilities given by $P(j|i) = \frac{\mu(\Phi P_i \cap P_j)}{\mu(P_i)}$.

2.3 Information theoretic notions on average description length

Up to this point, in the current presentation, the entropic properties of processes have been seen as something that is defined by the mathematical expressions regulating the process itself, and computable *from* said expressions. In a physical context we would like to have a way of estimating entropic quantities from experimental data, especially in the case when the phase flow is particularly difficult to work with or unknown at all. Information theory, in particular the side products of its results on data compression, come to our avail.

The object of Information Theory is the study of the quantification, storage and communication of information. Information is intended in the sense of the amount of data needed, on average, to reliably transmit the symbols produced from a process called *source* to a *receiver*. The source appears to the receiver as a probabilistic process, in the sense that the latter is instructed on the admissible symbols and on their probability, but has no deterministic, *a priori* knowledge of the sequence they're about to receive. In order to transmit the symbols produced by the source, it is necessary to encode them. Note that it is impossible to avoid coding, in this sense, as the very fact of having labelled the states constitutes an inevitable coding.

Definition 2.3.1: Coding of a random process
Let $\mathbf{X}$ *be a random variable with alphabet* χ*. A* D*-ary code* C *is any map* $C : \chi \to \mathrm{D}^*$*, with* D^* *the set of finite-length strings of characters from a finite set* D*, such that* $\#\mathrm{D} = D$*. We denote with* $C(x)$*,* $x \in \chi$ *the encoding of each outcome, and with* $l(x)$*,* $x \in \chi$ *the length of the* D*-ary string associated with* x*. A code is said* non-singular *if* C *is 1-1.*

We're assigning to each outcome in χ a finite string in D^*, called *codeword*. The legitimate question of how many characters will be needed on average for each symbol transmitted, when using a certain code C, can then be answered by averaging $l(x)$ over $P(x)$.

Definition 2.3.2: Average description length
Let $\mathbf{X}$ *be a random variable with alphabet* χ*. Let* C *be a code for its outcomes. The* average description length *of* $\mathbf{X}$*, using* C*, is defined as:*

$$L(C, \mathbf{X}) = \langle l(x) \rangle_{\mathbf{X}} = \sum_{x \in \chi} P(x) l(x) \tag{2.54}$$

Non-singularity is an important feature, as it allows the reliable encoding and decoding of the strings representing each source symbol, if taken singularly.

In general, though, we are interested in the transmission of a sequence of symbols, therefore it is necessary for the receiver to be able to tell each character from the preceding and following one, in order to avoid the ambiguity that could occur if two codewords contained, when concatenated, a third one. An idea could be to select an element of D^* as a separator between characters, but this would mean to loose a symbol over the alphabet and potentially to almost duplicate the size of any message. A better idea would be to devise a *self-punctuating code*, that is a code where no separator is needed between characters.

Definition 2.3.3: Prefix Code
A code C is said to be a prefix code *or an* instantaneous code *if for each finite sequence used as a codeword, all those sequences that begin with said sequence are not codewords. In other words, if no codeword is contained as a prefix in another codeword.*

This kind of device solves the problem because as soon as the receiver recognises a sequence matching any codeword, he may start listening for the next one, since no other codeword could have started with that same sequence. This kind of code is subject to an inequality determined by its prefix-free structure.

Theorem 2.3.1 (Kraft's inequality)**.** *Let C be a D-ary, istantaneous code, with word lengths l_1, l_2, l_3 ..., l_{max}. The following inequality holds:*

$$\sum_i D^{-l_i} \leq 1 \tag{2.55}$$

And conversely, given any set of lengths l_1, l_2, l_3 ..., l_{max} satisfying Kraft's inequality, it is possible to construct a prefix code with those lengths.

Proof. Given a finite maximal length l_{max}, the number of D-ary words, l_{max} characters long, is $D^{l_{max}}$. Some of these are acceptable codewords and some are not. In order to create the codewords we start from no character at all, and add one character at a time: we keep one of the D possible words as a proper codeword, thus removing all those to which it would be a prefix, and we use as prefixes the remaining $D-1$. Thus, for each l_i we are removing $D^{l_{max}-l_i}$, l_{max} character long possible words. Eventually, though, we are keeping D of them, since there are D l_{max}-long codewords which have none of the previous sequences as prefix. Therefore the total number of l_{max}-long words removed is inferior to the total number of potentially available l_{max}-long words:

$$\begin{aligned} &\sum_i D^{l_{max}-l_i} \leq D^{l_{max}} \\ \implies &\sum_i D^{-l_i} \leq 1 \end{aligned} \tag{2.56}$$

□

The inequality can be generalised to a countable set of instantaneous codewords. The advantage of using prefix codes is that they have codewords of a number of different lengths. This permits distributing cleverly the shortest codewords on the most frequent symbols, thus effectively reducing the average description length. These prefix code are indeed a good starting point in the problem of designing the *optimal code* for a given source, that is the code that has the absolutely least average description length. The work of Information theory continues by minimising $L(C, \mathbf{X})$ on the constraints set by Kraft's inequality and by the request that the lengths $l_1, l_2, \ldots$ be positive integers. I will bring up only a result, without proving it, that has been of cardinal importance in the present work. Although I presented some ideas from information theory for symbols generated by a single random variable, the concepts can be extended to stochastic sources. Indeed the case of i.i.d. (independent identically distributed) variables is that of a process without memory[‡].

Theorem 2.3.2 (Bounds on the average description length for an optimal coding)**.** *Let $\{\mathbf{X}_i\}$ be a stochastic process, and let C^* be the optimal code for such process. There exists a bound on L^*, the average description length of $\{\mathbf{X}_i\}$ with C^*.*

$$\frac{H(\mathbf{X}_1,\, \mathbf{X}_2,\, \ldots,\, \mathbf{X}_n)}{n} \leq L^* \leq \frac{H(\mathbf{X}_1,\, \mathbf{X}_2,\, \ldots,\, \mathbf{X}_n) + 1}{n} \qquad \forall n \qquad (2.57)$$

In particular, if $\{\mathbf{X}_i\}$ is stationary, the condition is stronger: introducing L_n^*, the mean description length over n symbols one has the following.

$$\lim_{n\to\infty} L_n^* = H(\chi) \qquad (2.58)$$

For a proof of this Theorem the reader is redirected to [7].

This result is fundamental. Indeed, after having defined a stochastic process from a dynamical system, by introduction of a partition, it is possible to use it as a source of symbols. Then by building an optimal code for the process, it is possible to get arbitrarily good estimates of the system's entropy, given that one has access to sufficiently long strings of symbols generated by the system. Nevertheless, in many cases this is not enough. This is because the construction of an optimal code for a process still requires to be in possession of a lot of information on it. An idea, in this case, is to build a perfectly invertible compression algorithm that asymptotically achieves the optimal average description length. The class of algorithms that achieve this result is that

[‡]Indeed for a process with PMF defined on n i.i.d. variables it is easy to prove $H(\chi) = H(\mathbf{X})$ by use of the n-ary chain rule.

of *optimal compression algoritms*. I will present the algorithm I used on the data, slightly adapted from its presentation in [7], in order to produce correct estimates for an n-ary process. In the same book, the interested reader will also find a proof of optimality for said algorithm. The algorithm is generally known as Lempel-Ziv 78 (LZ78) or Tree-Structured Lempel-Ziv algorithm, as it is due to the computer scientists Abraham Lempel and Yaakov Ziv.

The Lempel-Ziv 1978 algorithm

The basic idea of LZ78 is to run over the input stream of symbols and identify, at each time, the shortest string never-before-seen upstream. In other words, it is performing what is called a *distinct parsing* of the stream. Once it has found one, it means that the string is formed by an already-seen string with an extra character appended in the end, therefore the never-before-seen string will be encoded by the position of the prefix in the coded message and the last character. The algorithm's correct start, stop and ability to deal with new characters at runtime are granted by the *a priori* insertion among the already-seen strings of the empty set $\emptyset$. It is used as a prefix for new characters (including the first ever of the stream) and as a postfix in case the last string is an already seen one. Let's follow the algorithm's work, step by step, on an example: let the input stream be the one pictured below. The cursor will be represented by a vertical line after the current character.

$$\text{Input: } |AABBAABABBBA \qquad \text{Output:}$$

The algorithm reads the first character: since it is a new character, it appends $(\emptyset, A)$ to output.

$$\text{Input: } A|ABBAABABBBA \qquad \text{Output: } (\emptyset, A)$$

The algorithm advances one step further, finding another A. As A is an already known string, it goes forward again, thus reading AB, which hasn't been seen before. The algorithm writes to output: $(1, B)$, meaning that to reconstruct the string it has seen, it is necessary to get the 1st string in the compressed message and append a B to it.

$$\text{Input: } AAB|BAABABBBA \qquad \text{Output: } (\emptyset, A), (1, B)$$

Moving another step further the algorithm finds letter B, which it hasn't seen alone, as yet: it writes $(\emptyset, B)$ to output.

$$\text{Input: } AABB|AABABBBA \qquad \text{Output: } (\emptyset, A), (1, B), (\emptyset, B)$$

Going forward the next suitable sequence is AA, the pair inserted on output is $(1, A)$.

$$\text{Input: } AABBAA|BABBBA \qquad \text{Output: } (\emptyset, A), (1, B), (\emptyset, B), (1, A)$$

After an adequate number of steps we obtain the following:

$$\text{Input: } AABBAABABB|BA \qquad \text{Output: } (\emptyset, A), (1, B), (\emptyset, B), (1, A), (3, A), (3, B)$$

At this point the cursor will reach the end-of-input marker without having found any new string. This means that the string currently under examination has already been seen, therefore the pair appended to output will contain the pointer to the former occurrence of said string and the empty set as a postfix.

$$\text{Input: } AABBAABABBBA| \qquad \text{Output: } (\emptyset, A), (1, B), (\emptyset, B), (1, A), (3, A), (3, B), (5, \emptyset)$$

A flow-chart demonstrating the procedure followed by the algorithm in more practical terms is shown in Fig. 2.3. It can be proved that LZ78 is an optimal compression scheme for stationary and ergodic sources. A proof is presented in Paragraph 13.5.2 of [7] for binary processes. Since the data on which the algorithm has been used weren't from a binary source, but indeed from a D-ary one, I would like to present a version of the optimality theorem slightly modified to fit the needs of the study.

Theorem 2.3.3 (Optimality of LZ78)**.** *Let $\{\mathbf{X}_i\}$ be a stationary and ergodic stochastic process on a D-ary alphabet. By defining the LZ78 codeword length for a given input string $(x_1,\ x_2,\ \ldots,\ x_n)$ as:*

$$l(x_1,\ x_2,\ \ldots,\ x_n) = c(n)(\log_2(c(n)) + \log_2(D)) \tag{2.59}$$

Where $c(n)$ represents the number of strings in which the algorithm divides the input $(x_1,\ x_2,\ \ldots,\ x_n)$.
The following holds:

$$\limsup_{n\to\infty} \frac{l(x_1,\ x_2,\ \ldots,\ x_n)}{n} \leq H(\chi) \quad \text{with probability } 1^{\S} \tag{2.60}$$

With $H(\chi)$ the entropy rate of the process.

This theorem completes the theoretical background necessary for the carrying out of the measurements. Indeed it provides an important tool, since it

[§]For the several definitions of convergence for random variables the reader can look into [8] and [7].

allows to estimate the entropy rate of a system from its encoded experimental trajectories, by measuring the average description length per-symbol after their lossless compression. The estimate can be carried out up to a desired degree of approximation, determined also by the amount of available data. Among several available algorithms for lossless compression LZ78 has been chosen also for its relatively low computational complexity that goes as $O(N \log(N))$, with N number of symbols. For an optimality evaluation of my implementation of the algorithm, see Appendix C.

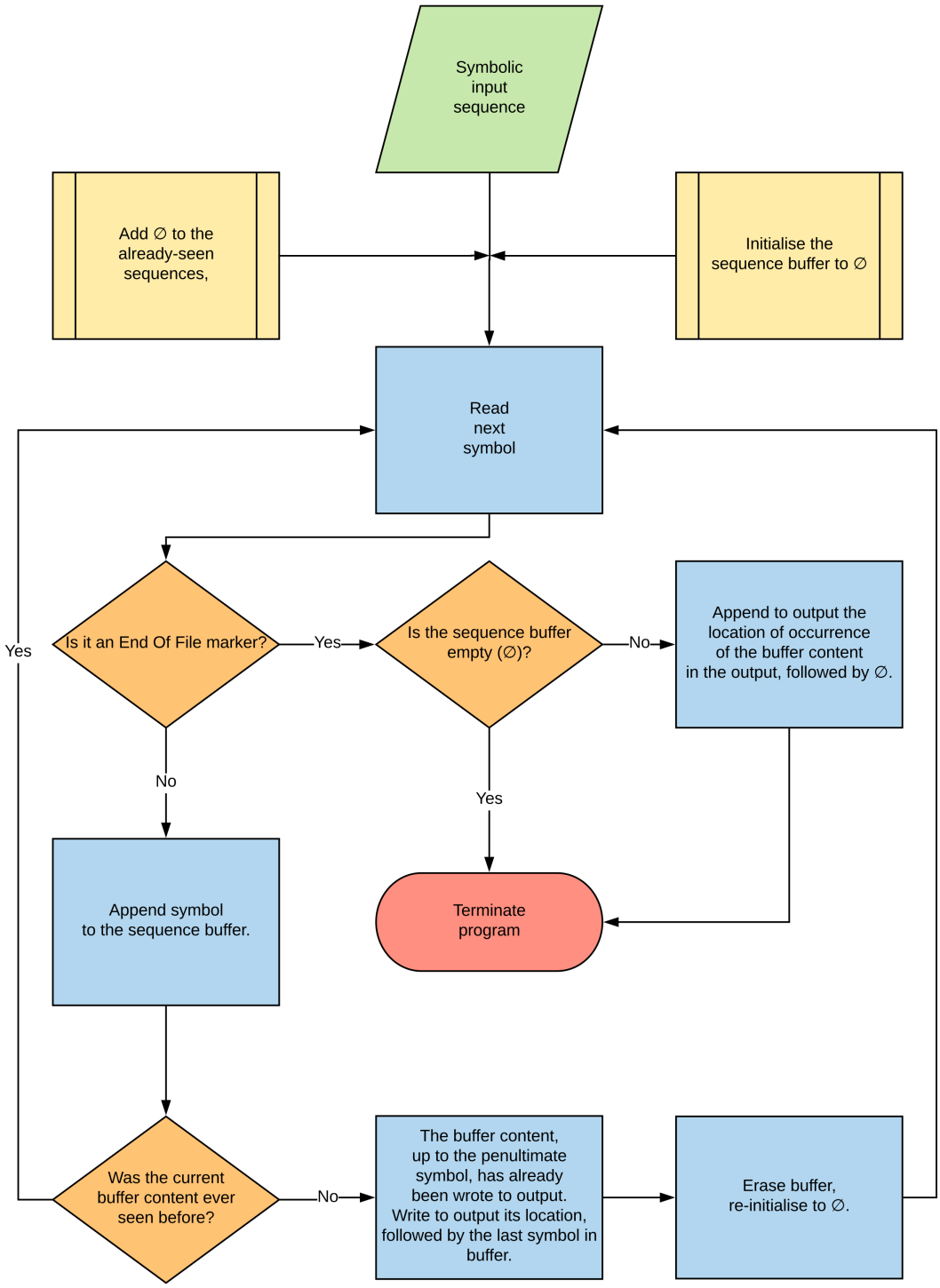

Figure 2.3: A flow-chart portraying the LZ78 algorithm.

Chapter 3

Results

3.1 Mobility network overview

An interesting preliminary observation, is to see to what degree the overall most connected cells are all interconnected among themselves. To this end, I built an undirected graph, increasing of one a link's weight between two cells, whenever a travel started from one and ended on the other, or vice versa.

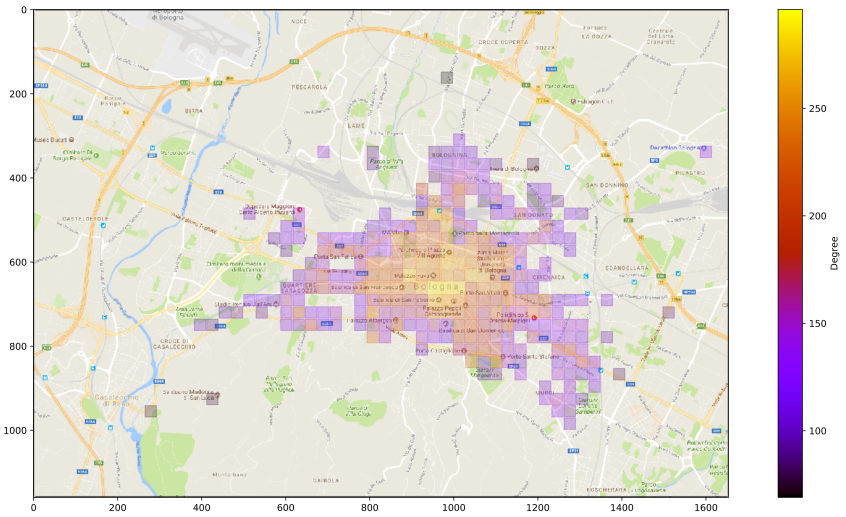

Figure 3.1: A colour-coded map of the set of cells generating the 86.4% of the total mobility. The weights are referred only to the generating set.

One says that a certain set of cells *generates the X% of the mobility* when,

by keeping only the travels that start or finish in a cell in the set, one preserves the $X\%$ of all the travels. Figure 3.1 displays the most connected nodes of said graph. From this analysis, the city's dynamics appears to be generated by a very highly connected nucleus in the city centre, and by some more peripheric branches, that run along the streets coming out of the most important gates. There are also some highly connected, but geographically isolated cells. These are mainly outdoors points of interest, common destinations during the summer and spring, such as a number of gardens or the Basilica di San Luca.

3.2 Further preparation of the timed patterns

By inspecting some of the actual strings generated from the data, it appeared that many travels contained stationary periods of different length. In order to have proper mobility patterns, it is necessary to spot the stationary periods in timed patterns. Such periods need to be removed, and the patterns connecting them are to be considered as separated travels. To do so in practice, it is necessary to devise a criterion. I chose to set a numeric threshold for character repetitions. This way if a symbol is contiguously repeated more times than it is suitable in actual mobility, the travel is broken down into an appropriate number of sequences. This selection, thus, allows to set a resolution for social activities. In fact any actual social activity, shorter than this resolution, is regarded as mobility. After having performed this operation on the timed patterns, the pattern length distribution appears as in Figure 3.2. Since the

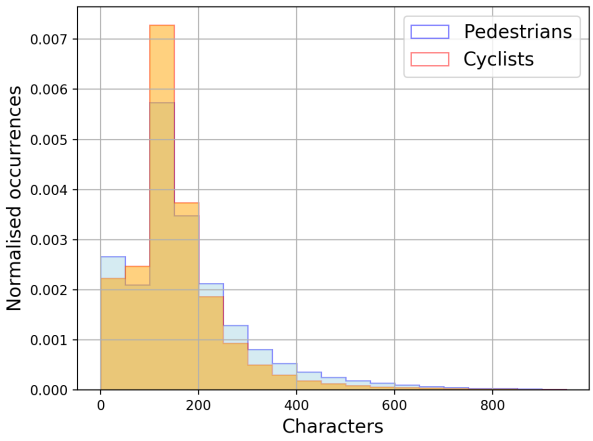

Figure 3.2: The length distribution after breaking down all the sequences, when a stop longer than 15 minutes was detected.

LZ78 algorithm is *asymptotically* optimal, on short strings its estimate tends

to yield always the same value, independently of the sample* Therefore it is necessary to select a length threshold that sequences need to surpass in order to enter the entropy calculation. In timed patterns this translates also into a duration requisite, because of the uniform time resampling the sequences underwent.

3.3 Timed patterns analysis

Two relevant quantities used in the filter were used as control parameters. The first is the minimal amount of time a travel has to last in order to be included in the entropy computation, we'll call this the *Time Threshold (T.T.)*. The second is the maximal amount of time an individual can stay in the same cell before triggering the break down of the string in several travels, we'll call this the *Breaking Time (B.T.)*. All of the following are histograms obtained from the timed patterns. In each entropy rate histogram I will indicate both of the control quantities. I now will show three histograms, obtained for a B.T. of 15 minutes and varying the T.T.

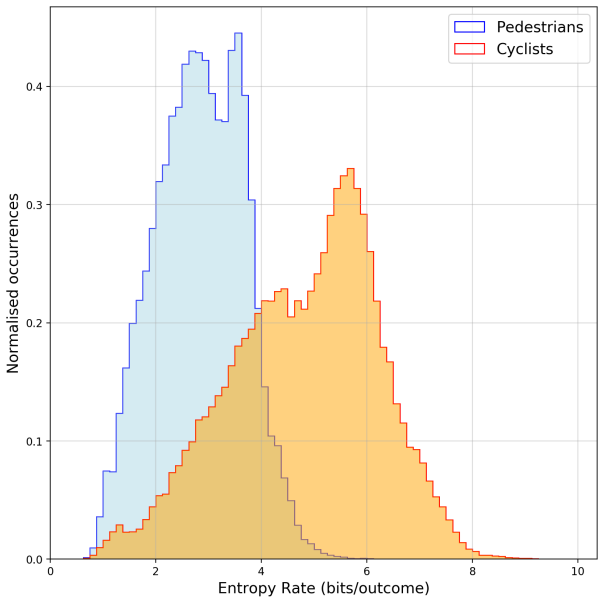

Figure 3.3: Timed entropy rate for 15 minutes T.T. and 15 minutes B.T.

*The independence on the sample has been spotted because of the presence, in the first run, of a very high entropy spike on an exact value close to 3.78 bits/outcome, which turned out to be the value assigned by the algorithm to any two-character string of two different characters.

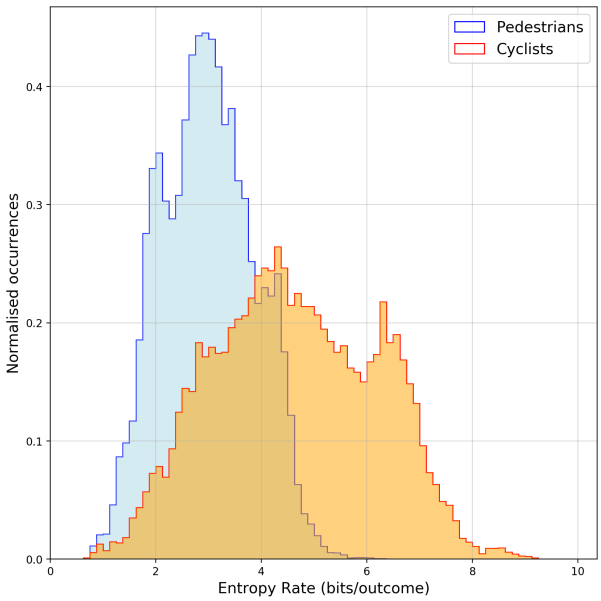

Figure 3.4: Timed entropy rate for 30 minutes T.T. and 15 minutes B.T.

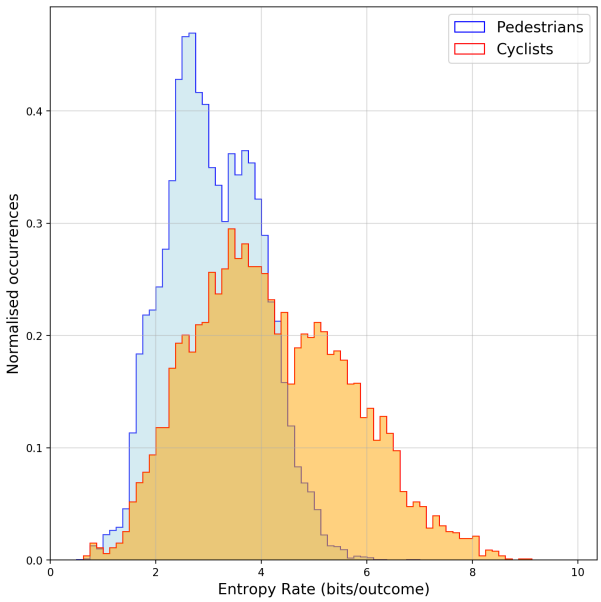

Figure 3.5: Timed entropy rate for 45 minutes T.T. and 15 minutes B.T.

In Figure 3.3, Pedestrians show two peaks very close to each other, respectively around 2.75 and 3.5 bits/outcome, almost of the same height. Cyclists, on the other hand, have two uneven peaks, the most prominent at around 5.5

bits/outcome and the other around 4 bits/outcome. Thus both distributions are bimodal. In Figure 3.4, pedestrians have a single principal peak around 3 bits/outcome, and cyclists have one around 4 bits/outcome. This points out that the most entropic activities for cyclists happen in the 15 to 30 minutes time scale. Indeed, increasing the T.T. to 45 minutes in Figure 3.5, only the lower entropy peaks remain: around 2.5 bits/outcome for pedestrians, and around 3.5 bits/outcome for cyclists. Short travels, therefore, show higher entropy rates, indicating a wider set of visited locations. I present now two histograms obtained by fixing the T.T. to 30 minutes, for values of B.T. of 30 and 45 minutes, in order to select travels that link social activities of different durations.

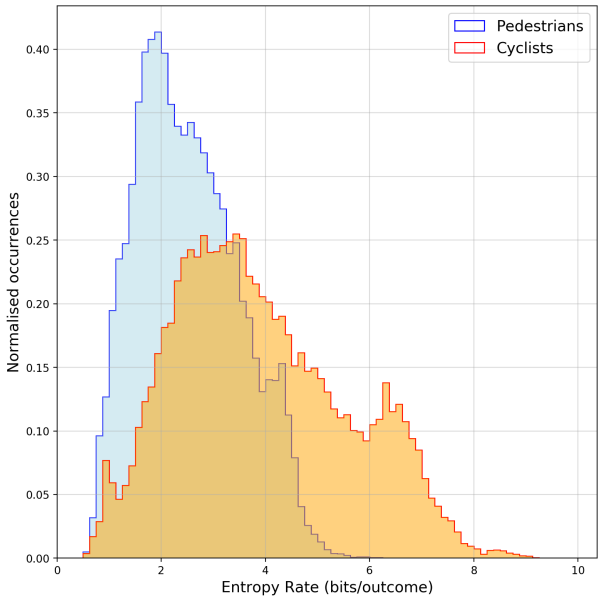

Figure 3.6: Timed entropy rate for 30 minutes T.T. and 30 minutes B.T.

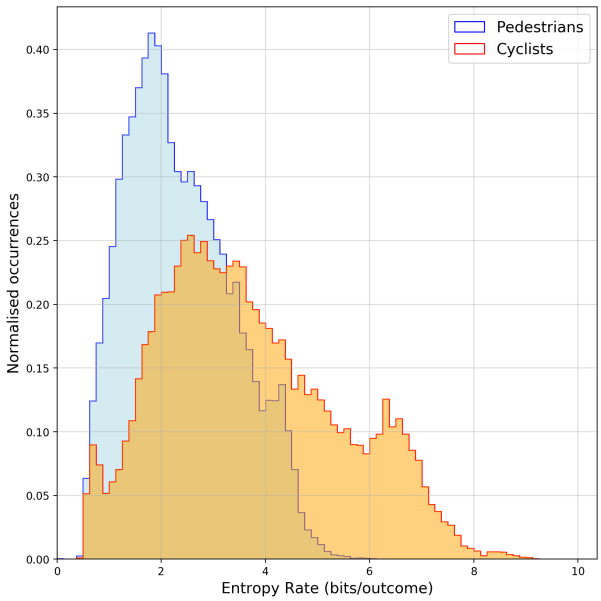

Figure 3.7: Timed entropy rate for 30 minutes T.T. and 45 minutes B.T.

We can compare these histograms with 3.4 as well, since they all share the T.T. of 30 minutes. In Figure 3.6, pedestrians have essentially a single peak roughly around 2 bits/outcome. Cyclists have two peaks, the principal one being around 3 bits/outcome. In Figure 3.7 both distributions keep roughly the same peaks. By raising the B.T. one chooses to view as separated travels only the strings that connect stopping periods longer than B.T.. If one assumes that prolonged stopping periods correspond to social activities, Figures 3.4, 3.6 and 3.7 represent the entropy distributions for travels connecting activities of durations respectively greater than 15, 30 and 45 minutes. I found that entropy is lower, for travels that connect longer activities. This is true for both cyclists and pedestrians. In Table 3.1, I display the number of entries of each histogram in this Section.

Time Threshold (min)	Breaking Threshold (min)	Pedestrians	Cyclists
15	15	24007	15703
30	15	35799	28998
45	15	13637	7859
30	30	51651	42579
30	45	56684	46175

Table 3.1: The number of samples used to fill each histogram.

Conclusions

In the present work, I analysed the entropic properties of human mobility in the Bologna city area. The data I used was made out of the BellaMossa 2017 database pedestrian and cyclist records in the Bologna Metropolitan City. The data have been filtered with a simple procedure taking into account the inter-record speeds. The resulting database has been encoded using a 200m sided square grid, covering the whole ROI. I implemented a version of the Lempel-Ziv 1978 compression algorithm, modified to compute entropy rates on N-ary stochastic processes, and used it on the encoded, uniformly interpolated, trajectories. The results, presented in Chapter 3, point out some interesting features of pedestrian and bicycle mobility. I plotted the entropy rate distributions for several values of the control parameters *Time Threshold* and *Breaking Time*. By comparing them I found out that entropy tends to be higher on the small scale, in the sense that considering short travels that connect short activities, the entropy distribution is peaked on higher values. In particular the peak of highest entropy for cyclists vanishes when one doesn't consider the travels shorter than 15 minutes, indicating that the most entropic bicycle travels happen in this time scale. The pedestrians' distribution, instead, loses the most entropic peak on the 30 minutes time scale. This is reasonable, taking into account the higher speed attainable by a cyclist with respect to a pedestrian. If one neglects the shorter activities, considering them a part of mobility, entropy decreases. Looking at the same trajectories, but interrupting them only for activities as long as 45 minutes, both distributions tend to be peaked around 2 bits. This means that looking at activities on the long time scale allows to perceive their origin-destination nature, overlooking the local, minor deviations that naturally incur. This highlights the short-range-disordered, long-range-ordered nature of human urban mobility. A continuation of the work could consist in the analysis of the trajectories separating them by day of the week, time of the day or district of origin. In order to highlight the geometric properties of the trajectories, it could also be interesting to work with the *jump patterns*.

Appendix A

Distances calculations

In order to measure the distance between two different geographical points it is necessary to take into account the shape and curvature of the earth. The longer the distances, the more precise the used approximation has to be. For the length scales considered in this work a spherical approximation was deemed sufficient. The paths traveled by the users are described by polygonal chains, where the length of each segment generally much shorter than the overall length of the chain. For this reason it has also been considered satisfactory to approximate the length of each element of the chain, to that of an element of infinitesimal length on the surface of a sphere. The infinitesimal element of

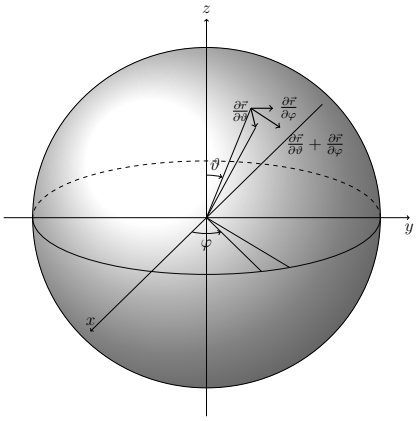

Figure A.1: The tangent space to a sphere.

length can be obtained from the parametrisation of the sphere's surface. The metric matrix for the spherical coordinates on a sphere is the following:

$$G = \begin{bmatrix} \frac{\partial \vec{r}}{\partial \varphi} \cdot \frac{\partial \vec{r}}{\partial \varphi} & \frac{\partial \vec{r}}{\partial \varphi} \cdot \frac{\partial \vec{r}}{\partial \vartheta} \\ \frac{\partial \vec{r}}{\partial \vartheta} \cdot \frac{\partial \vec{r}}{\partial \varphi} & \frac{\partial \vec{r}}{\partial \vartheta} \cdot \frac{\partial \vec{r}}{\partial \vartheta} \end{bmatrix} = \begin{bmatrix} r^2 \sin^2 \vartheta & 0 \\ 0 & r^2 \end{bmatrix} \tag{A.1}$$

The element of length, therefore, is defined by:

$$\mathrm{d}s^2 = \begin{bmatrix}\mathrm{d}\varphi & \mathrm{d}\vartheta\end{bmatrix} \begin{bmatrix} r^2 \sin^2\vartheta & 0 \\ 0 & r^2 \end{bmatrix} \begin{bmatrix}\mathrm{d}\varphi \\ \mathrm{d}\vartheta\end{bmatrix} = r^2 \sin^2\vartheta \, \mathrm{d}\varphi^2 + r^2 \mathrm{d}\vartheta^2 \tag{A.2}$$

In order to pass from standard spherical coordinates to a latitude-longitude coordinate system in degrees, the following conversions apply, for points in the northern hemisphere.

$$\begin{aligned} \varphi &\mapsto \frac{\pi}{180^\circ} \,\mathrm{lon} \\ \vartheta &\mapsto \frac{\pi}{2} - \frac{\pi}{180^\circ} \,\mathrm{lat} \end{aligned} \tag{A.3}$$

With these expressions, and turning the differentials into finite differences, we can build the final formula as used in the programs implemented for this study.

$$\Delta s = \sqrt{k^2 r_{Earth}^2 \cos^2(k \,\mathrm{lat}) \Delta\mathrm{lon}^2 + k^2 r_{Earth}^2 \Delta\mathrm{lat}^2} \tag{A.4}$$

Where, to unburden the notation, it has been set $k = \frac{\pi}{180^\circ}$.

Appendix B

Stable and unstable manifolds of a point

In the definition of Markov partitions we encountered the concepts of stable and unstable manifolds. They are defined as follows.

Definition B.0.1: Stable manifold of a point
Let $(\mathcal{M}, \mu, \Phi)$ be a dynamical system, let $x \in \mathcal{M}$ be a point of its phase space. We define the stable manifold *of x with respect to Φ, $W^s(\Phi, x)$ as follows:*

$$W^s(\Phi, x) \equiv \{y \in \mathcal{M} \mid \lim_{n \to +\infty} d(\Phi^n(y), \Phi^n(x)) = 0\} \tag{B.1}$$

The stable manifold of a point therefore is the set of points attracted to its evolution as time evolves.

Definition B.0.2: Unstable manifold of a point
Let $(\mathcal{M}, \mu, \Phi)$ be a dynamical system, let $x \in \mathcal{M}$ be a point of its phase space. We define the unstable manifold *of x with respect to Φ, $W^s(\Phi, x)$ as its stable manifold with respect to Φ^{-1}:*

$$W^u(\Phi, x) = W^s(\Phi^{-1}, x) \tag{B.2}$$

The unstable manifold, conversely, is the set of points which are repelled by the point's evolution as time flows. These concepts represent a generalisation of the stable and unstable manifolds of a fixed point of the phase flow.

Appendix C

Optimality assessment of the implemented LZ78 algorithm

Theorem 2.3.3 guarantees *asymptotical* convergence, but in experimental applications one faces finite strings. To find out the magnitude of the finite string effect, I performed tests on samples of known entropy. This way of testing the algorithm was inspired by [10], where several estimates are done on binary processes. My implementation of the algorithm can be found on the project repository [1]. I selected sequences of independent, identically distributed (i.i.d.) uniform discrete distributions as testing processes. For a process of i.i.d. variables, the entropy rate is equal to the entropy of a single variable, by Theorem 2.2.1:

$$\begin{aligned} H(\chi) &= \lim_{n\to\infty} \frac{1}{n} H(\mathbf{X}_1,\, \mathbf{X}_2,\, \mathbf{X}_3,\, \ldots,\, \mathbf{X}_n) \\ &= \lim_{n\to\infty} \frac{\sum_{i=1}^{n} H(\mathbf{X}_i|\mathbf{X}_{i-1},\, \ldots, \mathbf{X}_1)}{n} \\ &= \lim_{n\to\infty} \frac{n\, H(\mathbf{X})}{n} = H(\mathbf{X}) \end{aligned} \tag{C.1}$$

Furthermore, for a uniform distribution on an N-ary alphabet:

$$H(\mathbf{X}) = -\sum_{i=1}^{N} \frac{1}{N} \log_2(\frac{1}{N}) = \log_2(N) \tag{C.2}$$

Therefore, generating uniform samples on an N-ary alphabet, we expect to measure values close to $\log_2(N)$. Because of the asymptotic convergence, we also expect the estimate to get better as the size of the sample increases. I generated samples from uniform distributions on 2, 4, 8, 16, 32, 64, 128 and 256 characters, repeating the generation five times, with increasing sample lengths,

from 10^5 up to 9×10^5 characters. For each sample length I performed a linear regression, I report the results in Table C.1.

Sample length (chars)	Slope	Intercept(bits/char)	r-value
1×10^5	1.270	-0.2267	0.9978
3×10^5	1.219	-0.1207	0.9993
5×10^5	1.201	-0.08788	0.9996
7×10^5	1.194	-0.07861	0.9996
9×10^5	1.192	-0.07845	0.9996

Table C.1: The linear regression values for each number of samples used.

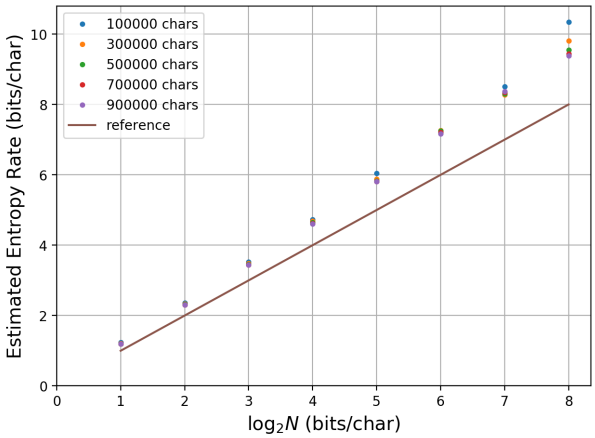

Figure C.1: The experimental entropy rates versus the logarithm of the alphabet size.

In Figure C.1, I plotted the experimental entropy rates versus the expected entropy value. A convergence towards the expected values is visible, but it appears very slow, thus requesting extremely long sequences to make the estimate better. The algorithm, therefore, doesn't seem to yield good results on *absolute* entropy estimation. Nevertheless, as the dependence seems to be linear, the algorithm remains reliable for evaluating *relative* entropy. Therefore, given its reasonably low computational cost ($N \log N$), it has been deemed good for this study.

Ringraziamenti

Ringrazio i miei genitori per avermi supportato in questo percorso di studi, e per avermi incoraggiato ad affrontarlo, in particolare al suo inizio. Ringrazio i miei amici di casa, per essermi rimasti vicini, per quanto possibile, nonostante le mie assenze sempre più lunghe. Ringrazio i miei amici di Bologna: questa città, senza di loro, non mi risulterebbe una casa come mi risulta ora. Ringrazio il mio relatore, e tutto il gruppo di Fisica dei Sistemi Complessi per il supporto nella stesura di questa tesi.

Bibliography

[1] Project repository on github. `https://www.github.com/GColom/Bolo_BM/`.

[2] Srm bologna. `http://www.srmbologna.it/`.

[3] Vladimir Igorevich Arnold and André Avez. *Ergodic Problems of Classical Mechanics (The Mathematical physics monograph series)*. Benjamin, 1968.

[4] Erdinc Avaroğlu. Pseudorandom number generator based on arnold cat map and statistical analysis. *Turkish Journal of Electrical Engineering and Computer Sciences*, 25:633–643, 2017.

[5] Satish Chandra and Anish Kumar Bharti. Speed distribution curves for pedestrians during walking and crossing. *Procedia - Social and Behavioral Sciences*, 104:660 – 667, 2013. 2nd Conference of Transportation Research Group of India (2nd CTRG).

[6] Pierre Collet. Dynamical systems and stochastic processes. In *Métodos Estocásticos en Sistemas Dinámicos*.

[7] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. A Wiley-Interscience publication. Wiley, 2006.

[8] William Feller. *An introduction to probability theory and its applications*. Wiley series in probability and mathematical statistics: Probability and mathematical statistics. Wiley, 1971.

[9] Riccardo Gallotti, Armando Bazzani, Mirko Degli Esposti, and Sandro Rambaldi. Entropic measures of individual mobility patterns. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(10):P10022, 2013.

[10] Ulrich Speidel, Mark Titchener, and Jia Yang. How well do practical information measures estimate the shannon entropy? In *Fifth International Symposium on Communication Systems, Networks, and Digital Signal Processing (CSNDSP2006)*, 2006.