Randomly Searching the Law: Mean First-Passage Times and Complexity of Legal Trees

Förster, Yanik-Pascal

Awarding institution:
King's College London

Randomly Searching the Law: Mean First-Passage Times and Complexity of Legal Trees

Yanik-Pascal Förster

Supervisor: Dr Pierpaolo Vivo
Dr Alessia Annibale

The Department of Mathematics
King's College London

This dissertation is submitted for the degree of
Doctor of Philosophy

December 2022

’Tis time
I should inform thee farther. Lend thy hand
And pluck my magic garment from me.
- Prospero

Acknowledgements

I acknowledge support by the EPSRC Centre for Doctoral Training in Cross-disciplinary Approaches to Non-Equilibrium Systems (CANES EP/L015854/1), and the use of the research computing facility at King's College London, *Rosalind* (https://rosalind.kcl.ac.uk).

Special thanks goes to (in alphabetical order by last name) Mark Crumpton, Dr Diede Fennema and Jan-Paul Lerch for proofreading all or parts of this thesis, and to DF twice over for suggesting the verb "to crimp" instead of "to necklacify" and for taking my lame jokes about the Dutch with humour. To Gioia Boschi, Pierpaolo Modugno, Vito A.R. Susca and Claudio Zeni for enduring my inflexibility in our small flat in Waterloo. And finally to Luca Gamberi for excellent companionship during the ups and downs of the PhD.

Abstract

This thesis proposes computational modelling to further the quantitative understanding of legal problems. We design a model for the behaviour of a law user retrieving information hidden in legal texts. The latter are typically organised in a hierarchical structure, which the reader needs to explore down to the 'deepest' level (Articles, Clauses, *etc*), until they have identified an answer to their question. Following previous works on transport properties of networks, the mean first-passage time (MFPT) taken by a random reader to retrieve information planted in the leaves is nominated as a measure of structural complexity of legal trees. The reader is assumed to initially skim the contents of a text, identifying keywords based on their interests, and be drawn towards the sought information based on keyword affinity. That is, they estimate how well the Chapter and Section headers of the hierarchy seem to match the informational content they expect to see in their target-answer, and follow the more promising ones with higher probability. Using randomly generated keyword patterns, we investigate the effect of two main features of the text – the horizontal and vertical coherence – on the searching time, and derive a few plausible high-level consequences. We obtain numerical and analytical results, the latter by approximating the biases imposed on the reader by the keyword patterns with the average bias obtained by averaging over the distribution of keyword patterns. This method leads to an explicit expression for the complexity of legal trees as a function of the structural parameters of the model.

We present in a simple workflow how one can prepare a real Act of Parliament for calculating its complexity and how the model parameters, for the text ensemble, can be estimated to obtain the complexity of the ensemble, as well. This is demonstrated explicitly on the example of the Housing Act 2004. We briefly discuss potential steps to test the validity of the various assumptions and results of our model.

Our analytical results are powered by a novel method to calculate MFPTs for random walks on graphs. This is a general result based on a dimensionality reduction technique for Markov state models, known as local-equilibrium (LE). We can prove that for a broad class of graphs, LE coarse-graining preserves the MFPTs between certain nodes if the coarse-grained states (or clusters) are suitably chosen. The amenable class of graphs includes trees.

This relation is exact for graphs that can be coarse-grained into a one-dimensional lattice, where each cluster connects to the lattice only through a single node of the original graph. The proof produces a generalisation of a well-known lemma about MFPTs along bridges, or essential edges. This essential edge lemma (EEL) is valid for reversible random walks, whereas our generalisation also applies to irreversible walkers. It leads to explicit formulae for the MFPTs between certain nodes in the necklace-class of graphs. We first demonstrate our method for the simple random walk on the c-ary tree, then we consider other graph structures and more general random walks, including irreversible random walks. This result is used to facilitate the analytical calculations for the reader model mentioned above.

For graphs that do not fall within the necklace class, we show that the generalised EEL provides useful approximations if the graph allows a one-dimensional coarse-grained representation and the clusters are either weakly interconnected or have a single dominating edge qualifying as a nearly essential edge. The error made in this approximation can be efficiently estimated by means of a perturbative expansion, given that the random walker is reversible. We provide references to other works on perturbations of Markov chains that may be useful in bounding the approximation error.

Table of contents

List of figures

List of tables

List of symbols and abbreviations

Roman Symbols

a	Keyword activation rate of root pattern, section 4.2
a_h	Specific keyword activation rate of pattern, eq. (4.9)
a_l	Generic keyword activation rate of pattern, eq. (4.9)
C	Complexity of a legal tree, eq. (4.21)
c	Offspring number of a c-ary tree (chapters 4 and 5)
C_{MF}	Approximate complexity of a legal tree, eq. (4.22)
e_{ij}	Weight of the edge ij
h	Height a c-ary tree (chapters 4 and 5)
k_i	Degree of node i
$\widetilde{k}_i$	Strength of node i, section 2.2
L	Pattern length, section 4.2
M_{IJ}	Coarse-grained mean first-passage time of node J starting from I
m_{ij}	Mean first-passage time of node j starting from i, eq. (2.6)
n	Number of nodes of a graph or of states in a state space
$\boldsymbol{Q}$	Coarse grained Markov chain transition matrix
$\mathbf{q}$	Markov chain transition matrix, eq. (2.2)
t	Target node (chapters 4 and 5)
t_{ij}	First-passage time of node j starting from i, eq. (2.5)

List of symbols and abbreviations

$v_0, \ldots, v_H$	Nodes on the backbone of a necklace, section 2.3

Greek Symbols

Δ	Pattern overlap, eq. (4.9)
Γ	Pattern mutation rate, section 4.2
Π^T	Steady state of coarse-grained Markov chain
$\boldsymbol{\pi}^T$	Steady state of Markov chain, eq. (2.3)
τ	Tightness, eq. (4.20)
$\boldsymbol{\xi}$	Pattern, section 4.2

Superscripts

$\mathbf{q}^0$	Necklacification of **q**, section 3.1
$\widehat{\mathbf{A}}_i$	Matrix **A** with row and column *i* removed
$\mathbf{A}^{-1}$	Inverse of the matrix **A**
$\mathbf{A}^T$	Transpose of the matrix or vector **A**

Other Symbols

$\mathbf{1}_n$	Vector of length *n* with all entries equal to 1
$\mathbb{1}_n$	Unit matrix of size *n*
∂i	Neighbourhood of node *i*
$\mathbb{E}$	Expected value
$\langle \cdot \rangle$	Expected value taken over pattern realisations, section 4.2
$\mathbb{E}(\cdot \vert Y)$	Conditional expected value given *Y*
$\mathbb{P}\{\cdot\}$	Probability measure
$\mathbb{P}\{\cdot \vert Y\}$	Conditional probability given *Y*

Acronyms / Abbreviations

EEL	Essential edge lemma, eq. (2.33)

FPT	First-passage time
GEEL	Generalised essential edge lemma, eq. (2.18)
LE	Local equilibrium, eq. (2.9)
MFPT	Mean first-passage time
PMF	Probability mass function

Chapter 1

Introduction

Recent years have seen a surging interest in quantitative studies of legal systems, one of the most elusive and intriguing properties being their complexity. As it stands, conventional wisdom tells us that dealing with legal systems is a complex task. However, the word "complexity" can have different meanings depending on the user of the word and what kind of object they are describing. The present thesis puts forward an innovative approach to define and address the complexity of a small component of legal systems - codified law.

In this chapter, we present a non-exhaustive overview of the literature related to our work. We begin by reviewing some important achievements of social physics, before moving over to the literature on legal network science and legal complexity. We finish the introduction by contextualising our work within these previously mentioned topics.

1.1 Social physics

Our work follows the movement of methods from statistical physics diffusing into a large variety of other sciences. Many of these advances have been remarkably successful, begetting a field of research (often referred to by the very broad term *socio-physics* or *social physics*) in its own right. It is fair to comment that the use of computational models in some fields (such as linguistics) is not limited to "physicists invad[ing] en masse" [1]. There is, therefore, a subjective component to whether a model is deemed "mathematical", "physical" or computational but from "within" the relevant community. For this reason, we largely follow the presentation of some authoritative reviews, in particular [2, 3] on a selection of topics. Withal, there are tell-tale signs of statistical physics described in the following paragraph, which we use as a rule of thumb to assemble a few additional works.

The effort of a social physicist on a given problem may be broadly divided into two steps: firstly, to define a model with meaningful microscopical rules, and secondly, to tease out macroscopic observables therefrom. In doing so, the researcher needs to find the balance between the refinement of the microscopic and its amenability to be studied numerically or analytically. Frequently, the main interest lies in systems of very large size (the thermodynamic limit), in which case the researcher can resort to statistical tools instead of trying to solve the dynamics exactly for finite size [2]. Additionally, phase transitions play an important role in social physics since the phases can often have a direct interpretation in the phenomenology of the real system. For instance in [4] the two phases in a system of Ising spins are interpreted as the situation in which either all staff work or all staff are on strike.

A classical example is the Voter Model (reviewed in [2] and references therein). This model describes an election with two candidates, in which every voter expresses a preference for one or the other while interacting with their neighbours' opinions. The model was initially studied on a regular lattice, which implies that every node has the same number of neighbours. In the simplest case, first introduced in [5], the system evolves by one time step by choosing a random voter and changing their preference to the preference of a randomly selected neighbour. The simplicity of this construction comes at the loss of fine-grain details about individual voters, but makes the model exactly solvable. As such, one can determine whether - in the thermodynamic limit of an infinite number of voters - the voters will reach consensus in finite time or not. Thus, a model with simple microscopic rules leads to macroscopic observables with obvious interpretations.

The common response to such results is to make the model gradually more interesting, general, or realistic while retaining enough regularity to be studied. In the case of the Voter Model, one can change the update rule for each voter (many examples are mentioned in [2]), or place the voters on any arbitrary network, as presented in great generality in [6]. Other generalisations include time-evolving networks and multistate opinions, e.g. when there are more than two candidates in an election [2], or the addition of a layer of "private" opinions [7].

Physical and mathematical models have found applications in *language dynamics*, too - that being the study of: evolution of, death of, and interactions between languages. Here – before the influx of physicists – simulations have aided challenges to conventional wisdom on reported occasions [1]. An important example contributed by mathematicians is the seminal paper [8], which uses a minimal model to describe the evolution of two competing languages based on each language's social status and fraction of speakers. Microscopic models (on the level of human interaction) for language evolution exist, in the form of so-called language games. A popular specimen of such models is the Naming Game [9], exhibited in its minimal form in [10]. Fundamentally, the agents of the Naming Game engage in pairwise conversations with a positive payoff if the exchange is successful [2].

In the minimal game, which is amenable to analysis and simulations, one can observe a drastic drop in the total number words in circulation after a transient period. After the decline, *all* conversations are successful (i.e. all agents agree on all object-word pairs) [10].

The evolution of language on the scale of populations is contrasted by its dynamics on the cognitive level of individuals. While the study of human cognition is not limited to language, it has proved a fertile environment for network science and statistical physics. The earliest source credited with connecting cognitive processes to a semantic network of words is [11]. Subsequently, this idea has led to a multitude of network-based studies. Examples of works that use the physical terminology of phase transitions are given by [12] and references therein. The authors argue that children's command over grammar has a sudden jump around the age of two, which is mirrored in a phase transition of their syntactic word-network from tree-like to scale-free. More generally, [13] identifies the sudden emergence (around the age of seven) of a large word-cluster that is connected on several layers of word relations (e.g. similar meaning and similarities in spelling). For a broad survey of cognitive (both semantic and lexical) network science we refer to [14, 15].

The idea of *human dynamics* is to model the spatial behaviour of humans as agents based on simple movement rules. Similar efforts have been made to understand movement patterns of foraging animals. Studies may focus on individual trajectories or on crowd behaviour, also including vehicular traffic [3]. The analysis of individual movement patterns especially has been marred by an absence of data, with only gradual improvements aided by technology. A prominent example is the article [16], which used the displacement of registered dollar bills over time as a proxy for human mobility. Reference [17] managed to zoom in onto individual people with the help of anonymised but individually tracked mobile phone locations. Their findings evidenced that individual trajectories are statistically highly regular and confined, but the authors argued that the convolution of trajectories with different scales may give rise to the scale-free statistics observed in [16]. We refer to [18] for a survey of the cornerstones in human dynamics. Later, in chapter 4, we find ourselves studying what are effectively individual human trajectories. However, our agent does not move in physical but in a semantic space.

Many of the early developments in social physics suffer from the aforementioned lack of data to utilise for comparison to theoretical predictions [2]. *Econophysics*, the application of statistical physics to economic problems, is an exception to this rule. This is because its beginnings were in the study of economic times series, for which there exists an ample supply of empirical data [3]. The landscape of research in this area is vast and varied. We refer to [19] for an extensive review on the recent developments and achievements of econophysics and limit ourselves to giving one example: An important point of interest is the collapse from an operational to a failed system; the constituents may be banks interconnected by their liabilities, or producers and service providers interconnected by products and infrastructure [3]. The core idea is that the failure of one node may disrupt the supply chain and cause a cascade of failures. In the instance of banking networks, a defaulting bank may cause a certain asset to be devalued substantially, which may in turn push other banks holding that asset into default. A comprehensive survey specifically on systemic risk in financial systems is given in [20].

Computational models for the spread of diseases have gained some prominence in recent years due to the well-known developments of the COVID-19 pandemic, when epidemiologists used computer simulations of disease-spreading models as part of their information package for the UK government [21]. So called *compartmental models* are widespread tools in epidemiological research. These models divide the population into groups (compartments) characterising their state with regards to the disease: One of the simplest models frequently considered is the *SIR* model, attributed to [22] but independently developed by Reed and Frost [23]. Therein an individual can be either “susceptible”, “infected” or “recovered” (and hence, immune). Placing the dynamics of human-to-human infections on a complex network is a natural way to allow for heterogeneity in the way agents interact [24]. Other generalisations, particularly important for policy decisions, include the refinement of compartments, e.g. by gender, age or occupation, or accounting for mobility between sub-populations [21]. At the extreme end of the fine-graining spectrum lie agent-based approaches, made feasible by increasingly powerful computers and vast amounts of data available through online sources [3]. As a matter of fact, COVID outbreak statistics in the UK have been partially produced by a large-scale agent-based model [21]. Much in the spirit of statistical physics, the analysis of these models in the context of social physics often focuses on stability and phase transitions. The phases in the case of epidemiological modelling differ in whether a small number of infections will or will not lead to a major outbreak [3].

We conclude the overview of social physics with a brief note on statistical network models. A number of these have proven highly relevant, partially because they exhibit key statistics found in social networks, and partially because one can often extract the asymptotic behaviour of macroscopic observables from their generative models. A comprehensive overview can be found in the survey [25].

A foundational network model is the Erdős-Rényi(-Gilbert) model [26], which is a network on a fixed set of N nodes, in which every possible link appears independently from the others with probability p. The independence of the edges often lends itself to analytical methods, at the expense of realism: every node degree in the network is a sample from a binomial random variable with the same parameters $N-1$ and p. As such, all nodes tend to have very similar degrees, whereas broad or even fat-tailed degree distributions are often considered key-characteristics that a model should exhibit [25, 27].

The Barabasi-Albert (preferential attachment) model [28] is one model giving rise to a scale-free degree distribution via an intuitive generating mechanism: at each time step, a new node is introduced with a set number of initial edges. The epithet "preferential attachment" refers to the fact that new nodes connect preferentially to old nodes with high degrees. It can then be shown that the degree distribution of this model approaches a power-law [25]. From the generating mechanism of this model one can sometimes extract asymptotic properties for the network ensemble in the thermodynamic limit: for instance [29] (with involvement of this author) studies the number of communities formed in a democratic society with two parties; the interaction network of the citizens is constructed following the Barabasi-Albert model.

Other models have been devised to produce network statistics other than degree distributions: famously, Watts and Strogatz [30] applied a rewiring algorithm to a ring-shaped lattice. The rewiring of a fraction of edges drastically reduces the diameter of the network while keeping a high clustering coefficient – a property now referred to as *small-world* [27].

We now progress from the overview of the broad field of social physics to the more specialised quantitative studies of legal systems, beginning with legal networks.

1.2 Legal network science

If we identify the relevant agents in a legal system and draw a dot (node or vertex) for each of them, and if we proceed to represent their interactions by lines (edges or links), we have represented the legal system by a network.

On a smaller scale, say for an Act of Parliament, the nodes represent pieces of the text, usually respecting some form of hierarchy. For instance, one may draw a node for every text Item on the level of Paragraphs and higher, and draw a line between two nodes if one of them contains the other, such as a given "section 1" may contain a "subsection 1.2". The network thus reflects the natural hierarchy of the text structure. Further edges may be employed to indicate, for instance, that one node contains text that modifies, refers to, or annuls the text in another node (these relationships are called *amendment*, *reference* and *repeal*, respectively). Bodies of text which have been modelled in this way include: codified law, court decisions (common law) and scientific citation networks. It is this kind of abstract system of documents that is predominantly studied in legal network science.

Several papers from the collaboration between Katz and Bommarito from the early 2010s, [31–33], have attained seminal status in legal network science. Reference [31] introduces a network-based distance measure for nodes in directed acyclic graphs; this measure is used to explore clustering phenomena on the citation network of US Supreme Court judgements. Similarly, [34] investigates the correlation between network statistics and the importance of precedents, as well as their time evolution, in the opinion network. In [32], the network-view of codified law described in the beginning of this section is employed for the first time. Here, the authors study the degree distribution of the US Code. In a later paper, [33], the authors significantly expand upon their work by conducting an extensive statistical study of its network "skeleton" (such as size and depth of its constituents, or interdependencies introduced by cross-references) and language (e.g. word entropy, size of vocabulary, or length of words). An algorithm is proposed to compose a complexity measure from their statistics that allows them to rank the Titles (highest level of organisation in the US Code) by their complexity.

Recent studies focus on the time-evolution of legal texts in the US and Germany, using a clustering algorithm [35]; similarly, [36] correlates changes in the Korean constitutional law to societal changes. Other references within [35] treat the time-evolution of national and super-national legal corpora from a network-perspective.

Several authors have noted important analogies between software and legal systems, for instance [37], analyses the US Code based on software engineering terms. More recently, [38] has expanded on the analogy, with a focus on symptoms and markers (called *smells*) of software that are likely to become problematic in the future, e.g. through long reference trees or duplicate phrases.

Further studies base their analysis on topic modelling, a family of machine learning algorithms that extract "topics" from a given text (one of the most important ones, *latent Dirichlet allocation*, introduced in [39]). For instance, [40] presents cases studies of the network of opinions expressed by the Supreme Court of the Unites States, tracking proxies for the change in topic proportions over time. The conceptual bridge to law search is built in [41]: the authors study again the corpus of US Supreme Court opinions, where two opinions can be connected either by a citation (directed link) or by textual similarity (undirected or symmetric link). Textual similarity is evaluated with the help of a topic model. The resulting graph defines a random walker whose stationary properties serve to define a distance measure between nodes in the network. Additionally, a geometric interpretation is used to assess the transport properties of the network, which give a measure of how quickly a reader following the links leaves the environment of the starting node. [42] proceeds from the work presented in [41] by testing law search models directly - the models are tested by predicting whether, of two given texts, one should cite the other, or whether a human would classify the pair as “related”. These works are among the first to study legal networks through the lens of “user experience”.

Besides explicitly network- or data-driven studies of the written law, legal scholars have been interested in the network of agents in a legal system. The next section indicates a number of notable works in this direction.

1.3 Legal complexity

As said in the introduction to the previous section, one can take a general stance by casting any legal systems into a network. Notable publications are pointed out in this section. These discuss complexity as it relates to the connection between the world, the policymaker, the legal practitioner, and the citizen that is subject to legal rules. Much of the work in this area is descriptive, such as the influential article [43]. The author considers how complexity may arise in institutional and public challenges, to what cost, and to what and whose benefit. They compile a number of principles by which to assess and potentially reduce complexity. Reference [44] explores the idea of beneficiaries of complexity further, coming to the conclusion that a lawyer’s interest may lie in an intermediate level of complexity: too high complexity deters individuals from taking cases to the courts, while too low complexity does not warrant hiring a lawyer.

Reference [45] pushes towards a more quantitative, information theoretical understanding of legal uncertainty and discusses the inequalities and biases caused by it. Expanding on the notion of uncertainty, [46] proposes to estimate the Shannon entropies of various aspects of legal systems and assign a total entropy based on these estimates. We also refer to [46] for an overview of the historical development of legal entropy.

By correlating network properties of legal systems and their potential outputs and effects (e.g. laws leading to the creation of commentary and by-laws), [47] understands complexity as a problem of knowledge management. One of the first to apply the language of complex adaptive systems to legal systems is [48]. The author proffers an extensive list of correspondences between the two. An earlier discussion and extensive review of the description of legal systems as complex adaptive systems is given in [49].

1.4 A parametric model for law search

The matter of this thesis touches on some of the above topics from an angle that has until now been largely absent from the field. Instead of data-driven studies, we follow the school of social physics by constructing a “physical” model based on intuitive and simple microscopic rules to study their effect on a macroscopic observable. The principal model presented in this thesis is, in part, a model for a reader navigating a heterogeneous semantic landscape, and in the other part a model of the semantic landscape itself. To our knowledge, such a combination of models has not been studied before, neither in the context of human dynamics nor in fields focusing on text models or text comprehension. Moreover, in contrast to most works in social physics, we can extract approximate results from networks of finite size.

In the review [50], the Office of the Parliamentary Council acknowledged the complexity of legal systems as a timely problem. The text outlines a number of specific causes for "excessive" complexity and concerns attached to them. With the present thesis, we extend this idea by proposing a complexity measure for Acts of Parliament that facilitates the comparison of different layouts and drafts. Our complexity measure depends on the structural layout (the topology) of the Act and can either incorporate a single instance of text (numerically) or some macroscopic parameters thereof (analytically).

For over a decade the field of legal network science has been maturing under the influence of researchers with a variety of backgrounds. We hope that a line of research looking to state quantifiable hypotheses will be widely appreciated. The remaining chapters describe the attempts of our own to do so.

In chapter 2, we propose a novel method to calculate mean first-passage times (MFPTs) for random walks on graphs. This method is based on a dimensionality reduction technique for Markov state models known as *local equilibrium* (LE). We show that for a broad class of graphs, LE coarse-graining preserves the MFPTs between certain nodes, upon making a suitable choice of the coarse-grained states (or clusters). Trees are included in this class of graphs. We prove that this relation is exact for graphs that can be coarse-grained into a one-dimensional lattice where each cluster connects to the lattice only through a single node of the original graph. The proof also produces a generalisation of the well-known essential edge lemma (EEL), which is valid for reversible random walks. The generalised EEL (GEEL) also applies to irreversible walkers. Such a generalised EEL leads to explicit formulae for the MFPTs between certain nodes in this class of graphs. We first demonstrate our method for the simple random walk on the c-ary tree, then we consider other graph structures and more general random walks including irreversible random walks. We refer the reader to section 2.2 for a brief review of basic facts and definitions concerning Markov chains on finite state spaces and in discrete time.

In chapter 3, we show evidence that if a graph allows a one-dimensional coarse-grained representation with clusters that are sparsely interconnected, the GEEL provides useful approximations even if the graph is not a necklace. This chapter comprises numerical studies alongside a method to estimate the error made in this approximation.

Chapter 4 introduces a model for the retrieval of information hidden in Acts of Parliament. These are typically organised in a hierarchical (tree) structure, which a reader interested in a given provision needs to explore down to the “deepest” level (Articles[1], Clauses, or lower). We assess the structural complexity of legal trees by computing the MFPT a random reader takes to retrieve information planted in the leaves. The reader is assumed to skim through the content of a legal text based on their interests and relating keywords, and be drawn towards the sought information based on keywords affinity. That is, they judge how well the Chapter and Section headers of the hierarchy seem to match the informational content of the leaves. Using randomly generated keyword patterns, we investigate the effect of two main features of the text – the horizontal and vertical coherence – on the searching time. We obtain numerical and analytical results, the latter based on a mean-field approximation on the level of patterns. This leads to an explicit expression for the complexity of legal trees as a function of the structural parameters of the model. We discuss the implications of our results on the policy drafting process, as well as their limitations and potential for further development.

In chapter 5, we go through the practical task of taking an Act of Parliament, reducing it to its most informative keywords, and calculating its complexity as defined in the previous chapter 4. The procedure is modular, such that various preprocessing and estimation steps can be easily switched for others. Building blocks and a Jupyter notebook in Python are provided for the interested reader.

Chapter 6 summarises our findings and highlights the connections between the technical and applied chapters. We also point out some of the open problems left or created by our work. Finally, we consider ways to test the results of chapter 4 experimentally using real legal texts.

[1] Hierarchical Items of a legal text will be capitalised to distinguish them from the items of the present thesis.

Chapter 2

Mean first-passage time formulae from dimensionality reduction

2.1 Introduction

In this first technical chapter, we develop a method that is useful for calculating the mean first-passage times of random walkers on certain networks. It bears great importance for the analytical calculations in chapter 4, but is interesting in its own right. We begin with a brief review of the literature around search processes and Markov models before going into technicalities of our results.

Random walks on networks are intuitive and highly general stochastic processes that enjoy attention in many different applications. Examples include models for foraging and predator-prey behaviour [51, 52] among numerous other biological applications [53], centrality measures such as the famous PageRank [54, 55], and search strategies [56] as in hide-and-seek games [57]. As a consequence, random walks on networks, especially the *simple* random walk, where the hopping probabilities from any network node to all adjacent nodes are uniform, are well studied.

First-passage times (FPTs), i.e. times at which certain events occur for the first time, are important observables of many stochastic processes, and in particular Markov processes, with random walks being no exception. Consider for instance the Gambler's ruin (the first time a Gambler's budget hits zero) or break-even points in trading (when the selling price of a stock exceeds the price paid for the first time) [58], and extinction events in birth-death-processes [59] (see [60–62] for further examples). The *mean* first-passage times (MFPTs) between states of a Markov process encapsulate fundamental properties of the system's kinetics. They are in fact closely related to the spectrum and eigenvectors of the transition matrix [60, 63, 64] and the relaxation times of the random walker, all of which are all of particular computational importance [65].

MFPTs have also been shown to provide important information about correlations and heterogeneity in complex systems [66], and optimal coarse-graining in Markov State Models [64].

MFPTs of random walks on networks encode *global* properties of the random walkers and the network they explore, hence their explicit and exact calculation can in general be hard for networks larger than a few nodes. There are many ways to express the full matrix of MFPTs theoretically; one of the classical and most general methods is to employ the so-called fundamental matrix, as proposed in [67]. The fundamental matrix is also connected to equilibrium properties and commute times of the walker and has been revisited and reformulated over time, for instance in [68]. Reversible random walkers, where hopping probabilities are in *detailed balance* with the equilibrium node-occupancy probabilities, are often more accessible. Here, some popular approaches include the network analogue of resistance theory (see [69] for application to trees and [60] for a general introduction) and the essential edge lemma, which applies when the graph consists of subgraphs that are connected by a single edge (see e.g. [62]).

These exact methods rarely lead to explicit results even in simple cases, when one would hope to express MFPTs, e.g. in terms of the graph parameters of a model. Only for specific problems, e.g. in the presence of a high degree of symmetry and hierarchy, can the problem be solved explicitly by successive “decimation” procedures [70–72]. This has led to the development of various approximation schemes – for instance, mean-field approaches based on node degree [73], or on the distance from a target [74]. For sufficiently dense networks with random weights, the information contained in the neighbourhood of the target node is sufficient to formulate an accurate rank-1 approximation for the MFPT from any other node [75]. For sparser networks, the approach presented in [76] exploits locally tree-like structures to derive asymptotic expressions for a large number of nodes. Moreover, tail-estimates for first-passages of rare events can be constructed for many different applications [61]. The approximations made in these works tend to be valid either in the limit of large graphs, $n \to \infty$, or are restrictive about the type of random walker to which they can be applied. For example, many are developed for *simple* (purely diffusive) random walks. For more general dynamics, explicit results are scarce.

In this chapter, we show that kinetic coarse-graining techniques, introduced to reduce the dimensionality of Markov State Models, can be used to derive *explicit* formulae for MFPTs in terms of the graph parameters. These are exact for a broad class of graphs, which includes tree-graphs.

The method is based on three key ideas: (i) upon kinetically coarse-graining random walks on graphs, the calculation of MFPTs simplifies due to the reduced dimensionality of the coarse-grained system, (ii) for certain graph structures, it is possible to adopt coarse-grained representations that drastically simplify the calculations, leading to explicit formulae for the MFPTs in the coarse-grained space, (iii) for the graph structures referred to in (ii), under some conditions, the MFPTs of the coarse-grained system match exactly certain MFPTs in the original system.

In particular, we prove that in graphs with special “necklace” topologies, certain MFPTs in the original dynamics can be calculated exactly using a coarse-graining technique known as *local-equilibrium* (LE). Our proof is valid for general random walks, including irreversible random walks, as long as they have a unique steady-state distribution. In addition to this proof, which leads to eq. (2.26), our analysis provides two main results. The first one, eq. (2.18), is a generalisation of the popular essential edge[1] lemma (EEL) [62], which is only valid for reversible random walks. It is retrieved here as a special case of a more general equation, which does not require dynamical reversibility. This result leads to explicit formulae for MFPTs in simple random walks on graphs with necklace topologies. The second main result, eq. (2.22), provides a way of obtaining approximate MFPTs when the graph topology is not an exact necklace. We explore the potential of this method in chapter 3.

In section 2.2, we review the notions of MFPTs and LE coarse-graining. In section 2.3, we show that LE coarse-graining preserves certain MFPTs for general random walks on a broad class of graphs with “necklace” structure. We also provide a generalisation of the essential edge lemma, valid for reversible random walks, to irreversible random walks. In section 2.4, we demonstrate how the LE coarse-graining method can be used to derive explicit MFPT formulae for the simple random walk on c-ary trees. These expressions are consistent with those resulting from the essential edge lemma. In section 2.5, we apply the generalised EEL, derived in section 2.3, to simple random walks on non-tree graphs with necklace structures. In section 2.6, we apply the method to irreversible random walkers, where the popular EEL does not apply. We summarise results in section 2.7. Some of the technical definitions of our derivations are elaborated in appendix 2.A. In appendix 2.B, we briefly explore the possibility to extend our results to higher moments and full distributions of FPTs.

[1] In modern literature the term *bridge* seems to be preferred.

2.2 Definitions: Markov chains and MFPTs

We begin by reviewing a few fundamental notions about finite, stationary Markov chains in *discrete time*. A thorough introduction can be found in the popular albeit unfinished textbook [62], in references therein and most introductory texts on stochastic processes. Given a finite set $S = \{1, \ldots, n\}$ of "states", a Markov chain on those states is a sequence of random variables (X_t) taking values in S, and with the additional property that the probability distribution of X_t depends only on X_{t-1}, that is

$$\mathbb{P}\{X_t = i \,|\, X_0, \ldots, X_{t-1}\} = \mathbb{P}\{X_t = i \,|\, X_{t-1}\} \ , \tag{2.1}$$

for all $i \in S$. We will assume throughout this thesis that this quantity does not depend in t. In that case, the Markov chain is called *stationary*. Thus, conditional on the previous state X_{t-1}, we can encode the distribution of the next state in a constant matrix $\mathbf{q}$ with entries

$$q_{ij} \coloneqq \mathbb{P}\{X_t = j \,|\, X_{t-1} = i\} \ . \tag{2.2}$$

The entries of $\mathbf{q}$ are non-negative, and the elements in each row sum to one, as they contain the transition probabilities given a certain X_{t-1}. In the notation chosen above, let the row vector $\boldsymbol{x}^T$ be the vector of occupation probabilities $x_i = \mathbb{P}\{X_{t-1} = i\}$ at time $t-1$. Then, the occupation probabilities at the next time step t are given by the row vector $\boldsymbol{x}^T\mathbf{q}$. Inductively, the transition matrix for s time steps is given by the matrix power $\mathbf{q}^s$. Note that here and below, we think of vectors as columns, referring to row vectors as the transpose $(-)^T$ of a column.

A finite Markov chain is called *aperiodic* if for every i, the greatest common divisor of the set $\{t|(\mathbf{q}^t)_{ii} > 0\}$ is equal to one. It is called *irreducible* or *ergodic* if there is an N such that for any i and j the chain can reach j from i in less than N steps with a positive probability. For an irreducible Markov chain, by virtue of the Perron-Frobenius theorem [77], $\mathbf{q}$ has a unique left eigenvector $\boldsymbol{\pi}^T$ with non-negative entries and unit eigenvalue. As this implies

$$\boldsymbol{\pi}^T\mathbf{q} = \boldsymbol{\pi}^T \ , \tag{2.3}$$

this vector is called the *stationary distribution*, or *steady-state*, of the chain. Moreover, given the chain is aperiodic, its distribution will converge to $\boldsymbol{\pi}^T$ irrespective of the initial conditions. Finally, $\boldsymbol{\pi}^T$ is called the *equilibrium* distribution if the *detailed balance* condition

$$\pi_i q_{ij} = \pi_j q_{ji} \tag{2.4}$$

is satisfied for all pairs of two states i and j. An irreducible Markov chain satisfying detailed balance is also called *reversible*.

The *first-passage time* (FPT) t_{ij} of (X_t) from i to j is a non-negative integer random variable. It is defined as the number of steps required by the chain to reach j given that it was initially at i,

$$t_{ij} := \min_{t\geq 0}\{t|X_t = j, X_0 = 0\} , \tag{2.5}$$

if the minimum exists. Note that in this convention $t_{ii} = 0$. The FPT t_{ij} exists (i.e. is finite) if there is a sequence of states connecting i to j, and the probability that this sequence is followed is positive. As we focus on irreducible Markov chains, this is the case for every pair of states i and j. In that case, the expected value, given by the MFPT $m_{ij} := \mathbb{E}(t_{ij})$, exists as well. The set of all MFPTs m_{ij} from any state i to any state j is determined by the recurrence equations

$$m_{ij} = q_{ij} + \sum_{k\neq j} q_{ik}(m_{kj} + 1) . \tag{2.6}$$

The first term of eq. (2.6) accounts for the chain going from i to j directly (which occurs with probability q_{ij}), while the second term accounts for the chain visiting any other state k first and starting a first-passage process from there (at the next time step). Using the fact that the rows of $\mathbf{q}$ are normalised, eq. (2.6) can be rearranged into matrix notation for the vector $\boldsymbol{m}_j$ of MFPTs to j starting from all other states. $\boldsymbol{m}_j$ is then given by [78]

$$\boldsymbol{m}_j = \left(\mathbb{1}_{n-1} - \widehat{\mathbf{q}}_j\right)^{-1} \mathbf{1}_{n-1} , \tag{2.7}$$

where $\mathbb{1}_n$ and $\mathbf{1}_n$ are the identity matrix and the all-1 vector of size n, respectively, and $\widehat{\boldsymbol{q}}_j$ is the transition matrix of the walker from which the j-th row and column have been removed. For reversible Markov chains, a number of exact methods to obtain MFPTs do exist, see e.g. [60, 62]. However, in this work, we will not assume that eq. (2.4) is satisfied. We will simply assume that the system has a unique steady-state.

In this thesis, we largely use the terminology of random walkers on finite (possibly directed) graphs that are equivalent to finite Markov chains. We consider random walkers on n-node graphs, with vertices labelled as $i, j, \ldots$ and transition probability matrix $\mathbf{q}$. If there is no edge from node i to node j, then $q_{ij} = 0$. Thus, the matrix $\mathbf{q}$ defines a directed graph in which every directed edge (or *arc*) (i, j) is weighted by the hopping probability q_{ij} of the walker. Note that we use the term “node” as synonymous with “vertex” and “edge” as synonymous with “link”. It is worth mentioning that a random walk on an undirected graph (where the stepping probabilities at i are proportional to the weights of the edges connecting to i) automatically defines a reversible chain, and vice-versa [62]. Moreover, it is well known and easy to check that the steady-state probabilities on undirected graphs are given by the *node strengths*

$$\pi_i = \frac{\tilde{k}_i}{Z} \tag{2.8}$$

with the normalising factor Z ensuring that $\sum_{i=1}^{n} \pi_i = 1$. The *strength* $\tilde{k}_i = \sum_j e_{ij}$ is the sum of weights of all edges connecting to i, and e_{ij} is the weight of the edge (ij) ($e_{ij} = 0$ if there is no edge connecting i to j).

Given a random walk $\mathbf{q}$ on n nodes, one can define a *coarse-grained* random walk $\boldsymbol{Q}$ on N nodes, where $N < n$, by grouping together the nodes $i, j, \ldots$ of the original network into N subgraphs, or *clusters*, labelled by upper case indices $I, J, \ldots$. This operation can be encoded into an $n \times N$ matrix $\mathbf{C}$, whose elements $C_{iI} \in \{1, 0\}$ denote whether (1) or not (0) node i belongs to subgraph I, for all $i = 1, \ldots, n$ and $I = 1, \ldots, N$. Given a coarse-graining protocol, we will use small letters – e.g. $\boldsymbol{q}$, m_{ij}, $\boldsymbol{\pi}$ – to refer to properties of the original random walker, and capital letters – $\boldsymbol{Q}$, M_{IJ}, Π – for corresponding properties of the coarse-grained walker.

Recently, there has been active research into how to optimally define the transition matrix $\boldsymbol{Q}$ of hopping probabilities between clusters, for a given choice of the clustering $\mathbf{C}$ [79]. For $\boldsymbol{Q}$ to retain the equilibrium properties of the original dynamics, its left eigenvector associated to the unit eigenvalue must satisfy $\Pi^T = \boldsymbol{\pi}^T \mathbf{C}$. I.e. the steady-state occupancy probability of a cluster must equate to the sum of the steady-state occupancy probabilities of the nodes in that cluster. This, however, does not determine $\boldsymbol{Q}$ uniquely and further conditions must be imposed. A popular prescription, known in the literature as the *local-equilibrium* (LE) clustering [79], requires that the probability flow from cluster I to cluster J be equal to the sum of the probability flows from any node i in cluster I to any node j in cluster J

$$Q_{IJ} = \frac{1}{\Pi_I} \sum_{ij} C_{iI} \pi_i q_{ij} C_{jJ} \ . \tag{2.9}$$

Due to the reduced dimensionality of $\boldsymbol{Q}$, when compared to $\mathbf{q}$, certain observables may be easier to calculate in the coarse-grained graph.

In the next section, we focus on a broad class of graphs where subgraphs can be arranged in a line, such that each subgraph connects to the line only through one vertex. We prove that the MFPTs between these vertices in the original dynamics are equal to the MFPTs between the corresponding clusters in the LE coarse-grained dynamics, for which we are able to derive explicit formulae. The result is general, in particular it does not require reversible dynamics. Hence, our method provides quick access to an explicit MFPT formula whenever information on the steady-state cluster occupancy probability Π is available.

2.3 Conservation of MFPTs under coarse-graining

In this section, we prove that coarse-graining according to LE preserves certain MFPTs of the random walker exactly, if the graph has a special "necklace" structure, i.e. if it can be regarded as a one-dimensional "chain of graphs". More precisely, we consider graphs consisting of $H+1$ disjoint, connected subgraphs $0,\ldots,H$ hanging from the line $v_0,\ldots,v_H$ (the *backbone*) of distinguished vertices $v_I \in I$, with $I = 0,\ldots,H$; see the top of fig. 2.1 for an illustration. The subgraphs can have any arbitrary structure, as long as their only interconnections are the links in the backbone. We are interested in MFPTs to the target node v_H, from another node in the backbone, initially set to v_0. Without loss of generality, we define the target subgraph as containing only the target node, $H = \{v_H\}$, as we are only concerned with MFPTs to v_H from outside H. We denote vertices *within* the subgraph I other than v_I by v_{Ii} with $i = 1,\ldots,|I|-1$, where $|I|$ is the number of vertices in I. To keep the notation uniform, we denote v_I as v_{I0} if necessary.

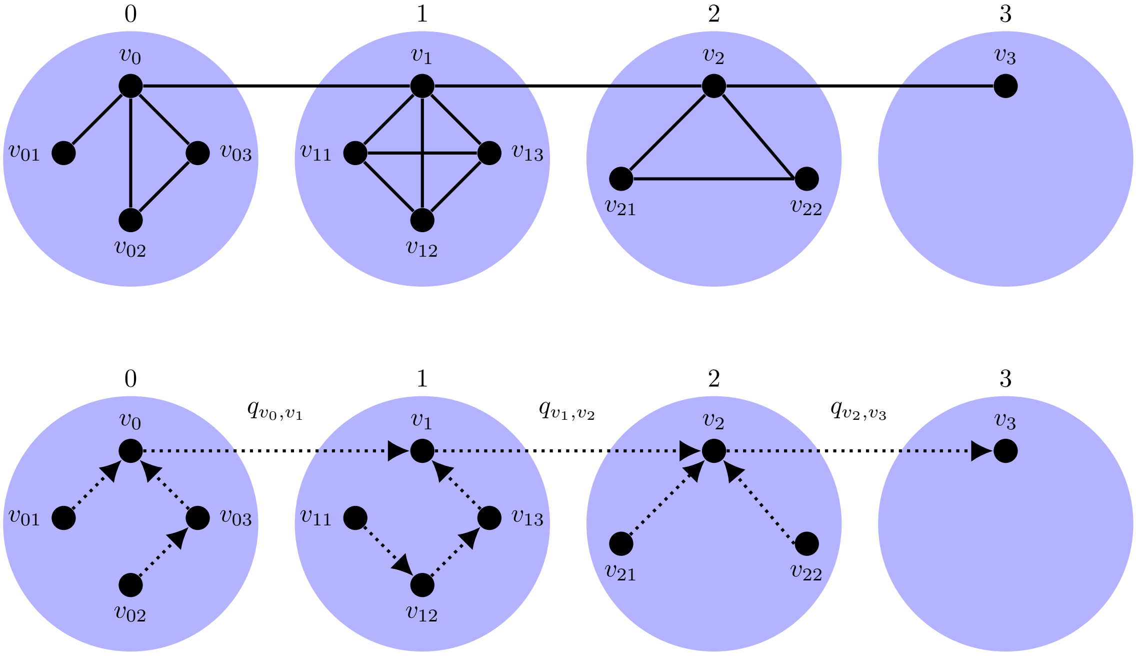


Fig. 2.1 Top: Example of a necklace with $H = 3$. Bottom: Dotted arrows form a spanning tree with root v_3. Since the subgraph interconnections consist of single edges, every such spanning tree must contain the edges (v_0,v_1), (v_1,v_2) and (v_2,v_3), with weight q_{v_0,v_1}, q_{v_1,v_2} and q_{v_2,v_3}, respectively.

In the following, we derive a general formula for the MFPT $m_{v_0v_H}$ from v_0 to v_H that outmanoeuvres the inversion formula in eq. (2.7), and we demonstrate that this matches exactly the MFPT M_{0H} of a walker in the coarse-grained graph, where each subgraph is regarded as a cluster, and the hopping probabilities between clusters are defined according to the LE prescription. The results can be generalised immediately to arbitrary pairs of nodes in the backbone.

To the best of our knowledge, our result is the first to lead to explicit and exact formulae for MFPTs in graphs with necklace structure. This is a broad class of graphs, which includes tree-graphs, as we will show in the next sections 2.4, 2.5 and 2.6. Similar graph structures were considered in [80], where expected escape-times from clusters connected to a one-dimensional lattice each by a single edge, were calculated. This approach, however, did not rely on coarse-graining techniques, which broaden the usability of our formula.

Our proof relies on a combinatorial approach to the calculation of MFPTs, which consists in finding all the spanning trees and two-tree forests in the graph hosting the random walk (see appendix 2.A for the definition of spanning trees and forests). Upon defining the weight of a tree $\mathfrak{t}$ as the product of all its edge weights

$$w(\mathfrak{t}) = \prod_{(ij)\in\mathfrak{t}} q_{ij} \ , \tag{2.10}$$

where the product runs over the edges of the tree, the MFPT from node i to j is found within this approach as [81, 82]

$$m_{ij} = \frac{s_{ij}}{s_j} \ . \tag{2.11}$$

Here, s_j is the sum of the weights of all spanning trees rooted in j, which we denote by $\mathfrak{t} \rightarrow j$ (as, by definition of root, all edges “point toward” the root),

$$s_j = \sum_{\mathfrak{t}\rightarrow j} w(\mathfrak{t}) \ . \tag{2.12}$$

Moreover, s_{ij} is the sum of the weights of all two-tree forests $(\mathfrak{t}, \mathfrak{s})$, such that $\mathfrak{t}$ has root j and $\mathfrak{s}$ contains i (but can have any root; cf. fig. 2.2 and further examples in appendix 2.A)

$$s_{ij} = \sum_{\substack{\mathfrak{t}\rightarrow j;\\ i\in\mathfrak{s}}} w(\mathfrak{t})w(\mathfrak{s}) \ . \tag{2.13}$$

Conveniently, one can express the stationary probabilities of an irreducible random walker in terms of the same quantities [62]

$$\pi_j = \frac{s_j}{\sum_{k=1}^{n} s_k} \ . \tag{2.14}$$

We apply eq. (2.11) to the sites v_0 and v_H. For graphs with necklace structure, as shown in fig. 2.1, any spanning tree with root v_H must contain all edges of the backbone pointing in the direction of v_H, as these are *essential edges*. That is, the graph becomes disconnected when any one of them is removed. This means that the path weight $q_{v_0 v_1} \cdot \cdots \cdot q_{v_{H-1} v_H}$ is a common factor in the sum in eq. (2.12). We will use the shorthand

$$w_{IJ} = \begin{cases} \prod_{K=I}^{J-1} q_{v_K v_{K+1}} & \text{if } I < J \\ \prod_{K=J}^{I-1} q_{v_{K+1} v_K} & \text{if } I > J \\ 1 & \text{else} \end{cases} \tag{2.15}$$

for products of the hopping probabilities $q_{v_I v_J}$, and W_{IJ} for the corresponding products of the hopping probabilities Q_{IJ} between clusters.

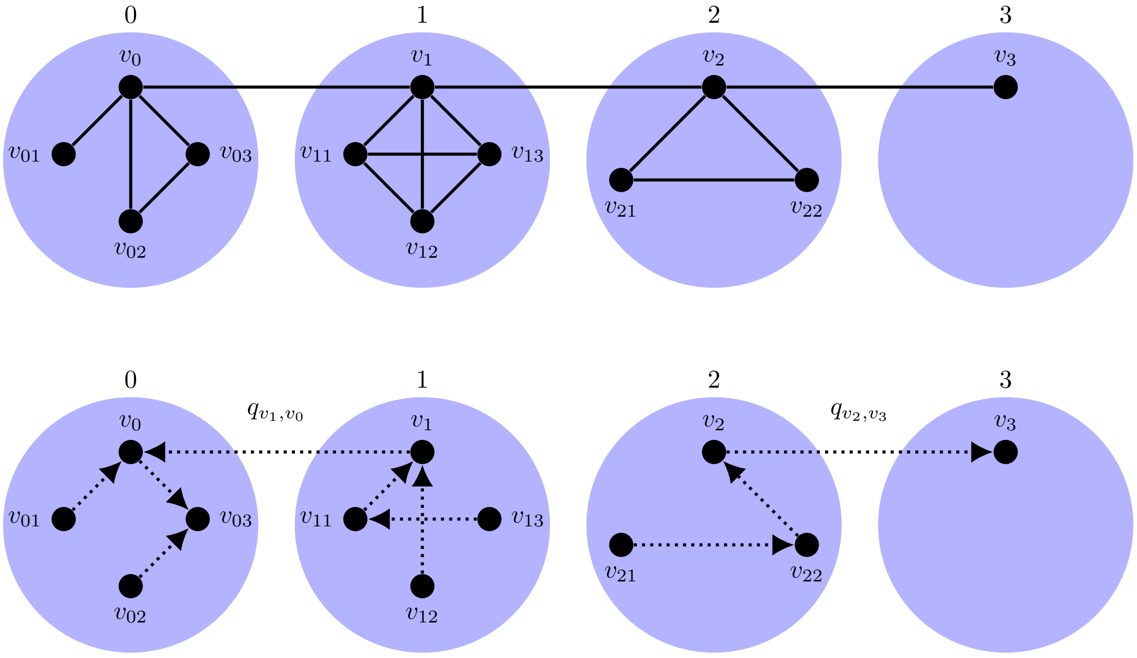


Fig. 2.2 Top: Example of a necklace with $H = 3$, as in fig. 2.1. Bottom: Dotted arrows show a two-tree forest with one component rooted in v_3, the other containing v_0. In this case, the v_0-component has root v_{03}. For a v_0-component with any other root in subgraph 0, all edges outside 0 stay the same, as any path coming in to 0 must go through v_0.

Furthermore, every subgraph I is connected to the backbone only via v_I, forcing the sub-spanning trees within each I to be rooted in v_I. Denoting the sum of weights of all sub-spanning trees of I by $w(I \to v_I)$, and noting that the backbone and the attached subgraphs account for all of the vertices in the graph, we can write s_{v_H} in the factorised form

$$s_{v_H} = w_{0H} \prod_{I=0}^{H} w(I \to v_I) \; . \tag{2.16}$$

Similarly, we can decompose the two-tree forest weight $s_{v_0 v_H}$ defined in eq. (2.13). Each relevant two-tree forest consists of a tree rooted in v_H, and a tree that contains v_0 but can be rooted in any of its vertices, including vertices that are not on the backbone. Firstly, we observe that whenever v_H and v_0 lie in different trees of a spanning forest, one of the (undirected) edges of the backbone, say (v_{I-1}, v_I) with $I > 1$, must have been omitted in the forest, e.g. in fig. 2.2, it is (v_1, v_2). All other edges of the backbone must necessarily be included, each in one direction. Consequently, the v_H-component contributes the weight $w_{IH} \prod_{J=I}^{H} w(J \to v_J)$ for the forest in which subgraphs I, $I+1, \ldots, H$ are included in the v_H-component. The other component contributes three factors:

1. for any fixed $K = 0, \ldots, I-1$ there is a weight $w_{0K} w_{I-1,K}$ for the backbone edges pointing towards subgraph K,
2. for any vertex $v_{Km} \in K$ with K fixed as above, we have a weight $w(K \to v_{Km})$ for the spanning trees of K pointing towards the vertex,
3. the remaining subgraphs $J = 0, \ldots,\ K-1,\ K+1,\ \ldots, I-1$ give rise to $w(J \to v_J)$.

For instance, the v_0-component in fig. 2.2 is rooted in the node v_{03} of 0 and has weight $w(0 \to v_{03}) w(1 \to v_1) q_{v_1 v_0}$; the other component has weight $w(2 \to v_2) w(3 \to v_3) q_{v_2 v_3}$. Summing the product of these weights over I, J, K and m, one obtains

$$\begin{aligned} s_{v_0 v_H} &= \sum_{I=1}^{H} \left[w_{IH} \prod_{J=I}^{H} w(J \to v_J) \right. \\ &\quad \left. \times \sum_{K=0}^{I-1} \left(\sum_{m=0}^{|K|-1} w(K \to v_{Km}) w_{0K}\, w_{I-1,K} \prod_{J=0, J\neq K}^{I-1} w(J \to v_J) \right) \right] \\ &= \sum_{I=1}^{H} \sum_{K=0}^{I-1} \left[w_{IH}\, w_{0K}\, w_{I-1,K} \frac{\prod_{J=0}^{H} w(J \to v_J)}{w(K \to v_K)} \sum_{m=0}^{|K|-1} w(K \to v_{Km}) \right] . \end{aligned} \tag{2.17}$$

To apply eq. (2.11), we divide this expression by the one in eq. (2.16). Due to the factorisation $w_{0H} = w_{0K} w_{KI} w_{IH}$ implied by eq. (2.15), the factors w_{0K} and w_{IH} in the numerator are cancelled by w_{0H}; we therefore arrive at the simplified formula

$$\begin{aligned} m_{v_0 v_H} &= \sum_{I=1}^{H} \sum_{K=0}^{I-1} \frac{w_{IH}\, w_{0K}\, w_{I-1,K}}{w_{0H} \prod_{J=0}^{H} w(J \to v_J)} \frac{\prod_{J=0}^{H} w(J \to v_J)}{w(K \to v_K)} \sum_{m=0}^{|K|-1} w(K \to v_{Km}) \\ &= \sum_{I=1}^{H} \sum_{K=0}^{I-1} \frac{w_{I-1,K}}{w_{KI}} \frac{\Pi_K}{\pi_{v_K}} , \end{aligned} \tag{2.18}$$

where we have written $\boldsymbol{\pi}$ in terms of tree weights via eq. (2.14) and used the identity of $\Pi_K = \sum_{m=0}^{|K|-1} \pi_{v_{Km}}$ for the equilibrium cluster occupancy probability of the random walker. Eq. (2.18) is a useful result in its own right, which we demonstrate in the remaining sections 2.4 to 2.6 of this chapter. Due to its connection to the EEL shown at the end of this section, we refer to eq. (2.18) as the *generalised essential edge lemma* (GEEL).

We now make the same simplification for the coarse-grained walker. Retaining our convention of using capital letters for reference to coarse-grained dynamics, we are interested in

$$M_{0H} = \frac{S_{0H}}{S_H} \ . \tag{2.19}$$

Upon choosing the subgraphs $I = 0, \ldots, H$ as the clusters of the coarse-grained dynamics, these are collapsed into single vertices, and all spanning trees become lines (see as illustrations figs. 2.1 and 2.2 where all nodes within the same shaded area are identified). Consequently, S_H is the weight of the directed path from 0 to H,

$$S_H = \prod_{I=0}^{H-1} Q_{I,I+1} = W_{0H} \ . \tag{2.20}$$

On the other hand, all two-tree forests contributing to S_{0H} can again be found by omitting one edge of the path $0, \ldots, H$, directing the H-component towards H and having the 0-component point anywhere. As above, these two requirements can be condensed into

$$S_{0H} = \sum_{I=1}^{H} \sum_{K=0}^{I-1} W_{0K} \, W_{I-1,K} \, W_{IH} \ . \tag{2.21}$$

In the quotient of eqs. (2.20) and (2.21), we can again use the factorisation $W_{0H} = W_{0K} W_{KI} W_{IH}$ in the denominator as above, such that the term $W_{0K} W_{IH}$ is cancelled. Thus eq. (2.11) becomes

$$M_{0H} = \sum_{I=1}^{H} \sum_{K=0}^{I-1} \frac{W_{IH} \, W_{0K} \, W_{I-1,K}}{W_{0H}} = \sum_{I=1}^{H} \sum_{K=0}^{I-1} \frac{W_{I-1,K}}{W_{KI}} \ . \tag{2.22}$$

For the comparison with eq. (2.18), we express the LE path weights W in terms of the unclustered weights w. To this end, we apply the LE definition of coarse-graining, eq. (2.9), and the chain structure of the coarse-grained graph, which implies

$$Q_{IJ} = \left(\delta_{I,J+1} + \delta_{I,J-1}\right) \frac{\pi_{v_I}}{\Pi_I} q_{v_I v_J} \ . \tag{2.23}$$

Substituting the transition probabilities of the coarse-grained dynamics in the definition of W_{IJ} given in eq. (2.15), we can rewrite the fraction in eq. (2.22) as follows

$$\begin{aligned}\frac{W_{I-1,K}}{W_{KI}} &= \left[\prod_{J=K+1}^{I-2} \frac{Q_{J,J-1}}{Q_{J,J+1}}\right] \frac{Q_{I-1,I-2}}{Q_{K,K+1}Q_{I-1,I}} = \left[\prod_{J=K+1}^{I-2} \frac{q_{v_{J,J-1}}}{q_{v_{J,J+1}}} \frac{q_{v_{I-1,I-2}}}{q_{v_{K,K+1}}}\right] \frac{\Pi_K}{q_{v_{I-1,I}}\pi_{v_K}} \\ &= \frac{w_{I-1,K}}{w_{KI}} \frac{\Pi_K}{\pi_{v_K}} \ . \end{aligned} \tag{2.24}$$

Substituting eq. (2.24) in eq. (2.22) and comparing with the right-hand side of eq. (2.18), we finally get

$$m_{v_0 v_H} = M_{0H} \ . \tag{2.25}$$

We remark that eq. (2.25) generalises directly to arbitrary pairs of vertices v_I, v_J along the backbone and is not restricted to v_0 and v_H, thus it holds that

$$m_{v_I v_J} = M_{IJ} \tag{2.26}$$

for arbitrary I, J. The proof is valid for arbitrary random walkers; it implies that along the backbone of the necklace, coarse-graining according to LE preserves MFPTs.

Since random walks on graphs with necklace structure can be coarse-grained into one-dimensional random walks, explicit MFPT formulae can be derived for such random walks, using eq. (2.22). In the next section, section 2.4, we demonstrate this method in detail for the simple random walk on c-ary trees.

In addition, we note that for graphs with necklace structure, where eq. (2.26) holds exactly, MFPTs can also be computed from eq. (2.18). Conversely, when graphs do not have necklace structure, neither eq. (2.18) nor eq. (2.26) hold exactly, however, when deviations from the necklace structure are small, one may expect eq. (2.26) to hold approximately. This means that MFPTs for the coarse-grained dynamics can be used as proxies for certain MFPTs in the original dynamics. Importantly, the coarse-grained MFPTs can still be computed exactly via eq. (2.22), as long as the coarse-grained graph is a one-dimensional lattice. Hence, the LE coarse-graining method can be used to get a reliable estimate of MFPTs in graphs with a more general structure than that of necklaces, as long as they can be coarse-grained into one-dimensional lattices.

We conclude this section by showing that for reversible random walks on non-directed graphs, with edge weights $\boldsymbol{e}$, such that $e_{ij} = e_{ji} \ \forall \ i, j$, and the matrix of transition probabilities

$$q_{ij} = \frac{e_{ij}}{\tilde{k}_i} \tag{2.27}$$

with the *node strength* $\tilde{k}_i = \sum_j e_{ij}$, eq. (2.18) retrieves the EEL as given in Lemma 5.1 of [62].

Firstly, as the MFPTs between the nodes of the backbone are additive, i.e. $m_{v_I v_J} + m_{v_J v_K} = m_{v_I v_K}$ for $I < J < K$, we may also write eq. (2.18) as

$$m_{v_0 v_H} = \sum_{I=1}^{H} m_{v_{I-1} v_I} \ , \tag{2.28}$$

such that for each $I \geq 1$

$$m_{v_{I-1} v_I} = \sum_{K=0}^{I-1} \frac{w_{I-1,K}}{w_{KI}} \frac{\Pi_K}{\pi_{v_K}} \ . \tag{2.29}$$

Due to the symmetry of $\boldsymbol{e}$, the equilibrium occupancy probabilities are given by

$$\pi_{v_K} = \frac{\tilde{k}_{v_K}}{Z} = \frac{1}{Z}\left(e_{v_K v_{K-1}} + e_{v_K v_{K+1}} + \sum_{\ell=1}^{|K|-1} e_{v_K v_{K\ell}} \right) \ , \tag{2.30}$$

where $Z = \sum_{K=0}^{H} \sum_{\ell=0}^{|K|-1} \tilde{k}_{v_{K\ell}}$ is the normalising factor and we stipulate that $e_{v_0, v_{-1}} = 0$ (for $K = 0$). Similarly, the equilibrium occupation probability for the cluster K is the sum of the corresponding probabilities of its vertices

$$\begin{aligned}
\Pi_K &= \pi_{v_K} + \frac{1}{Z} \sum_{\ell=1}^{|K|-1} \sum_{m=0}^{|K|-1} e_{v_{K\ell} v_{Km}} \\
&= \frac{1}{Z}\left(e_{v_K v_{K-1}} + e_{v_K v_{K+1}} + \sum_{m=0}^{|K|-1} e_{v_K v_{Km}} + \sum_{\ell=1}^{|K|-1} \sum_{m=0}^{|K|-1} e_{v_{K\ell} v_{Km}} \right) \\
&= \frac{1}{Z}\left(e_{v_K v_{K-1}} + e_{v_K v_{K+1}} + \sum_{\ell,m=0}^{|K|-1} e_{v_{K\ell} v_{Km}} \right) \ .
\end{aligned} \tag{2.31}$$

Moreover, the first factor in the sum of eq. (2.29) can be simplified by expanding the w's according to their definition in eq. (2.15)

$$\frac{w_{I-1,K}}{w_{KI}} = \prod_{J=K+1}^{I-1} \frac{e_{v_J, v_{J-1}}}{\tilde{k}_{v_J}} \prod_{J=K}^{I-1} \frac{\tilde{k}_{v_J}}{e_{v_J, v_{J+1}}} = \frac{\tilde{k}_{v_K}}{e_{v_{I-1} v_I}} \ . \tag{2.32}$$

This expression substituted into eq. (2.29) produces the EEL derived in Lemma 5.1 of [62]

$$\begin{aligned} m_{v_{I-1}v_I} &= \frac{1}{e_{v_{I-1}v_I}} \sum_{K=0}^{I-1} \left(e_{v_K v_{K-1}} + e_{v_K v_{K+1}} + \sum_{\ell,m=0}^{|K|-1} e_{v_{K\ell} v_{Km}} \right) \\ &= 1 + \frac{2}{e_{v_{I-1}v_I}} \left(\sum_{K=0}^{I-2} e_{v_K v_{K+1}} + \sum_{K=0}^{I-1} \sum_{0 \le \ell < m \le |K|-1} e_{v_{K\ell} v_{Km}} \right), \end{aligned} \tag{2.33}$$

where in the last step we have used again the symmetry of $\boldsymbol{e}$.

Eq. (2.33) replaces the matrix inversion in eq. (2.7) by a sum over edge weights, a much less expensive operation. It provides particularly great leverage when a graph has many essential edges, the limiting case being a tree, for which every edge is essential.

For non-directed *unweighted* graphs, $\boldsymbol{e}$ is replaced by an adjacency matrix $\mathbf{A}$, with entries $A_{ij} \in \{1, 0\}$ denoting presence (1) or absence (0) of links. Then, the inner sum in eq. (2.33) counts the number of edges E_K within subgraph K, and the sum $\sum_{K=0}^{I-2} e_{v_K v_{K+1}}$ counts the number of backbone edges connecting v_0 and v_{I-1}, amounting to $I-1$. Therefore, eq. (2.33) yields in this case

$$m_{v_{I-1}v_I} = 2I - 1 + 2\sum_{K=0}^{I-1} E_K \ . \tag{2.34}$$

Hence, for simple random walks on non-directed unweighted graphs, all that is required to compute MFPTs between the "hanging points" of two clusters I and J is the number of edges in the subgraphs $0, \ldots, I$. It is important to stress, however, that eqs. (2.33) and (2.34) apply only to reversible random walks, while eq. (2.18) applies to general random walks.

In the following section 2.4, we demonstrate how the LE coarse-graining method can be used to derive explicit MFPT formulae for the simple random walk on c-ary trees. In the subsequent sections 2.5 and 2.6, we apply the GEEL, eq. (2.18), to non-tree graphs with necklace structure and more general random walks, including irreversible random walks.

2.4 Mean first passage times in c-ary trees: exact results

In this section, we apply the LE coarse-graining method to the simple random walk on an unweighted, non-directed c-ary tree of height H, which consists of a root with degree c, $H-1$ levels of descendants with degree $c+1$ (one of which corresponds to the "upward" edge) and a bottom level of leaves with unit degree (see fig. 2.3 for an illustration). The

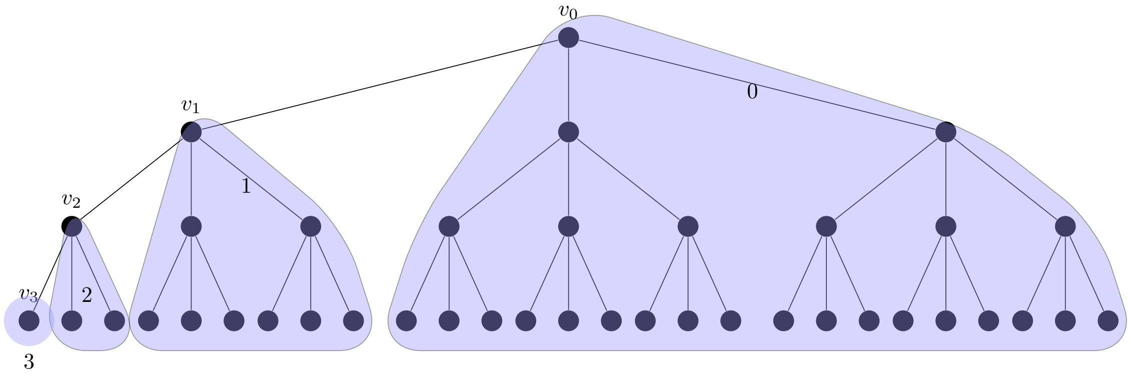


Fig. 2.3 Ternary tree with height $H = 3$. The root is v_0 and the target is v_3. Shaded areas show clusters $0, 1, 2, 3$.

transition matrix has elements

$$q_{ij} = \frac{A_{ij}}{k_i} \,, \tag{2.35}$$

where $\boldsymbol{A}$ is the adjacency matrix and $k_i = \sum_j A_{ij}$ is the degree of node i. The equilibrium occupancy probability of node i is $\pi_i = k_i/(2E)$ where $E = \frac{1}{2}\sum_{i=1}^{n} k_i$ is the total number of links. The MFPTs m_{ij} between any two nodes i, j can in principle be obtained by solving numerically the system of equations (2.6), however, in this work we are concerned with the derivation of explicit formulae. Since the simple random walker defined by eq. (2.35) is reversible, and every link is essential, the EEL in eq. (2.33) is also applicable here (as are other methods for reversible random walkers). However, we apply here the LE coarse-graining method, with the purpose of demonstrating it on a simple example, where results are available via other methods and can be easily validated. Our first objective is to calculate the MFPT from the root to a target leaf, then we turn to MFPTs between arbitrary vertices.

2.4.1 MFPT from root to leaf: exact results in the coarse-grained tree

Consider a c-ary tree of height H, with root v_0, as shown in fig. 2.3. Without loss of generality, we set the target in the first leaf, v_H, noting that, due to the symmetry of the tree, it is always possible to draw the diagram in such a way that the target is the first leaf.

The starting point of our derivation consists in reducing the dimensionality of the problem by coarse-graining the tree according to the LE method. We coarse-grain the tree into $H+1$ subgraphs, in such a way that every node in the path $v_0, v_1, \ldots, v_H$ (see fig. 2.3) is assigned its own subgraph. We define each subgraph $I = 0, \ldots, H$ as containing v_I, and all the vertices of the tree rooted in v_I, excluding the branches through v_{I-1} and v_{I+1}, as shown by the shaded areas in fig. 2.3. Note that subgraph H contains only the node v_H.

By regarding fig. 2.3, it is clear that c-ary trees belong to the class of graphs with necklace structure considered in section 2.3, hence the MFPT from root to leaf can be computed as the MFPT from cluster 0 to cluster H in the LE coarse-grained dynamics.

We thus define the hopping probability between clusters according to the LE definition eq. (2.9), where $\Pi_I = \sum_i C_{iI}\pi_i$. Since the $H+1$ clusters are sitting on a one-dimensional lattice, the transition matrix $\boldsymbol{Q}$ of the coarse-grained dynamics will be a $(H+1)\times(H+1)$ tridiagonal matrix. Its elements are obtained from eq. (2.9) as follows: Writing $\sum_i C_{iI}\cdots = \sum_{i\in I}\cdots$ and $\sum_{j\in I} = \sum_j - \sum_{j\notin I}$, we have

$$Q_{II} = \frac{1}{\sum\limits_{i\in I} k_i}\sum_{i\in I}\left(\sum_j A_{ij} - \sum_{j\notin I} A_{ij}\right) . \tag{2.36}$$

Noting that $k_i = \sum_j A_{ij}$ and that the number of links between cluster I and any other cluster is $\sum_{i\in I, j\notin I} A_{ij} = 2$ for all I (except clusters $I = 0, H$ that have a single out-going edge, each), we have

$$Q_{II} = \begin{cases} 1 - \frac{1}{\sum\limits_{i\in I} k_i} & : I \in \{0, H\} \\ 1 - \frac{2}{\sum\limits_{i\in I} k_i} & : 1 \leq I \leq H-1 , \end{cases} \tag{2.37}$$

and similarly

$$Q_{IJ} = \frac{1}{\sum\limits_{i\in I} k_i}(\delta_{I,J-1} + \delta_{I,J+1}) \quad : I, J = 0 \ldots H , \tag{2.38}$$

with the understanding that $\delta_{I,-1} = \delta_{I,H+1} = 0$. The sums $\sum_{i\in I} k_i$ can be derived using the following facts:

1. Every subgraph I consists of a top node at height $H-I$, and $c-1$ copies of the c-ary tree with height $H-I-1$.
2. A c-ary tree with height H has

$$n = (c^{H+1} - 1)/(c-1) \tag{2.39}$$

vertices.

3. The sum of degrees of any graph is twice the number of its edges.
4. Any tree with n vertices has exactly $n-1$ edges.

Using these observations, we find

$$\sum_{i\in I} k_i = \begin{cases} 2(|I|-1)+1 = 2c^{H-I}-1 & : I \in \{0,H\}\,, \\ 2(|I|-1)+2 = 2c^{H-I} & : 1 \le I \le H-1\,, \end{cases} \tag{2.40}$$

where we have denoted the size of subgraph I, i.e. the number of its vertices, by $|I|$. Hence, the local-equilibrium transition probabilities read

$$Q_{IJ} = \begin{cases} (\delta_{J,I-1}+\delta_{J,I+1})\frac{1}{2c^{H-I}-1} & : I \in \{0,H\}\,, \\ (\delta_{J,I-1}+\delta_{J,I+1})\frac{1}{2c^{H-I}} & : 1 \le I \le H-1\,, \end{cases} \tag{2.41}$$

$$Q_{II} = \begin{cases} \frac{2c^{H-I}-2}{2c^{H-I}-1} & : I \in \{0,H\}\,, \\ \frac{c^{H-I}-1}{c^{H-I}} & : 1 \le I \le H-1\,. \end{cases} \tag{2.42}$$

Eq. (2.25) from section 2.3 implies that the MFPT $m_{v_0 v_H}$ from root to target in the original tree matches the MFPT M_{0H} between the first and last cluster in the LE coarse-grained dynamics, when the clusters are defined as above. Thus, we can now calculate $m_{v_0 v_H}$ by appealing to eqs. (2.25) and (2.22). In the latter, the fractions can be cancelled efficiently since $Q_{I,I-1} = Q_{I,I+1}$ for $I = 1, \ldots, H-1$,

$$\frac{W_{I-1,K}}{W_{KI}} = \prod_{J=K+1}^{I-1} \frac{Q_{J,J-1}}{Q_{J,J+1}} \frac{1}{Q_{K,K+1}} = \frac{1}{Q_{K,K+1}}\,, \tag{2.43}$$

leading us to

$$M_{0H} = \sum_{I=1}^{H} \sum_{K=0}^{I-1} \frac{1}{Q_{K,K+1}}. \tag{2.44}$$

Substituting the transition probabilities from eq. (2.41), we obtain the final result

$$\begin{aligned} M_{0H} &= 2\sum_{j=1}^{H-1} jc^j + H(2c^H-1) \\ &= H\left(\frac{2c^{H+1}}{c-1}-1\right) - 2c\frac{c^H-1}{(c-1)^2}\,. \end{aligned} \tag{2.45}$$

The above expression matches exactly the MFPT $m_{v_0 v_H}$ in the original system (i.e. without coarse-graining), as derived in Example 5.14 of [62] by using the EEL on every edge between v_0 and v_H and adding up the results. We note that thanks to the tridiagonal nature of the coarse-grained transition matrix $\boldsymbol{Q}$, one could have also pursued the matrix inversion in eq. (2.7), however, computations via this route are more involved.

2.4.2 MFPTs between arbitrary vertices of the c-ary tree

In this section, we complement the results obtained in section 2.4.1 for the MFPT from root to leaf by deriving explicit formulae for the MFPTs between any two vertices s and t of a c-ary tree. In contrast to the previous section 2.4.1, here we appeal directly to eq. (2.18), which provides an equivalent route to eq. (2.22).

Firstly, we can always permute the branches in such a way that s lies on an outer branch of the diagram, as shown in fig. 2.4. We can then proceed by (i) finding their common ancestor a, which also lies on the outer branch, on the path between s and the root, (ii) calculating the MFPTs from the source s to the ancestor a and from the ancestor to the target t separately, and finally (iii) adding up the results.

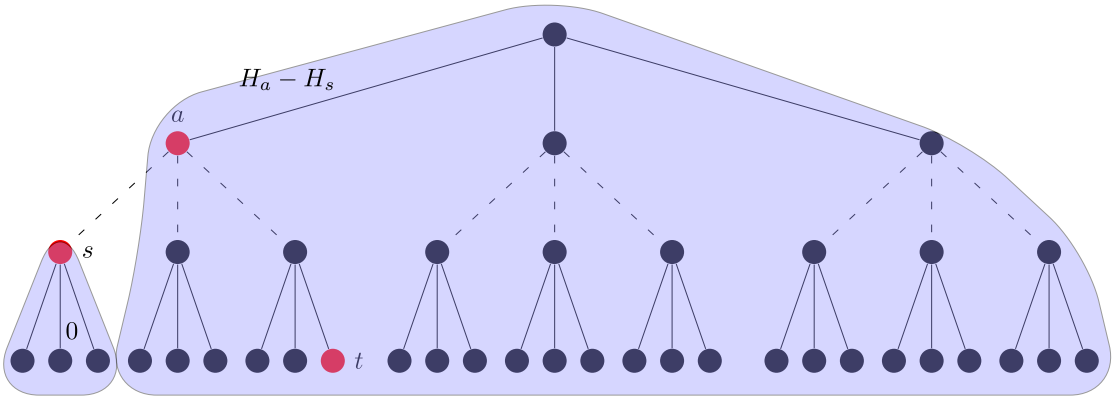


Fig. 2.4 Ternary tree as shown in fig. 2.3, with an example of source s, target t and common ancestor a marked in red. Shaded areas enclose the subgraphs 0 and $H_a - H_s$ defined for the first-passage process from s to a (intermediate clusters not shown). For this example, $H_s = 1$ and $H_t = 0$. For the first-passage process from a to s, subgraphs are labelled in reverse order. Dashed lines indicate potentially omitted levels.

For the purpose of this section, H denotes the height of the tree, not the number of clusters employed in the coarse-graining approach. For this, we denote by H_s the height of s (defined as the distance to the leaves of the tree rooted in s), and by H_a the height of a. The "upward" MFPT m_{sa} can be obtained by defining $H_a - H_s + 1$ subgraphs as follows: subgraph 0 contains the tree rooted in s excluding the branch pointing towards a. Subgraph $H_a - H_s$ is formed by the tree rooted in a excluding the branch leading to s. For instance, in the diagram shown in fig. 2.4, the subgraph 0 consists of the children of s, and $H_a - H_s$ contains all branch-offs at a leading away from s. Following the notation introduced in section 2.3, we identify $v_0 = s$, $v_{H_a - H_s} = a$ and enumerate the vertices along the line connecting s to a as $v_1, \ldots, v_{H_a - H_s - 1}$. Each intermediate subgraph I for $I \in \{1, \ldots, H_a - H_s - 1\}$ contains the tree rooted in v_I, excluding both branches leading to s and a.

In order to apply eq. (2.18), we need to compute for each $I = 1,\dots,H_a - H_s$ and $K = 0,\dots,I-1$ the summands

$$\frac{w_{I-1,K}}{w_{KI}} \frac{\Pi_K}{\pi_{v_K}} = \prod_{J=K+1}^{I-1} \frac{q_{v_J v_{J-1}}}{q_{v_J v_{J+1}}} \frac{\Pi_K}{q_{v_K v_{K+1}} \pi_{v_K}} \tag{2.46}$$

which follows from the definition of the weights w given in eq. (2.15). To handle the product, we notice that leaves and the root can only appear on the backbone as the source $s = v_0$ and the ancestor $a = v_{H_a - H_s}$, respectively. Meanwhile, the vertex v_J with the product index $J = K+1,\dots,I-1$ runs from v_1 if $K = 0$ to $v_{H_a-H_s-1}$ if $I = H_a - H_s$. Therefore, the degree of v_J is always $k_{v_J} = c+1$, and thus $q_{v_J v_{J+1}} = q_{v_J v_{J-1}} = \frac{1}{c+1}$. Consequently, the product in the above formula can be cancelled. Inserting eq. (2.46) into eq. (2.18) then gives

$$m_{sa} = \sum_{I=1}^{H_a-H_s} \sum_{K=0}^{I-1} \prod_{J=K+1}^{I-1} \frac{q_{v_J v_{J-1}}}{q_{v_J v_{J+1}}} \frac{\Pi_K}{q_{v_K v_{K+1}} \pi_{v_K}} = \sum_{I=1}^{H_a-H_s} \sum_{K=0}^{I-1} \frac{1}{q_{v_K v_{K+1}}} \frac{\Pi_K}{\pi_{v_K}} . \tag{2.47}$$

Since for the simple random walker the equilibrium probabilities of the vertices are proportional to their degrees, we can expand

$$\Pi_K = \sum_{\ell=0}^{|K|-1} \pi_{v_{K\ell}} = \frac{1}{Z} \sum_{\ell=0}^{|K|-1} k_{v_{K\ell}} , \tag{2.48}$$

with normalising factor Z. Similarly to the argument in section 2.4.1, the sum of the degrees in K counts the number of edges leaving K, and double-counts the edges within it. Using the fact that each inner subgraph $K = 1,\dots H_a - H_s$ is connected to two neighbouring subgraphs, while the outer subgraphs $K = 0,\ H_a - H_s$ only are connected to one, each, we have

$$\Pi_K = \begin{cases} \frac{1}{Z}[2(|K|-1)+1] & : K \in \{0, H_a - H_s\} , \\ \frac{1}{Z}[2(|K|-1)+2] & : 1 \le K \le H_a - H_s - 1 . \end{cases} \tag{2.49}$$

The sizes of the subgraphs K are now given by

$$|K| = \begin{cases} \frac{c^{H_s+1}-1}{c-1} & : K = 0 , \\ \frac{c^{H+1}-c^{H_a}}{c-1} & : K = H_a - H_s , \\ c^{H_s+K} & : 1 \le K \le H_a - H_s - 1 \end{cases} \tag{2.50}$$

with the same reasoning as in section 2.4.1. Moreover, the equilibrium occupancy probabilities $\pi_{v_K} = \frac{k_{v_K}}{Z}$ as well as the hopping probabilities $q_{v_K v_{K+1}} = \frac{1}{k_{v_K}}$ can be combined into

$$\frac{1}{q_{v_K v_{K+1}}} \frac{\Pi_K}{\pi_{v_K}} = \begin{cases} 2(|K|-1)+1 & : \; K \in \{0, H_a - H_s\} \; , \\ 2(|K|-1)+2 & : \; 1 \le K \le H_a - H_s - 1 \; . \end{cases} \tag{2.51}$$

Substituting this expression together with the subgraph sizes $|K|$ from eq. (2.50) into eq. (2.47), we arrive at the result

$$\begin{aligned} m_{sa} &= \sum_{I=1}^{H_a-H_s} \left(\sum_{K=1}^{I-1} 2|K| + 2|0| - 1 \right) \\ &= \sum_{I=1}^{H_a-H_s} \left(\sum_{K=0}^{I-1} 2c^{H_s+K} + 2\frac{c^{H_s+1}-1}{c-1} - 1 \right) \\ &= 2\frac{c^{H_a+1}-c^{H_s+1}}{(c-1)^2} - (H_a - H_s)\frac{c+1}{c-1} \; . \end{aligned} \tag{2.52}$$

In the opposite direction, downward from a to s, we have to consider the v_I's in reverse order

$$m_{as} = \sum_{I=1}^{H_a-H_s} \sum_{K=0}^{I-1} \frac{1}{q_{v_{H_a-H_s-K}, v_{H_a-H_s-K-1}}} \frac{\Pi_{H_a-H_s-K}}{\pi_{v_{H_a-H_s-K}}} \; , \tag{2.53}$$

into which we can again substitute eq. (2.51) to obtain

$$\begin{aligned} m_{as} &= \sum_{I=1}^{H_a-H_s} \left(\sum_{K=1}^{I-1} 2c^{H_a-K} + 2\frac{c^{H+1}-c^{H_a}}{c-1} - 1 \right) \\ &= (H_a - H_s)\left(\frac{2c^{H+1}}{c-1} - 1 \right) - 2\frac{c^{H_a+1}-c^{H_s+1}}{(c-1)^2} \; . \end{aligned} \tag{2.54}$$

Note that the limiting case $H_s = 0$, $H_a = H$ reproduces eq. (2.45) for the MFPT from the root to any leaf, as it should.

In order to obtain m_{st}, we need to add m_{sa} and m_{at}; the latter is obtained by replacing t for s in eq. (2.54). The final result can be written as

$$\begin{aligned} m_{st} &= m_{sa} + m_{at} \\ &= 2(n-1)(H_a - H_t) + 2\frac{c^{H_t+1}-c^{H_s+1}}{(c-1)^2} + (H_s - H_t) + \frac{2}{c-1}(H_s - H_t) \; , \end{aligned} \tag{2.55}$$

where n is the total number of vertices of the tree, eq. (2.39). This result is in agreement with those derived in Example 5.14 of [62].

2.5 Exact results on non-tree graphs with necklace structure

This section applies the GEEL eq. (2.18) on two further examples of the necklace type, namely the c-star and a concatenation of $H+1$ cliques of size c. We note that the c-star graph is, in fact, a c-ary tree of height 1, so the results obtained here also follow trivially from those obtained in section 2.4, via eq. (2.22).

2.5.1 Star graph

In this section, we consider a star graph of c vertices around a middle vertex v_0 as shown in fig. 2.5.

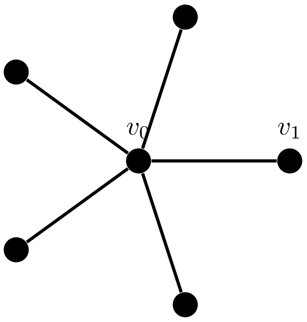


Fig. 2.5 A star with $c = 5$ vertices. The subgraph 1 is just the vertex v_1; all other vertices form the subgraph 0.

In analogy to what was done in section 2.4, we define subgraph 1 as containing only vertex v_1, and subgraph 0 as containing all other vertices. This makes clear that the star graph belongs to the family of necklace graphs, as required for our approach to work.

Proceeding as in the previous calculations, we apply the GEEL, eq. (2.18), to determine $m_{v_0v_1}$ for the simple random walker on the c-star. As with two clusters we have $H = 1$, the GEEL in this case contains but a single summand,

$$m_{v_0v_1} = \sum_{I=1}^{1} \sum_{K=0}^{I-1} \frac{w_{I-1,K}}{w_{KI}} \frac{\Pi_K}{\pi_{v_K}} = \frac{1}{q_{v_0v_1}} \frac{\Pi_0}{\pi_{v_0}}, \tag{2.56}$$

using that $w_{00} = 1$ and $w_{01} = q_{v_0v_1}$. The equilibrium probabilities on the right hand side of eq. (2.56) are given by $\pi_{v_0} = \frac{c}{2c}$ and $\Pi_0 = \frac{2c-1}{2c}$, because there is one central vertex v_0 with degree c, and $c-1$ outer vertices with degree 1 in cluster 0. Substituting these values together with the hopping probability $q_{v_0 v_1} = \frac{1}{c}$ into eq. (2.56), we find the MFPT

$$m_{v_0 v_1} = \frac{c(2c-1)}{c} = 2c - 1 \; . \tag{2.57}$$

Given the simplicity of the star graph, MFPTs can be calculated explicitly, through a variety of methods. For instance, eq. (2.57) could have been alternatively derived by noticing that the random walker steps to any outer node with uniform probability at every second step, and from a leaf back to the central vertex at every other step [62]. On the other hand, as noted above, the star graph is a c-ary tree with height 1, therefore it adheres to eq. (2.45). Finally, the matrix inversion in eq. (2.7) can also be performed directly [75].

2.5.2 Cliques on a necklace

Consider $H+1$ cliques, i.e. complete subgraphs, of size c, arranged in a chain passing through the hanging points $v_0, \ldots, v_H$; an example for $c = 4$ is shown in fig. 2.6. As this graph has necklace structure, we will again apply the GEEL, eq. (2.18), to determine $m_{v_0 v_H}$ for the simple random walker on this graph.

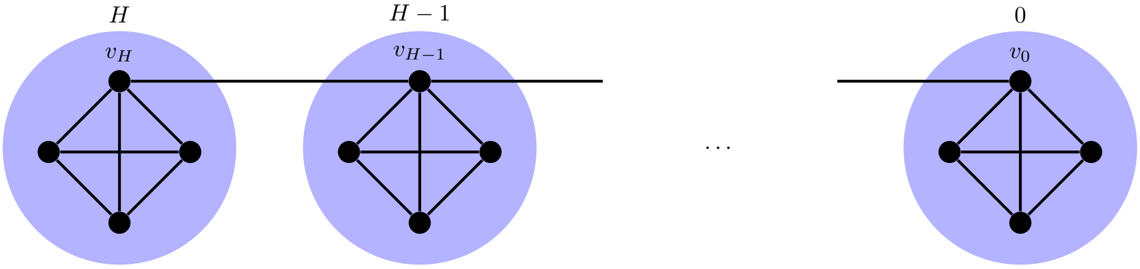


Fig. 2.6 A necklace with $H+1$ clusters, each of which is a clique with $c = 4$ vertices.

The transition probabilities between the nodes of the chain for the simple random walker are given by

$$q_{v_I v_J} = \begin{cases} (\delta_{I,J-1} + \delta_{I,J+1})\frac{1}{c} & : I \in \{0, H\} \, , \\ (\delta_{I,J-1} + \delta_{I,J+1})\frac{1}{c+1} & : 1 \leq I \leq H-1 \, , \end{cases} \tag{2.58}$$

as the degrees k_{v_I} are either c (first case) or $c+1$ (other cases). As in eq. (2.47) for the c-ary tree, this implies that the summands in eq. (2.18) simplify,

$$m_{v_0 v_H} = \sum_{I=1}^{H} \sum_{K=0}^{I-1} \frac{w_{I-1,K}}{w_{KI}} \frac{\Pi_K}{\pi_{v_K}} = \sum_{I=1}^{H} \sum_{K=0}^{I-1} \frac{\Pi_K}{q_{v_K v_{K+1}} \pi_{v_K}} \; . \tag{2.59}$$

To determine the equilibrium probabilities, we notice that within each subgraph there are $c-1$ vertices v_{Ki} ($i \neq 0$) with degree $k_{v_{Ki}} = c-1$, while the hanging vertex v_K has degree $k_{v_K} = c+1$ if $K = 1, \dots, H-1$ or $k_{v_K} = c$ if $K \in \{0, H\}$. Hence, the stationary probability ratios amount to

$$\frac{\pi_{v_K}}{\Pi_K} = \frac{k_{v_K}}{k_{v_K} + (c-1)^2} = \begin{cases} \frac{c}{c+(c-1)^2} & : K = 0, H \,, \\ \frac{c+1}{c+1+(c-1)^2} & : 1 \leq K \leq H-1 \,. \end{cases} \tag{2.60}$$

Given that the random walker is the simple random walker, we can now write the summands in eq. (2.59) as

$$\frac{\Pi_K}{q_{v_K v_{K+1}} \pi_{v_K}} = k_{v_I} + (c-1)^2 \,, \tag{2.61}$$

which allows us to conclude

$$\begin{aligned} m_{v_0 v_H} &= \sum_{I=1}^{H} \left(\sum_{K=1}^{I-1} \left(c + 1 + (c-1)^2 \right) + c + (c-1)^2 \right) \\ &= \frac{c(c-1)H(H+1)}{2} + H^2 \,. \end{aligned} \tag{2.62}$$

For a simple path of length H we have $c = 1$, thus reproducing the well-known $m_{v_0 v_H} = H^2$ [63]. For later reference, we also note that for $H = 1$ and arbitrary c, we find $m_{v_0 v_1} = c(c-1)+1$.

2.6 Applications to irreversible random walks

In this section, we verify that the GEEL, eq. (2.18), can also be applied to irreversible random walks (on necklace graphs), where the classical EEL, eq. (2.33), is not applicable. To this purpose, we consider below two simple examples where results can be validated by direct computations.

2.6.1 Example 1 – irreversible random walk

Consider the Markov chain in fig. 2.7, representing a random walk with transition matrix

$$\mathbf{q} = \begin{pmatrix} 0 & \frac{2}{3+\alpha} & \frac{1}{3+\alpha} & \frac{\alpha}{3+\alpha} \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} , \tag{2.63}$$

where the order of the states has been chosen as v_0, v_{01}, v_{02}, v_1. For the avoidance of doubt, in fig. 2.7 we have labelled edges using unnormalised weights e_{ij}.

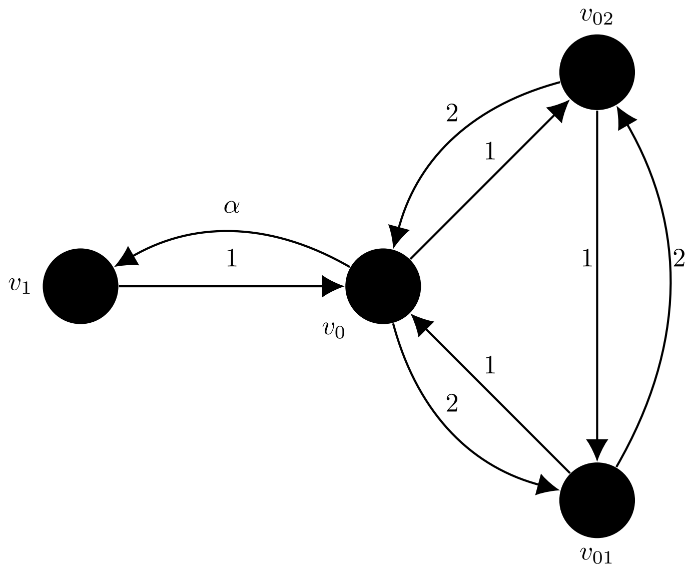


Fig. 2.7 A weighted graph with asymmetrical edge weights that define an irreversible Markov chain on its vertices.

Given the stationary probability vector

$$\boldsymbol{\pi}^T = \frac{1}{9+2\alpha}(3+\alpha, 3, 3, \alpha) \tag{2.64}$$

we can confirm that detailed balance with the transition probabilities above is not satisfied, e.g. for the nodes v_{02} and v_0, where the probability flows

$$\pi_{v_0} q_{v_0 v_{02}} = \frac{3+\alpha}{9+2\alpha}\frac{1}{3+\alpha} = \frac{1}{9+2\alpha} \tag{2.65}$$

and

$$\pi_{v_{02}} q_{v_{02} v_0} = \frac{3}{9+2\alpha}\frac{2}{3} = \frac{2}{9+2\alpha} \tag{2.66}$$

do not equate. As a consequence, the EEL, eq. (2.33), is not applicable. On the other hand, we can calculate MFPTs via the GEEL, eq. (2.18), which does not rely on dynamical reversibility. With the subgraphs for eq. (2.18) defined as $0 = \{v_0, v_{01}, v_{02}\}$ and $1 = \{v_1\}$, we have $H = 1$, which means that as in eq. (2.56), there is a single summand in the GEEL. Into this, we substitute the details of this example, eqs. (2.63) and (2.64)

$$m_{v_0 v_1} = \frac{1}{q_{01}}\frac{\Pi_0}{\pi_{v_0}} = \frac{9+\alpha}{\alpha}\,. \tag{2.67}$$

This result is easily validated by computing MFPTs directly, using eq. (2.7). This requires the first row sum of the inverse

$$\left(\mathbb{1}_3 - \widehat{\mathbf{q}}_{v_1}\right)^{-1} = \frac{1}{\alpha}\begin{pmatrix} 3+\alpha & 3 & 3 \\ 3+\alpha & \frac{9\alpha+21}{7} & \frac{6\alpha+21}{7} \\ 3+\alpha & \frac{3\alpha+21}{7} & \frac{9\alpha+21}{7} \end{pmatrix}, \tag{2.68}$$

which gives the same result as eq. (2.67).

Incidentally, a naïve appeal to the EEL can, despite its inapplicability, return the correct result here, if we replace the summands $2e_{v_{0\ell}v_{0m}}$ in eq. (2.33) by $e_{v_{0\ell}v_{0m}} + e_{v_{0m}v_{0\ell}}$:

$$\frac{e_{v_0v_{01}} + e_{v_{01}v_0} + e_{v_0v_{02}} + e_{v_{02}v_0} + e_{v_{01}v_{02}} + e_{v_{02}v_{01}}}{e_{v_0v_1}} + 1 = \frac{2+1+1+2+1+2}{\alpha} + 1 = \frac{9+\alpha}{\alpha}\ . \tag{2.69}$$

This observation can be traced back to the fact that for this particular example, the sum of edge weights

$$e_{0,01} + e_{01,0} + e_{0,02} + e_{02,0} + e_{01,02} + e_{02,01} + e_{01} = 9+\alpha \tag{2.70}$$

is actually the proportional weight of the cluster 0 in the steady-state. Also by coincidence, we have $q_{v_0v_1}\pi_{v_0} = e_{v_0v_1}$. Therefore, the numerator in eq. (2.69) is actually Π_0 (up to normalisation), while the denominator is $q_{v_0v_1}\pi_{v_0}$. Thus, we are back to the GEEL for two clusters. This equality is coincidental, particularly because $q_{v_0v_1}\pi_{v_0} = e_{v_0v_1}$ is only a given for reversible dynamics. Doing the same check in the following example shows that the GEEL and the EEL generally disagree for irreversible dynamics.

2.6.2 Example 2 – irreversible random walk

As a second example, we consider the graph in fig. 2.8. The edge weights define a random walker with transition matrix (in the order $v_0, v_{01}, v_{02}, v_{03}, v_1$)

$$\mathbf{q} = \begin{pmatrix} 0 & \frac{2}{3+\alpha} & 0 & \frac{1}{3+\alpha} & \frac{\alpha}{3+\alpha} \\ \frac{1}{5} & 0 & \frac{2}{5} & \frac{2}{5} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.71}$$

which has the stationary distribution

$$\boldsymbol{\pi}^T = \frac{1}{20\alpha+141}(30+10\alpha, 40, 27, 44, 10\alpha)\ . \tag{2.72}$$

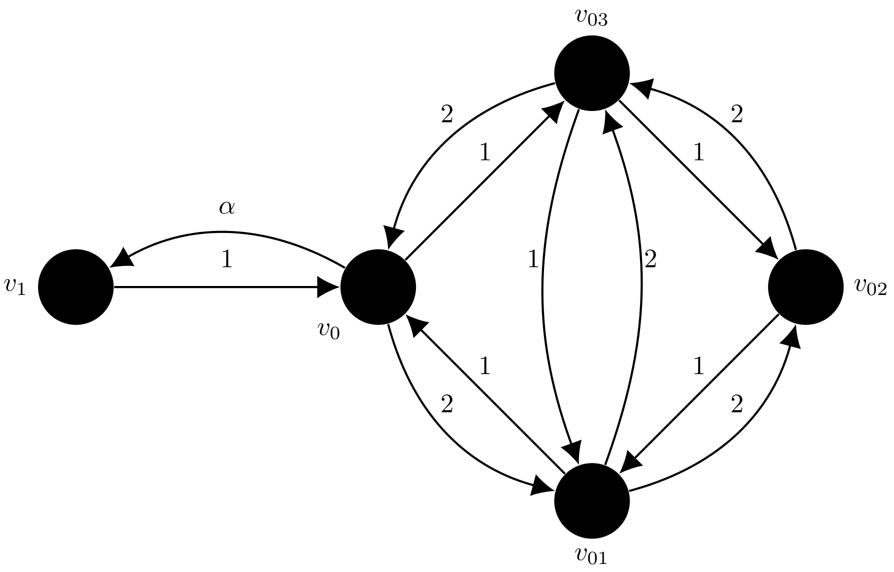


Fig. 2.8 A weighted graph extending the example given in fig. 2.7.

Defining the subgraphs $0 = \{v_0, v_{01}, v_{02}, v_{03}\}$, $1 = \{v_1\}$ for eq. (2.18), we proceed as in the previous example to find the MFPT

$$m_{v_0 v_1} = \frac{3+\alpha}{\alpha}\frac{\Pi_0}{\pi_{v_0}} = \frac{3+\alpha}{\alpha}\frac{10\alpha+141}{10\alpha+30} = 1 + \frac{141}{10\alpha} \,. \tag{2.73}$$

This result again is easily validated using eq. (2.7) and the first row sum of the inverse

$$\left(\mathbb{1}_3 - \widehat{\mathbf{q}}_{v_1}\right)^{-1} = \frac{1}{\alpha}\begin{pmatrix} 3+\alpha & 4 & \frac{27}{10} & \frac{22}{5} \\ 3+\alpha & \frac{5\alpha+12}{3} & \frac{10\alpha+27}{10} & \frac{20\alpha+22}{15} \\ 3+\alpha & \frac{2\alpha+12}{3} & \frac{7\alpha+27}{10} & \frac{130\alpha+330}{75} \end{pmatrix} . \tag{2.74}$$

For the comparison to the EEL, eq. (2.33), let us again add up all internal edge weights of cluster 0:

$$\sum_{i,j\in 0} e_{ij} = 5 \cdot 3 = 15 \,, \tag{2.75}$$

giving us $\frac{15+\alpha}{\alpha} \neq m_{v_0 v_1}$. It is clearly visible by comparison with the previous example, that in the present example the sum of edge weights within cluster 0 is simply not the steady-state mass of 0. Similarly, $\pi_{v_0} q_{v_0 v_1} \neq e_{v_0 v_1}$, i.e. the steady-state probability of v_0 is not the node strength, either. This leads us to conclude that the EEL and GEEL agree for irreversible dynamics if the sum of internal edge weights of 0 accidentally gives the correct Π_0 and if the node strength of v_0 is accidentally given by $\pi_{v_0} q_{v_0 v_1}$ (up to normalisation).

2.7 Conclusions

In this chapter, we explored the behaviour of MFPTs of random walkers on graphs under LE approximation. We show that for graphs resembling a necklace, the end-to-end MFPT of the "natural" coarse-grained graph is equal to the MFPT between the vertices of the line connecting the subgraphs. The defining property of a necklace is that its subgraphs are arranged linearly and each one hangs via a single vertex from a one-dimensional chain. Cayley trees, the T-graph, c-ary trees *etc.* – being trees – all fall into this class. To the best of our knowledge, conservation of MFPTs under LE coarse-graining for the necklace type of graphs was previously unknown.

In virtue of the LE approach, we are able to generalise the essential edge lemma (EEL) to non-reversible walkers and produce explicit and exact formulae for the MFPTs in cases where the EEL is inapplicable. We have checked our exact analytical formulae against well-known results (or limits thereof) where available, or against results obtained by the established standard formula, eq. (2.7).

Explicit formulae in terms of network parameters for MFPTs are hard to come by: our LE approach offers a way to outmanoeuvre the infamous matrix inversion in eq. (2.7), which in most cases can only be tackled numerically. Applications abound where explicit formulae are called for, as MFPTs are used as a low-order quantitative indicator in many different contexts. There is for instance an interest in MFPTs and FPTs to evaluate search strategies and transport for random walks, and models for diffusion on complex media ([56] and references therein). Further fields include the description of ill-mixed gene regulatory network models [83] and kinetics of reactions in high-dimensional potentials [64]. In addition, MFPTs have recently been applied to assess the heterogeneity of complex social systems [66]. Moreover, [84] shows that the numerical error incurred using eq. (2.7) – or other theoretically exact but numerically expensive methods – may lead to large errors for MFPTs between different communities of vertices. For this reason, explicit albeit approximate formulae may be preferable over exact but expensive or imprecise methods. We address the potential of our method to deliver useful approximations in chapter 3.

There are several interesting pathways for future work. First, in this work we have only considered graphs that can be coarse-grained into a one-dimensional lattice. However, one may envisage to extend this framework to graphs that can be coarse-grained into loop-less graphs, i.e. unbalanced trees.

In this chapter, we have focused on nodes along the "backbone" of graphs with necklace structure. MFPTs between nodes residing "far" from the backbone may be poorly described by the LE coarse-graining and other frameworks may be better suited for them. Recently, it has been shown that a coarse-graining method, which was proposed by Hummer and Szabo

[79], preserves MFPTs *averaged* over node pairs of different clusters [64]. This coarse-graining may provide more accurate estimates for nodes residing far from the backbone, however, it leads to a more complex relation between the transition matrices of the original and clustered network. It would be interesting to see whether analytical progress can be made for graph structures that allow one-dimensional coarse-grained representations, for such coarse-graining protocols.

Finally, we have focused entirely on *mean* first-passage times. Higher moments and full distributions of FPTs are considerably less tractable than their mean, such that only specialised results limited to certain moments or as approximations in specific problems are available [69, 85]. However, numerical simulations presented in appendix 2.B suggest that our LE-coarse graining method might preserve higher moments of FPTs approximately if the graph is an exact necklace.

In the next chapter, we proceed to show that we can capitalise on the LE approach in a second way. We demonstrate that it provides accurate and explicit (though approximate) formulae for MFPTs on graph structures that are not exact necklaces. Thus, we extend the virtues of the results of this chapter to a much larger class of graphs.

Appendix

Appendix 2.A Spanning trees, spanning forests and essential edges

As explained in section 2.3, one can calculate the MFPTs on a graph by solving the combinatorial problem of finding all the spanning trees and forests of certain kinds (i.e. two-tree forests) in the graph (see eq. (2.11)). Below, we provide the definitions of spanning trees and forests for the reader who is not familiar with these concepts.

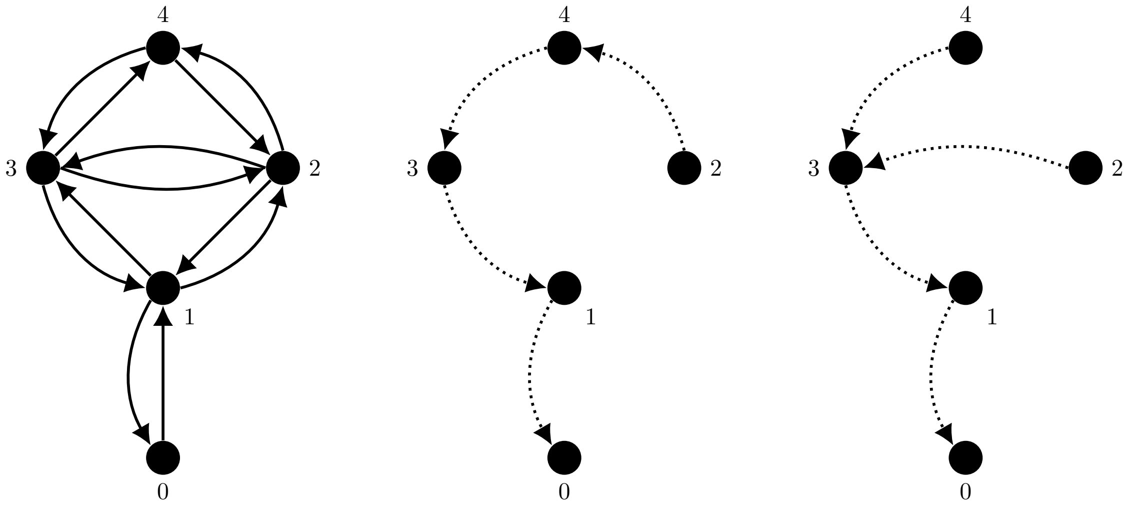


Fig. 2.A.1 Directed graph (left panel) and two examples of spanning trees with root 0 (middle and right panels). Note that the undirected edge $(0,1)$ is essential: Every spanning tree has to contain either $0 \to 1$ or $1 \to 0$, depending on the root.

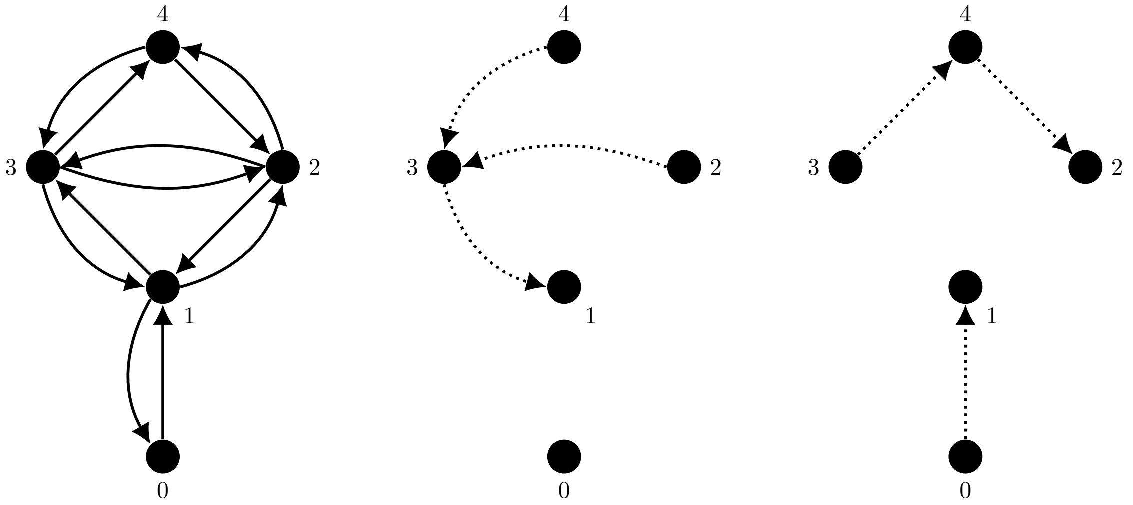


Fig. 2.A.2 Directed graph (left panel) and two examples of spanning forests (middle and right panels). The middle forest has roots 0 and 1, the right forest has roots 1 and 2.

Given a directed graph, one defines a *spanning forest* as a loopless directed subgraph that covers all vertices, while every vertex has at most one outgoing edge. Those vertices without outgoing edges are the *roots* of the forest. There is always at least one root, and if there are several, they define the different components, or trees, of the forest. In particular, a single-component spanning forest is called a *spanning tree*. For instance, fig. 2.A.1 shows two different directed spanning trees with root 0. Similarly, fig. 2.A.2 shows two directed spanning forests with roots $\{0,1\}$ and $\{1,2\}$, respectively.

The problem of finding the spanning trees and two-tree forests of a graph is in general a formidable combinatorial task for large graphs. However, it may become feasible for special graph structures. For instance, if the graph itself has the structure of a tree, it will possess exactly one spanning tree for every root.

Appendix 2.B Higher moments of FPTs

In this section, we present some evidence for a generalisation of eq. (2.26) to higher moments, and in fact full distributions, of FTPs. We recall from eq. (2.5) that the FPT t_{ij} is the first time step at which the walker is in state j, after having started from state i. We denote the m-th moment of t_{ij} by

$$\lambda_{ij}^m := \mathbb{E}(t_{ij}^m) = \sum_{s=0}^{\infty} s^m \mathbb{P}\left\{t_{ij} = s\right\} \tag{2.76}$$

for $m \geq 0$. Evidently, $\lambda_{ij}^0 = 1$ due to normalisation, and $\lambda_{ij}^1 = m_{ij}$ by definition of the MFPT from i to j.

For the first-passage process to j, we may without loss of generality assume that j is an absorbing state. In that case, the j-th row of the transition matrix $\mathbf{q}$ has a unit entry in the j-th column and 0 everywhere else. The probability that $t_{ij} \leq s$ is then given by the probability that the walker is in state j at time s (as it can have entered j either before time s and never left, or entered at time s). In terms of $\mathbf{q}$, this reads

$$\mathbb{P}\left\{t_{ij} \leq s\right\} = (\mathbf{q}^s)_{ij} \ , \tag{2.77}$$

or for the probability mass function (PMF)

$$\mathbb{P}\left\{t_{ij} = s\right\} = (\mathbf{q}^s)_{ij} - \left(\mathbf{q}^{s-1}\right)_{ij} \ . \tag{2.78}$$

Analogously, we denote FPTs and their moments on the coarse-grained graph by T_{IJ} and Λ_{IJ}^m, respectively.

Led by eq. (2.26), we now test if on a necklace the original and coarse-grained walker have the same FPT distributions or moments, i.e. if $\mathbb{P}\left\{t_{v_0 v_H} = s\right\} = \mathbb{P}\left\{T_{0H} = s\right\}$ or $\lambda_{v_{I-1}v_I}^m = \Lambda_{I-1,I}^m$ for any $m > 1$. We do this by way of example, considering a necklace of five cliques with five vertices, each, following the examples in section 2.5.2. Additionally, every edge is weighted by a number drawn independently and uniformly from the unit interval, and both edge directions are taken to be independent as well. That is, the edges (i, j) and (j,i)

are both present and weighted independently for each edge (i,j) present in the necklace as per section 2.5.2).

For a single realisation of edge weights, the PMFs obtained using eq. (2.78) are shown in fig. 2.B.1 along with their Kullback-Leibner (KL) divergence. To this end, both PMFs were truncated such that

$$\mathbb{P}\left\{t_{v_0 v_H} \leq t_{\text{trunc}}\right\} \approx \mathbb{P}\left\{T_{0H} \leq t_{\text{trunc}}\right\} \approx 0.9999 \tag{2.79}$$

to avoid numerical problems with the Kullback-Leibler divergence when the PMFs range close to zero. The two PMFs show an excellent agreement, with only a slight relative shift of the mode of T_{0H} to higher values.

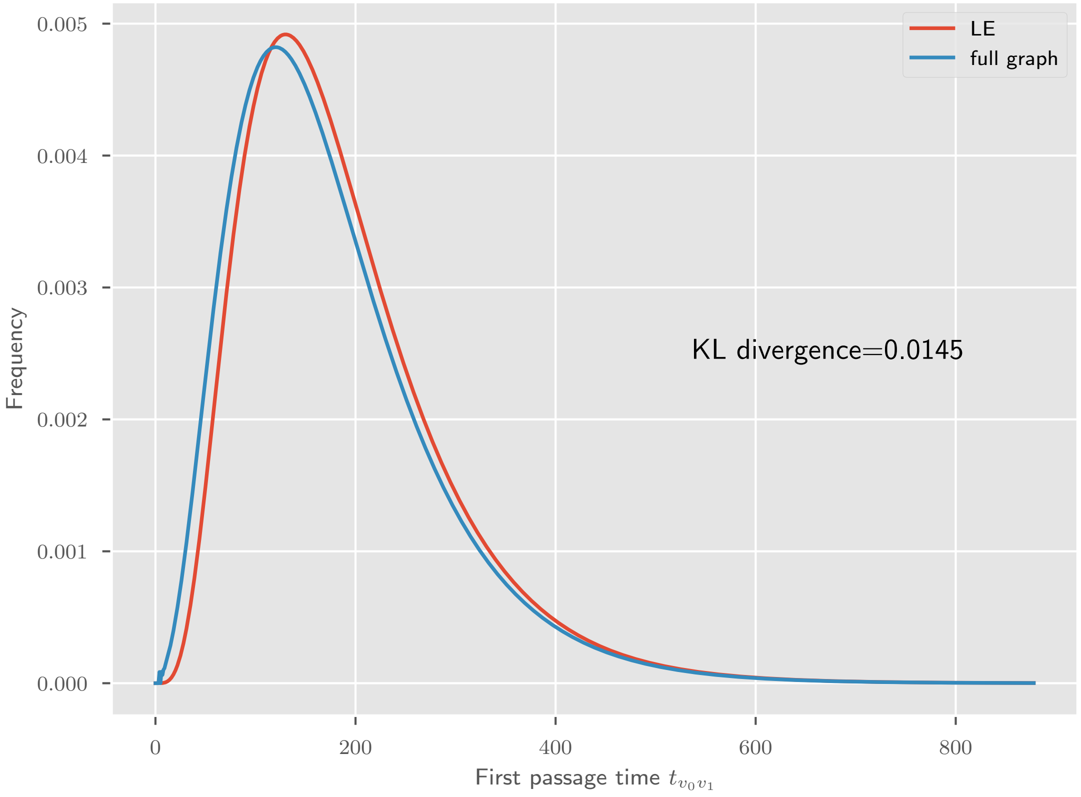


Fig. 2.B.1 Probability mass functions of the FPTs of the coarse-grained and full walker, obtained using eq. (2.78). The shown part of the mass functions account for a fraction of 0.9999 of the total mass, each. The Kullback-Leibner (KL) divergence measures the deviation between the curves.

Similarly, in fig. 2.B.2, the root-moments $\sqrt[m]{\Lambda_{0H}^m}$ of the coarse-grained walker are plotted against the root-moments $\sqrt[m]{\lambda_{v_0 v_H}^m}$ of the original walker for $m = 2, 3, 4, 5, 10$ and 15 over 200 realisations of edge weights. The moments were computed by truncating the PMFs, eq. (2.78), at $t = 880$ and applying eq. (2.76) (again truncated at $t = 880$).

Fig. 2.B.2 shows an excellent – though not exact – agreement between the moments Λ_{0H}^m and $\lambda_{v_0 v_H}^m$. There is a substantial disagreement between the two only for comparatively low values of $\lambda_{v_0 v_H}^m$.

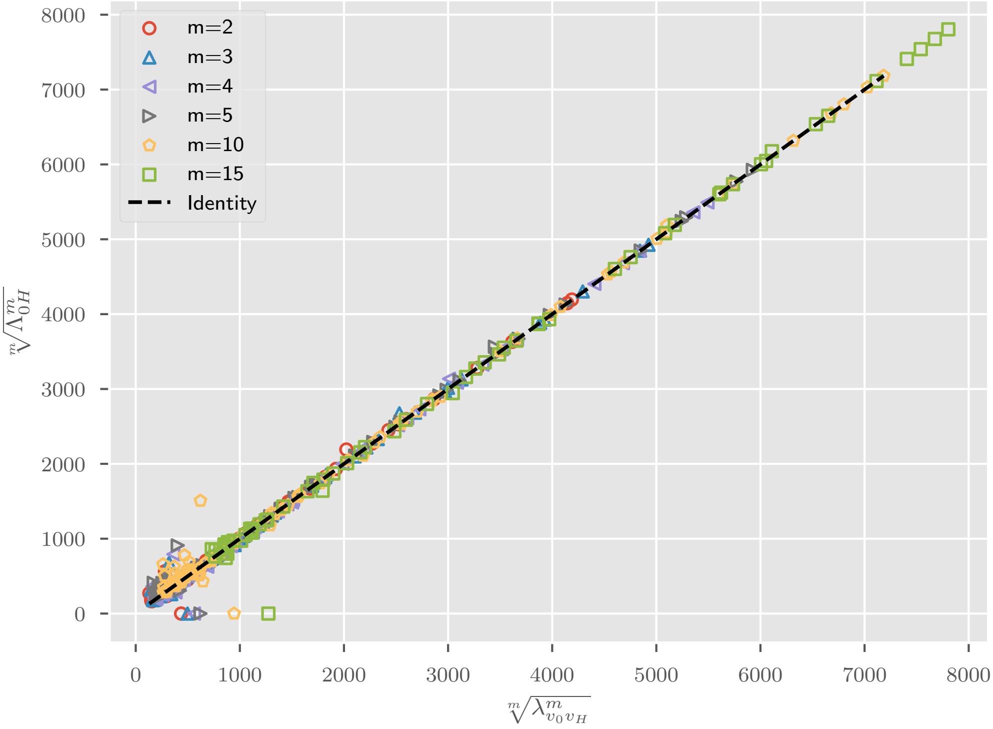


Fig. 2.B.2 *m*-th root-moments of the FPTs of the coarse-grained and the full walker for 200 realisations of edge weights.

Chapter 3

Beyond necklaces: A perturbative error approximation

In chapter 2, we derived an exact equation for MFPTs in graphs with necklace structure, eq. (2.18), and have proven its equivalence with eq. (2.22). We have shown that both equations lead to explicit MFPT formulae when steady-state probabilities are known. This chapter addresses ways to apply the method of chapter 2 to obtain approximate MFPTs when the system under study deviates from this ideal setting. An obvious generalisation consists of a necklace to which additional links with small weights have been added between adjacent clusters. In a slight abuse of nomenclature, we continue referring to the inter-cluster links of the original necklace as the backbone edges. Regarding the newly added edges, as they "bypass" the backbone edges, we refer to them as *leaks*. If the total leakage between two clusters is small compared to the weight of their backbone edge, we expect eq. (2.25) to be approximately true. Our aim is to investigate the error made in that approximation.

The two sides of eq. (2.25), $m_{v_0v_1}$ and M_{01}, are difficult to compare directly using otherwise powerful algebraic methods – the reason being that they are properties of graphs with different sets of nodes. We propose an approach that consists of first mapping the problematic non-necklace to a related necklace, its *necklacification*, with the same set of nodes. A perturbative expansion is then available to compare $m_{v_0v_1}$ and M_{01} to the corresponding quantities of the necklacification. The choice of a necklacification is far from unique; we present one choice that is tailored to simplify the analysis for reversible walkers.

Investigating the sensitivity of Markov chains under perturbations has been an active field of research for a number of decades, focusing mostly on the effects on the steady-state distribution. The earliest two representatives are the references [86, 87], treating perturbations in absolute size and relative to the original hopping probabilities, respectively. An assortment of condition numbers of the steady-state are compared in [88]. [89] is one of the first to study the sensitivity of MFPTs under perturbations. We refer to [90, 88, 89] and references therein for an overview of the results available. Our approach presented here is largely about *constructing* an appropriate approximating Markov chain whose perturbation gives rise to the chain of interest. This method is tailored towards necklace-like graphs and leverage their properties to use matrix-forest theorems, e.g. eqs. (2.11) and (2.14), to our advantage.

The analysis in the present chapter is most appropriate for reversible random walkers, i.e. for non-directed weighted graphs. After introducing the general method (section 3.1), we demonstrate the technique with examples of walkers on undirected (section 3.2) and directed (section 3.3) graphs and uncover a limitation of the method in the latter case. We finish with three numerical studies of our theoretical results in section 3.4.

3.1 Necklacification: General method and linear approximation of error

As in the previous chapter, let $\mathbf{q}$ be the transition matrix of a walker on a graph with two clusters 0 and 1, and a dominant edge (*backbone*) (v_0, v_1) connecting the two. Let $K_1, \ldots, K_{\ell_0}$ and $L_1 \ldots, L_{\ell_1}$ be the *leaking nodes* of the two clusters, respectively, which is to say that each link between 0 and 1 that is not the backbone[1] can be written as (K_r, L_s) for some r and s; these links we call *leaks*.

Next, we define a *necklacification* with transition matrix $\mathbf{q}^0$. At this point we only require that in the necklacification all the leaks are removed, $q^0_{K_r,L_s} = 0 = q^0_{L_s,K_r}$ for all r and s, and that the only non-zero rows of the perturbation $d\mathbf{q} := \mathbf{q} - \mathbf{q}^0$ correspond to the *surface nodes* $v_0, v_1, \{K_1, \ldots, K_{\ell_0}\}$, and $\{L_1, \ldots, L_{\ell_1}\}$. The superscript $-^0$ is used to denote quantities pertaining to a given necklacification throughout this chapter.

Our aim is to interpret eq. (2.25) as an approximation for the MFPT m_{v_0,v_1} by M_{01} on the coarse-grained network (with hopping probabilities defined by the local equilibrium rule eq. (2.9)). To quantify the resulting error, we compare m_{v_0,v_1} and M_{01} to the corresponding quantities, $m^0_{v_0,v_1}$ and M^0_{01}, calculated for the necklacified walker. For the latter, eq. (2.25) manifests as

$$m^0_{v_0 v_1} = M^0_{01} \,, \tag{3.1}$$

and is once again true.

[1] However, for any given leak, either v_0 or v_1 - not both - may participate.

We can now split the error-bound on m_{v_0,v_1} in two upon inserting $0 = M^0_{01} - m^0_{v_0 v_1}$ and using the triangle inequality,

$$\left|m_{v_0,v_1} - M_{01}\right| = \left|m_{v_0,v_1} - m^0_{v_0,v_1} + M^0_{01} - M_{01}\right| \leq \left|m_{v_0,v_1} - m^0_{v_0,v_1}\right| + \left|M_{01} - M^0_{01}\right| \ . \qquad (3.2)$$

This allows us to bound the errors on the right hand side individually. In the following two sections 3.1.1 and 3.1.2, we study these two separately, beginning with the contribution from the "microscopic" dynamics, $m_{v_0 v_1} - m^0_{v_0 v_1}$.

3.1.1 Contribution from microscopic dynamics

By virtue of the necklacification introduced above, we now have two Markov chains on the same state space to compare. In this section, we provide a general perturbative way of doing so. Essentially, we will treat the necklacification as the "ground state" Markov chain (with a benign topology) that is perturbed by a matrix $d\mathbf{q}$, leading to a more complicated topology. For convenience, we denote the inverse matrices $\left(\mathbb{1}_{n-1} - \widehat{\mathbf{q}}_{v_1}\right)^{-1}$ and $\left(\mathbb{1}_{n-1} - \widehat{\mathbf{q}^0}_{v_1}\right)^{-1}$ by $\boldsymbol{G}$ and $\boldsymbol{G}^0$, respectively.

The first summand in eq. (3.2) can be tackled by some basic matrix calculus, using the derivative rule

$$\frac{d}{dt}\boldsymbol{X}^{-1}(t) = -\boldsymbol{X}^{-1}(t)\left(\frac{d}{dt}\boldsymbol{X}\right)(t)\boldsymbol{X}^{-1}(t) \ . \qquad (3.3)$$

Eq. (3.3) allows us to expand the vector of MFPTs of v_1, given by eq. (2.7). There are two conceptually different cases that we formally distinguish: in the first one, we assume that the total leakage ε takes continuous values in an interval starting at 0. In this case, we assume that the perturbation $d\mathbf{q}(\varepsilon)$ is smoothly parametrised by ε such that all entries of $d\mathbf{q}(0)$ vanish. In the second case, we consider leak weights as discrete; we may only control the number of leaks and how many leak edges attach to each leaking node. Again, we should have $d\mathbf{q} = 0$ in the absence of leakage. As both setups can be treated similarly, we provide the details of the former, *continuous-leakage* problem first and state the necessary modifications for the latter problem afterwards.

In the continuous-leakage situation, expanding eq. (2.7) at $\varepsilon = 0$, the zeroth order coefficient of $m_{v_0 v_1}$ is just $m^0_{v_0 v_1}(0)$. Taking the first derivative using eq. (3.3), we obtain

$$\begin{aligned} m_{v_0,v_1}(\varepsilon) - m^0_{v_0,v_1}(\varepsilon) &= -\varepsilon\left[\boldsymbol{G}(0)\left(-\frac{d}{d\varepsilon}\widehat{d\mathbf{q}}_{v_1}(0)\right)\boldsymbol{G}^0(0)\mathbf{1}_{n-1}\right]_{v_0} + R(\varepsilon) \\ &= \varepsilon\left[\boldsymbol{G}(0)\frac{d}{d\varepsilon}\widehat{d\mathbf{q}}_{v_1}(0)\boldsymbol{m}^0_{v_1}(0)\right]_{v_0} + R(\varepsilon) \qquad (3.4) \end{aligned}$$

for the error of the microscopic approximation, where we used eq. (2.7) to rewrite $\boldsymbol{G}^0 \mathbf{1}_{n-1}$ as $\boldsymbol{m}^0_{v_1}$. The remainder of the expansion, $R(\varepsilon)$, is of order $O(\varepsilon^2)$. We deal with R at the end of the section and turn towards rewriting the linear term now.

As stated above, the only non-zero rows of $d\mathbf{q}$ are those corresponding to the surface nodes, for which we find

$$\left[\frac{d}{d\varepsilon}\widehat{d\mathbf{q}}_{v_1}\boldsymbol{m}^0_{v_1}\right]_{v_0} = \sum_{j\in 0}\frac{d}{d\varepsilon}dq_{v_0,j}m^0_{jv_1} + \sum_{s}\frac{d}{d\varepsilon}dq_{v_0,L_s}m^0_{L_s v_1}\,, \tag{3.5}$$

$$\left[\frac{d}{d\varepsilon}\widehat{d\mathbf{q}}_{v_1}\boldsymbol{m}^0_{v_1}\right]_{K_r} = \sum_{j\in 0}\frac{d}{d\varepsilon}dq_{K_r,j}m^0_{jv_1} + \sum_{s}\frac{d}{d\varepsilon}dq_{K_r,L_s}m^0_{L_s v_1}\,, \tag{3.6}$$

and

$$\left[\frac{d}{d\varepsilon}\widehat{d\mathbf{q}}_{v_1}\boldsymbol{m}^0_{v_1}\right]_{L_s} = \sum_{j\in 1}\frac{d}{d\varepsilon}dq_{L_s,j}m^0_{jv_1} + \sum_{r}\frac{d}{d\varepsilon}dq_{L_sK_r}m^0_{K_r v_1} + \frac{d}{d\varepsilon}dq_{L_s v_0}m^0_{v_0 v_1}\,. \tag{3.7}$$

Moreover, one can use a matrix forest theorem (related to the one used in the proof in section 2.3) to express the elements of $\boldsymbol{G}^0 = \left(\mathbb{1}_{n-1} - \widehat{\mathbf{q}^0}_{v_1}\right)^{-1}$ in terms of spanning forest weights [82]: given two nodes i and j, let $\{\mathfrak{f}\,|\,\mathrm{roots}(\mathfrak{f}) = \{v_1, j\}, i \to j\}$ be the set of all spanning forests with two trees rooted in v_1 and j such that i and j are in the same tree. For our convention of rooted spanning forests, see appendix 2.A. We then have

$$\left(\mathbb{1}_{n-1} - \widehat{\mathbf{q}^0}_{v_1}\right)^{-1}_{ij} = \frac{1}{w^0(v_1)} \sum_{\substack{\mathfrak{f};\, i\to j,\\ \mathrm{roots}(\mathfrak{f})=\{v_1,j\}}} w^0(\mathfrak{f}) \tag{3.8}$$

wherein

$$w^0(v_1) := \sum_{\substack{\mathfrak{t};\\ \mathrm{roots}(\mathfrak{t})=\{v_1\}}} w^0(\mathfrak{t})\,, \tag{3.9}$$

with $w^0(\cdot)$ denoting the weight of a tree or forest consisting of edges of the necklacification.

As we are estimating the error for the MFPT between v_0 and v_1, we only need to obtain the entries of this matrix for $i = v_0$. For the v_0-th row, it is easy to see that all L_s elements have to vanish: as the only connection between clusters 0 and 1 in the necklacified network is the backbone (v_0, v_1), no spanning forests exists with roots v_1 and L_s such that v_0 and L_s are in the same component[2]. For this reason, and because the non-zero rows $\widehat{d\boldsymbol{q}}$ correspond

[2] Alternatively, $\widehat{\mathbf{q}}^0$ is block-diagonal – with blocks given by the partition of the nodes into their clusters – as the only connection between the clusters in the necklacification is encoded in the v_1-th row and column of $\mathbf{q}^0$. Therefore $\left(\mathbb{1} - \widehat{\mathbf{q}}^0\right)^{-1}$ is block-diagonal.

exactly to the surface nodes, it suffices to find the v_0, v_0 and v_0, K_r entries of the matrix $\boldsymbol{G}^0$. Due to the presence of the essential edge (v_0, v_1), the spanning forest and tree weights must factorise

$$\begin{aligned} w^0(v_1) &= w^0(0 \to v_0) q_{v_1 v_0} w^0(1 \to v_1) \ , \\ w^0(\mathfrak{f}\colon \text{roots}(\mathfrak{f}) = v_1, K_r,\ v_0 \to K_r) &= w^0(0 \to K_r) w^0(1 \to v_1) \ , \end{aligned} \tag{3.10}$$

using the notation for weights of sub-spanning trees of the clusters defined in section 2.3 above eq. (2.16). The above factorisation and the tree-formula for the steady-state probabilities in eq. (2.14) directly imply that

$$\boldsymbol{G}^0_{v_0,K_r} = \frac{w^0(0 \to K_r)}{w^0(0 \to v_0) q^0_{v_0 v_1}} = \frac{\pi^0_{K_r}}{\pi^0_{v_0} q^0_{v_0 v_1}} \ . \tag{3.11}$$

Similarly, we obtain

$$\boldsymbol{G}^0_{v_0,v_0} = \frac{w^0(0 \to v_0)}{w^0(0 \to v_0) q^0_{v_0 v_1}} = \frac{1}{q^0_{v_0 v_1}} \tag{3.12}$$

for the v_0, v_0 element. Finally, on the necklacified graph, the first-passage processes starting anywhere within cluster 0 must pass through v_0, i.e. for all $j \in 0 \setminus \{v_0\}$ we have $m^0_{j,v_1} = m^0_{j,v_0} + m^0_{v_0,v_1}$. In summary, the above can be combined to yield (all quantities evaluated at $\varepsilon = 0$)

$$\begin{aligned} &\left[\boldsymbol{G} \frac{d}{d\varepsilon} \widehat{d\mathbf{q}}_{v_1} \boldsymbol{G}^0 \mathbf{1}_{n-1}\right]_{v_0} \\ =& \frac{1}{q^0_{v_0 v_1}} \left[\frac{d}{d\varepsilon} \widehat{d\mathbf{q}}_{v_1} \boldsymbol{m}^0_{v_1}\right]_{v_0} + \sum_r \frac{\pi^0_{K_r}}{\pi^0_{v_0} q^0_{v_0 v_1}} \left[\frac{d}{d\varepsilon} \widehat{d\mathbf{q}}_{v_1} \boldsymbol{m}^0_{v_1}\right]_{K_r} \\ =& \sum_r \frac{\pi^0_{K_r}}{\pi^0_{v_0} q^0_{v_0 v_1}} \left[\sum_{j \in 0 \setminus \{v_0\}} \frac{d}{d\varepsilon} dq_{K_r,j} \left(m^0_{j v_0} + m^0_{v_0,v_1}\right) + \frac{d}{d\varepsilon} dq_{K_r,v_0} m^0_{v_0,v_1} + \sum_s \frac{d}{d\varepsilon} dq_{K_r,L_s} m^0_{L_s v_1}\right] \\ &+ \frac{1}{q^0_{v_0 v_1}} \left[\sum_{j \in 0 \setminus \{v_0\}} \frac{d}{d\varepsilon} dq_{v_0,j} \left(m^0_{j v_0} + m^0_{v_0,v_1}\right) + \frac{d}{d\varepsilon} dq_{v_0,v_0} m^0_{v_0,v_1} + \sum_s \frac{d}{d\varepsilon} dq_{v_0,L_s} m^0_{L_s v_1}\right] \ , \end{aligned} \tag{3.13}$$

giving the coefficient for the first order in ε of $m_{v_0 v_1} - m^0_{v_0 v_1}$. For $d\mathbf{q}$ based on discrete leakage, we use eq. (3.3) by setting $\boldsymbol{G}(\varepsilon) = \left(\mathbb{1}_{n-1} - \widehat{\mathbf{q}}^0_{v_1} - \varepsilon \cdot \widehat{d\mathbf{q}}_{v_1}\right)^{-1}$. This leads to the same expressions as just given, with $\frac{d}{d\varepsilon} dq_{ij}$ replaced by dq_{ij}.

For the microscopic contribution to eq. (3.2), we are only left to discuss the remainder, R, in the first order expansion in eq. (3.4). As exploited in previous steps, the error $m_{v_0v_1} - m^0_{v_0v_1}$ is given by the v_0-entry of $\left(\boldsymbol{G} - \boldsymbol{G}^0\right)\mathbf{1}_{n-1}$ for which we can easily compute the first and second derivatives as

$$\frac{d}{d\varepsilon}(\boldsymbol{G} - \boldsymbol{G}^0) = \boldsymbol{G}\left(\frac{d}{d\varepsilon}\mathbf{q}\right)\boldsymbol{G} - \boldsymbol{G}^0\left(\frac{d}{d\varepsilon}\mathbf{q}^0\right)\boldsymbol{G}^0\,, \tag{3.14}$$

$$\begin{aligned}\frac{d^2}{d\varepsilon^2}(\boldsymbol{G} - \boldsymbol{G}^0) =& \boldsymbol{G}\left(\frac{d^2}{d\varepsilon^2}\mathbf{q}\right)\boldsymbol{G} + 2\boldsymbol{G}\left(\frac{d}{d\varepsilon}\mathbf{q}\right)\boldsymbol{G}\left(\frac{d}{d\varepsilon}\mathbf{q}\right)\boldsymbol{G} - \boldsymbol{G}^0\left(\frac{d^2}{d\varepsilon^2}\mathbf{q}^0\right)\boldsymbol{G}^0\\ &- 2\boldsymbol{G}^0\left(\frac{d}{d\varepsilon}\mathbf{q}^0\right)\boldsymbol{G}^0\left(\frac{d}{d\varepsilon}\mathbf{q}^0\right)\boldsymbol{G}^0\,.\end{aligned} \tag{3.15}$$

To obtain the corresponding remainder, we need to compute this previous second derivative for general $\varepsilon \neq 0$. However, doing so requires explicit knowledge of $\boldsymbol{G}$ first, at which point we could as well have used eq. (2.7) to calculate $m_{v_0v_1}$ without any approximations. We have, however, full knowledge of $\mathbf{q}$ and assume that we can do the easier computations relating to $\boldsymbol{G}^0$. We therefore propose to replace all $\boldsymbol{G}$'s by $\boldsymbol{G}^0$'s to approximate the equation above, while keeping the correct $\mathbf{q}$'s and $\mathbf{q}^0$'s. This way we can identify some of the summands above, arriving at the simplified expression

$$\begin{aligned}\frac{d^2}{d\varepsilon^2}(\boldsymbol{G} - \boldsymbol{G}^0) \approx& \boldsymbol{G}^0\left(\frac{d^2}{d\varepsilon^2}d\mathbf{q}\right)\boldsymbol{G}^0\\ &+ 2\boldsymbol{G}^0\left[\left(\frac{d}{d\varepsilon}\mathbf{q}\right)\boldsymbol{G}^0\left(\frac{d}{d\varepsilon}\mathbf{q}\right) - \left(\frac{d}{d\varepsilon}\mathbf{q}^0\right)\boldsymbol{G}^0\left(\frac{d}{d\varepsilon}\mathbf{q}^0\right)\boldsymbol{G}^0\right]\boldsymbol{G}^0\,.\end{aligned} \tag{3.16}$$

We now hold all the ingredients of the remainder term $R(\varepsilon)$. As usual, if we can bound $\left|\left(\frac{d^2}{d\varepsilon^2}(\boldsymbol{G} - \boldsymbol{G}^0)\mathbf{1}_{n-1}\right)_{v_0}\right|$ (or its approximation) by a constant C, we can write down a (conservative) bound

$$|m_{v_0,v_1}(\varepsilon) - m^0_{v_0,v_1}(\varepsilon)| \leq \varepsilon\left|\left[\boldsymbol{G}^0(0)\frac{d}{d\varepsilon}\widehat{d\mathbf{q}}_{v_1}(0)\boldsymbol{m}^0_{v_1}(0)\right]_{v_0}\right| + \frac{\varepsilon^2}{2}C\,. \tag{3.17}$$

As in the previous paragraph under eq. (3.13), the version for unparametrised leaks is obtained by replacing the derivative of $\widehat{d\mathbf{q}}_{v_1}$ with $\widehat{d\mathbf{q}}_{v_1}$ itself.

As a final remark in this section, we recall that the MFPTs $\boldsymbol{m}^0_{v_1}$ must decompose into $m^0_{v_{0i}v_1} = m^0_{v_{0i}v_0} + m^0_{v_0v_0}$ for all nodes v_{0i} in cluster 0 as a consequence of the necklacification. For the intra-cluster MFPTs, approximations such as the one discussed in [75] can provide useful estimates for $m^0_{v_{0i}v_0}$ and $m^0_{v_{1j}v_1}$, provided the spectrum of $\widehat{\mathbf{q}^0}_{v_1}$ has a large gap between its largest two eigenvalues.

We continue our treatment of eq. (3.2) with the contribution from the coarse-grained dynamics.

3.1.2 Contribution from macroscopic dynamics

For completeness, we include a general discussion for the second summand of the right hand side of the inequality (3.2). In full generality, however, it is of limited use since not all involved quantities can be bounded in the required way. Recall from section 2.5.2 that in the coarse-grained systems with two clusters the MFPTs are given by

$$\begin{aligned} M_{01}^0 =& \frac{1}{Q_{01}^0} = \frac{\Pi_0^0}{\pi_{v_0}^0 q_{v_0 v_1}^0} \,, \\ M_{01} =& \frac{1}{Q_{01}} = \frac{\Pi_0}{\sum_{i\in 0, j\in 1} \pi_i q_{ij}} = \frac{\Pi_0}{\pi_{v_0} q_{v_0 v_1} + \sum_{r,s} \pi_{K_r} q_{K_r L_s}} \,. \end{aligned} \tag{3.18}$$

This implies that the difference between the two macroscopic MFPTs is given by

$$\begin{aligned} M_{01} - M_{01}^0 = & \Bigg\{ d\Pi_0 \pi_{v_0}^0 q_{v_0 v_1}^0 - \Pi_0^0 \sum_{r,s} \pi_{K_r}^0 q_{K_r L_s} \\ & -\Pi_0^0 \left[d\pi_{v_0} dq_{v_0 v_1} + \pi_{v_0}^0 dq_{v_0 v_1} + d\pi_{v_0} q_{v_0 v_1}^0 + \sum_{r,s} d\pi_{K_r} q_{K_r L_s} \right] \Bigg\} \\ & \times \left[\pi_{v_0}^0 q_{v_0 v_1}^0 \left(\pi_{v_0} q_{v_0 v_1} + \sum_{r,s} \pi_{K_r} q_{K_r L_s} \right) \right]^{-1} \,. \end{aligned} \tag{3.19}$$

Knowing the relationship between $\mathbf{q}$ and $\mathbf{q}^0$, we thus have to quantify the differences $d\Pi := \Pi - \Pi^0$ and $d\boldsymbol{\pi} := \boldsymbol{\pi} - \boldsymbol{\pi}^0$ in order to compare M_{01} and M_{01}^0.

Calculating $d\boldsymbol{\pi}$ exactly can be challenging. However, many results exist bounding its norm by the norm of the perturbation $d\mathbf{q}$ [88]

$$d\pi := ||d\boldsymbol{\pi}|| \leq k\left(\mathbf{q}^0\right) ||d\mathbf{q}|| \,, \tag{3.20}$$

where k is a suitable condition number, depending on the necklacification, and $||-||$ is any norm. If $||-||$ is the maximum-norm, it clearly follows that

$$d\Pi_0 := ||d\Pi_0|| \leq |\mathbf{0}| \cdot k\left(\mathbf{q}^0\right) ||d\mathbf{q}|| \,, \tag{3.21}$$

yielding the bound

$$\begin{aligned}|M_{01}-M^0_{01}| \leq \Bigg\{ & q^0_{v_0v_1}\left(d\Pi_0\pi^0_{v_0}+\Pi^0_0 d\pi\right)+\Pi^0_0\sum_{r,s}\pi^0_{K_r}q_{K_rL_s} \\ & +\Pi^0_0\left[d\pi dq_{v_0v_1}+\pi^0_{v_0}dq_{v_0v_1}+\sum_{r,s}d\pi q_{K_rL_s}\right]\Bigg\} \\ & \times\left[\pi^0_{v_0}q^0_{v_0v_1}\left[\pi_{v_0}q_{v_0v_1}+\sum_{r,s}\pi_{K_r}q_{K_rL_s}\right]\right]^{-1} . \end{aligned} \tag{3.22}$$

At this point, we require a lower bound for either $d\pi$ or $d\Pi$ to use in the denominator. In general, such a (non-trivial) bound can't exist, as remarked for birth-death processes by [91].

In the next section, we apply the results of this discussion to a useful necklacification for non-directed graphs. This necklacification has the desirable property that $M_{01} = M^0_{01}$, such that eq. (3.17) is the only bound that we need to consider. In fact, it satisfies $\Pi = \Pi^0$, and therefore provides another example for a situation in which the optimal lower bound is the trivial one, $0 \leq d\Pi$.

3.2 Reversible chains

For reversible walkers, the difficulty outlined at the end section 3.1.2 can be easily avoided. In fact, provided the walker is reversible, we can construct necklacifications $\mathbf{q}^0$ that have the same coarse-grained MFPT M^0_{01} as $\mathbf{q}$. In this section, we show by example how to construct this necklacification and the error calculation that follows.

For ease of presentation, we consider a simple example where we could even do exact calculations. It is a prototype model for block-model situations where one has densely connected communities that are only weakly interconnected. Thus, let us consider two clusters 0 and 1 formed by fully connected subgraphs (cliques) of size n and m, respectively. Let the clusters be connected by the backbone (v_0, v_1) with unit weight and a single leak (v_{0K}, v_{1L}) weighted ε, and let all other edges in the graph carry unit weight. A sketch of this graph with $m = n = 5$ is given in the upper diagram of fig. 3.2.1.

Clearly, all nodes within the bulk of cluster 0, i.e. $0 \setminus \{v_0, v_{0K}\}$, are interchangeable. For this reason, we can aid the calculation by inserting an additional coarse-graining step: we replace the bulk by a single node, say v_{0i}, with a weighted self-loop. If the edges connecting to v_{0i} are to represent the correct dynamics of the walker entering, leaving or

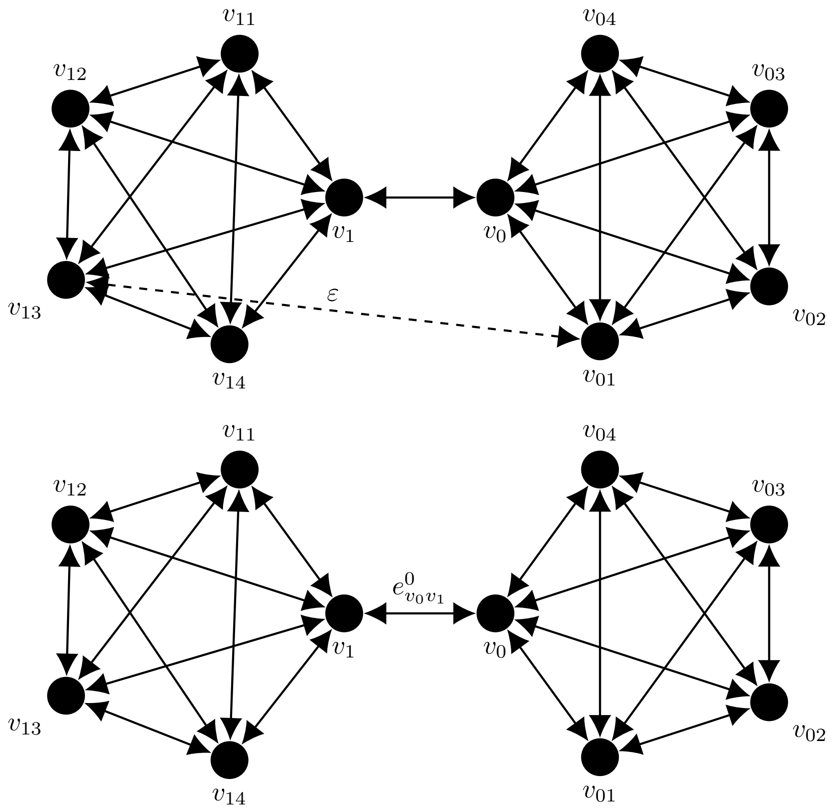


Fig. 3.2.1 Necklacification (bottom) of an undirected graph (top). The walkers on both graphs have the same coarse-grained stationary distributions ($\Pi = \Pi^0$) and coarse-grained MFPT ($M_{01} = M^0_{01}$) if $e^0_{01} = 1 + \varepsilon$. Unlabelled edges have unit weight.

remaining in the bulk, we naturally arrive at the transition matrix

$$\mathbf{q} = \begin{pmatrix} 0 & \frac{1}{n} & \frac{n-2}{n} & 0 & 0 & \frac{1}{n} \\ \frac{1}{n-1+\varepsilon} & 0 & \frac{n-2}{n-1+\varepsilon} & \frac{\varepsilon}{n-1+\varepsilon} & 0 & 0 \\ \frac{1}{n-1} & \frac{1}{n-1} & \frac{n-3}{n-1} & 0 & 0 & 0 \\ 0 & \frac{\varepsilon}{m-1+\varepsilon} & 0 & 0 & \frac{m-2}{m-1+\varepsilon} & \frac{1}{n-1+\varepsilon} \\ 0 & 0 & 0 & \frac{1}{m-1} & \frac{m-3}{m-1} & \frac{1}{m-1} \\ \frac{1}{m} & 0 & 0 & \frac{1}{m} & \frac{m-2}{m} & 0 \end{pmatrix} . \tag{3.23}$$

Here, we have made the same replacement of the bulk nodes of cluster 1 by the representative v_{1j}, and consider nodes in the order $v_0, v_{0K}, v_{0i}, v_{1L}, v_{1j}, v_1$.

For future reference, we apply eq. (2.7) to obtain, with the help of Mathematica, the true MFPT

$$m_{v_0v_1} = \frac{4\varepsilon^2 n + mn(n^2-n+1) + \varepsilon(m^2 n + 2n(n^2-n+1) + m(n^2-n+2))}{2\varepsilon(n+m) + nm(1+\varepsilon)} . \tag{3.24}$$

The steady-state probabilities are simply proportional to the *node strengths* (the sum of weights connecting a given node, self-loops counted once only), such that the LE-approximated quantity can be derived with ease[3]

$$M_{01} = \frac{n(n-1)+1+\varepsilon}{1+\varepsilon}. \tag{3.25}$$

For this model, we now introduce a useful necklacification that admits obvious generalisations.

3.2.1 Necklacification with preserved coarse-grained MFPT

A necklacification for which the equality $M_{01} = M^0_{01}$ holds is given in the bottom diagram in fig. 3.2.1: one simply removes the leak and adds its weight to the backbone weight. Symbolically, this mapping amounts to

$$e^0_{v_0v_1} := e_{v_0v_1} + e_{v_{0K}v_{1L}}, \quad e^0_{v_{0K}v_{1L}} := 0 \tag{3.26}$$

and all other edge weights remain unchanged. To see why this mapping preserves M_{01}, let e_{ij} denote the weight of the edge (i,j) (setting $e_{ij} = 0$ if the edge (i,j) does not exist). Using again that stationary probabilities are proportional to the node strengths, we observe that $\Pi^0 = \Pi$ since

$$\Pi_0 = \frac{1}{Z}\sum_{i\in 0} k_i = \frac{1}{Z}\left(\sum_{i,j\in 0} e_{ij} + \varepsilon + 1\right). \tag{3.27}$$

Here, the summand ε arises from the leak (v_{0K}, v_{1L}). For the necklacified walker, we obtain the same expression, ε now being a summand in $w^0_{v_0,v_1} = 1+\varepsilon$, while no other edge weights change. Similarly, the probability flowing out of 0 is the same for both models, as

$$\sum_{\substack{i\in 0,\\ j\in 1}} \pi_i q_{ij} = \frac{1}{Z}(1+\varepsilon) = \sum_{\substack{i\in 0,\\ j\in 1}} \pi^0_i q^0_{ij} \tag{3.28}$$

by construction.

This way we can show the equality for M_{01} and M^0_{01} of the necklacified walker,

$$M^0_{01} = \frac{\Pi^0_0}{\pi^0_{v_0} q^0_{v_0,v_1}} = \frac{\Pi_0}{\pi_{v_0} q_{v_0,v_1} + \pi_{v_{0K}} q_{v_{0K},v_{1L}}} = M_{01}\ . \tag{3.29}$$

[3] In 0, there is one node with strength n, one with strength $n-1+\varepsilon$ and $n-2$ nodes with strength $n-1$ each.

Moreover, a straightforward generalisation of the above argument shows that $M^0_{01} = M_{01}$ for the analogous necklacification with an *arbitrary number* of leaks, where all leaks have been removed and their total weight is added to the backbone.

We may write the necklacification of our example in terms of the transition matrix

$$\mathbf{q}^0 = \begin{pmatrix} 0 & \frac{1}{n+\varepsilon} & \frac{n-2}{n+\varepsilon} & 0 & 0 & \frac{1+\varepsilon}{n+\varepsilon} \\ \frac{1}{n-1} & 0 & \frac{n-2}{n-1} & 0 & 0 & 0 \\ \frac{1}{n-1} & \frac{1}{n-1} & \frac{n-3}{n-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{m-2}{m-1} & \frac{1}{m-1} \\ 0 & 0 & 0 & \frac{1}{m-1} & \frac{m-3}{m-1} & \frac{1}{m-1} \\ \frac{1+\varepsilon}{m+\varepsilon} & 0 & 0 & \frac{1}{m+\varepsilon} & \frac{m-2}{m+\varepsilon} & 0 \end{pmatrix} \tag{3.30}$$

or, given in the form of a perturbation,

$$d\mathbf{q} = \varepsilon \cdot \begin{pmatrix} 0 & \frac{1}{n(n+\varepsilon)} & \frac{n-2}{n(n+\varepsilon)} & 0 & 0 & -\frac{n-1}{n(n+\varepsilon)} \\ -\frac{1}{(n-1)(n-1+\varepsilon)} & 0 & -\frac{n-2}{(n-1)(n-1+\varepsilon)} & \frac{1}{n-1+\varepsilon} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m-1+\varepsilon} & 0 & 0 & -\frac{m-2}{(m-1)(m-1+\varepsilon)} & -\frac{1}{(m-1)(m-1+\varepsilon)} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{m-1}{m(m+\varepsilon)} & 0 & 0 & \frac{1}{m(m+\varepsilon)} & \frac{m-2}{m(m+\varepsilon)} & 0 \end{pmatrix} . \tag{3.31}$$

It is straightforward to apply eq. (3.13) to this model by using the derivatives of $d\mathbf{q}$ in eq. (3.31), given by

$$dq'_{v_0 v_{0K}}(0) = \frac{1}{n^2} \, , \quad dq'_{v_0 v_{0i}}(0) = \frac{n-2}{n^2} \, , \tag{3.32}$$

$$dq'_{v_{0K} v_0}(0) = -\frac{1}{(n-1)^2} \, , \quad dq'_{v_{0K} v_{0i}}(0) = -\frac{n-2}{(n-1)^2} \, , \quad dq'_{v_{0K} v_{1L}}(0) = \frac{1}{n-1} \, , \tag{3.33}$$

and the following quantities for the present necklacification

$$\frac{\pi^0_{v_{0K}}}{\pi^0_{v_0}} = \frac{n-1}{n+\varepsilon} \, , \tag{3.34}$$

$$m^0_{v_{0i} v_0} = m^0_{v_{0K} v_0} = n-1 \, , \tag{3.35}$$

$$m^0_{v_{1L} v_1} = m-1 \, . \tag{3.36}$$

These quantities substituted into eq. (3.13) we obtain

$$m_{v_0 v_1}(\varepsilon) - m^0_{v_0 v_1}(\varepsilon) = \varepsilon(m-n) + O(\varepsilon^2) \tag{3.37}$$

to first order in ε.

An analogous calculation allows us to generalise to the case with multiple leaking nodes $K_r \in 0$ and $L_s \in 1$ weighted by discrete non-negative numbers $e_{K_r L_s}$. We begin by only assuming that the clusters are close to fully connected, such that $m_{v_{0i}v_0} \approx n-1$, $m_{v_{1j}v_1} \approx m-1$, but allow the node degrees k_v^0 to vary slightly. This makes the formula more widely applicable. We introduce the following symbols for different sums of leakage terms

$$\ell_{K_r} = \sum_s e_{K_r L_s} \ , \tag{3.38}$$

$$\tilde{\ell}_{K_r} = \frac{\ell_{K_r} k^0_{K_r}}{\ell_{K_r} + k^0_{K_r}} \ , \tag{3.39}$$

$$\ell = \sum_r \ell_{K_r} \ , \tag{3.40}$$

$$\tilde{\ell} = \sum_r \tilde{\ell}_{K_r} \ , \tag{3.41}$$

$$\tilde{\ell}_{\partial v_0} = \sum_{K_r \in \partial v_0} \frac{\ell_{K_r} k^0_{K_r}}{\ell_{K_r} + k^0_{K_r}} \ . \tag{3.42}$$

In this notation, error to first order reads

$$\left[\boldsymbol{G}^0 \widehat{d\mathbf{q}} \boldsymbol{G}^0 \mathbf{1}_{n-1}\right]_{v_0} \approx \frac{m^0_{v_0 v_1}}{1+\ell}\left[\frac{\ell(k^0_{v_0}-1)}{k^0_{v_0}} - \tilde{\ell}\right] + \frac{n-1}{1+\ell}\left[\frac{\ell(k^0_{v_0}-1)}{k^0_{v_0}} + \tilde{\ell}_{\partial v_0}\right] + \frac{\tilde{\ell}}{1+\ell}\left[m-n\right] \ , \tag{3.43}$$

where $m^0_{v_0 v_1} \approx 1 + \frac{n(n-1)}{1+\ell}$. If the clusters are complete (sub)graphs, then all degrees are given by $k^0_{v_0} = n$, $k^0_{v_{0i}} = n-1$, $k^0_{v_{1j}} = m-1$ and $k^0_{v_1} = m$, and the above formula reduces to

$$\left[\boldsymbol{G}^0 \widehat{d\mathbf{q}} \boldsymbol{G}^0 \mathbf{1}_{n-1}\right]_{v_0} = \frac{m^0_{v_0 v_1}}{1+\ell}\left[\frac{\ell(n-1)}{n} - \tilde{\ell}\right] + \frac{n-1}{1+\ell}\left[\frac{\ell(n-1)}{n} + \tilde{\ell}\right] + \frac{\tilde{\ell}}{1+\ell}\left[m-n\right] \ , \tag{3.44}$$

which is an exact equality.

Upon setting the number of leaks to a single one with weight ε, we retrieve the result for small ε, eq. (3.37), when differentiating with respect to ε at 0. It is possible to calculate the remainder of the expansion in $d\mathbf{q}$ if the spectra of $\mathbf{q}$ and $\mathbf{q}^0$ are known; we do not follow this route, but refer to [92] for a general treatment. In numerical experiments later in this chapter, we make liberal use of eq. (3.43), especially in section 3.4.3, where the distinction between leak and backbone is merely formal as all edges have unit weight.

From a close look at eqs. (3.43) and (3.44), we notice that – contrary to our initial hopes – the error does necessarily approach 0 as n and m increase, unless the leak weights are scaled down accordingly. The reason for this is that the probability of following a leak and the backbone scale in the same way with n, so the graph does not approach a necklace as n increases. However, as $m^0_{v_0v_1} = 1 + \frac{n(n-1)}{1+\ell}$ is still dominated by n, the *relative error* $\left(m_{v_0v_1} - m^0_{v_0v_1}\right)/m_{v_0v_1}$ approaches 0 as n increases.

Returning to the present case with one leak, we are to determine the remainder via eq. (3.16). Aided by Mathematica, we find

$$\frac{d^2}{d\varepsilon^2}\left(m_{v_0v_1} - m^0_{v_0v_1}\right)(\varepsilon) \approx -\frac{2(n-1)N_\varepsilon}{(\varepsilon+1)^3 mn(\varepsilon+m-1)^2(\varepsilon+n-1)^4}\ , \tag{3.45}$$

where the factor N_ε in the numerator is given by

$$\begin{aligned}
&N_\varepsilon\\
&=(\varepsilon+1)m^4(n-1)\left(2(\varepsilon+1)+n^3+2\varepsilon n^2+\left(\varepsilon^2-3\varepsilon-3\right)n\right)\\
&\quad+m^3\Big[-4(\varepsilon-1)(\varepsilon+1)^2+(\varepsilon-3)n^5+\left(7\varepsilon^2-2\varepsilon+9\right)n^4+\left(11\varepsilon^3-6\varepsilon^2-2\varepsilon-13\right)n^3\\
&\quad+\left(5\varepsilon^4-17\varepsilon^3-15\varepsilon^2+7\varepsilon+15\right)n^2-2\left(2\varepsilon^4-5\varepsilon^3-9\varepsilon^2+4\varepsilon+6\right)n\Big]\\
&\quad+m^2\Big[-2\left(\varepsilon^2-1\right)^2-2(\varepsilon+1)n^6-4\left(\varepsilon^2+2\varepsilon-3\right)n^5+\left(\varepsilon^3-19\varepsilon^2+21\varepsilon-31\right)n^4\\
&\quad+\left(6\varepsilon^4-29\varepsilon^3+23\varepsilon^2-11\varepsilon+43\right)n^3+\left(3\varepsilon^5-20\varepsilon^4+23\varepsilon^3+23\varepsilon^2+4\varepsilon-33\right)n^2\\
&\quad-\left(\varepsilon^5-14\varepsilon^4-5\varepsilon^3+27\varepsilon^2+4\varepsilon-13\right)n\Big]\\
&\quad+mn\Big[-2(\varepsilon-1)^2\left(4\varepsilon^2+9\varepsilon+5\right)+4(\varepsilon+1)n^5+\left(\varepsilon^3+7\varepsilon^2+7\varepsilon-15\right)n^4\\
&\quad+\left(3\varepsilon^4+18\varepsilon^2-20\varepsilon+31\right)n^3+\left(3\varepsilon^5-8\varepsilon^4+29\varepsilon^3-25\varepsilon^2+4\varepsilon-43\right)n^2\\
&\quad+\left(\varepsilon^6-5\varepsilon^5+14\varepsilon^4-28\varepsilon^3-18\varepsilon^2+3\varepsilon+33\right)n\Big]\\
&\quad-2(\varepsilon+1)(n-1)n(\varepsilon+n-1)^2\left(2\varepsilon+n^2+\varepsilon n+2\right)\ .
\end{aligned} \tag{3.46}$$

While this expression is not very handy, by plotting it for various values of n and m we observe that it is maximal at $\varepsilon = 0$, where it attains the value

$$\frac{d^2}{d\varepsilon^2}\left(m_{v_0v_1} - m^0_{v_0v_1}\right)(0) \approx \frac{4\left(n^2+2\right)}{m} - \frac{2m(n+2)}{n} + 6n - 4\ . \tag{3.47}$$

Thus, we arrive at the result that

$$|m_{v_0v_1} - m^0_{v_0v_1}| \lessapprox |m-n|\varepsilon + \left|\frac{4\,(n^2+2)}{m} - \frac{2m(n+2)}{n} + 6n - 4\right|\frac{\varepsilon^2}{2}\,. \tag{3.48}$$

Fig. 3.2.2 shows this last expression in comparison with the true error. We can see that, indeed, eq. (3.48) provides an upper bound to the LE coarse-graining error $|m_{v_0v_1} - M_{01}|$, and indeed a good approximation for the error at low values of ε.

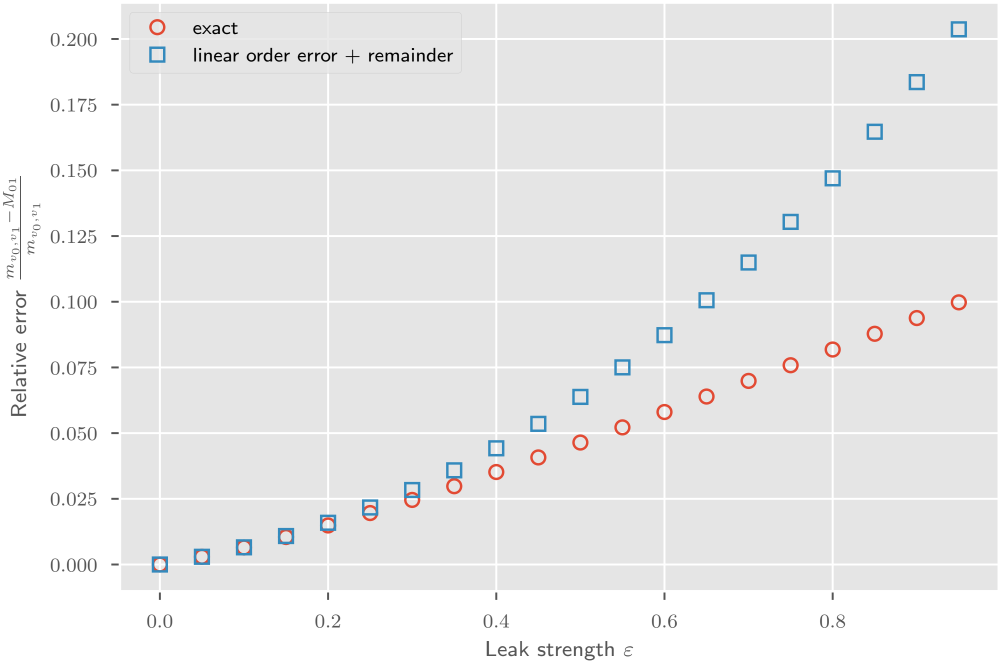


Fig. 3.2.2 Comparison of eq. (3.48) to the true error $|m_{v_0,v_1} - M_{01}|$. In this example, we set $n = 20, m = 40$.

We will test the above formulae in different incarnations in section 3.4.

3.3 Irreversible chains: Problems induced by directed leaks

By way of example, consider the directed graph at the top of fig. 3.3.1, consisting of two cliques of equal size n, that we will regard as cluster 0 and 1, respectively. As before, we assume that there is a backbone-like edge represented by the directed arcs (v_0, v_1) and (v_1, v_0) with unit weight. In addition, each vertex v_{0i} in cluster 0 is paired with a unique vertex v_{1i} in cluster 1 by two directed edges, (v_{0i}, v_{1i}) and (v_{1i}, v_{0i}), with weight ε and δ, respectively, which are assumed to be small. In this example, we lose the symmetry given by the undirected graph, but we have introduced another by turning all nodes into surface nodes and by giving identical weights to all leaks. Thus, there are four classes of nodes, represented by v_0, v_{01}, v_1, v_{11}, and the nodes in each class share the same steady state probability. To compute $\boldsymbol{\pi}$, it is then convenient to use a reduced representation of the dynamics, in terms of classes (rather than nodes). To this purpose, we note that the hopping probability between nodes of different classes is given by the 4×4 matrix

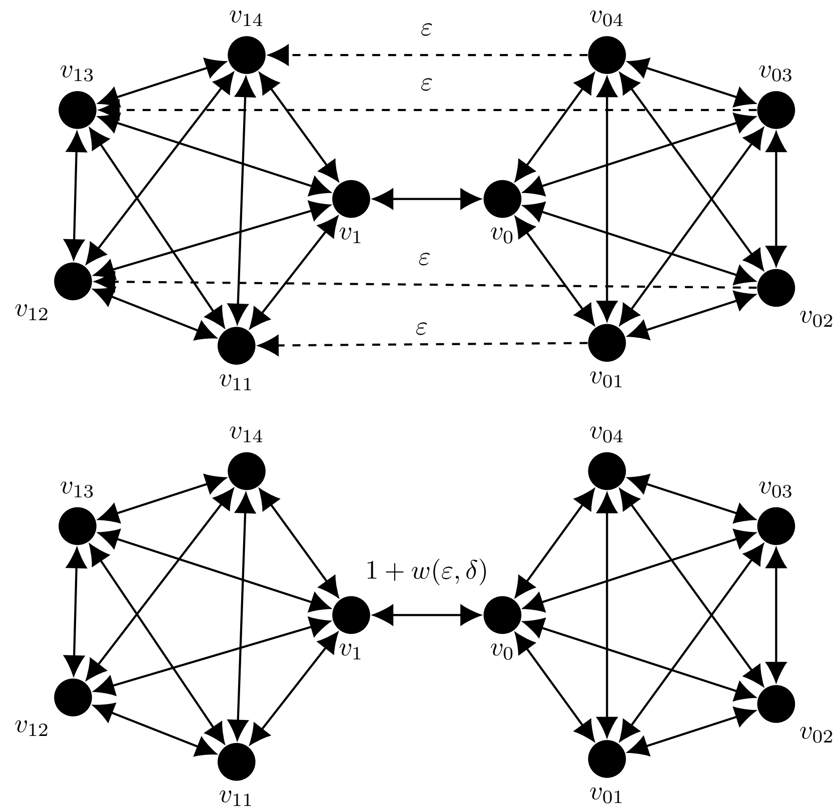


Fig. 3.3.1 Two cliques $0 = \{v_0, v_{01}, v_{02}, v_{03}, v_{04}\}$ and $1 = \{v_1, v_{11}, v_{12}, v_{13}, v_{14}\}$ of size $n = 5$, connected to the backbone v_0, v_1. Top: Additional directed edges (v_{0i}, v_{1i}) with weight ε ("leaks", dashed) interconnect the cliques; for every edge with weight ε the reverse edge has weight δ. Unlabelled edges carry unit weight. Bottom: Necklacification of the top graph, where the leaks have been absorbed into an increment w of the backbone weight.

$$\mathsf{q}^{\text{red}} = \begin{pmatrix} 0 & \frac{1}{n} & \frac{1}{n} & 0 \\ \frac{1}{n-1+\varepsilon} & 0 & 0 & \frac{\varepsilon}{n-1+\varepsilon} \\ \frac{1}{n} & 0 & 0 & \frac{1}{n} \\ 0 & \frac{\delta}{n-1+\delta} & \frac{1}{n-1+\delta} & 0 \end{pmatrix}. \tag{3.49}$$

This matrix is not row-normalised as it only shows the hopping probabilities between the representatives v_0, v_{01}, v_1, v_{11}. Additionally, between nodes of the same class the hopping probabilities amount to

$$q_{v_{0i}v_{0j}} = \frac{1}{n-1+\varepsilon} \,, \tag{3.50}$$

$$q_{v_{1i}v_{1j}} = \frac{1}{n-1+\delta} \,. \tag{3.51}$$

Writing the eigenvector equation $\boldsymbol{\pi}^T = \boldsymbol{\pi}^T \mathbf{q}$ for π_{v_0}, for instance, we then find

$$\begin{aligned} \pi_{v_0} &= \sum_{j=0}^{n-1} \pi_{v_{0j}} q_{v_{0j}v_0} + \sum_{j=0}^{n-1} \pi_{v_{1j}} q_{v_{1j}v_0} \\ &= \pi_{v_0} q_{v_0 v_0} + (n-1)\pi_{v_{01}} q_{v_{01}v_0} + \pi_{v_1} q_{v_1 v_0} + (n-1)\pi_{v_{11}} q_{v_{11}v_0} \,. \end{aligned} \tag{3.52}$$

In the second equality, we used the observation that there are $n-1$ nodes in each of the symmetry classes each represented by v_{01} and v_{11}, respectively. With the hopping probabilities between classes given in eq. (3.49), π_{v_0} simplifies to

$$\pi_{v_0} = \frac{c-1}{c-1+\varepsilon}\pi_{v_{01}} + \frac{1}{c}\pi_{v_1} \,. \tag{3.53}$$

The remaining elements of $\boldsymbol{\pi}$ for each symmetry class can be expanded in the same way, leading to the system of equations

$$\pi_{v_{01}} = \frac{1}{n}\pi_{v_0} + \frac{n-2}{n-1+\varepsilon}\pi_{v_{01}} + \frac{\delta}{n-1+\delta}\pi_{v_{11}} \,, \tag{3.54}$$

$$\pi_{v_1} = \frac{1}{n}\pi_{v_0} + \frac{n-1}{n-1+\delta}\pi_{v_{11}} \,, \tag{3.55}$$

$$\pi_{v_{11}} = \frac{\varepsilon}{n-1+\varepsilon}\pi_{v_{01}} + \frac{1}{n}\pi_{v_1} + \frac{n-2}{n-1+\delta}\pi_{v_{11}} \,. \tag{3.56}$$

Solving the above (reduced) set of equations gives

$$\begin{pmatrix} \pi_{v_0} \\ \pi_{v_{01}} \\ \pi_{v_1} \\ \pi_{v_{11}} \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} n(1+n\delta+\varepsilon) \\ (1+\delta+n\delta)(n-1+\varepsilon) \\ n(1+\delta+n\varepsilon) \\ (n-1+\delta)(1+\varepsilon+n\varepsilon)) \end{pmatrix} , \tag{3.57}$$

with normalising constant Z.

We can now calculate the LE hopping probabilities, eq. (2.9), again by grouping nodes by their symmetry classes

$$Q_{01} = \frac{1}{\Pi_0} \sum_{i,j=0}^{n-1} q_{v_{0i}v_{1j}} \pi_{v_{0i}} = \frac{1}{\Pi_0} \left(q_{v_0 v_1} \pi_{v_0} + (n-1) q_{v_{01}v_{11}} \pi_{v_{01}} \right) . \tag{3.58}$$

Substituting the entries for $\boldsymbol{\pi}$ obtained in eq. (3.57) and the hopping probabilities from eq. (3.49), we find the expression

$$Q_{01} = \frac{1 + n\delta + \varepsilon + \varepsilon(n-1)(1+\delta+n\delta)}{n(1+\varepsilon+n\delta) + (n-1)(n-1+\varepsilon)(1+\delta+n\delta)} , \tag{3.59}$$

which, due to eq. (2.18), is again the reciprocal of the MFPT

$$M_{01} = \frac{1}{Q_{01}} = \frac{n(1+\varepsilon+n\delta) + (n-1)(n-1+\varepsilon)(1+\delta+n\delta)}{1 + n\delta + \varepsilon + \varepsilon(n-1)(1+\delta+n\delta)} . \tag{3.60}$$

We also record the correct value

$$m_{v_0 v_1} = \frac{-2\delta\varepsilon + \delta + n^2(\delta+\varepsilon+1) + n(\delta(2\varepsilon-1)-1) + 1}{\delta + \varepsilon n + 1} . \tag{3.61}$$

of the MFPT for this simple model, obtained with the help of Mathematica.

In light of the previous section 3.2, it seems natural to necklacify this walker by removing the leaks and adding an appropriate weight to the backbone. And since the arc $v_1 \longrightarrow v_0$ does not affect the first-passage process, we can make the backbone symmetrical. The correct additional weight should be a function of ε as well as δ, as in the original dynamics the walker can follow leaks back and forth without being absorbed at v_1. To find the appropriate weight function, we first consider a general variable w and define the necklacification

$$\mathbf{q}^0 = \begin{pmatrix} 0 & \frac{1}{n+w} & \frac{n-2}{n+w} & 0 & 0 & \frac{w+1}{n+w} \\ \frac{1}{n-1} & 0 & \frac{n-2}{n-1} & 0 & 0 & 0 \\ \frac{1}{n-1} & \frac{1}{n-1} & \frac{n-3}{n-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{n-2}{n-1} & \frac{1}{n-1} \\ 0 & 0 & 0 & \frac{1}{n-1} & \frac{n-3}{n-1} & \frac{1}{n-1} \\ \frac{w+1}{n+w} & 0 & 0 & \frac{1}{n+w} & \frac{n-2}{n+w} & 0 \end{pmatrix} , \tag{3.62}$$

which is reversible, and thus exhibits the MFPT

$$m^0_{v_0 v_1} = M^0_{01} = \frac{1 + n(n-1) + w}{1 + w} . \tag{3.63}$$

We may then optimise $w = w(\varepsilon, \delta)$ such that[4] $M_{01} = m^0_{v_0 v_1}$, which produces in an elementary but error-prone calculation

$$w(\varepsilon, \delta) = \frac{(n-1)\left(-\delta + \varepsilon + \delta\varepsilon n^2 + \delta\varepsilon n + \varepsilon n\right)}{-\delta + \varepsilon + \delta n^2 + \delta n + n} \ . \tag{3.64}$$

We have now restored the situation given in section 3.2 where the coarse-grained MFPTs are identical and we need only to consider the microscopic error discussed in section 3.1.1.

Computing the matrix products $\boldsymbol{G}^0 \left(\frac{d}{d\varepsilon}\widehat{d\mathbf{q}}\right) \boldsymbol{G}^0$ and $\boldsymbol{G}^0 \left(\frac{d}{d\delta}\widehat{d\mathbf{q}}\right) \boldsymbol{G}^0$ supported by Mathematica, we notice that the contribution involving $\frac{d}{d\delta}\widehat{d\mathbf{q}}$ vanishes! For the other summand, we find at $\varepsilon = \delta = 0$

$$\left[\boldsymbol{G}^0 \left(\frac{d}{d\varepsilon}\widehat{d\mathbf{q}}\right) \boldsymbol{G}^0 \mathbf{1}_{n-1}\right]_{v_0}\Bigg|_{\varepsilon=\delta=0} = -\frac{1}{n^2} \ . \tag{3.65}$$

The linear error in ε is non-zero, but quadratically suppressed by the cluster size.

We compare the results of this section to simulations in section 3.4.2. From the results of the present section, however, we can already anticipate a weakness of the presented method: LE relies on equilibrium properties, which are implicitly encoded in the weights of undirected graphs. Even if an essential edge has nominally different weights in its two directions, only the weight of the arc pointing from cluster 0 to 1 matters for the first-passage process. This arc can be symmetrised without loss of generality. In the presence of leaks, however, the reverse *leak* weights matter as they allow switching between the clusters without the walker being absorbed. In the numerical study in section 3.4.2, we see that M_{01} still offers a good approximation for $m_{v_0 v_1}$. However, it appears that our method of determining corrections to this approximation is unable to pick up any contribution from the leaks in the reverse direction.

3.4 Numerical studies

3.4.1 Bidirectional leaks with random weights

We begin with the situation of section 3.2, depicted in fig. 3.2.1, where two clusters 0 and 1 are given by unweighted cliques which are connected by an unweighted backbone and some symmetrically weighted leaks.

Fig. 3.4.1 compares the value of the clustered MFPT M_{01} to the true value of the original MFPT $m_{v_0 v_1}$, obtained by solving eq. (2.7) numerically for different numbers of

[4] It would certainly be preferable to optimise w such that $m^0_{v_0 v_1} = m_{v_0 v_1}$. However, if $m_{v_0 v_1}$ is known, we need not approximate it.

leaks k, and different ε's, i.i.d. sampled from the probability density

$$f_\mu(x) = \frac{1}{\mu} e^{-x/\mu}, \quad x \geq 0\,. \tag{3.66}$$

Here, we fix μ to the inverse number of leaks, $\mu = 1/k$, and the size of both cliques to $n = m = 40$.

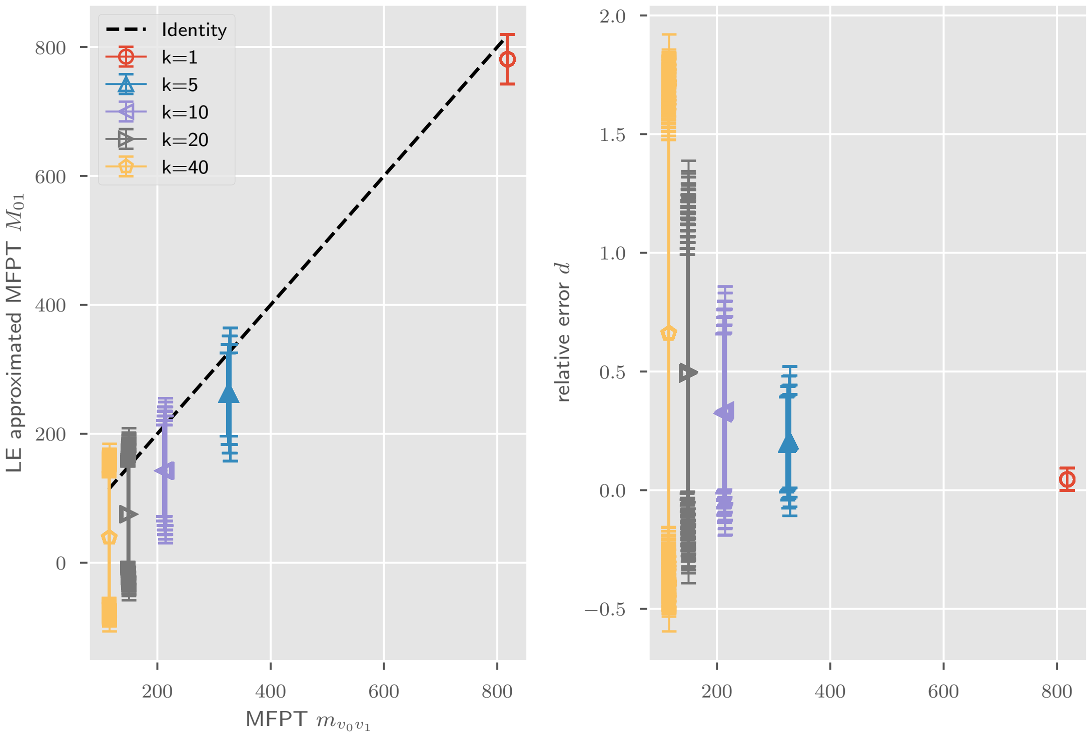


Fig. 3.4.1 Left: Exact MFPTs $m_{v_0v_1}$ according to eq. (2.7) vs M_{01} as per eq. (3.25), for the random walker with k leaks (see section 3.4.1) and clique size $m = n = 40$. Right: Relative deviation, see eq. (3.67). The k leaks are drawn without replacement from all possible pairings of vertices (v_{0i}, v_{1j}); the values for the $\varepsilon_{v_{0i}v_{0j}}$'s are drawn independently with density $f_{1/k}$ as in eq. (3.66). For each value of k, 1000 realisations of leaks and $\varepsilon_{v_{0i}v_{0j}}$'s were sampled. The error bars on both plots show the linear order expression from eq. (3.44); the error is divided by $m_{v_0v_1}$ in the right panel.

Firstly, we notice that the relative deviation

$$d = \frac{m_{v_0v_1} - M_{01}}{m_{v_0v_1}} \tag{3.67}$$

in the right panel of fig. 3.4.1 increases with k, as the total leakage increases. This is in fact expected, since higher leakage moves the graph further from having necklace structure. Secondly, M_{01} systematically underestimates the true MFPT; this is reasonable to expect, as the coarse-grained walker is absorbed immediately when it crosses over to block 1, whereas in the microscopic model it has the chance to cross via a leak, hit a non-absorbing node in 1, and return. With $k = 0$, only the backbone exists and we recover the necklace-case. Moreover, all values of M_{01} lie within the linear order error in eq. (3.44), which decreases accordingly as the number of leaks decreases to a single one.

3.4.2 Directed leaks with random weights

Fig. 3.4.2 compares the approximation derived in section 3.3 to the true value of $m_{v_0 v_1}$, computed by solving eq. (2.7) numerically for different values of n and different values of ε and δ, i.i.d. sampled from the exponential probability density in eq. (3.66) with mean $\mu = 1/n^2$.

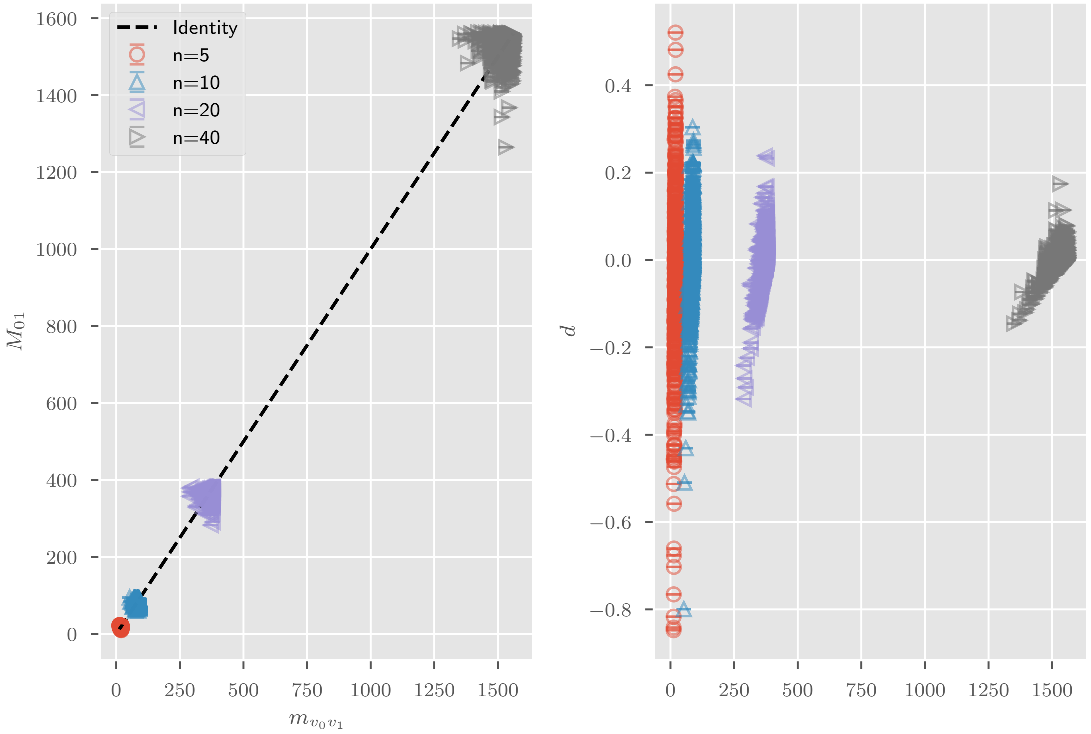


Fig. 3.4.2 Left: Exact MFPTs $m_{v_0 v_1}$ according to eq. (2.7) vs approximate results M_{01} as per eq. (3.60), for the random walker with leaks of section 3.3. Right: Relative deviation, see eq. (3.67). Each data point represents a pair of ε and δ drawn independently with density f_{1/n^2} as in eq. (3.66). Colours indicate the clique size n. For each c, 1000 samples for ε and δ were drawn.

The right panel of fig. 3.4.2 shows that the relative deviations, eq. (3.67), decrease with increasing n. As the total mean leakage amounts to $1/n$ (in any direction, so to and from 0, respectively), this means that the approximation becomes better as less relative weight is accumulated on the leaks. This is as expected, for the lower the total leakage, the closer the graph is to having necklace structure, and the more precise the approximation in eq. (2.26) becomes. Note, however, that our linear approximation for the error is unusable in this example: the error bars in fig. 3.4.2 are shorter than the symbols.

3.4.3 Stochastic Block Model

In this last section of the more abstract part of this thesis, we apply the methods outlined above to a popular basic model for social networks with communities, the Stochastic Block Model (SBM) [93]. In the simplest SBM with two blocks, the set of nodes is partitioned in two, V_0 and V_1 say, with sizes n_0 and n_1, respectively. Stochasticity enters the model as the edges are sampled randomly: two nodes are connected with probability p_0 if they are both in V_0, and with probability p_{01} otherwise. Similarly, two nodes in V_1 are connected with probability p_1. Usually, one chooses p_{01} to be much smaller than p_0 and p_1, leading to the formation of two communities.

Let us denote the (random) number of edges within blocks 0 and 1, and between 0 and 1 as E_0, E_1 and E_{01}, respectively. Clearly, there can only be a first-passage event within finite time if there is at least one edge between the blocks. Our discussion shall therefore be conditional on $|E_{01}| > 0$. Since all edges have unit weight in this model, we pick the backbone $(v_0 v_1)$ randomly from E_{01}. In analogy to the previous discussions, $\boldsymbol{\pi}$ and Π are essentially the node strengths (or degrees, in this unweighted case), which allows us to immediately write

$$M_{01} = \frac{E_{01} + 2E_0}{E_{01}} = 1 + 2\frac{E_0}{E_{01}} \ . \tag{3.68}$$

We can proceed to calculate the expected value of M_{01}, averaging over the ensemble of edge sets in the block model. As all edges are independent, the expectation can be written as

$$\mathbb{E}\left(M_{01} \,|\, E_{01} > 0\right) = 1 + n_0(n_0 - 1)\mathbb{E}\left(\frac{1}{E_{01}} \middle| E_{01} > 0\right) \ . \tag{3.69}$$

The number of edges within and between clusters follow binomial distributions, which we briefly review in the appendix 4.A. Using basic properties of the binomial distribution, the conditional expectation on the right can be rewritten as the unconditional expectation of $\frac{1}{(X+1)^2}$, where X follows a binomial distribution with parameters $n = n_0 n_1 - 1$ and $p = p_{01}$, in the notation of eq. (4.28). There exists a closed form representation for the latter expectation, involving a generalised hypergeometric function. For simplicity's sake, we replace the expected value using $\mathbb{E}\left(\frac{1}{(X+1)^2}\right) \approx \left[\mathbb{E}\left(\frac{1}{X+1}\right)\right]^2$. We can thus write

$$\begin{aligned}\mathbb{E}\left(M_{01}\,|\,E_{01}>0\right) \approx& 1+n_0(n_0-1)\left[\frac{1-(1-p_{01})^{n_0 n_1}}{p_{01}n_0 n_1}\right]^2 \\ =& 1+\frac{(n_0-1)\left[1-(1-p_{01})^{n_0 n_1}\right]^2}{n_0\left(p_{01}n_1\right)^2}\,, \end{aligned} \tag{3.70}$$

using the first negative moment of a binomial variable, for instance reported in [113]. We test the quality of eq. (3.68) by simulating an ensemble of SBMs with block sizes $n_0 = 60$ and $n_1 = 80$, and $p_0 = p_1 = 0.9$, $p_{01} = 2/(n_0 n_1)$. This way we can assume to have on average two edges connecting the clusters. The results of this simulation are depicted in fig. 3.4.3. In this simulation, we discard all network realisations with $E_{01} = 0$, for the reason described above. Furthermore, we do not show values obtained for $E_{01} = 1$ as this is the trivial case in which eq. (3.68) is certain to be exact.

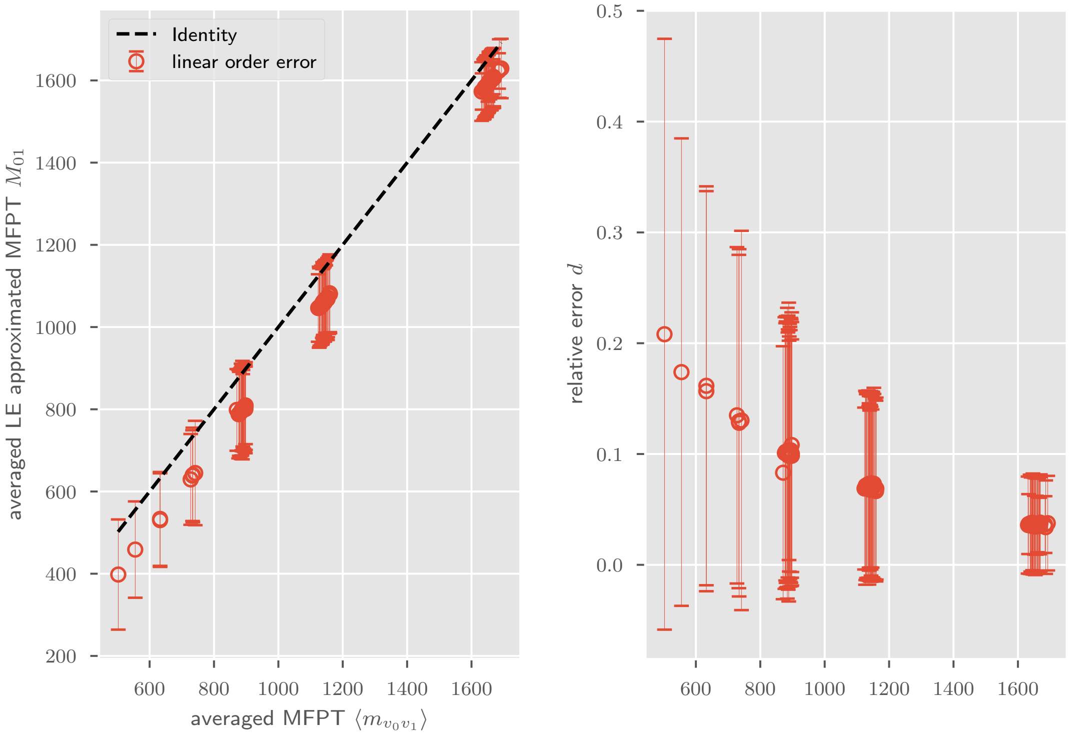


Fig. 3.4.3 MFPTs on 100 realisations of a SBM with block sizes $n_0 = 60$ and $n_1 = 80$, $p_0 = p_1 = 0.9$ and $p_{01} = 2/(n_0 n_1) \approx 4.2\times 10^{-4}$. Each data point is generated by choosing every inter-block edge connecting 0 to 1 as the backbone once, and averaging the results obtained for the different choices of backbone. Left: LE-approximated MFPT given in eq. (3.68) over true MFPT, calculated as per eq. (2.7). Right: relative deviation (see eq. (3.67)) over true MFPTs. The error bars on both plots show the linear order expression from eq. (3.43)

Similar to section 3.4.1, we observe that M_{01} systematically underestimates $m_{v_0v_1}$ as expected. Moreover, the approximation error lies within the linear error estimate, eq. (3.43).

3.5 Conclusions

We have shown how the method developed in chapter 2 can be applied to graphs whose necklace-structure is only approximate. More precisely, we considered clusters that had more than one interconnection. Of these interconnections, one link qualified as the backbone, whereas the others were either sparse as compared to the density of edges within the clusters, or carried low weights as compared to the backbone. Furthermore, we have quantified the error made in this approximation for archetypical test cases, including the Stochastic Block Model, and shown that our error-estimate markedly improves the agreement between MFPTs based on microscopic and macroscopic dynamics.

In essence, our error approximation relies on constructing a new microscopic model that reproduces the dynamics of the coarse-grained walker on the nodes of the original graph. This enables us to use matrix calculus to quantify the deviation between their MFPTs. However, we show that this approach is problematic if leaks are directed (i.e. if their weights in both directions differ). While the original approximation derived from eq. (2.25) still seems usable, our error estimate to linear order does not depend on the "returning" weight of the leaks and vastly underestimates the true error. Devising a coarse-graining procedure that allows us to estimate its effect on MFPTs therefore remains open for future research.

Among the many remaining open problems we highlight the following. We have only clustered graphs in such a way leading to loop-less graphs. This will not always be possible or desirable, if the nodes of interest are not located at a boundary between suitable clusters. This may frequently be the case, for instance in small-world models where "long-distance" edges may introduce loops between clusters [30]. Moreover, we have for the sake of clarity focused on almost-necklaces with two clusters. What does not appear in this setting is the question whether the dominant links connecting cluster I to $I-1$ and $I+1$ attach to the same node in I. It is a requirement for exact necklaces that this be the case, but it will need to be relaxed in approximate settings. A natural extension of our necklacification technique should encompass such problems, too.

Finally, extensive research has already been done on perturbations of Markov chains with general statements: see for instance [89] and works cited within that article. It is likely that our method can be supplemented by some of these results.

This chapter concludes the abstract half of the thesis. In the next chapter we deliver the announced model for an information-seeking user of the law.

Chapter 4

Complexity of codified law: Searching times

4.1 Introduction

What is the maximum number of tenants that a UK landlord may let a property of a given size to, without incurring in penalties? Faced with a legal question of this nature, a layperson would naturally resort to consulting the institutional repository www.legislation.gov.uk, where a quick keyword search (say, "overcrowding") would return – as most recent reference – the Housing Act 1985. The requested information will eventually be found in the Articles 325 *et sqq.*, which can be located after following the path "UK Public General Acts", "Housing Act 1985", "Part X: Overcrowding", "Definition of Overcrowding", "324 Definition of Overcrowding", through several Part and Section headers of the Act.

Arguably, a definition of how "complex" a piece of legislation is should reflect how fast and reliably information hidden in its text can be retrieved. The concept of "legal complexity" and quantitative measures thereof – in one of its many incarnations – have been considered by legal scholars and – to a lesser extent – by scientists in recent years (as discussed in chapter 1), so far without reaching a satisfactory and widely accepted consensus on the best framework to use. In this chapter, we develop a detailed model for the search process of information hidden in a legal text – organised in a hierarchical fashion – by a reader unfamiliar with the text, who needs to extract a precise answer out of a potentially messy structure of semantic dependencies. The level of detail of the model is such that the reader is defined by intuitive probabilistic behaviour that can be controlled at several levels of the hierarchy.

We consider a tree structure that mimics the organisation of a typical Act of Parliament, with primary focus on the usual structure of legislative bills in the United Kingdom. To standardise labels, we always use the capitalised terms “Act”, “Part”, “Chapter”, “Section”, “Article”, and “Paragraph” for ease of reference. If applicable, the list can be extended by adding “sub”-Items, e.g. Sub-Sections between Sections and Articles. For the sake of clarity, references to items of this thesis have and will be made in small letters. Each Item in the hierarchy is identified by a header or set of keywords, which ideally reflect the general content of the corresponding sub-tree: for example, the tree in fig. 4.1.1 represents a selection of the Housing Act 1985, with some of the nodes labelled by their textual content.

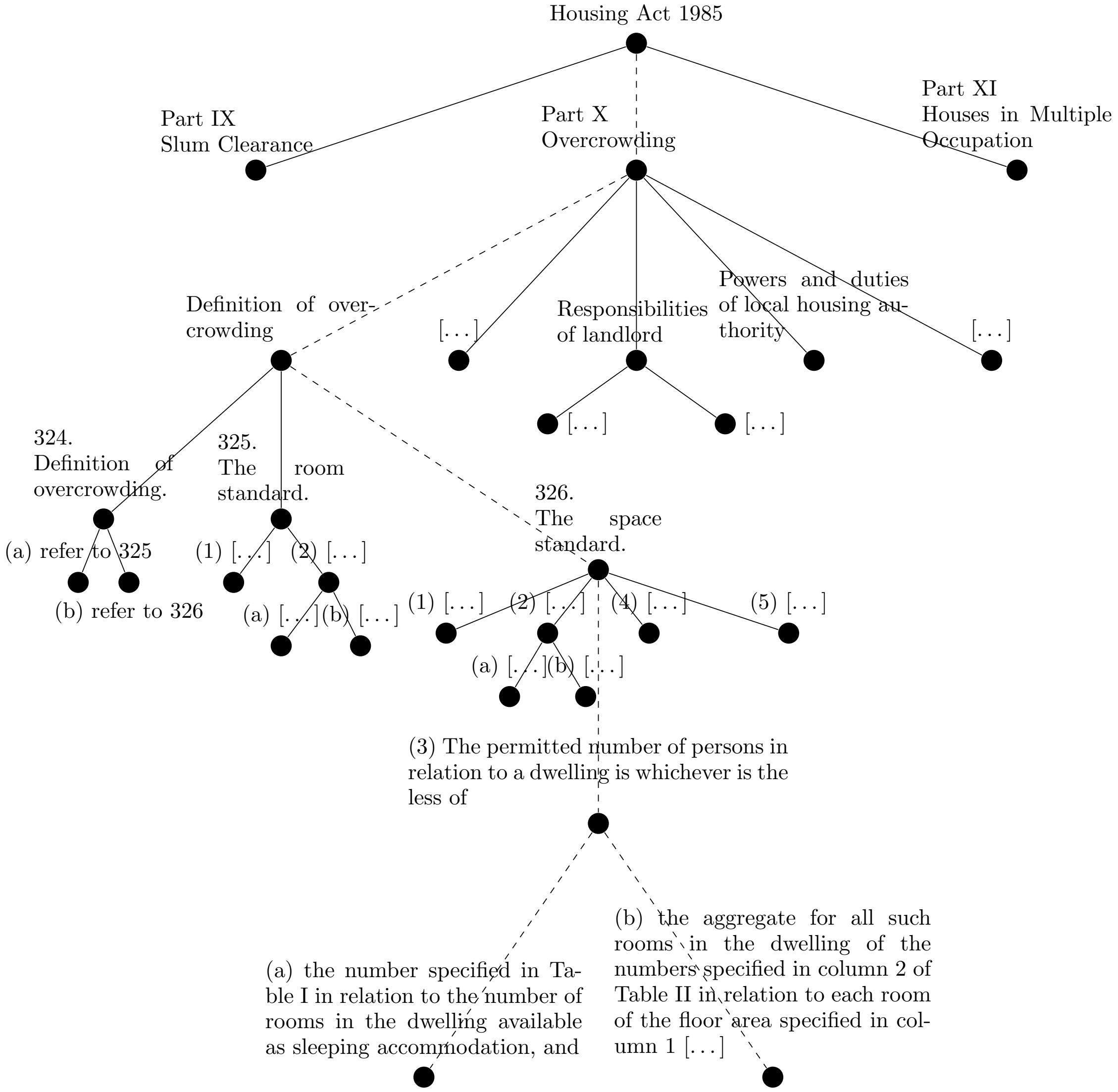

Housing Act 1985
Part IX
Slum Clearance
Part X
Overcrowding
Part XI
Houses in Multiple Occupation
Definition of overcrowding
[...]
Responsibilities of landlord
Powers and duties of local housing authority
[...]
[...]
[...]
324. Definition of overcrowding.
325. The room standard.
326. The space standard.
(a) refer to 325
(b) refer to 326
(1) [...]
(2) [...]
(a) [...]
(b) [...]
(1) [...]
(2) [...]
(4) [...]
(5) [...]
(a) [...]
(b) [...]
(3) The permitted number of persons in relation to a dwelling is whichever is the less of
(a) the number specified in Table I in relation to the number of rooms in the dwelling available as sleeping accommodation, and
(b) the aggregate for all such rooms in the dwelling of the numbers specified in column 2 of Table II in relation to each room of the floor area specified in column 1 [...]

Fig. 4.1.1 Example from an excerpt of the UK Housing Act 1985, c.68, to be found at https://www.legislation.gov.uk/ukpga/1985/68/contents. The nodes of the tree represent structural Items of the text such as the Act itself, its Parts, Sections, etc, and two Items are linked if one is contained in the other. The dashed edges label the ideal path of a reader researching the question “What is the maximum number of tenants that a UK landlord may let a property with a given number of rooms to, without incurring in penalties?”

Higher up in the hierarchy, the textual content mainly consists of short titles, containing only a small number of keywords (e.g. "Housing" and "Occupation"), while lower nodes often contain full sentences. The Chapter "Definition of Overcrowding" of Part X "Overcrowding" in fig. 4.1.1 contains the text [94]

> ***324 Definition of overcrowding.***
> *A dwelling is overcrowded for the purposes of this Part when the number of persons sleeping in the dwelling is such as to contravene—*
> *(a) the standard specified in section 325 (the room standard), or*
> *(b) the standard specified in section 326 (the space standard).*

Based on the headers and keywords, and assuming they reflect the content underneath, the reader will be more or less inclined to follow a certain path rather than another in their search for a piece of information, planted in one of the leaves of the tree. For the sake of simplicity, we do not consider more complicated network structures with cycles and long-range connections, determined for instance by cross-references or internal amendments. The relative number of edges thus ignored depends on the legislative style of the country: for the UK Housing Act 2004, [115] finds that the entirety of the Act roughly encompasses 3,500 elements when paragraphs are considered as leaves; meanwhile, its five most-cited sections receive a total of about 100 references from within the Act, which corresponds to about 3% of the number of edges of the backbone tree. Reference [35] reports that for the US code, roughly 10%-12% of the edges are cross-references. On the other had, in the German federal legislation 60% (in 1994) to 86% (in 2018) text elements contain a reference, such that the number of references are comparable to the size of the tree in that system. Clearly, the approximation made by omitting cross-references is hardly justified in the latter case, although we do not study the quality of this approximation in the present thesis.

We characterise the complexity of a legal tree in terms of the time a random reader takes on average to reach the sought information by hopping through the nodes of the tree, guided by the Items' keywords. The hopping probabilities reflects the search strategy of the reader and is defined in section 4.2. As our observable of choice, we will therefore focus on the MFPTs to reach the target stating from the root, for which we will give analytical estimates. Reintepreting the MFPTs of random readers of "legal trees", we will be able to formulate a closed-form expression for their *structural complexity* in terms of the network parameters, and draw some real-life policy implications for the drafting of legal texts.

4.1.1 Studies of (randomised) search models

Since the previous chapters have focused largely on the technical aspects of MFPTs, in this section, we present an overview of the literature relating to search models – legal and more general ones. After that, we review a few works on information retrieval and text models that capitalise on document interrelations.

The searching behaviour of agents in a diverse range of systems has been the subject of intensive study for many decades. Following similar developments in economics [95, 96] and biology [97, 98], understanding law-search has been gathering traction recently [41, 99, 42]. In these works, searching is framed as an optimal stopping problem: upon sampling a resource (information, trading opportunities, food) a number of times in one location, how does the searcher decide when to stop and change location? Our approach will be different in that our searcher has sufficient information to know exactly when to stop. The modelling of such problems in terms of random walks (on networks or grids, say) has proven successful in many cases – see for instance [100] and references therein for search strategies involving resets of the random walker to its starting position. Another, similar class of problems concerns moving and hiding targets and optimal strategies for a random-walk searcher [57].

The references [41, 99, 42], further an approach based on a joint empirical analysis of structure and contents of legal text networks. This is done by means of network-based *topic modelling*, a method largely developed in [39, 101, 102] to extract a set of topics given a sufficiently large body of text, and assign one or more topic labels to each section. For instance, this analysis can be based on the movements of a random walker in the textual landscape, studying in which regions the walker tends to sojourn for longer periods of time, as well as the overlap between such regions. In [42], the authors demonstrate that this analysis may be useful to predict citations in US legal opinions and statutory law. Moreover, they propose a law-search model based on their findings on link prediction, and compare it with human law-searchers. Further references on this topic can be found in [42, 99].

Other lines of research have been more interested in the particular behaviour of an information-seeker. Important foundations to this field are laid out in [103], leading to further studies in various contexts. In particular, [104] (and references therein) examines the information-seeking behaviour of legal professionals. These are shown to primarily use informal sources of information, including their own professional experience where applicable, or consulting knowledgeable colleagues and acquaintances regarding topics outside their own expertise. Comprehensive reviews of the area are given in [105] and [106].

Shifting our focus towards our own contributions, in section 4.2 we provide the main definitions of our model for a keyword-driven 'random reader', hopping through Sections of a legal tree according to suitably defined probabilistic rules. Section 4.3 analyses some of the statistical properties of this model. Our main results are shown in section 4.4: these are based on a mean-field – or rather, 'mean-text' – approximation for the MFPT between the starting point and a target node of our random reader model. This approximation serves as the definition of our 'complexity function' for legal trees. We compare analytical and numerical results in section 4.5, and summarise our findings in section 4.6. Technical derivations of our results are shown in the appendices 4.A to 4.E. The supplementary material contains an animation of the model discussed in this chapter.

4.2 Definition of the model

We consider a finite tree of N nodes – in which every node stands for an "Item" in the law as described in section 4.1 – and two nodes are connected as parent and daughter if one contains the other (e.g. a Section within a Chapter, or an Article within a Section). There are no limitations on the exact shape of the tree. For simplicity of the analysis, however, we consider c-ary trees, i.e. trees in which a designated *root*, r, has degree c and every other node is either a leaf, or has degree $c+1$.

We model the textual content of every node v in terms of a binary string of length L, that we will refer to as *pattern*. We denote patterns as $\boldsymbol{\xi}^v = (\xi_1^v, \ldots, \xi_L^v)$ where $\xi_i^v \in \{0,1\}$ encodes presence (1) or absence (0), in the textual content of node v, of keyword i, from a predetermined *glossary* of L keywords (which is typically defined by the user). A reduced glossary for the example in fig. 4.1.1 may be the list {"slum", "demolition", "clearance", "overcrowding", "room", "space", "responsibilities", "occupation", "escape"}, which would lead to the assignment of patterns shown in fig. 4.2.1. We argue that these are sufficient for a rough model of reader behaviour on structured text. In a recent eye-fixation study, [107] looked at the attention patterns of participants to conclude that humans, when searching for specific information in a text, do not read it word for word. Instead, they search it in a two-stage process that involves skimming and only reading the thus pre-selected Items afterwards. Thus is seems reasonable to assume that our model reader uses rapidly extracted keyword patterns rather than the full textual information for navigating the text.

We assume that a reader is interested in the information hidden in a particular leaf of the tree, which we call the *target* t. We model the search process of the reader as a random walker that moves randomly along the links of the tree, starting from the root. We assume

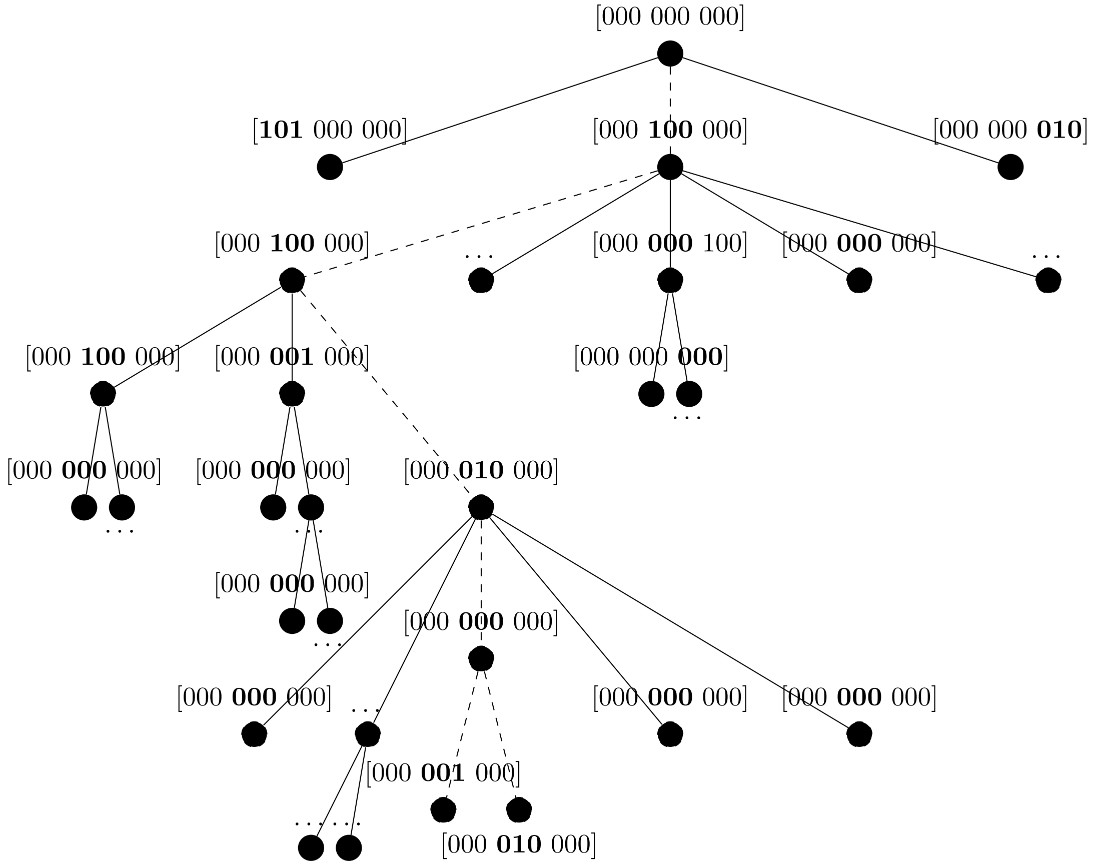


Fig. 4.2.1 Representation of the excerpt of the UK Housing Act 1985 c.68 shown in fig. 4.1.1 by binary patterns for the (shortened) glossary {“slum”, “demolition”, “clearance”, “overcrowding”, “room”, “space”, “responsibilities”, “occupation”, “escape”} of length $L = 9$. Every node is assigned a binary pattern of length L, whose bits represent the presence or absence of the corresponding keyword. The boldface bits stand for keywords specific to the Part in which their node is located. Dots are used to omit some of the all-0 patterns.

that the reader is more likely to step on a node when the text associated with it has a higher semantic similarity with the sought (target) information. Hence, we assume that when on a node, the walker will step to one of the neighbouring nodes v with a probability that depends on the semantic similarity between the node v and the target node, and it does not depend on the starting node. The semantic *dis*similarity of two nodes is measured as the *Hamming distance* of their patterns

$$d(\boldsymbol{\zeta},\boldsymbol{\xi}) := d_H(\boldsymbol{\zeta},\boldsymbol{\xi}) := \sum_{\ell=1}^{L} |\zeta_\ell - \xi_\ell| \,, \tag{4.1}$$

which equals the number of bits on which the two patterns disagree. We will refer to such distance as “pattern-distance”. This distance is not to be confused with the “edge-distance” between the corresponding nodes on the graph, defined as the number of edges constituting the shortest path between them. As the probability to step onto a node v does not depend on the node from which the walker is stepping, we can define, for any link pointing to v, a weight

$$\omega(v) = \frac{1}{d\left(\boldsymbol{\xi}^{v}, \boldsymbol{\xi}^{t}\right) + 1} \tag{4.2}$$

which increases with the semantic similarity between the patterns in v and t. Using these non-negative weights, we can define a matrix of transition probabilities between two connected nodes u and v as

$$q_{u,v} = \frac{\omega(v)}{\sum_{x \in \partial u} \omega(x)} \,, \tag{4.3}$$

where ∂u is the neighbourhood of node u.

We will characterise the complexity of the search process in terms of the average number of steps taken until the target is first found. Our object of study will be the dependence of this quantity on the way patterns are assigned to nodes as well as on the properties of the tree itself.

We will assume a stochastic set-up, where patterns are regarded as quenched random binary vectors, with statistics controlled by two tunable parameters, which we call *tightness* and *overlap*, representing the vertical and horizontal coherence of the legal text, respectively. In particular, we will assume that the components of the root pattern are independent random variables with fixed expected value a. All patterns on the lower levels are generated from their parent node via a Markov process, according to which the entries of the patterns in the child node are mutated with a given rate with respect to those of the parent node. The idea that content similarity and structural closeness are covariates follows recent developments in topic modelling (e.g. structural topic models [108]) and information retrieval [109]. The *tightness* τ is defined as a decreasing function of the mutation rate such that tighter sets of patterns are generated by lower rates of mutation. Additionally, assuming that each part covers a unique topic with corresponding specific keywords, we define the *overlap* (denoted 2Δ), quantifying the number of keywords that are expected to be shared by two successive Parts.

In later sections, we will study the impact of τ and Δ on the complexity of the defined search process. We will first quantify their role on the statistics of pattern distances (section 4.3), then we will study their role on the average number of steps taken by the walker to first reach the target node where the information of interest is hidden (sections 4.4 and 4.5).

In the remainder of the present section, we provide details about the Markov process used to generate patterns. These reflect the hierarchy of the tree and have the desired properties of tightness and overlap.

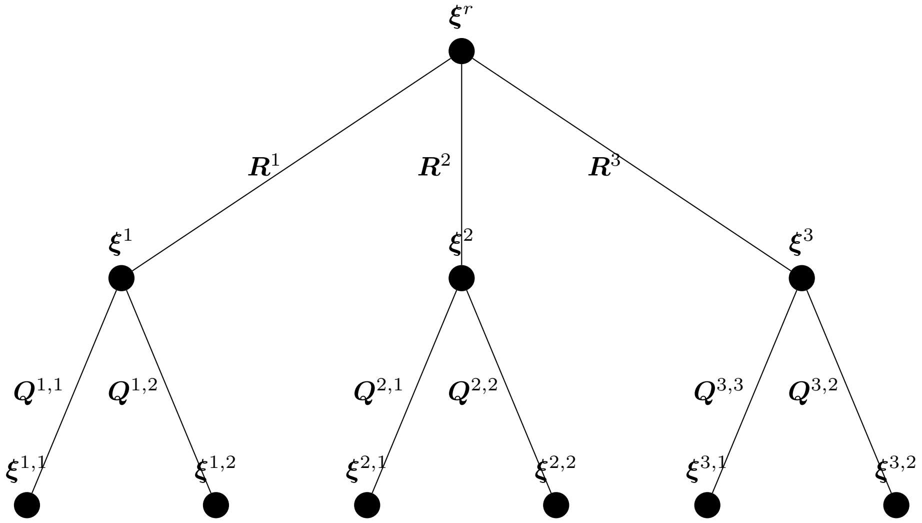


Fig. 4.2.2 Pattern hierarchy and their relations. The i-th bit of each pattern is generated from the parent bit by a two-state Markov process, the transition matrix of which is indicated next to the edge between the two patterns. For instance, the transition matrix relating ξ_i^2 to ξ_i is $\boldsymbol{R}_i^2$.

We will index the vertices by lexicographical labels. Thus, the c descendants of the root will be denoted by a single index $\mu_1 \in \{1,\dots,c\}$; the μ_2-th descendant of the μ_1-th descendant of the root will be denoted by the two indices μ_1,μ_2, and a node v in the k-th generation will be denoted by k indices $\mu_1,\dots,\mu_k$. As the tree is c-ary, we have $\mu_j \in \{1,\dots,c\}$ for all $1 \le j \le k$. We refer to fig. 4.2.2 for a schematic representation of the genealogy of patterns. We denote by $\boldsymbol{\xi}$ the textual pattern at the root. For the root pattern, we assume the entries to be drawn from a factorised distribution

$$P(\boldsymbol{\xi}) = \prod_{i=1}^{L} P_i(\xi_i) \tag{4.4}$$

with

$$P_i(\xi_i) = a\xi_i + (1-a)(1-\xi_i)\ , \tag{4.5}$$

so for all $i = 1,\dots,L$ the expectation is $\langle \xi_i \rangle = a$. For the first level of hierarchy (i.e. Part-level), we assume that the patterns are generated with distribution

$$P^{\mu}(\boldsymbol{\xi}^{\mu}|\boldsymbol{\xi}) = \prod_{i=1}^{L} R_i^{\mu}(\xi_i^{\mu}|\xi_i) \tag{4.6}$$

from the root pattern. Here, the $R_i^{\mu}(\xi_i^{\mu}|\xi_i)$ for $\xi_i^{\mu},\xi_i \in \{0,1\}$ form the entries of the 2×2 transition matrix of the Markov process generating the Part-pattern entry ξ_i^{μ} from the root.

By the law of total probability, ξ_i^μ is described by the marginal probability distribution

$$P_i^\mu(\xi_i^\mu) = \sum_{\xi_i=0}^{1} P_i(\xi_i) R_i^\mu(\xi_i^\mu|\xi_i) \ . \tag{4.7}$$

Fixing $\left\langle \xi_i^\mu \right\rangle = a_i^\mu$, $\boldsymbol{R}_i^\mu$ has the elements

$$\boldsymbol{R}_i^\mu = \begin{pmatrix} P_i^\mu(0|0) & P_i^\mu(1|0) \\ P_i^\mu(0|1) & P_i^\mu(1|1) \end{pmatrix} = \begin{pmatrix} 1-\Gamma' & \Gamma' \\ 1 - \frac{a_i^\mu - (1-a)\Gamma'}{a} & \frac{a_i^\mu - (1-a)\Gamma'}{a} \end{pmatrix} , \tag{4.8}$$

where the parameter $\Gamma' \in [0,1]$ determines the rate of mutation from ξ_i to ξ_i^μ.

We take the a_i^μ to satisfy some constraints, motivated by the idea that different Parts treat individual topics. If a given keyword is highly related to the topic of some Part, it will have a high probability to appear in that Part. If each keyword is related to one topic only, it will appear with high probability in the Part treating that topic, and with low probability in all other Parts. This situation is represented in the top panel of fig. 4.2.3. However, we allow for a degree of topic similarity between two successive Parts μ and $\mu+1$, realised by a subset of size 2Δ of keywords that appear with high probability in μ, as well as in $\mu+1$.

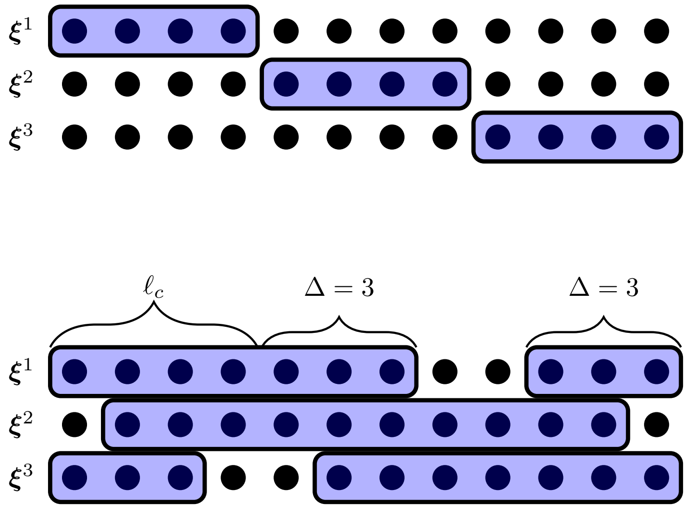


Fig. 4.2.3 A schematic drawing of patterns with $L = 12$ and $c = 3$, with boxes marking bits with high expectation $\left\langle \xi_i^\mu \right\rangle = a_h$. Top: $\Delta = 0$, i.e. specific keywords of different Parts do not overlap. Bottom: $\Delta = 3$, i.e. two neighbouring Parts have $2\Delta = 6$ specific keywords in common. For $\Delta = 0$ there are $\ell_c(c-1)$ keywords that are generic for any Part, therefore the maximum value for Δ is $\ell_c \frac{(c-1)}{2} = 4$.

This situation for $2\Delta = 6$ is shown in the bottom panel of fig. 4.2.3, representing the appearance of topic-specific keywords in the Parts. In particular, we introduce Δ as a parameter that controls the number of Part-specific keywords shared by neighbouring Parts

$\mu, \mu+1$. We say that the Parts have an *overlap* and define

$$a_i^\mu = \begin{cases} a_h & : \ (\mu-1)\ell_c - \Delta < i \le \mu\ell_c + \Delta \ , \\ a_l & : \ \text{else} \ , \end{cases} \tag{4.9}$$

where ℓ_c is the ratio L/c, $\Delta \in [0, \ell_c \frac{c-1}{2}]$ and we consider periodic boundaries, i.e. $a_i^\mu = a_{kL+i}^\mu$ for all $k \in \mathbb{Z}$ (see fig. 4.2.3 for a schematic representation). Moreover, we assume $0 < a_l < a_h < 1$ (the indices l and h stand for "low" and "high"). However, since elements of $\boldsymbol{R}_i^\mu$ are probabilities, one can derive the stricter bounds

$$(1-a)\Gamma' \le a_i^\mu \le (1-a)\Gamma' + a \tag{4.10}$$

for all $\mu = 1, \dots, c$ and $j = 1, \dots, L$. For our purposes, it will be useful to define

$$a_l = (1-a)\Gamma' + \beta_l \tag{4.11}$$

$$a_h = (1-a)\Gamma' + a \tag{4.12}$$

for some β_l satisfying $0 < \beta_l < a$ to enforce $a_l < a_h$. The set of indices i such that $a_i^\mu = a_h$ we refer to as the a_h*-domain* of μ, and to the complementary set as a_l*-domain*.

We now extend the above prescription to generate patterns at the lower levels of the hierarchy. Consider the pattern $\boldsymbol{\xi}^{\mu_1,\cdots\mu_k}$ on level k, generated with probabilities

$$P^{\mu_1\cdots\mu_k}(\boldsymbol{\xi}^{\mu_1\cdots\mu_k}|\boldsymbol{\xi}^{\mu_1\cdots\mu_{k-1}}) = \prod_{i=1}^{L} Q_i^{\mu_1\cdots\mu_k}(\xi_i^{\mu_1\cdots\mu_k}|\xi_i^{\mu_1\cdots\mu_{k-1}}) \tag{4.13}$$

for a 2×2 transition matrix $\boldsymbol{Q}_i^{\mu_1\cdots\mu_k}$, so the marginal probabilities can be written in terms of the marginal of the parent pattern

$$P_i^{\mu_1\cdots\mu_k}(\xi_i^{\mu_1\cdots\mu_k}) = \sum_{\xi_i^{\mu_1\cdots\mu_{k-1}}=0}^{1} P_i^{\mu_1\cdots\mu_{k-1}}(\xi_i^{\mu_1\cdots\mu_{k-1}}) Q_i^{\mu_1\cdots\mu_k}(\xi_i^{\mu_1\cdots\mu_k}|\xi_i^{\mu_1\cdots\mu_{k-1}}) \ . \tag{4.14}$$

For simplicity, we assume that $a_i^{\mu_1\cdots\mu_k} := \langle \xi_i^{\mu_1\cdots\mu_k} \rangle$ is inherited from the parent, i.e.

$$a_i^{\mu_1\cdots\mu_k} = a_i^{\mu_1\cdots\mu_{k-1}} = \cdots = a_i^{\mu_1} \ , \tag{4.15}$$

which equals either a_h or a_l, according to eq. (4.9). Then, $\boldsymbol{Q}_i^{\mu_1\cdots\mu_k}$ takes the form

$$\boldsymbol{Q}_i^{\mu_1\ldots\mu_k} = \begin{pmatrix} P_i^{\mu_1\ldots\mu_k}(0|0) & P_i^{\mu_1\ldots\mu_k}(1|0) \\ P_i^{\mu_1\ldots\mu_k}(0|1) & P_i^{\mu_1\ldots\mu_k}(1|1) \end{pmatrix} = \begin{pmatrix} 1-a_i^{\mu_1}\Gamma & a_i^{\mu_1}\Gamma \\ (1-a_i^{\mu_1})\Gamma & 1-(1-a_i^{\mu_1})\Gamma \end{pmatrix} . \tag{4.16}$$

Here, the parameter $\Gamma \in [0,1]$ controls the level of noise in the patterns below Part-level. The relation between patterns in terms of the transition matrix families $\boldsymbol{R}$ and $\boldsymbol{Q}$ is summarised in fig. 4.2.2.

With all model parameters defined, in the next section we study the statistical properties of the distances between patterns sitting on different nodes of the network, as these will determine the kinetics of the random walk and, in particular, the complexity of the search process that the random walk is meant to model. We add as a final note that if the ground-truth glossary of relevant words of an Act and its Parts where known, one could immediately compute the pattern distances and other statistics described above. This way one could potentially classify Acts by the statistical properties of their pattern distances. Modern keyword extraction methods can be used to estimate the glossary, as is briefly described in chapter 5.

4.3 Expected pattern-distances along and across branches

In this section, we provide analytical expressions, in terms of the model control parameters τ and Δ, for the expected values of two classes of pattern-distances. To be precise, we study the distances of (i) adjacent Part-level patterns $\boldsymbol{\xi}^{\mu}$ and $\boldsymbol{\xi}^{\mu+1}$ and (ii) any Part-level pattern $\boldsymbol{\xi}^{\mu_1}$ and leaf patterns of the same Part, $\boldsymbol{\xi}^{\mu_1\cdots\mu_h}$. For clarity of presentation, we state here the main results and present their derivations in appendix 4.A.

4.3.1 Overlap: Distance between neighbouring patterns

Let $\boldsymbol{\xi}^{\mu}$, $\boldsymbol{\xi}^{\mu+1}$ be two child patterns of the root pattern $\boldsymbol{\xi}$, with marginal expectations as described by eq. (4.9). With d being the Hamming distance, eq. (4.1), we are interested in the properties of the distance on Part-level $d^{\mu,\mu+1} := d\left(\boldsymbol{\xi}^{\mu}, \boldsymbol{\xi}^{\mu+1}\right)$ as we vary Δ. Appendix 4.A shows that the expectation of the pattern-distance over the distribution of patterns $\bar{d}(\Delta) := \left\langle d^{\mu,\mu+1} \right\rangle$ is given by the expression

$$\bar{d}(\Delta) = \begin{cases} \bar{d}_0 + 2\ell_c(a-\beta_l)\left((c-2)\frac{\beta_l}{a}+1\right) - 4\Delta\frac{(a-\beta_l)\beta_l}{a} & : \Delta \le \Delta_{\text{trans}} , \\ \bar{d}_0 + 2\ell_c(c-1)(a-\beta_l) - 4\Delta(a-\beta_l) & : \text{else} , \end{cases} \tag{4.17}$$

with $\Delta_{\text{trans}} = \ell_c \frac{(c-2)}{2}$ and $\bar{d}_0 = 2(1-a)\Gamma'(1-\Gamma')L$. We recall that $\ell_c = L/c$, where L and c are the pattern length and the number of children of each node, respectively.

Fig. 4.3.1 compares eq. (4.17) to the numerical average of distances $d^{\mu,\mu+1}$. The agreement is excellent, showing the accuracy of our calculation.

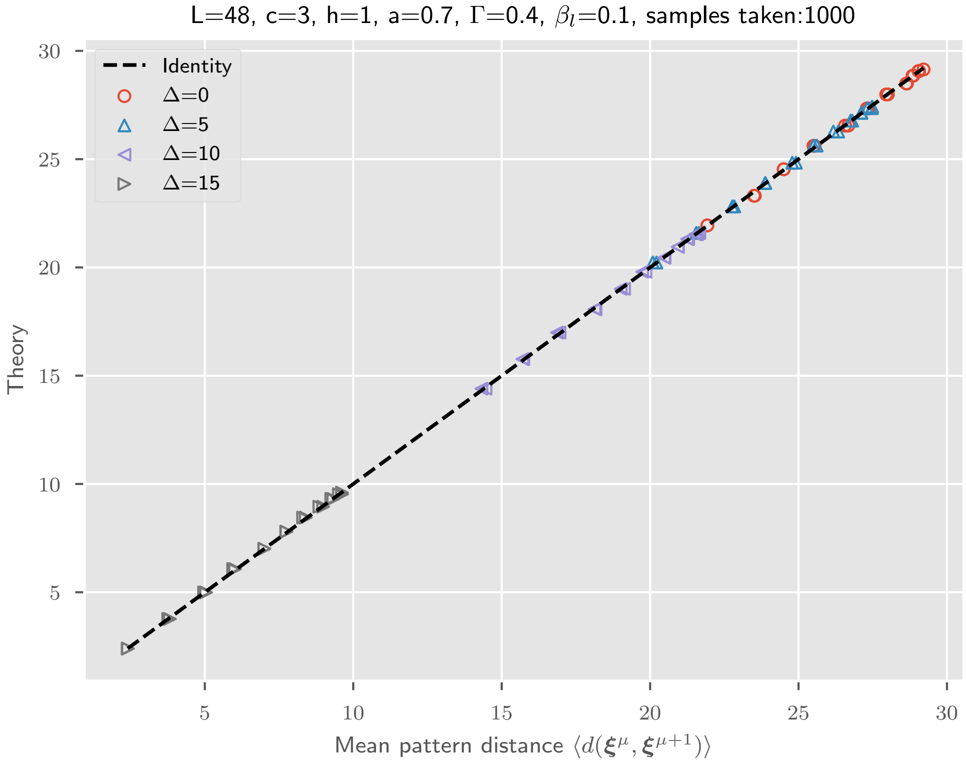


Fig. 4.3.1 Scatter plot for the expected distance $\bar{d}$ on the horizontal axis, compared to eq. (4.17) on the vertical axis. $\bar{d}$ is sampled in simulations with the parameters shown at the top.

We observe that $\bar{d}$ is a decreasing function of Δ, in the physical range of parameters $\beta_l < a$ that we identified after eq. (4.11). Hence, the parameter Δ controls the "topical overlap" between adjacent Parts as described in the previous section 4.2.

4.3.2 Tightness: Distance between ancestor and descendant patterns

Having examined the "horizontal" variation of patterns in the previous section, we now turn towards the "vertical" pattern-distance; that is to say, the expected pattern-distance between the "Part" node at the top of a certain branch, and any leaf in the same branch. To this end, let $\boldsymbol{\xi}^{\mu_1}$ be any Part-level pattern, and let $\boldsymbol{\xi}^{\mu_1\cdots\mu_{h-1}}$ be the pattern of any leaf in the same branch, i.e. descending from μ_1. Appendix 4.A shows that we have for the expected pattern-distance between $\boldsymbol{\xi}^{\mu_1}$ and $\boldsymbol{\xi}^{\mu_1\cdots\mu_{h-1}}$

$$\langle d(\boldsymbol{\xi}^{\mu_1},\boldsymbol{\xi}^{\mu_1\cdots\mu_{h-1}})\rangle = g^{(h-1)}(a_l)\,(\ell_c(c-1)-2\Delta)+g^{(h-1)}(a_h)\,(\ell_c+2\Delta)\ , \tag{4.18}$$

with

$$g^{(k)}(x) := 1-2x(1-x)\,(1-\Gamma)^k\ . \tag{4.19}$$

In fig. 4.3.2, we compare eq. (4.18) to the numerical average of $d(\boldsymbol{\xi}^{\mu_1},\boldsymbol{\xi}^{\mu_1\cdots\mu_{h-1}})$, observing an excellent agreement between the two.

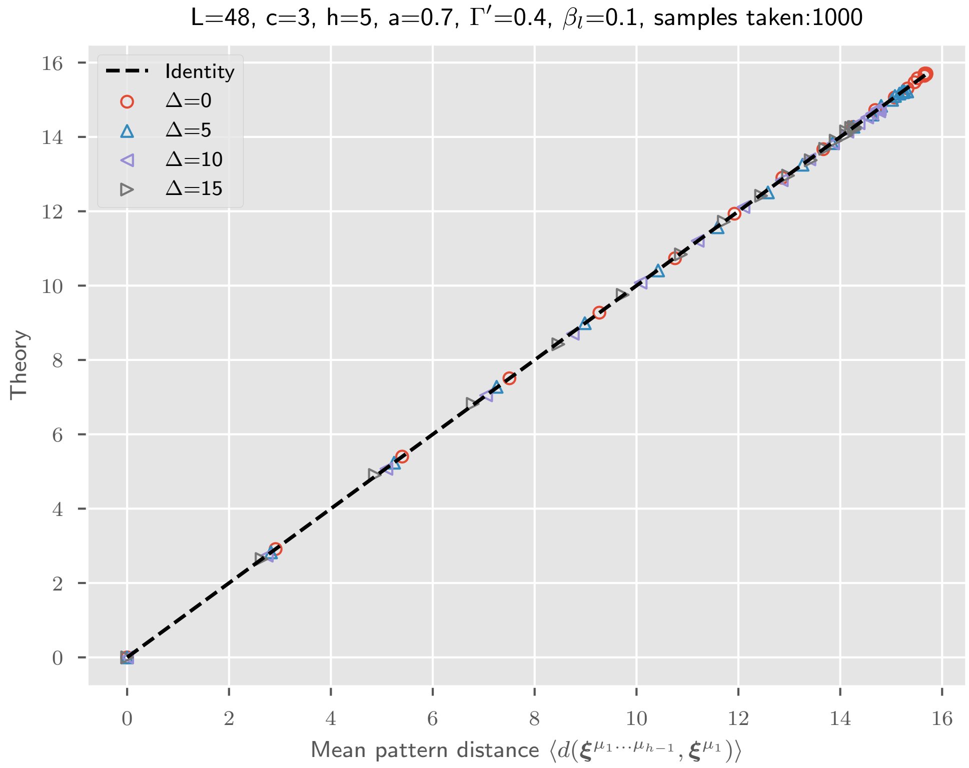


Fig. 4.3.2 Mean pattern-distance across $h=4$ levels, simulated with the shown parameters on the x-axis, and the predicted expectation value according to eq. (4.18) on the y-axis.

In contrast to Δ, the corresponding pattern-distance does not depend linearly on Γ. However, as it strictly decreases as Γ increases, we see that Γ acts as expected from a mutation rate, namely, the higher the mutation rate, the higher the distance between Part- and leaf-level patterns. Eq. (4.16) shows that upon setting $\Gamma=1$, the $\xi_i^{\mu_1}$'s and $\xi_i^{\mu_1\cdots\mu_{h-1}}$'s become independent. This is the state of least tightness and also the maximum of $\langle d(\boldsymbol{\xi}^{\mu_1},\boldsymbol{\xi}^{\mu_1\cdots\mu_{h-1}})\rangle$. Similarly, $\Gamma=0$ enforces $\langle d(\boldsymbol{\xi}^{\mu_1},\boldsymbol{\xi}^{\mu_1\cdots\mu_{h-1}})\rangle=0$, representing highest tightness. We define the *tightness* as the following monotonically decreasing function of Γ

$$\tau(\Gamma) = 1 - g^{(h-1)}(a_j)/[2a_j(1-a_j)] = (1-\Gamma)^{h-1} \ . \tag{4.20}$$

As discussed in section 4.2, this parameter controls the (expected) similarity between an ancestor-descendant pair of patterns, just as Γ controls the level of mutation from the former to the latter. Tuning τ (as opposed to Γ) allows us to achieve a better resolution in simulations for $\Gamma \to 1$.

In analogy to the relation between Δ and $\bar{d}$, the distance $\langle d(\boldsymbol{\xi}^{\mu_1}, \boldsymbol{\xi}^{\mu_1 \cdots \mu_{h-1}}) \rangle$ is linearly decreasing in τ.

4.4 Complexity measure for legal trees from approximate MFPTs

In this section, we present an expression for the *complexity* of the Act represented by the model introduced in section 4.2. For the definition of the complexity, recall that every assignment of patterns defines a transition matrix $\mathbf{q}$ for the random walker due to eq. (4.3). Denoting by $m_{rt}(\mathbf{q})$ the MFPT from root r to target t given such a transition matrix, we define the complexity as the average of $m_{rt}(\mathbf{q})$ over the distribution of patterns,

$$C := \langle m_{rt}(\mathbf{q}) \rangle \ . \tag{4.21}$$

The quantity C does not depend on any particular realisation of patterns; it reflects the "higher-level" properties encoded in the model parameters and the tree. However, evaluating the expectation in eq. (4.21) based on the formulae in eq. (2.7) or eq. (2.18) analytically is a formidable task. We avoid this difficulty by introducing the mean-field approximation

$$C_{\mathrm{MF}} := m_{rt}\left(\langle \mathbf{q} \rangle\right) \approx \langle m_{rt}(\mathbf{q}) \rangle = C \ , \tag{4.22}$$

i.e. we calculate m_{rt} for the random walker subject to the averaged transition matrix $\langle \mathbf{q} \rangle$, with the average taken over the pattern distribution. Since $\mathbf{q}$ is always a stochastic matrix, so is $\langle \mathbf{q} \rangle$, and it does indeed define a random walker on the tree. Eq. (4.22) is an approximation because m_{rt} is a non-linear function of $\mathbf{q}$, which can be seen from eq. (2.7). We dub the approximate, left-hand quantity in eq. (4.22) the *approximate complexity*.

C_{MF} is an explicit (though complicated) function of all model parameters, though we are mostly interested in its dependencies on Δ (defined in eq. (4.9)), τ (defined in eq. (4.20)) and a (defined in eq. (4.5)). Furthermore, recall the parameters h and c of the tree itself, being the height of the tree, and the number of children to a (non-leaf) node, respectively. The derivation of C_{MF} is deferred to appendix 4.E; in the following, we only summarise the results. The expressions are also implemented in Python in the `mean_field` module of the `pattern_walker` package provided in the supplementary material of this thesis.

C_{MF} decomposes into h summands

$$C_{\mathrm{MF}} = C_{\mathrm{MF}0} + C_{\mathrm{MF}1} + \sum_{K=2}^{h-1} C_{\mathrm{MF}K} \ . \tag{4.23}$$

The constituents are given by

$$\begin{aligned} C_{\mathrm{MF}0} = \frac{\Pi_0}{\pi_{v_0}} \\ \times \frac{\prod_{\mu=2}^{c} \varepsilon\left(f_\Delta^{(h-1,0,0,0;1)}, f_\Delta^{(h-1,1,1,0;\mu)}\right) + \sum_{\mu=2}^{c} \prod_{\nu=2;\nu\neq\mu}^{c} \varepsilon\left(f_\Delta^{(h-1,0,0,0;1)}, f_\Delta^{(h-1,1,1,0;\nu)}\right)}{\delta_0} \\ \times \sum_{I=1}^{h-1} \prod_{J=2}^{I-1} \frac{1}{\varepsilon\left(f_\Delta^{(h-J-1,0,0,0;1)}, f_\Delta^{(h-J+1,0,0,0;1)}\right)} \ , \end{aligned} \tag{4.24}$$

with the denominator

$$\begin{aligned} \delta_0 =& \varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h-1,1,0,0;1)}\right) \varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h,0,0,0;1)}\right) \\ &\times \prod_{\mu=2}^{c} \varepsilon\left(f_\Delta^{(h-1,0,0,0;1)}, f_\Delta^{(h-1,1,1,0;\mu)}\right) \end{aligned} \tag{4.25}$$

for $K = 0$ and

$$\begin{aligned}
C_{\mathrm{MF}1} =& \frac{\Pi_1}{\pi_{v_1}} \frac{1}{\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h-1,1,0,0;1)}\right)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h,0,0,0;1)}\right)} \\
&\times\left((c-1)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h-1,1,0,0;1)}\right) + \varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h,0,0,0;1)}\right)\right. \\
&\quad\left. + \varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h-1,1,0,0;1)}\right)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h,0,0,0;1)}\right)\right) \\
&\times \sum_{I=2}^{h-1} \prod_{J=2}^{I-1} \frac{1}{\varepsilon\left(f_\Delta^{(h-J-1,0,0,0;1)}, f_\Delta^{(h-J+1,0,0,0;1)}\right)}, \qquad (4.26)
\end{aligned}$$

$$\begin{aligned}
C_{\mathrm{MF}K} =& \frac{\Pi_K}{\pi_{v_K}} \left[c + \varepsilon\left(f_\Delta^{(h-K-1,0,0,0;1)}, f_\Delta^{(h-K+1,0,0,0;1)}\right)\right] \\
&\times \sum_{I=K+1}^{I-1} \prod_{J=K}^{I-1} \frac{1}{\varepsilon\left(f_\Delta^{(h-J-1,0,0,0;1)}, f_\Delta^{(h-J+1,0,0,0;1)}\right)}, \qquad (4.27)
\end{aligned}$$

where the fractions Π_K/π_{v_K} for $K > 1$, Π_1/π_1 and Π_0/π_0 are given in appendix 4.E in eqs. (4.74), (4.75) and (4.77), respectively. Moreover, the weight function ε are defined in eq. (4.66), and the f_Δ's in eqs. (4.51) and (4.60).

We shall now verify our expression for C_{MF} by comparing it to C_{MF} obtained from numerical simulations. In the simulations below, we fixed a tree with $c = 3$ and $h = 4$, as well as the parameters $L = 48$, $a = 0.7$, $\beta_l = 0.07$ and $\Gamma' = 0.3$. For each given pair of values for Δ and τ, we calculate C_{MF} with the following procedure: Generate a set of patterns, and subsequently a transition matrix $\mathbf{q}$ as described in section 4.2; repeat 100 times and take the average of the resulting matrices, denoted $\langle\mathbf{q}\rangle_{\mathrm{emp}}$; this average approximates $\langle\mathbf{q}\rangle$. The value of $m_{rt}(\langle\mathbf{q}\rangle_{\mathrm{emp}})$ calculated numerically using eq. (2.7) is our benchmark for C_{MF} in eq. (4.23). The two values of C_{MF} are plotted against each other in fig. 4.4.1. We see from the figure that the agreement is excellent, in spite of the fact that the derivation in appendix 4.D uses two approximations (eqs. (4.51) and (4.68)) to obtain explicit expressions for the entries of $\langle\mathbf{q}\rangle$.

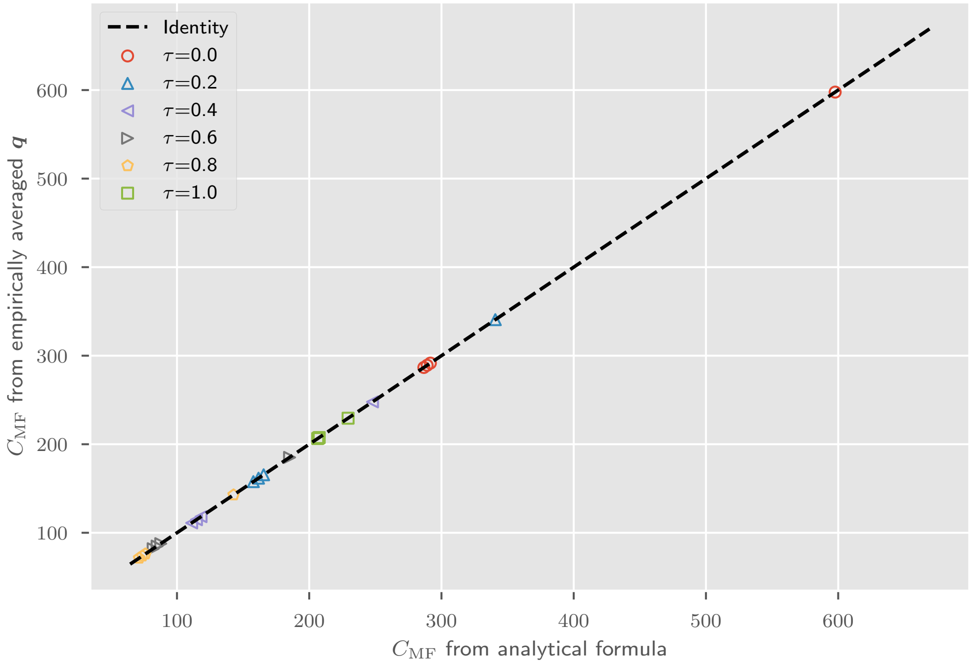


Fig. 4.4.1 Complexity C_{MF} obtained from eq. (4.23) vs C_{MF} calculated using eq. (2.7) with an empirically transition matrix $\langle \mathbf{q} \rangle_{\mathrm{emp}}$ averaged over 100 realisations for each set of parameters. Each data point refers to a pair (Δ, τ), grouped by colour and symbol according to τ. For all points, we fixed $a = 0.7$ and $\Gamma' = 0.3$.

In the next section, we proceed by testing the goodness of the approximation eq. (4.22) numerically.

4.5 Simulations and observations

This section contains numerical validations of the approximation in eq. (4.22) by comparing C_{MF} to C, computed numerically as an average over quenched MFPTs $m_{rt}(\mathbf{q})$. We then proceed to consider the behaviour of C and C_{MF} as we vary the model parameters.

For given values of the parameters a, Δ and τ, we calculate $C(a, \tau, \Delta)$ by repeating the following steps 500 times: Generate a set of patterns, and subsequently the transition matrix $\mathbf{q}$ according to eq. (4.3); record the value $m_{rt}(\mathbf{q})$ calculated numerically using eq. (2.7). The average of these values is approximately (due to the finite size of the sample) equal to $C(a, \tau, \Delta)$. We fix a tree with $c = 3$ and $h = 4$, as well as the parameters $L = 48$, $\beta_l = 0.07$ and $\Gamma' = 0.3$. With these values for c and h, the total number of nodes is $N = 121$ and the MFPT for the diffusive random walker (i.e. all edges unweighted) is $m_{rt}^{\mathrm{diff}} = 848$.

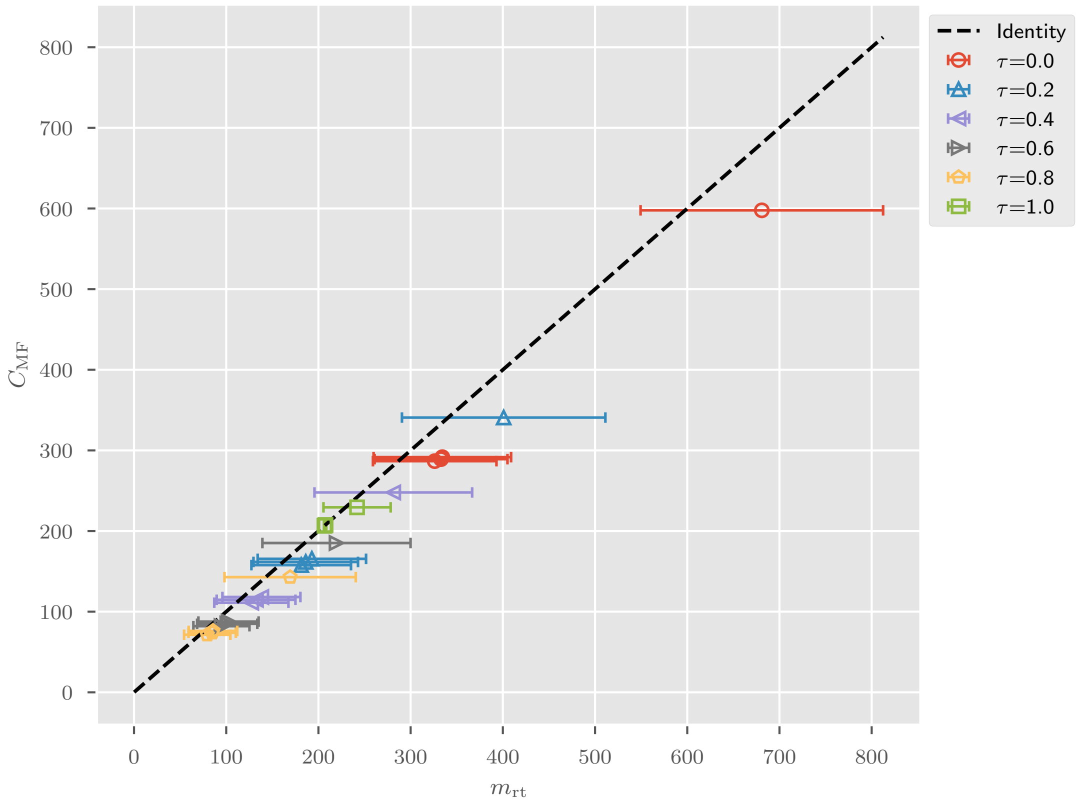


Fig. 4.5.1 Complexity C_{MF} obtained via section 4.4 vs C, being the average MFPT $\langle m_{rt}(\mathbf{q})\rangle$, where the average is taken over the distribution of patterns. For each realisation of $\mathbf{q}$, $m_{rt}(\mathbf{q})$ was computed using eq. (2.7). Error bars represent one standard deviation of $m_{rt}(\mathbf{q})$. Each data point refers to a pair (Δ, τ), grouped by colour and symbol according to τ. For all points, we fixed $a = 0.7$ and $\Gamma' = 0.3$.

In fig. 4.5.1, we directly compare C_{MF} as per eq. (4.23) to C in a scatterplot for fixed a. The standard deviation of $m_{rt}(\mathbf{q})$ with respect to variations in $\mathbf{q}$ is indicated as error bars. The plot confirms for all parameters considered that eq. (4.22) leads to a systematic underestimation while accurately reflecting the correct trend.

Next, we analyse the dependence of C_{MF} on the different parameters of our model. Fig. 4.5.2 plots C_{MF} and C as a function of $a = \langle \xi_j \rangle$, the expectation of root-level bits ξ_j. The dashed lines represent the mean-field approximation C_{MF} in eq. (4.23), the symbols the value of C as obtained in the beginning of this section. Fig. 4.5.2 confirms that C_{MF} tracks C faithfully, also for varying a.

For small values of τ, i.e. little vertical coherence between patterns, the complexity C_{MF} first shows a slight increase as a function of a (approximately in the interval $0 < a < 0.2$) before a more pronounced decrease to about $1/3$ of its maximum (for $a > 0.2$). As τ approaches 1, the curves for C_{MF} first become notably flatter and lower, which is as

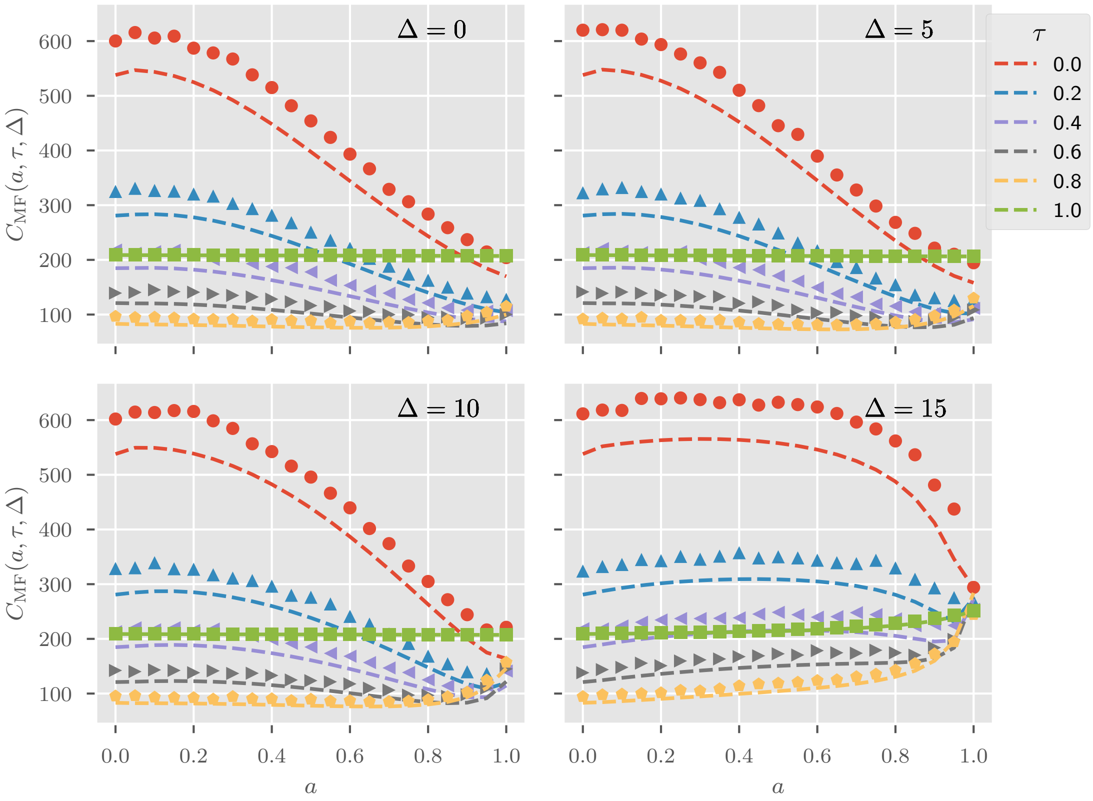


Fig. 4.5.2 Complexity C_{MF} as a function of a for different values of τ and Δ. The points show the numerical average for C on the right hand side of eq. (4.22), taken over 500 realisations of $\mathbf{q}$.

expected since higher vertical coherence is more likely to put the reader on the right track faster. This effect is only seen up to $\tau = 0.8$, as too high coherence – given if $\tau \to 1$ – forces all patterns within the same Part to be equal, which does not help the reader navigate at all.

Fig. 4.5.2 suggests the following conclusion: For fixed, low values of τ, the complexity can be minimised by increasing a as much as possible. Since a represents the keyword density of the root pattern, this means that the Title of the represented Act should reference as many keywords as possible. For high values of τ, the complexity increases with a, though the increase is far less pronounced than the decrease at low τ.

Fig. 4.5.3 shows C_{MF} as dashed lines and C as symbols as functions of Δ. The panels and different curves per panel correspond to different values for a and τ, respectively. We make the same observation as above about the systematic underestimation incurred in eq. (4.22), although in addition to τ, the offset also seems to decrease with $a \to 1$.

C is largely constant in Δ for $\Delta \leq 8$. Beyond this value, C begins to increase with Δ, except for the lowest tested value $a = 0.05$. C increases by about a factor of 2 for $\Delta > 8$. Strikingly, C_{MF} and C show a slight *decrease* up to $\Delta \leq 8$ for $a = 0.8$, which is contrary to

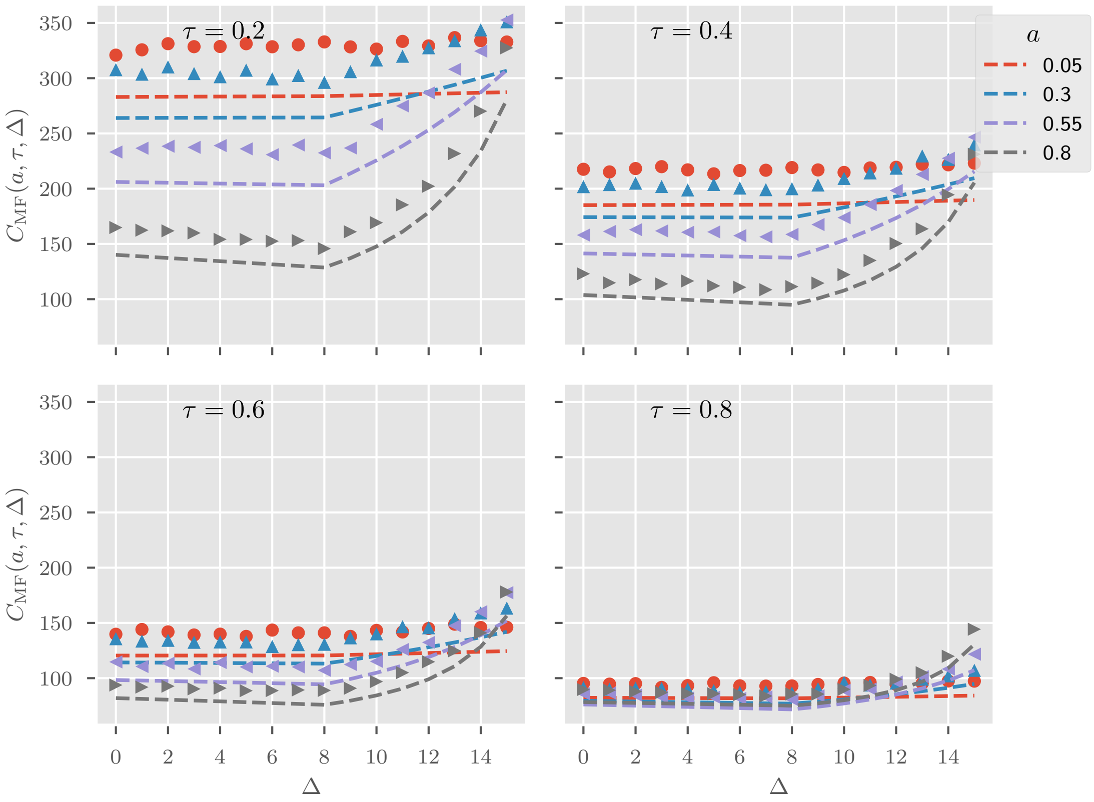


Fig. 4.5.3 Complexity C_{MF} as a function of Δ for different values of a and τ. The points show the numerical average for C on the right hand side of eq. (4.22), taken over 500 realisations of $\mathbf{q}$.

our intuition that higher overlap between adjacent Parts should increase the complexity as it leads to more initial missteps of the random walker. However, the observed decrease is minor compared to the observed increase exhibited at higher Δ. The fact that such increase in complexity is less significant for higher τ is again in line with our expectation that a more “vertically coherent” text should be overall easier to navigate. We note that the range of values of C over Δ is less than the one over a, shown in fig. 4.5.2, demonstrating that Δ has weaker influence.

We deduce from fig. 4.5.3 that in order to reduce C, Δ should not be chosen too high. Further simulations suggest that depending on the coordination number c of the tree, the complexity can also rise if Δ is chosen too low. This means that C has a local minimum in Δ, which represents the optimal keyword overlap between adjacent Parts.

Fig. 4.5.4 presents C_{MF} as a function of τ as dashed lines, with different curves and panels representing different values for Δ and a, respectively. Different symbols are used to represent the values of $C = \langle m_{rt}(\mathbf{q})\rangle$. The plot shows that C_{MF} has a pronounced local minimum in τ between 0.8 and 0.9 for all values of Δ and a tested. At $\tau = 1$, the random walker becomes diffusive whenever inside a Part, because all patterns within a given Part

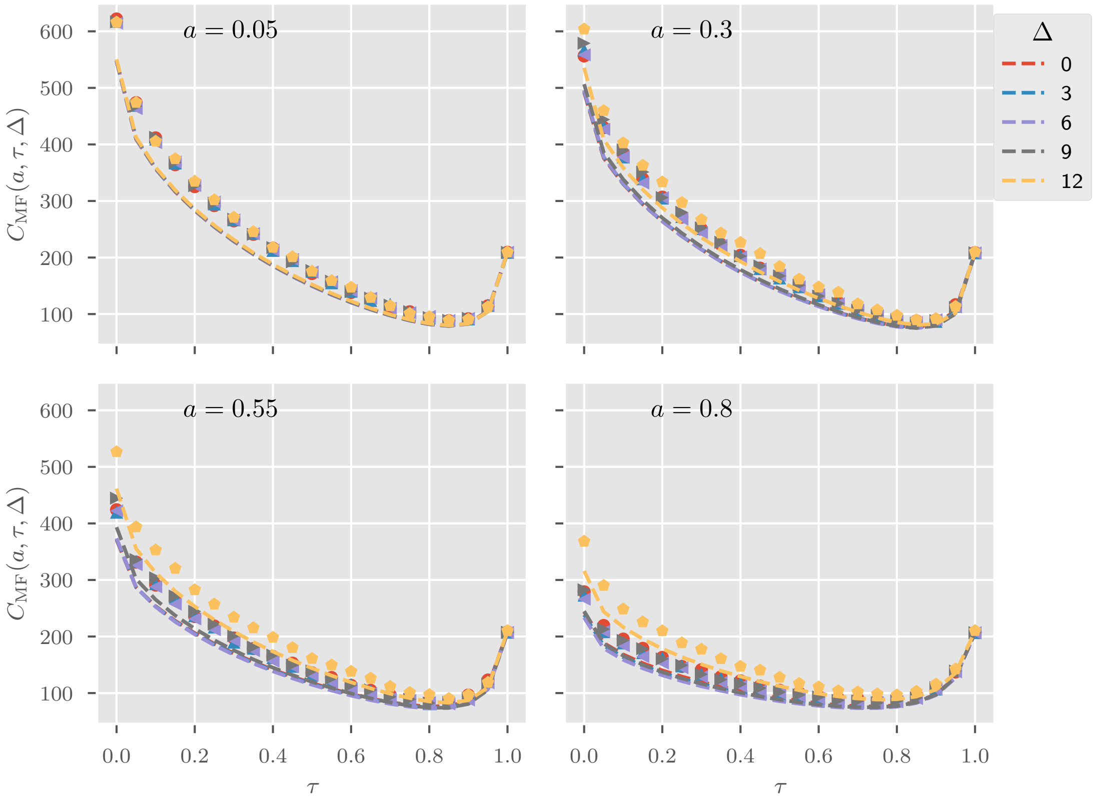


Fig. 4.5.4 Complexity C_{MF} as a function of τ for different values of Δ and a. The points show the numerical average for C on the right hand side of eq. (4.22), taken over 500 realisations of $\mathbf{q}$.

are identical. Therefore, it makes sense that the minimal complexity should not be realised at this value, since the patterns cease to guide the walker to the target node t. The range of variation of C over τ is comparable to that over a, as shown in fig. 4.5.2.

To conclude the analysis of fig. 4.5.4, we summarise that C may be minimised by choosing an appropriately high value for τ, which should not be too close to 1. This means that one should allow the keywords within one Part to vary slightly, to avoid keyword patterns that are either almost identical or approximately independent.

Figs. 4.5.2, 4.5.3 and 4.5.4 suggest that an increasing Δ, or decreasing τ or a, results in a higher rate of mistakes made by the walker, thus leading to a higher searching time. Since Δ has its primary effect on the root (Act) level, it is dominated by τ, which controls noise on all (bar the Act) levels. Since a also affects the values of a_h and a_l, it has an effect on all levels as well; accordingly, we observe that varying a and τ have comparable effects on C, and dominate variations over Δ. Form here, we conclude that the priority should be on maximising the tightness τ and keyword frequency a to reduce the complexity of the modelled ensemble of Acts.

The present section shows that C_{MF} faithfully reproduces the trends of C for varying τ, Δ and a. This entails a significant benefit because it allows us to optimise the parameters of the model with respect to C_{MF} without the need for costly simulations. As a consequence, one can imagine optimising a real legal text, by estimating its parameter values for our model and tweaking the text and layout to minimise C_{MF}.

Here, we have not considered the effects of the parameters c, h, L and Γ'. The former two of these deserve a word of caution: The number of nodes increases as c^{h+1}, and m_{rt} for the regular random walker as hc^{h+1}. Therefore, without appropriate rescaling, the values of C and C_{MF} do not allow for the comparison of graphs of different size.

4.6 Conclusions and outlook

We presented a quantitative theory of informational complexity of legal trees by analysing a random walker model for the retrieval of information planted in the leaves of a legal tree. The model assumes that the reader proceeds by keyword affinity, such that they are drawn towards nodes whose content looks similar to the target information. The searched text is generated randomly, with two main parameters controlling its horizontal and vertical coherence. Our analysis and numerical simulations show that these properties of the text have the desired effect on the random walker: With high vertical coherence, the content of the leaves is well-reflected in the top Items (Parts) of the text, and the reader finds their target more quickly. High horizontal coherence, on the other hand, means that different Parts are difficult to discern, leading to more initial errors by the reader.

As a measure of complexity, we propose the MFPT of the random reader from the root of the tree to the predefined target information; it gives an intuitive account of how difficult it is for a typical reader to navigate the legal text by following only local information. Similarly, MFPTs have also been employed successfully to assess the heterogeneity and transport properties of social and other complex networks [66].

So far, we have limited our analysis to trees, where we were able to compute our complexity measure analytically using simple approximations. However, considering the various relations between different paragraphs of a legal text, legal networks usually exhibit interconnections on top of the tree-like backbone. A direct generalisation of the present work can be the inclusion of cross-references which introduce a dependency between the citing and the cited paragraph, and have to to be considered as the law is maintained. Cross-references can also lead to detours *and* act as shortcuts. In fact, studies of European civil law have found that these legal systems can exhibit small-world properties [110]. In other studies, more general directed acyclic graphs are used to represent citation networks, e.g. the network of precedents in the US [31] or of the total citation network in a system of statute law [32, 35]. Recently, [38] has elaborated on the similarity between legal and software systems, drawing from best practices on the latter to propose improvements on the former. The framework developed in the present chapter may prove useful in providing a more quantitative ground to assess the methods in these lines of research as well.

Moreover, our model definitions rely on a number of assumptions on the details of how keywords are distributed over the text: firstly, the definition of overlap assumes that the Parts of an Act are ordered in such a way that consecutive pairs realise the maximal overlap in that Act, and that this overlap is the same for all adjacent pairs. Secondly, we assume that below the Part level, the activation probability for every keyword is fixed within each Part, which might be unrealistic for "deep" laws with many levels. Relaxing these assumptions – introduced for the sake of computational simplicity – may render the model even more realistic and general.

We have modelled a reader as a Markovian random walker, that is to say that it is "memoryless". To replicate the behaviour of a real reader more closely, more general types of walks (e.g. self-avoiding walks [112]) might be appropriate.

Finally, on the side of our analysis, it should be possible to refine the approximation in eq. (4.22). Perhaps, more of the information contained in the pattern-dependent $m_{rt}(\mathbf{q})$ can be exploited by carrying the analysis beyond its mean to higher moments.

We can derive three broad and intuitive lessons from the results in section 4.5 to reduce the "complexity" of a legal tree: (i) Keywords at the lowest levels should be reflected at higher levels, i.e. a legal text should be "tightly" formulated. Yet, it is possible to make it overly tight, which happens when all text Items look too similar to one another. This situation is identical to giving no clues at all to the reader. (ii) Parts should be well separated by their keywords (and hence by topic); some keyword overlap is acceptable, as long as sufficiently many Part-specific keywords remain to guide the reader. (iii) Text at higher levels should not be too sparse. If high-level entries contain only a small number of keywords (such as a short headline), little information about its subordinate Items can be conveyed (except by interpretation, e.g. through association of keywords and related words). A higher keyword frequency at the top levels saves the reader time-costly detours into wrong Parts.

After this thorough treatment of theoretical aspects of our model, in the final chapter 5, we describe a procedure by which one can actually extract the keywords from a given Act and estimate the model parameters. We can then compute C_{MF} and related quantities for an existing hierarchical collection of legal documents.

Appendix

Appendix 4.A Expected pattern distances

In this appendix, we derive the expected pattern distances eqs. (4.17) and (4.18) in section 4.3. Before we start, it will be useful to recall that X is a binomial random variable (denoted $X \sim \text{Binom}\,(p\,,n)$) if it has the probability mass function (PMF)

$$\mathbb{P}\{X = x\} = \binom{n}{x} p^x (1-p)^{n-x} \tag{4.28}$$

with expectation

$$\mathbb{E}(X) = np\ . \tag{4.29}$$

We begin with the expectation of the Hamming distance on Part-level, $d^{\mu,\mu+1} := d\left(\boldsymbol{\xi}^{\mu}, \boldsymbol{\xi}^{\mu+1}\right)$. From the model definitions in section 4.2, we see that this is a combination of binomial random variables, for either $\xi^{\mu}_j = \xi^{\mu+1}_j$ or $\xi^{\mu}_j \neq \xi^{\mu+1}_j$, and the probabilities of both events depend on whether both bits have the same marginal expectation a_h (or a_l), or if one is equal to a_h and the other equal to a_l.

Let the numbers of indices j such that $\left\langle \xi^{\mu}_j \right\rangle = \left\langle \xi^{\mu+1}_j \right\rangle = a_h$ and $\left\langle \xi^{\mu}_j \right\rangle = \left\langle \xi^{\mu+1}_j \right\rangle = a_l$ be L_{hh} and L_{ll}, respectively, and let the number of indices with $\left\langle \xi^{\mu}_j \right\rangle \neq \left\langle \xi^{\mu+1}_j \right\rangle$ be L_{hl}. The L's are functions of Δ (cf. fig. 4.2.3)

$$\begin{aligned}
L_{hh}(\Delta) &= \begin{cases} 2\Delta & : \Delta \le \ell_c \frac{(c-2)}{2}\ , \\ 4\Delta - \ell_c(c-2) & :\ \text{else}\ , \end{cases} \\
L_{ll}(\Delta) &= \begin{cases} \ell_c(c-2) - 2\Delta & : \Delta \le \ell_c \frac{(c-2)}{2}\ , \\ 0 & :\ \text{else}\ , \end{cases} \\
L_{hl}(\Delta) &= \begin{cases} 2\ell_c & : \Delta \le \ell_c \frac{(c-2)}{2}\ , \\ 2\ell_c(c-1) - 4\Delta & :\ \text{else}\ , \end{cases}
\end{aligned} \tag{4.30}$$

where L is the number of bits per patterns, c is the number of Parts, and $\ell_c = L/c$. The reason for the presence of different cases is that the inequality

$$(\mu-1)\ell_c - \Delta + L < (\mu+1)\ell_c + \Delta \tag{4.31}$$

is true if and only of $\Delta > \ell_c \frac{c-2}{2}$. Now $(\mu-1)\ell_c - \Delta + L$ is the "left" boundary of the a_h-domain of Part μ after applying the L-periodicity of eq. (4.9), and $(\mu+1)\ell_c + \Delta$ is the "right" boundary of the a_h-domain of Part $\mu+1$; therefore, $\Delta \le \ell_c \frac{c-2}{2}$ implies $\left\langle \xi^{\mu}_j \right\rangle = \left\langle \xi^{\mu+1}_j \right\rangle = a_h$ if and only if $\mu\ell_c - \Delta < j \le \mu\ell_c + \Delta$, while for $\Delta > \ell_c \frac{c-2}{2}$ there is a second set of solutions given by $(\mu-1)\ell_c - \Delta + L < j \leq (\mu+1)\ell_c + \Delta$ (refer to fig. 4.2.3 for a schematic illustration). Notice that with $\Delta > \ell_c \frac{c-2}{2}$, we necessarily have $L_{ll} = 0$, i.e. the a_h-domains of both patterns together cover all of $\{1, \ldots, L\}$.

The probabilities of $\{\xi_j^\mu \neq \xi_j^{\mu+1}\}$ can be derived using the law of total probability and the conditional independence of the Part patterns given the root pattern $\boldsymbol{\xi}$,

$$\begin{aligned}
\mathbb{P}\left\{\xi_j^\mu \neq \xi_j^{\mu+1}\right\} =& \sum_{z=0}^{1} \mathbb{P}\left\{\xi_j^\mu \neq \xi_j^{\mu+1} \middle| \xi_j = z\right\} \mathbb{P}\left\{\xi_j = z\right\} \\
=& \sum_{z=0}^{1} \mathbb{P}\left\{\xi_j = z\right\} \Big[\mathbb{P}\left\{\xi_j^\mu = z \middle| \xi_j = z\right\} \mathbb{P}\left\{\xi_j^{\mu+1} \neq z \middle| \xi_j = z\right\} \\
&+ \mathbb{P}\left\{\xi_j^\mu \neq z \middle| \xi_j = z\right\} \mathbb{P}\left\{\xi_j^{\mu+1} = z \middle| \xi_j = z\right\} \Big] , \qquad (4.32)
\end{aligned}$$

where the factors in the square brackets are given by the elements of the $\boldsymbol{R}_j$'s in eq. (4.8) for each of the combinations of a_h and a_l. In fact, by definition of $\boldsymbol{R}_j$, we have

$$\begin{aligned}
\mathbb{P}\left\{\xi_j^\mu \neq \xi_j^{\mu+1} \middle| \xi_j = 0\right\} =& \boldsymbol{R}_j^\mu(0|0)\boldsymbol{R}_j^{\mu+1}(1|0) + \boldsymbol{R}_j^\mu(1|0)\boldsymbol{R}_j^{\mu+1}(0|0) \\
=& 2(1-\Gamma')\Gamma' \\
\mathbb{P}\left\{\xi_j^\mu \neq \xi_j^{\mu+1} \middle| \xi_j = 1\right\} =& \boldsymbol{R}_j^\mu(0|1)\boldsymbol{R}_j^{\mu+1}(1|1) + \boldsymbol{R}_j^\mu(1|1)\boldsymbol{R}_j^{\mu+1}(0|1) \qquad (4.33) \\
=& \frac{a - a_j^\mu + (1-a)\Gamma'}{a} \frac{a_j^{\mu+1} - (1-a)\Gamma'}{a} \\
&+ \frac{a_j^\mu - (1-a)\Gamma'}{a} \frac{a - a_j^{\mu+1} + (1-a)\Gamma'}{a} . \qquad (4.34)
\end{aligned}$$

With the decomposition in eq. (4.32), and using that $\langle \xi_j \rangle = \mathbb{P}\{\xi_j = 1\} = a$ for all j, this produces the marginal probability

$$\begin{aligned}
\mathbb{P}\left\{\xi_j^\mu \neq \xi_j^{\mu+1}\right\} =& 2(1-a)\Gamma'(1-\Gamma') \\
&+ \frac{1}{a} \Big[(a - a_j^\mu + (1-a)\Gamma')(a_j^{\mu+1} - (1-a)\Gamma') \\
&+ (a_j^\mu - (1-a)\Gamma')(a - a_j^{\mu+1} + (1-a)\Gamma') \Big] . \qquad (4.35)
\end{aligned}$$

Considering now all L_{hh} bits j for which $\left\langle \xi_j^\mu \right\rangle = \left\langle \xi_j^{\mu+1} \right\rangle = a_h$, the above expression reduces to $2(1-a)(1-\Gamma')\Gamma'$ since $a_h = (1-a)\Gamma' + a$ (given in eq. (4.12)) sets the other two summands to zero. The sum of the distances $|\xi_j^\mu - \xi_j^{\mu+1}|$ for these bits then forms a binomial random variable with "success" probability $2(1-a)(1-\Gamma')\Gamma'$. Similarly, we can treat the other bits in two groups of size L_{lh} and L_{ll}, respectively, as described in the beginning of this appendix. For this purpose, it is useful to recall from eq. (4.11) that $a_l - (1-a)\Gamma' = \beta_l$. The total distance $d^{\mu,\mu+1}$ is then given by a sum of binomial random variables

$$\begin{aligned} d^{\mu,\mu+1} \sim & \mathrm{Binom}\left(2(1-a)\Gamma'(1-\Gamma'), L_{hh}\right) \\ & + \mathrm{Binom}\left(2(1-a)\Gamma'(1-\Gamma') + a - \beta_l, L_{hl}\right) \\ & + \mathrm{Binom}\left(2(1-a)\Gamma'(1-\Gamma') + 2\frac{(a-\beta_l)\beta_l}{a}, L_{ll}\right) , \end{aligned} \tag{4.36}$$

expressed in terms of β_l for conciseness.

The expected pattern-distance between neighbours is readily calculated using the linearity of $\langle\cdot\rangle$, the above characterisation for $d^{\mu,\mu+1}$ and the expectation for binomial random variables, eq. (4.29). These ingredients give the result reported for $\bar{d} = \left\langle d^{\mu,\mu+1}\right\rangle$ in eq. (4.17). We can see that $\bar{d}$ is a decreasing function in Δ with maximum and minimum

$$\bar{d}_{\max} = \bar{d}_0 + 2\ell_c(a-\beta_l)\left((c-2)\frac{\beta_l}{a} + 1\right) , \tag{4.37}$$

$$\bar{d}_{\min} = \bar{d}_0 , \tag{4.38}$$

with $\bar{d}_0 = 2(1-a)\Gamma'(1-\Gamma')L$. Moreover, $\bar{d}$ is a combination of two affine linear function, with the transition between the two occuring at $\Delta_{\text{trans}} = \ell_c\frac{c-2}{2}$ with value

$$\bar{d}_{\text{trans}} = \bar{d}(\Delta_{\text{trans}}) = \bar{d}_0 + 2\ell_c(a-\beta_l) . \tag{4.39}$$

We now examine the expected distance of two patterns $\boldsymbol{\xi}^\mu$, $\boldsymbol{\zeta}$, where the former represents Part μ, and the latter is a descendant of the former, at distance k. We are particularly interested in the situation where $\boldsymbol{\zeta}$ is a leaf-level pattern, i.e. the edge-distance between the two is $k = h-1$. To determine $\langle d(\boldsymbol{\zeta},\boldsymbol{\xi}^\mu)\rangle$, we need to know the probabilities of the events $\{\xi_j^\mu \neq \zeta_j\}$ which we calculate from the k-th power of $\boldsymbol{Q}$ defined in eq. (4.16)

$$\boldsymbol{Q}^k(a_j^\mu) = \begin{pmatrix} 1 - a_j^\mu\left(1-(1-\Gamma)^k\right) & a_j^\mu\left(1-(1-\Gamma)^k\right) \\ (1-a_j^\mu)\left(1-(1-\Gamma)^k\right) & 1-(1-a_j^\mu)\left(1-(1-\Gamma)^k\right) \end{pmatrix} , \tag{4.40}$$

where a_j^μ is the marginal expectation $\left\langle \xi_j^\mu \right\rangle = a_j^\mu$. As a_j^μ can take the values a_l and a_h, as defined in eqs. (4.11) and (4.12), the j-th bits are different with probability

$$\begin{aligned}\mathbb{P}\left\{\xi_j^\mu \neq \zeta_j\right\} &= \mathbb{P}\left\{\zeta_j = 0|\xi_j^\mu = 1\right\}\mathbb{P}\left\{\xi_j^\mu = 1\right\} + \mathbb{P}\left\{\zeta_j = 1|\xi_j^\mu = 0\right\}\mathbb{P}\left\{\xi_j^\mu = 0\right\} \\ &= \boldsymbol{Q}^k(0|1)a_j^\mu + \boldsymbol{Q}^k(1|0)(1-a_j^\mu) \\ &= 2a_j^\mu(1-a_j^\mu)\left[1-(1-\Gamma)^k\right] =: g^{(k)}(a_j^\mu)\ . \end{aligned} \tag{4.41}$$

For the full pattern-distance composed by all bits, we have to take both possible values, a_l and a_h, for a_j^μ into account. Again, the full pattern-distance is a sum of two independent binomial random variables

$$d(\boldsymbol{\zeta},\boldsymbol{\xi}^\mu) \sim \mathrm{Binom}\left(g^{(k)}(a_h)\,, \ell_c + 2\Delta\right) + \mathrm{Binom}\left(g^{(k)}(a_l)\,, \ell_c(c-1) - 2\Delta\right)\ , \tag{4.42}$$

with expectation given by eq. (4.29),

$$\langle d(\boldsymbol{\zeta},\boldsymbol{\xi}^\mu)\rangle = g^{(k)}(a_l)\,(\ell_c(c-1)-2\Delta) + g^{(k)}(a_h)\,(\ell_c+2\Delta)\ . \tag{4.43}$$

For $k = h-1$, we obtain the result given in the main text in eq. (4.18).

Appendix 4.B 5-Tuple labels for paths and nodes

The statistical properties of the pattern-distance of any node to the target are determined by the location of the node in the tree. The determining criteria are (i) the edge-distance between the node and the target, (ii) whether the root lies on the shortest path between the node and the target, and (iii) the Part containing the node, as Part patterns have different switching probabilities over the edges incident to the root.

We will label directed paths along the edges of the tree by 5-tuples, which encode features of the paths that influence the statistical properties of pattern-distance between the start and end nodes of these paths.

Let P be a directed path of length ℓ such that at no point following its direction, one moves closer (in the sense of edge-distance) to t (see fig. 4.B.1 for examples). Without loss of generality, we may assume that the Parts μ are enumerated such that 1 is the Part containing t. The remaining Parts may be in any order. We assign to P a 5-tuple $P \sim (P_1, P_2, P_3, P_4; P_5)$, where P_1 is the number of edges of P connecting its starting node to the Part node $\mu = 1$. We set $P_2 = 1$ if the edge $(1, r)$ from Part 1 to the root lies on P, and otherwise we have $P_2 = 0$. We define $P_3 = 1$ if P has an edge (r, μ) from the root to a Part node with $\mu \geq 2$, and otherwise we set $P_3 = 0$. P_4 counts the number of edges of P

connecting its end node to the nearest Part node $\mu \geq 2$. P_5 records the Part μ containing the end node of P.

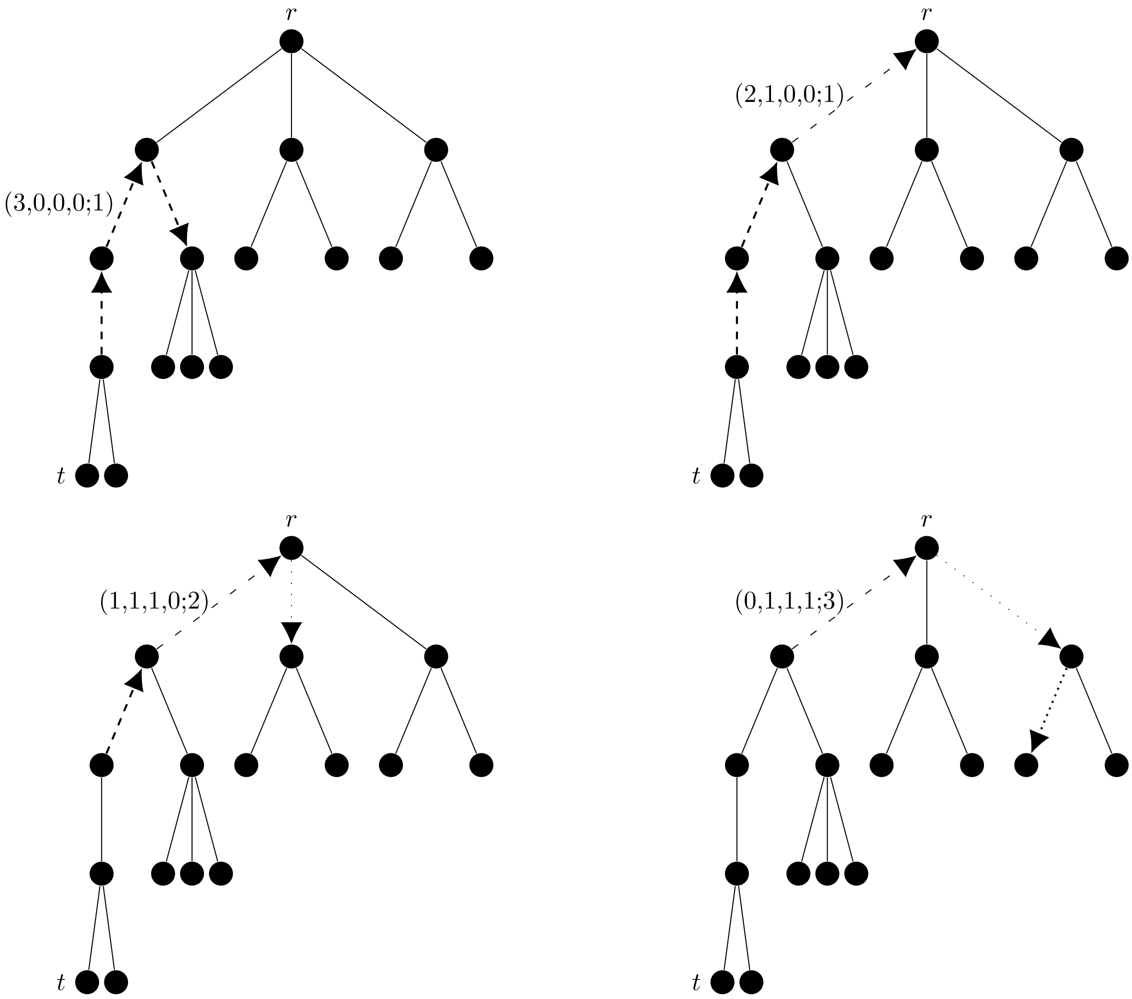


Fig. 4.B.1 Labels for different paths of length 3 oriented away from t, shown by arrows. The number of densely dashed and densely dotted arrows give the first and fourth coordinate, respectively. The number of thinner, loosely dashed and loosely dotted arrows give the second and third coordinates, respectively. The latter two are always associated with the root, so the second and third entries are both either 0 or 1. Note that the first four coordinates always sum to the path-length.

In practice, the label $(P_1, P_2, P_3, P_4; P_5)$ can be determined by making the following observation. If the root node r lies on P and ℓ is the length of the path P, there are two numbers $k, m \geq 0$ such that P consists of k edges in Part 1 and $m = \ell - k$ edges in exactly one other Part, say μ (we count the edge (r, μ) as belonging to Part μ). By construction, we have $k = 0$ if and only if P starts in the root node r and $m = 0$ if and only if P ends in r.

Then, we label P as

$$P \sim \begin{cases} (k-1,1,0,0;1) & : P \text{ ends with } r, \\ (0,0,1,m-1;\mu) & : P \text{ begins with } r, \\ (k-1,1,1,m-1;\mu) & : \text{else.} \end{cases} \tag{4.44}$$

If r does not lie on P, we assign the label

$$P \sim \begin{cases} (p,0,0,0;1) & : P \text{ lies in the target-Part }, \\ (0,0,0,p;\mu) & : P \text{ lies in Part } \mu > 2. \end{cases} \tag{4.45}$$

Various examples for labelled paths are given in fig. 4.B.1. In all cases, the sum of the first four indices equals the length of the path P.

We anticipate that our notation will not be well defined for most directed paths in the tree, but it will be well defined for those paths relevant to the analysis in this chapter. Moreover, the above definition does not identify paths uniquely; for instance, the label $(k,0,0,0;1)$ applies to all paths of length k not leaving the target-Part and with the additional constraint of being directed away from t. However, due to the constraint on the path direction, the pattern-distances between the start and end nodes of each path are identically distributed, and the distribution is determined by $(k,0,0,0;1)$.

We can now use these labels to refer to classes of nodes as well. Given a class of paths with label $(k,l,m,n;\mu)$, consider only those paths P starting at t. We may then label nodes v via the labels of the shortest path starting in t and ending in v. Examples are shown in fig. 4.B.2.

We stress that by $(k,l,m,n;\mu)$ we always refer to shortest paths, which are unique in trees, therefore the label for each node is always well-defined in this way. However, just as paths are not uniquely identified by their label, the same is true for nodes labelled in the way just introduced – for instance, fig. 4.B.2 shows two nodes that can be described by $(2,0,0,0;1)$. The only nodes fully identifiable by their labels are $t \sim (0,0,0,0;1)$ and $r \sim (h-1,1,0,0;1)$. However, the “resolution” provided by these labels is sufficient for a statistical description of the pattern-distances to t for each node in the graph. This is the subject of the next appendix.

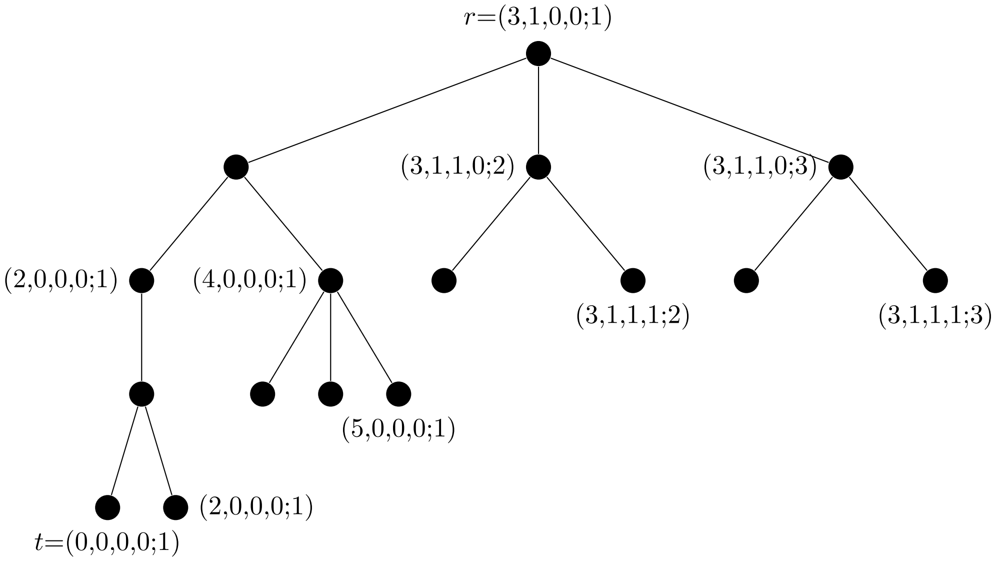


Fig. 4.B.2 5-tuple labels for various nodes in the tree; these are obtained by finding the shortest paths from t and forming the path labels as in fig. 4.B.1. Note that the first four coordinates always sum to the edge-distance to t.

Appendix 4.C Conditional pattern-distance along shortest paths

In this appendix, we derive the conditional PMF $\mathbb{P}\{d^v = x \,|\, d^u\}$ for the pattern-distance of node v to t given the pattern-distance to t of another node u, located on the shortest path between t and v. To this purpose, we first derive an expression for the probability of the event $\{\xi_j^v \neq \xi_j^t\}$ given d^u, which can be calculated appealing to Bayes' rule. Bayes' rule states that the conditional probability of the event A given an event B with $\mathbb{P}\{B\} \neq 0$ obeys

$$\mathbb{P}\{A \,|\, B\} = \mathbb{P}\{B \,|\, A\} \frac{\mathbb{P}\{A\}}{\mathbb{P}\{B\}} \; ; \tag{4.46}$$

consequently, we can write the probability of $\{\xi_j^v \neq \xi_j^t\}$ given d^u as

$$\mathbb{P}\left\{\xi_j^v \neq \xi_j^t \,\middle|\, d^u = y\right\} = \mathbb{P}\left\{d^u = y \,\middle|\, \xi_j^v \neq \xi_j^t\right\} \frac{\mathbb{P}\left\{\xi_j^v \neq \xi_j^t\right\}}{\mathbb{P}\{d^u = y\}} \; , \tag{4.47}$$

and we proceed calculating the terms on the right hand side individually.

Clearly, the pattern distance d^u to the target is a sum of binomial random variables, whose statistics depend on Δ. If $\Delta = \Delta_{\max}$, all bits of the pattern $\boldsymbol{\xi}^u$ have the same expectation $\left\langle \xi_j^u \right\rangle = a_h$, and are therefore identically distributed. As the bits are independent,

d^u is the binomial random variable

$$d^u \sim \text{Binom}\left(f^{tu}_{\Delta_{\max}}, L\right) \tag{4.48}$$

with "success probability"

$$f^{uv}_{\Delta_{\max}} := \mathbb{P}\left\{\xi^u_j \neq \xi^v_j\right\} \ . \tag{4.49}$$

In contrast to this, $\Delta < \Delta_{\max}$ implies that $a^u_j = \left\langle \xi^u_j \right\rangle$, and hence $\mathbb{P}\left\{\xi^t_j \neq \xi^u_j\right\}$, depends on j and on the Part containing u as prescribed by eq. (4.9). Therefore, instead of eq. (4.49) we consider

$$g^{uv}\left(a^u_j, a^v_j\right) = \mathbb{P}\left\{\xi^v_j \neq \xi^u_j\right\} \ . \tag{4.50}$$

In order to extend eq. (4.48) to general Δ, we make the simplifying assumption that the bits of a given pattern are identically distributed, with probabilities averaged over all bits of that pattern. That is, we make the approximation

$$\begin{aligned} d^u &\sim \text{Binom}\left(f^{tu}_{\Delta}, L\right) \ , \\ f^{uv}_{\Delta} &:= \sum_{x,y\in\{h,l\}} O^{\mu\nu}_{\Delta}(a_x, a_y) g^{uv}(a_x, a_y) \ , \end{aligned} \tag{4.51}$$

where μ and ν are the parts containing u and v, respectively, and $O^{\mu\nu}_{\Delta}(a_x, a_y)$ is the fraction of indices j such that $a^{\mu}_j = a_x$ while $a^{\nu}_j = a_y$. By definition of Δ in section 4.2, and with the help of fig. 4.2.3, we can write these fractions as

$$\begin{aligned} O^{\mu\nu}_{\Delta}(a_h, a_h) &= \begin{cases} \frac{1}{L}\left[\ell_c + 2\Delta\right] & : \mu = \nu \ , \\ \frac{1}{L}\left[\max\left(0, (2-\nu)\ell_c + 2\Delta\right) + \max\left(0, \nu\ell_c - L + 2\Delta\right)\right] & : \mu = 1 \neq \nu \ , \end{cases} \\ O^{\mu\nu}_{\Delta}(a_l, a_l) &= \begin{cases} \frac{1}{L}\left[L - \ell_c - 2\Delta\right] & : \mu = \nu \ , \\ \frac{1}{L}\left[\max\left(0, (\nu-2)\ell_c - 2\Delta\right) + \max\left(0, L - \nu\ell_c - 2\Delta\right)\right] & : \mu = 1 \neq \nu \ , \end{cases} \\ O^{\mu\nu}_{\Delta}(a_h, a_l) &= O^{\mu}_{\Delta}(a_l, a_h) = \frac{1}{2}\left(1 - O^{\mu}_{\Delta}(a_h, a_h) - O^{\mu}_{\Delta}(a_l, a_l)\right) \ . \end{aligned} \tag{4.52}$$

Notice that the definition of the f_{Δ}'s is consistent with eq. (4.49) because $O^{\mu\nu}_{\Delta_{\max}}(a_h, a_h) = 1$. Under the assumptions of eq. (4.51), the fraction in eq. (4.47) can be written as

$$\frac{\mathbb{P}\left\{\xi^v_j \neq \xi^t_j\right\}}{\mathbb{P}\left\{d^u = y\right\}} = \frac{f^{tv}_{\Delta}}{\binom{L}{y}\left(f^{tu}_{\Delta}\right)^y \left(1 - f^{tu}_{\Delta}\right)^{L-y}} \ , \tag{4.53}$$

using the expression for the binomial PMF in eq. (4.28).

To calculate the conditional probability $\mathbb{P}\left\{d^u = y \middle| \xi_j^v \neq \xi_j^t\right\}$ in eq. (4.47), we can use the fact that bits are independent, and consider the pattern-distance $d^u_{(j)}$ of u that disregards bit j. This allows us to split the event $\{d^u = y\}$ into the cases where $\xi_j^u = \xi_j^t$ and $\xi_j^u \neq \xi_j^t$, respectively:

$$\begin{aligned}\mathbb{P}\left\{d^u = y \middle| \xi_j^v \neq \xi_j^t\right\} =&\mathbb{P}\left\{d^u_{(j)} = y, \xi_j^u = \xi_j^t \middle| \xi_j^v \neq \xi_j^t\right\}\\ &+\mathbb{P}\left\{d^u_{(j)} = y-1, \xi_j^u \neq \xi_j^t \middle| \xi_j^v \neq \xi_j^t\right\}\\ =&\mathbb{P}\left\{d^u_{(j)} = y\right\}\mathbb{P}\left\{\xi_j^u = \xi_j^t \middle| \xi_j^v \neq \xi_j^t\right\}\\ &+\mathbb{P}\left\{d^u_{(j)} = y-1\right\}\mathbb{P}\left\{\xi_j^u \neq \xi_j^t \middle| \xi_j^v \neq \xi_j^t\right\} \ .\end{aligned} \tag{4.54}$$

In this expression, the marginal probabilities of $d^u_{(j)}$ are given by the binomial PMF, eq. (4.28), with n replaced by $L-1$ and $p = f_\Delta^{tu}$, whereas $\left\{\xi_j^u \neq \xi_j^t \middle| \xi_j^v \neq \xi_j^t\right\}$ is the same event as $\left\{\xi_j^u = \xi_j^v \middle| \xi_j^v \neq \xi_j^t\right\}$, which has probability $1 - f_\Delta^{uv}$. We can now combine the eqs. (4.53) and (4.54) into eq. (4.47) to obtain

$$\begin{aligned}\mathbb{P}\left\{\xi_j^v \neq \xi_j^t \middle| d^u = y\right\} =&\frac{(1-f_\Delta^{uv})f_\Delta^{tv}}{f_\Delta^{tu}}\frac{\binom{L-1}{y-1}}{\binom{L}{y}} + \frac{f_\Delta^{uv}f_\Delta^{tv}}{(1-f_\Delta^{tu})}\frac{\binom{L-1}{y}}{\binom{L}{y}}\\ =&\frac{(1-f_\Delta^{uv})f_\Delta^{tv}y}{f_\Delta^{tu}L} + \frac{f_\Delta^{uv}f_\Delta^{tv}(L-y)}{(1-f_\Delta^{tu})L} \ .\end{aligned} \tag{4.55}$$

Due to the assumption of eq. (4.51), this expression is independent of j, which implies that $d^v \mid d^u$ is a binomial random variable as well, with "success" probability as in eq. (4.55), and L trials. Therefore, we can calculate the conditional expectation

$$\langle d^v \mid d^u \rangle = d^u \left(\frac{(1-f_\Delta^{uv})f_\Delta^{tv}}{f_\Delta^{tu}} - \frac{f_\Delta^{uv}f_\Delta^{tv}}{(1-f_\Delta^{tu})}\right) + \frac{f_\Delta^{uv}f_\Delta^{tv}}{(1-f_\Delta^{tu})}L \ . \tag{4.56}$$

To finish this calculation, we have to find the g's defined in eq. (4.50), which will determine the probabilities f_Δ^u, f_Δ^v and f_Δ^{uv} by their definition in eq. (4.51). For this, the label-notation introduced in appendix 4.B will be useful.

We consider a pattern $\boldsymbol{\xi}^u$ at some node u in Part μ, and a path $P \sim (k,l,m,n;\nu)$ starting from u and ending in v. The transition probability from the j-th bit of pattern $\boldsymbol{\xi}^u$ to the j-th bit of the pattern at the end of the path P is given by

$$\boldsymbol{Q}_j^P := \left(\boldsymbol{Q}_j^{\uparrow\mu}\right)^k \left(\boldsymbol{R}_j^{\uparrow\mu}\right)^l \left(\boldsymbol{R}_j^{\nu}\right)^m \left(\boldsymbol{Q}_j^{\nu}\right)^n \ . \tag{4.57}$$

The $\boldsymbol{Q}$'s are as defined in eq. (4.16), with powers as in eq. (4.40). The matrix $\boldsymbol{R}_j^\nu$ is defined as in eq. (4.8) depending only on the Part-index of the label P, and that only if $\Delta < \Delta_{\max}$; if $\Delta = \Delta_{\max}$, the $\boldsymbol{R}_j^\nu$ are equal for all j and ν. Moreover, $\boldsymbol{R}^\uparrow$ is the family of transition matrices comprised by elements $\boldsymbol{R}_j^{\uparrow\mu}(\xi_j|\xi_j^\mu) = P_j^\mu(\xi_j|\xi_j^\mu)$ for $\xi_j^\mu, \xi_j \in \{0,1\}$, which are given by Bayes' rule, eq. (4.46). Consequently, the elements of $\boldsymbol{R}_j^{\uparrow\mu}$ read

$$\boldsymbol{R}_j^{\uparrow\mu} = \begin{pmatrix} \frac{1-a}{1-a_j^\mu}(1-\Gamma') & 1-\frac{1-a}{1-a_j^\mu}(1-\Gamma') \\ \frac{(1-a)\Gamma'}{a_j^\mu} & 1-\frac{(1-a)\Gamma'}{a_j^\mu} \end{pmatrix} . \tag{4.58}$$

Similarly, $\left(\boldsymbol{Q}_j^{\uparrow\mu}\right)^k$ is the matrix with elements $P_j^v(\xi_j^\mu|\xi_j^v)$ for $\xi_j^v, \xi_j^\mu \in \{0,1\}$. However, due to our stipulation that a_j^v depend only on the Part μ in which v is located (see eq. (4.15)), a quick calculation reveals that $\boldsymbol{Q}_j^{\uparrow\mu} = \boldsymbol{Q}_j^\mu$. Thus, the probabilities $g^P(a_j^\mu, a_j^\nu) = \mathbb{P}\left\{\xi_j^u \neq \xi_j^v\right\}$ are given by

$$g^{(k,l,m,n;\nu)}(a_j^\mu, a_j^\nu) := g^{uv}(a_j^\mu, a_j^\nu) = a_j^\mu Q_j^{(k,l,m,n;\nu)}(0|1) + (1-a_j^\mu) Q_j^{(k,l,m,n;\nu)}(1|0) \, , \tag{4.59}$$

where the dependence on a_j^ν is implicit in $\boldsymbol{Q}_j^\nu$. In the following paragraph, we report the relevant values for g by explicitly expanding eq. (4.59) in terms of the matrix elements given in eqs. (4.8), (4.58) and (4.40).

There are essentially five different cases to consider for eq. (4.59), each one corresponding to P being one of the paths $(k,0,0,0;1)$, $(k,1,0,0;1)$, $(k,1,1,m;\nu)$, $(0,0,1,m;\nu)$ or $(0,0,0,m;\nu)$. By direct calculation starting from eq. (4.59), we find the probabilities

$$\begin{aligned} &g^{(k,0,0,0;1)}(x,y) = 2x(1-x)\left[1-(1-\Gamma)^k\right] , \\ &g^{(k,1,0,0;1)}(x,y) = a + x - 2ax + 2(1-a)(1-\Gamma)^k(\Gamma'-x) \, , \\ &g^{(k,1,1,m;\nu)}(x,y) = \frac{2}{a}(1-\Gamma)^{k+m}\left[(1-a)\Gamma'(x+y-\Gamma') - xy\right] + x + y - 2xy(1-(1-\Gamma)^{k+m}) \, , \\ &g^{(0,0,1,m;\nu)}(x,y) = 2a(1-a) + 2(1-a)(1-\Gamma)^m\left[\Gamma' - a\right] \, , \\ &g^{(0,0,0,m;\nu)}(x,y) = 2y(1-y)\left[1-(1-\Gamma)^m\right] \, . \end{aligned} \tag{4.60}$$

The results presented in this appendix are the probabilistic building blocks required to calculate the elements of the expected transition matrix $\langle \mathbf{q} \rangle$, needed in section 4.4. This calculation is shown in the next appendix 4.D.

Appendix 4.D Local weights in the mean transition matrix

This appendix combines the results of the previous appendices 4.B and 4.C to find approximate expressions for the elements of $\langle \mathbf{q} \rangle$, which are needed to derive the main result of section 4.4, eq. (4.23).

In what follows, we will use again the convention to enumerate the Parts of the tree in such a way that t is a node descending from Part 1, which we call the *target-Part*. Also, given a node v, we will say that the neighbour closest (in terms of edge-distance) to t lies in the *target-wards* neighbourhood of v, and the neighbour closest to r lies in the *root-wards* neighbourhood of v. The remaining – non-target – Parts can be enumerated in any order.

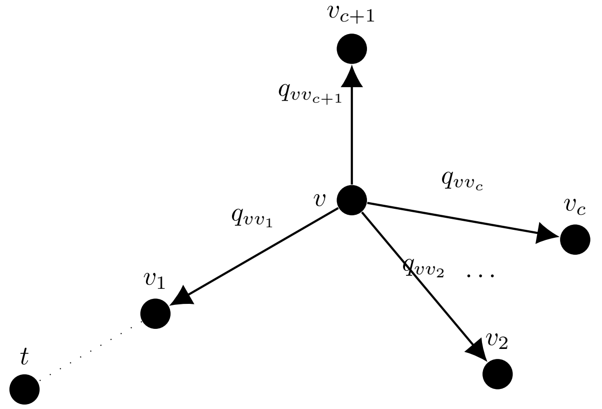


Fig. 4.D.1 Neighbourhood of the node v, separated into target-ward, root-ward and all other neighbours. The q's denote weights of edges pointing away from v. If v is the root, then there is no root-ward neighbour v_{c+1}, and if v is a leaf, there is *only* the neighbour v_{c+1}.

The derivation in this appendix is based on the observation that every node except t has exactly one target-wards neighbour, enumerated as v_1, with corresponding edge weight q_{vv_1}. We will denote the other neighbours of v by v_i with corresponding edge weights q_{vv_i} for $i > 1$, as depicted in fig. 4.D.1. By convention, if the root-wards node is different from v_1, then v_{c+1} denotes that root-wards neighbour. Our model definitions suggest that q_{vv_1} should on average exceed the other edge weights associated to v. However, evaluating $\left\langle q_{vv_i} \right\rangle$ is a complicated task due to the fact that it involves the pattern-distances of all nodes of the neighbourhood to normalise the q_{vv_i}'s (see eqs. (4.3) and (4.2)). It is, thus, more convenient to calculate the ratios of such q's,

$$\left\langle \frac{q_{vv_1}}{q_{vv_i}} \right\rangle =: \varepsilon_i \ . \tag{4.61}$$

This can be done by temporarily assuming that the pattern-distance $d^{v_1} = d(\boldsymbol{\xi}^{v_1}, \boldsymbol{\xi}^t)$ of the target-wards neighbour is given. Then, we can first calculate the conditional expectations

$$\left\langle \frac{q_{vv_1}}{q_{vv_i}} \middle| d^{v_1} \right\rangle = \left\langle \frac{d^{v_i}+1}{d^{v_1}+1} \middle| d^{v_1} \right\rangle = \langle d^{v_i}+1 \,|\, d^{v_1} \rangle \, \frac{1}{d^{v_1}+1} \ . \tag{4.62}$$

Moreover, under the assumption of eq. (4.51), d^{v_1} is a binomial random variable with parameters $p = f_\Delta^{tv_1}$ and $n = L$ in the notation of eq. (4.28). Hence, the expectation of $[d^{v_1}+1]^{-1}$ is known to be [113]

$$\left\langle \frac{1}{d^{v_1}+1} \right\rangle = \frac{1 - \left(1 - f_\Delta^{tv_1}\right)^{L+1}}{(L+1) f_\Delta^{tv_1}} \ . \tag{4.63}$$

Therefore, we can write $\left\langle q_{vv_1}/q_{vv_i} \right\rangle$ explicitly by substituting eqs. (4.56), (4.62) and (4.63) into eq. (4.61)

$$\begin{aligned}\left\langle \frac{q_{vv_1}}{q_{vv_i}} \right\rangle =& \frac{(1 - f_\Delta^{v_1 v_i}) f_\Delta^{tv_i}}{f_\Delta^{tv_1}} - \frac{f_\Delta^{v_1 v_i} f_\Delta^{tv_i}}{(1 - f_\Delta^{tv_1})} \\ &+ \frac{1 - \left(1 - f_\Delta^{tv_1}\right)^{L+1}}{(L+1) f_\Delta^{tv_1}} \left(1 + \frac{f_\Delta^{v_1 v_i} f_\Delta^{tv_i}}{(1 - f_\Delta^{tv_1})} L - \frac{(1 - f_\Delta^{v_1 v_i}) f_\Delta^{tv_i}}{f_\Delta^{tv_1}} + \frac{f_\Delta^{v_1 v_i} f_\Delta^{tv_i}}{(1 - f_\Delta^{tv_1})}\right) ,\end{aligned} \tag{4.64}$$

where $f_\Delta^{tv_1}$ and $f_\Delta^{tv_i}$ are given by the probabilities described in eq. (4.60), in terms of the paths connecting v_1 and v_i to t, respectively. $f_\Delta^{v_1 v_i}$ is given by f_Δ^P with P the labelled path connecting v_1 to v_i over v. In the notation of eq. (4.61), we thus have

$$\left\langle \frac{q_{vv_1}}{q_{vv_i}} \right\rangle = \varepsilon_i = \varepsilon\left(f_\Delta^{tv_1}, f_\Delta^{tv_i}\right) \tag{4.65}$$

with

$$\begin{aligned}\varepsilon(x,y) :=& \left(1 + \frac{Lyz(x,y)}{1-x} - \frac{y(1-z(x,y))}{1-x} + \frac{yz(x,y)}{1-x}\right) \frac{1-(1-x)^{L+1}}{(L+1)x} \\ &+ \frac{y(1-z(x,y))}{x} - \frac{yz(x,y)}{1-x}\end{aligned} \tag{4.66}$$

and $z(x,y)$ representing the f_Δ for the path connecting v_1 and v_i

$$z\left(f_\Delta^{tv_1}, f_\Delta^{tv_i}\right) = f_\Delta^{v_1 v_i} \ . \tag{4.67}$$

Having derived an expression for the ratios $\varepsilon_i = \langle q_{vv_1}/q_{vv_i}\rangle$, we now use these to compute the averages $\langle q_{vv_i}\rangle$ that we were originally interested in, by venturing the approximation

$$\left\langle q_{vv_1}\right\rangle = \left\langle \frac{q_{vv_1}}{q_{vv_i}} q_{vv_i}\right\rangle \approx \left\langle q_{vv_i}\right\rangle \varepsilon_i \tag{4.68}$$

for all $i > 1$ in the neighbourhood of v (see fig. 4.D.1). Now all expected weights $\left\langle q_{vv_i}\right\rangle$ in the neighbourhood of v are approximately determined by the system of equations

$$\begin{aligned} \left\langle q_{vv_1}\right\rangle - \varepsilon_i \left\langle q_{vv_i}\right\rangle &= 0 \quad (\forall i > 1)\ , \\ \sum_{i\geq 1} \left\langle q_{vv_i}\right\rangle &= 1\ . \end{aligned} \tag{4.69}$$

The unique solution of this linear system is given by

$$\begin{aligned} \left\langle q_{vv_1}\right\rangle &= \frac{1}{Z}\prod_{i>1}\varepsilon_i\ , \\ \left\langle q_{vv_i}\right\rangle &= \frac{1}{Z}\prod_{j>1;j\neq i}\varepsilon_j \quad (i > 1)\ , \end{aligned} \tag{4.70}$$

with Z being the normalising constant.

Important specialisations of this formula are those for which (i) all f^{tv_i}'s and $f^{v_iv_1}$'s for $i > 1$ are equal, i.e. when all v_i $(i > 1)$ have the same path label P_i relative to the target, and (ii) all v_i's for $(1 < i < c+1)$ are labelled by the same P_i, but $v_{c+1} = r$, which has its own unique label. Case (i) applies unless v is the root or the Part level node $\mu = 1$. Case (ii) applies if v is the Part node $\mu = 1$.

In the first case, all $\left\langle q_{vv_i}\right\rangle$'s and ε_i's for $i > 1$ have to be equal, which produces

$$\begin{aligned} \left\langle q_{vv_1}\right\rangle &\approx \frac{\varepsilon_i}{c+\varepsilon_i}\ , \\ \left\langle q_{vv_i}\right\rangle &\approx \frac{1}{c+\varepsilon_i}\ . \end{aligned} \tag{4.71}$$

In the second case, we have to distinguish $\left\langle q_{vv_i}\right\rangle$ for $1 < i < c+1$ and $\left\langle q_{vv_{c+1}}\right\rangle$ with the result

$$\begin{aligned} \left\langle q_{vv_1}\right\rangle &\approx \frac{\varepsilon_i\varepsilon_{c+1}}{(c-1)\varepsilon_{c+1}+\varepsilon_i+\varepsilon_i\varepsilon_{c+1}}\ , \\ \left\langle q_{vv_i}\right\rangle &\approx \frac{\varepsilon_{c+1}}{(c-1)\varepsilon_{c+1}+\varepsilon_i+\varepsilon_i\varepsilon_{c+1}}\ , \\ \left\langle q_{vv_{c+1}}\right\rangle &\approx \frac{\varepsilon_i}{(c-1)\varepsilon_{c+1}+\varepsilon_i+\varepsilon_i\varepsilon_{c+1}}\ . \end{aligned} \tag{4.72}$$

Appendix 4.E MFPT from mean transition matrix

This appendix combines the findings of the appendices 4.B, 4.C and 4.D with eq. (2.18) to state the main result of section 4.4 in eq. (4.23).

As laid out in the previous appendices, we can approximately calculate the elements of the transition matrix, averaged over realisations of patterns. The symmetries of this approximate matrix allow us to employ a relatively simple combinatorial argument for eq. (2.18), where the v_I's are the nodes of the path $(h-1,1,0,0;1)$ from t to r, though enumerated in reverse order, $r = v_0, \ldots, t = v_h$. In fact, we can express the fractions Π_J/π_{v_J} in terms of sums of fractions of ε's as described in the following paragraphs.

In section 2.3, we introduced the notation $\{\mathsf{t} \longrightarrow u\}$ for the set of all directed spanning trees rooted in u. For an exact tree, there is exactly one such tree for every u, which we denote t_u. Further dividing the tree into clusters (cf. fig. 2.3), let us label the nodes within a given cluster J as v_{Ji}, with $0 \leq i \leq |J|$ and $|J|$ being the size of cluster J. By convention, the index $i = 0$ is reserved for the node of J connecting J to the path $(h-1,1,0,0;1)$, i.e. $v_{J0} = v_J$. For any node $v_{Ji} \in J$, the set $\mathsf{t}_{v_{Ji}}$ differs from t_{v_J} only by the direction of the edges between v_J and v_{Ji}. For instance, let $J > 1$; if v_{Ji} is an immediate descendant of v_J, then the edge-distance between t and v_J is $h - J$, and

$$\frac{\pi_{v_{Ji}}}{\pi_{v_J}} = \frac{c + \varepsilon\left(f_\Delta^{(h-J,0,0,0;1)}, f_\Delta^{(h-J+2,0,0,0;1)}\right)}{\varepsilon\left(f_\Delta^{(h-J,0,0,0;1)}, f_\Delta^{(h-J+2,0,0,0;1)}\right)\left[c + \varepsilon\left(f_\Delta^{(h-J-1,0,0,0;1)}, f_\Delta^{(h-J+1,0,0,0;1)}\right)\right]}, \tag{4.73}$$

using eq. (4.71) and ε defined as in eq. (4.66). We also utilised that in this instance, the shortest paths from t to v_J and v_{Ji} have the form $(h-J,0,0,0;1)$ and $(h-J+1,0,0,0;1)$, respectively.

In Π_J/π_{v_J}, this term appears as a summand $c-1$ times, because v_J has that many immediate descendants that are not on the path to the target, i.e. that have label $(h-J+1,0,0,0;1)$. Repeating this analysis for all $h-J$ lower levels of the cluster J (where there are now c immediate descendants to each node that is not a leaf), we find the expression

$$\begin{aligned}\frac{\Pi_J}{\pi_{v_J}} =& 1+\frac{c-1}{c+\varepsilon\left(f_\Delta^{(h-J-1,0,0,0;1)},f_\Delta^{(h-J+1,0,0,0;1)}\right)}\\ &\times\left[\sum_{\ell=1}^{h-J-1} c^{\ell-1}\frac{c+\varepsilon\left(f_\Delta^{(h-J-1+\ell,0,0,0;1)},f_\Delta^{(h-J+1+\ell,0,0,0;1)}\right)}{\prod_{k=1}^{\ell}\varepsilon\left(f_\Delta^{(h-J-1+k,0,0,0;1)},f_\Delta^{(h-J+1+k,0,0,0;1)}\right)}\right.\\ &\left.+\frac{c^{h-J-1}}{\prod_{k=1}^{h-J-1}\varepsilon\left(f_\Delta^{(h-J-1+k,0,0,0;1)},f_\Delta^{(h-J+1+k,0,0,0;1)}\right)}\right]. \end{aligned} \tag{4.74}$$

The last summand within the brackets arises from the fact that all leaves have only one outgoing edge.

If $J=1$, then the mean edge weights at v_J are given by eq. (4.72), whereas lower edges inside the cluster still follow eq. (4.71),

$$\begin{aligned}\frac{\Pi_1}{\pi_{v_1}} &= 1\\ &+\frac{(c-1)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)},f_\Delta^{(h-1,1,0,0;1)}\right)}{d_1}\\ &\times\left[\sum_{\ell=1}^{h-2} c^{\ell-1}\frac{c+\varepsilon\left(f_\Delta^{(h-2+\ell,0,0,0;1)},f_\Delta^{(h+\ell,0,0,0;1)}\right)}{\prod_{k=1}^{\ell}\varepsilon\left(f_\Delta^{(h-2+k,0,0,0;1)},f_\Delta^{(h+k,0,0,0;1)}\right)}\right.\\ &\left.+\frac{c^{h-2}}{\prod_{k=1}^{h-2}\varepsilon\left(f_\Delta^{(h-2+k,0,0,0;1)},f_\Delta^{(h+k,0,0,0;1)}\right)}\right], \end{aligned} \tag{4.75}$$

with the denominator

$$\begin{aligned}d_1 =& (c-1)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)},f_\Delta^{(h-1,1,0,0;1)}\right)+\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)},f_\Delta^{(h,0,0,0;1)}\right)\\ &+\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)},f_\Delta^{(h-1,1,0,0;1)}\right)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)},f_\Delta^{(h,0,0,0;1)}\right). \end{aligned} \tag{4.76}$$

Finally, for $J = 0$, we have $v_0 = r$, which brings us back to eq. (4.70) for the edges connecting to $v_j = r$. This observation leads us to

$$
\begin{aligned}
\frac{\Pi_0}{\pi_{v_0}} = 1& \\
+&\frac{1}{\prod_{\mu=2}^{c} \varepsilon\left(f_\Delta^{(h-1,0,0,0;1)}, f_\Delta^{(h-1,1,1,0;\mu)}\right) + \sum_{\mu=2}^{c} \prod_{\nu=2;\nu\neq\mu}^{c} \varepsilon\left(f_\Delta^{(h-1,0,0,0)}, f_\Delta^{(h-1,1,1,0;\nu)}\right)} \\
\times&\sum_{\mu=2}^{c} \prod_{\nu=2;\nu\neq\mu}^{c} \varepsilon\left(f_\Delta^{(h-1,0,0,0)}, f_\Delta^{(h-1,1,1,0;\nu)}\right) \left[\frac{c+\varepsilon\left(f_\Delta^{(h-1,1,0,0;1)}, f_\Delta^{(h-1,1,1,1;\mu)}\right)}{\varepsilon\left(f_\Delta^{(h-1,1,0,0;1)}, f_\Delta^{(h-1,1,1,1;\mu)}\right)}\right. \\
&+\frac{c+\varepsilon\left(f_\Delta^{(h-1,1,1,0;\mu)}, f_\Delta^{(h-1,1,1,2;\mu)}\right)}{\varepsilon\left(f_\Delta^{(h-1,1,0,0;1)}, f_\Delta^{(h-1,1,1,1;\mu)}\right) \varepsilon\left(f_\Delta^{(h-1,1,1,0;\mu)}, f_\Delta^{(h-1,1,1,2;\mu)}\right)} \\
&\left.+\sum_{\ell=3}^{h-1} c^{\ell-1} \frac{c+\varepsilon\left(f_\Delta^{(h-1,1,1,l-2;\mu)}, f_\Delta^{(h-1,1,1,l;\mu)}\right)}{d_{0,\ell}^{\mu}} + \frac{c^{h-1}}{d_{0,h-1}^{\mu}}\right]
\end{aligned}
\tag{4.77}
$$

with the denominator terms

$$
\begin{aligned}
d_{0,\ell}^{\mu} =& \varepsilon\left(f_\Delta^{(h-1,1,0,0;1)}, f_\Delta^{(h-1,1,1,1;\mu)}\right) \varepsilon\left(f_\Delta^{(h-1,1,1,0;\mu)}, f_\Delta^{(h-1,1,1,2;\mu)}\right) \\
&\times \prod_{k=3}^{\ell} \varepsilon\left(f_\Delta^{(h-1,1,1,k-2;\mu)}, f_\Delta^{(h-1,1,1,k;\mu)}\right) .
\end{aligned}
\tag{4.78}
$$

For eq. (2.18), we need to combine these expression with appropriate path weights connecting the clusters. More precisely, we need the fractions

$$
\frac{\left\langle q_{v_{I-1}v_{I-2}}\right\rangle \ldots \left\langle q_{v_{K+1}v_K}\right\rangle}{\left\langle q_{v_K,v_{K+1}}\right\rangle \ldots \left\langle q_{v_{I-1}v_I}\right\rangle} = \frac{\left\langle q_{v_{I-1}v_{I-2}}\right\rangle}{\left\langle q_{v_{I-1}v_I}\right\rangle} \frac{\left\langle q_{v_{I-2}v_{I-3}}\right\rangle}{\left\langle q_{v_{I-2}v_{I-1}}\right\rangle} \cdots \frac{\left\langle q_{v_{K+1}v_K}\right\rangle}{\left\langle q_{v_{K+1}v_{K+2}}\right\rangle} \frac{1}{\left\langle q_{v_K v_{K+1}}\right\rangle} . \tag{4.79}
$$

If $K > 1$ this fraction can be written as

$$
\begin{aligned}
\frac{\left\langle q_{v_{I-1}v_{I-2}}\right\rangle \ldots \left\langle q_{v_{K+1}v_K}\right\rangle}{\left\langle q_{v_K,v_{K+1}}\right\rangle \ldots \left\langle q_{v_{I-1}v_I}\right\rangle} =& \left[c+\varepsilon\left(f_\Delta^{(h-K-1,0,0,0;1)}, f_\Delta^{(h-K+1,0,0,0;1)}\right)\right] \\
&\times \prod_{J=K}^{I-1} \frac{1}{\varepsilon\left(f_\Delta^{(h-J-1,0,0,0;1)}, f_\Delta^{(h-J+1,0,0,0;1)}\right)} .
\end{aligned}
\tag{4.80}
$$

Due to the distinct form of the weights close to r, the same fractions for $K = 1$ and $K = 0$ take the form

$$\begin{aligned}
&\frac{\left\langle q_{v_{I-1}v_{I-2}}\right\rangle \dots \left\langle q_{v_2 v_1}\right\rangle}{\left\langle q_{v_1,v_2}\right\rangle \dots \left\langle q_{v_{I-1}v_I}\right\rangle} = \\
&\frac{1}{\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h-1,1,0,0;1)}\right)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h,0,0,0;1)}\right)} && (4.81)\\
&\times\left((c-1)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h-1,1,0,0;1)}\right) + \varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h,0,0,0;1)}\right)\right.\\
&\quad\left. + \varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h-1,1,0,0;1)}\right)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h,0,0,0;1)}\right)\right)\\
&\times\prod_{J=2}^{I-1}\frac{1}{\varepsilon\left(f_\Delta^{(h-J-1,0,0,0;1)}, f_\Delta^{(h-J+1,0,0,0;1)}\right)} && (4.82)
\end{aligned}$$

and

$$\begin{aligned}
&\frac{\left\langle q_{v_{I-1}v_{I-2}}\right\rangle \dots \left\langle q_{v_1 v_0}\right\rangle}{\left\langle q_{v_0,v_1}\right\rangle \dots \left\langle q_{v_{I-1}v_I}\right\rangle} = \\
&\frac{\prod_{\mu=2}^{c}\varepsilon\left(f_\Delta^{(h-1,0,0,0;1)}, f_\Delta^{(h-1,1,1,0;\mu)}\right) + \sum_{\mu=2}^{c}\prod_{\nu=2;\nu\neq\mu}^{c}\varepsilon\left(f_\Delta^{(h-1,0,0,0;1)}, f_\Delta^{(h-1,1,1,0;\nu)}\right)}{\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h-1,1,0,0;1)}\right)\varepsilon\left(f_\Delta^{(h-2,0,0,0;1)}, f_\Delta^{(h,0,0,0;1)}\right)\prod_{\mu=2}^{c}\varepsilon\left(f_\Delta^{(h-1,0,0,0;1)}, f_\Delta^{(h-1,1,1,0;\mu)}\right)}\\
&\times\prod_{J=2}^{I-1}\frac{1}{\varepsilon\left(f_\Delta^{(h-J-1,0,0,0;1)}, f_\Delta^{(h-J+1,0,0,0;1)}\right)}\,, && (4.83)
\end{aligned}$$

respectively. Combining these expressions in the manner of eq. (2.18) produces the function C_{MF} shown in section 4.4.

Chapter 5

A step-by-step guide to estimating the complexity of a given Act of Parliament

The previous chapter developed a theoretical idea of complexity for Acts of Parliament based on some of their textual parameters. In this chapter, we demonstrate how one may practically estimate the complexity of a given Act employing the results of chapter 4. During the presentation we occasionally refer to code-snippets assembled in a Jupyter-notebook that is available in the supplementary material (see also appendix B) to this thesis and to download. All codes - including the `pattern_walker` modules - are available on the author's github page https://github.com/YPFoerster/pattern_walker. The reader is invited to play and experiment with the provided material.

Section 5.1 shows one of many ways of vectorising an Act of Parliament, i.e. to translate its text Items into keyword patterns. In section 5.2, we estimate the parameters of the random reader model from the vectorised Act. In section 5.3, we feed these pieces of information - either the patterns or the parameters - into the relevant Python classes which implement the complexity functions defined in chapter 4. The chapter is summarised in section 5.4, and appendix 5.A contains a brief error calculation pertaining to section 5.2.

5.1 Retrieving and vectorising the Act

The provided notebook retrieves a version of the Housing Act 2004 (parsed into JSON) from Graphie [115, 116] in a nested JSON-file. The nesting defines a hierarchy which is shown in as a tree in fig. 5.1.1. Some preprocessing follows in order to strip the text of stopwords (words not conveying any information on their own, e.g. "to"), special characters, punctuation marks and grammatical inflection [117, 118]. Within the module

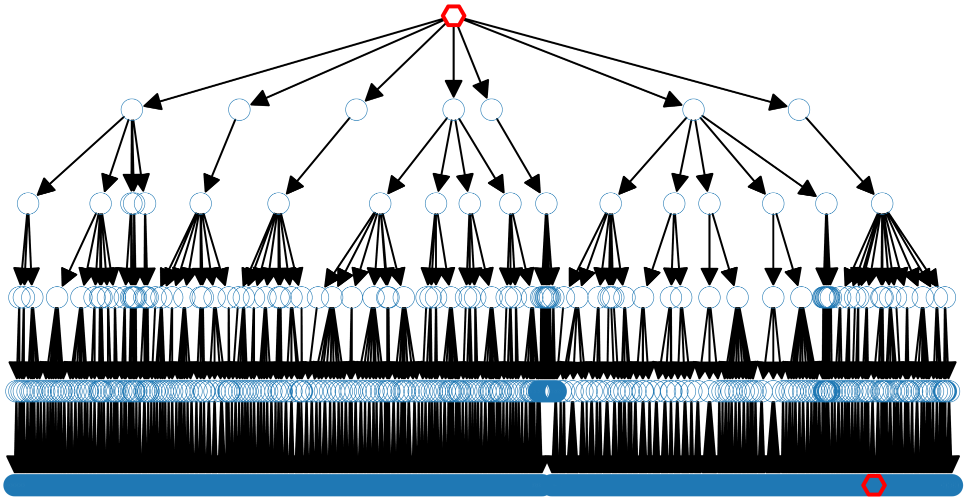


Fig. 5.1.1 Tree representation of the Housing Act 2004 [114]. Nodes marked in red are the root ("Housing Act") and an arbitrarily chosen target node, here Part 7, Section 239, Paragraph 10.

`utils`, one can see that we added a few custom stopwords such as "act", "provision" or "section", as we do not expect these to contribute to the semantic content either.

We are left with a corpus in which every document consists of so-called lemmas. Lemmatisation is required to allow for a sensible computer-based analysis, as otherwise two inflected versions of the same word or stem (say, "house" and "houses") would be classified as different although presumably, they should be understood as a single concept [117].

Having preprocessed the texts, the next task is to identify the important keywords that define our glossary. Below we focus on only one methods of doing so. The algorithm produces a vocabulary ranked by some measure of importance. The number of top-ranked words we decide to use for our glossary implicitly defines our first model parameter, the pattern length L. Given the glossary, it is then a straightforward task to vectorise the documents, i.e. to replace the texts by patterns.

Since we have some *a priori* information about the topology of the corpus, we use that knowledge to estimate the overlap between the different Parts. We do so by obtaining separate glossaries for each Part first; the overlap of these glossaries determines the overlap between the Parts (note that different Parts may overlap by different amounts, contrary to our simple model in chapter 4). Our full vocabulary is then the union of the Part-vocabularies.

Automated keyword extraction methods have been enjoying active research throughout recent years. Consequently, a variety of specialised methods exist, discussion of which is beyond the scope of the present chapter. We choose one method specifically, as it is based on intuitive statistics of term occurrences within and across documents. The outcome of the analysis will to an extent depend on the method used to extract keywords. However, these methods usually are tested on keyword annotated texts (cf. [120]), such that "high ranking" methods should produce almost identical results.

5.1.1 Term frequency statistics

In information retrieval, *term frequency, inverse document frequency* (tf-idf) is a statistic that assigns importance to terms (words or sequences of words, up to a given length) in a corpus. Its rationale is that important terms should appear often in certain documents, while terms appearing in many documents will be less important. The tf-idf of a given term t in a document d is the product $\mathrm{tf}(t,d) \cdot \mathrm{idf}(t)$. $\mathrm{tf}(t)$ is the proportional count of t among all terms in d. $\mathrm{idf}(t)$ is the logarithm of the inverse relative number of documents containing t [117]. The implementation we use in our Jupyter notebook looks for n-grams, i.e. sequences of words of length n; the minimal and maximal length of n-grams considered are parameters to be specified. We choose to include n-grams of lengths up to and including four in the example provided in the supplementary material.

To actually build the glossary and vectorise the text, we take the 100 terms with the highest tf-idf values from each Part; these are the terms *specific* to the Part from which they have been taken. The union of these sets represents the total glossary. Some of the terms might appear multiple times, as they are extracted from separate Parts independently. Consequently, the total glossary is shorter than the number of Parts multiplied by 100. This is an expression of the *overlap* between the Parts. In order to vectorise a text item, we associate with it a 0-1-vector: the i-th component of that vector simply records if the i-th word of the glossary appears in that piece of text (1) or not (0).

The top-five terms of each Part-vocabulary obtained via tf-idf are shown in tab. 5.1.1. Part 5 sticks out in that the algorithm finds substantially fewer key terms than in the other Parts. Moreover, from the table we can see that three of the five top key-terms overlap in two or three words. In fact, upon inspection we find that Part 5 has been repealed in 2011 such that the online version shows only the headlines and no contents [114, 119]. Arguably, the extracted key-terms don't necessarily invoke associations to any clear-cut topics. We refer to [38], which remarks that legal texts may suffer from "natural language obsession" (among other problematic textual patterns), which can pose a challenge to data extraction. This is an issue that may be alleviated using pattern-detection methods and a better keyword extraction algorithm. For reviews of state-of-the-art methods, see [120, 118].

Table 5.1.1 The five top-ranked terms (n-grams with n up to four) of each Part, obtained by applying tf-idf to the Parts separately.

Part	terms
1	prepare, premise landlord common, tenant premise landlord, purpose operation relate, order possible soon
2	authority publish, contract circumstance, deal application licence granting, application licence granting refusal, approval give relation description
3	include welfare, authority time time review, housing offence england, account follow, date specify authority
4	let breach banning order, come force order precede, remain, agreement instrument effect interim, recover virtue remain
5	supplementary, act estate agent, duty person act estate, person act estate agent, home information pack
6	c. security tenure, amendment effect month, discount apply take, repayment discount apply, require result notice give
7	group prescribe relationship, wale order connection, commit consent connivance, occupier premise, consent connivance attributable

We can now compute the pattern-distance between all neighbours in the tree. The resulting histogram is shown in fig. 5.1.2, which we compare to the frequencies predicted by a binomial distribution. For the sake of transparency, we point out that the extent to which the histogram agrees with a binomial distribution starkly depends on the parameters of the tf-idf procedure.

The steps above fix our values for L and the Part overlaps, and all documents are now vectorised. As such, the corpus is now intelligible for our random reader from chapter 4, allowing us to assign an value of complexity to the Act. For comparison with an ensemble of Acts with the same layout and textural parameters, we compute C_{MF} as well. To this end, we must estimate our model parameters first.

5.2 Estimating the parameters of the model

In this section, we use our knowledge about the model in chapter 4 to derive estimates for the relevant parameters. The order in which we do so (and thus introduce dependencies among them) is visualised in fig. 5.2.1, and the results of this process are summarised in tab. 5.2.1. We also report standard deviations of our estimators calculated via error propagation. The latter is briefly summarised in the appendix 5.A.

Some of the parameters, L, Δ, we have by virtue of the above vectorisation; L is the length of the merged vocabulary and Δ is the number of topic-specific keywords shared by two Parts. The latter will, in general, depend on the Parts compared, so we adopt an

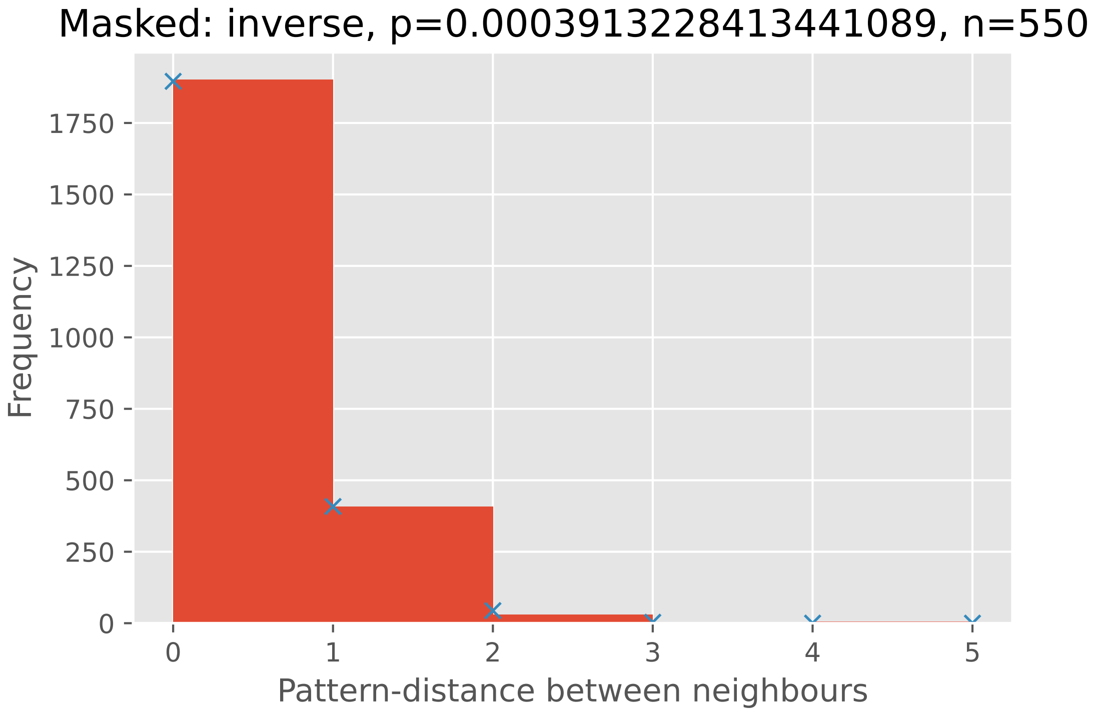


Fig. 5.1.2 Histogram of pattern-distances between neighbouring nodes of the tree, taking only the $n = 550$ non-specific keywords for each Part into account. Blue crosses show frequencies predicted by a binomial distribution with the same n and p estimated from the data.

average procedure – close to the definition of overlap in section 4.2 – for simplicity: In the absence of overlap, each Part has L/c topic-specific keywords, where c is the number of Parts. We estimate Δ as the excess of specific keywords over L/c, i.e. $\widehat{\Delta} = L_h - \frac{L}{c}$, where L_h is the number of specific keywords of a given Part (by the setup above, $L_h = 100$).

The parameters a_h and a_l, given in eq. (4.9) are marginal probabilities of keyword occurrence, depending on whether the word in question appertains to the topic of the current Part or to a different Part. Let a_h^v be the fraction of Part-specific keywords present in the text at v, then the Maximum Likelihood Estimator (MLE) $\widehat{a_h}$ of a_h is the average of the a_h^v over all nodes v in the graph. Similarly, the estimator $\widehat{a_l}$ is computed from the fraction a_l^v of non-specific keywords appearing in v, averaged over all v,

$$\widehat{a_l} = \frac{1}{nL_l} \sum_{v \neq \text{root}} \sum_{\substack{j; \\ \mathbb{E}\left(\xi_j^v\right) = a_l}} \xi_j^v \,, \tag{5.1}$$

with the outer sum running over all nodes excluding the root, and the inner sum selecting the non-specific keyword bits of each pattern. L_l is the number of such keywords for any

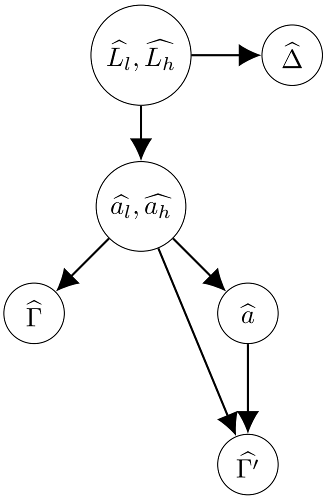


Fig. 5.2.1 Order or parameter estimation. Arrows indicate dependency, with the "dependent" parameter at the end of the arrow.

given pattern, which we assume to be a constant. We use the same estimator of a_h with the obvious replacements.

Subsequently, we can apply eq. (4.43) and the distance-histogram to estimate

$$\widehat{\Gamma} = \frac{p_j}{2\widehat{a}_j(1-\widehat{a}_j)} \,, \tag{5.2}$$

where j can be h or l and p_j is the MLE estimator of the parameter p for a binomial model (see eq. (4.28)) for the distances. If $j = l$, p_l is given by the average pattern-distance between two given neighbours, considering only non-specific keywords, divided by the number of non-specific keywords. That is,

$$p_l = \frac{1}{tL_l} \sum_{\substack{u\sim v;\\ u,v\neq \text{root}}} \sum_{\substack{j;\\ \mathbb{E}\left(\xi_j^v\right)=a_l}} \left|\xi_j^u - \xi_j^v\right| \,, \tag{5.3}$$

with the outer sum running over all t pairs of neighbours not involving the root. The analogous procedure yields an estimate for p_h. Since usually there are more generic than topic-specific keywords, it is advantageous to choose $j = l$ as that choice provides a larger sample.

The parameters a and Γ' are problematic because the sample is small (one sample for the former, seven for the latter). We may often observe the Title vector to have no or only singular entries[1], which would imply that a is zero or tiny! However, given eqs. (4.11) and

[1] In the example presented here, the Title vector indicates two present keywords, the title of the Act is "Housing Act".

(4.12), where we set $a_h \approx (1-a)\Gamma' + a$ and $a_l \approx (1-a)\Gamma'$, the spread between a_h and a_l should approximate a, thus

$$\widehat{a} = \widehat{a}_h - \widehat{a}_l \ . \tag{5.4}$$

In eq. (4.11), we introduced a small deviation β_l to the difference $a_h - a_l$. If we had a reliable way of estimating a directly, we could use eq. (5.4) to estimate β_l instead. In this chapter, for the sake of simplicity, we set $\beta_l = 0$ *a priori*.

Next we can, for instance, use eq. (4.8) to derive the estimator

$$\widehat{\Gamma'} = \frac{p'_j - \widehat{a} - \widehat{a}_j}{2(1-\widehat{a})} \ , \tag{5.5}$$

for Γ'. Here, p'_j is the estimator of the binomial model for the pattern-distances between root and Part nodes, analogous to p_j above. We report the values of all estimatiors in tab. 5.2.1.

Table 5.2.1 Estimators derived from the Part-wise models using the 100 "most relevant" terms of each Part, reported to two significant figures of the uncertainty. Uncertainties are standard deviations calculated in appendix 5.A, using error propagation for a, Γ and Γ'.

parameter	estimated value
L	650
Δ	3.57
p	$3.9 \times 10^{-4} \pm 1.6 \times 10^{-7}$
p'	$2.4 \times 10^{-4} \pm 3.4 \times 10^{-5}$
a_h	$1.9 \times 10^{-3} \pm 9.0 \times 10^{-4}$
a_l	$1.2 \times 10^{-3} \pm 7.2 \times 10^{-4}$
a	$6.8 \times 10^{-4} \pm 1.2 \times 10^{-3}$
Γ	$1.6 \times 10^{-1} \pm 9.2 \times 10^{-2}$
Γ'	$4.0 \times 10^{-4} \pm 2.7 \times 10^{-3}$

5.3 Computing complexities for the instance and the ensemble

Even before estimating the parameters of the model, we can feed our tree and vectors into an `empiricalPatternWalker`, which is essentially a container for the graph and pattern information, the root and the target node. For the mean-field calculation for the ensemble of texts with our estimated parameters, we employ the `MFPatterWalker_general` class, which accepts the tree and the estimators. As the hierarchy of the Housing Act 2004 is not a c-ary trees (cf. fig 5.1.1), the simplifications in the appendices of chapter 4 do not apply in full. However, since trees are sparse graphs, we can return to eq. (2.7), which can be solved efficiently using sparse matrices. To this end, we use the function `utils.mfpt` with keyword arguments `method='grounded_Laplacian'` and `sparse='True'`. Passing the keyword argument `weight_str='weight'` for the walker using the empirical pattern data, we obtain $C_{\text{emp}} := m_{rt}(\mathbf{q})$ as discussed in section 4.4. Passing `weight_str='mean_weight'` together with the above instance of `MFPatterWalker_general`, we obtain C_{MF} as defined in eq. (4.22).

We can thus compare the actual complexity of the Act to the approximate complexity of the ensemble and to the complexity of the tree itself by passing the parameters appropriate to enforce unit edge weights (which makes the walker diffusive). The numerical values for each are given in tab. 5.3.1 as absolute values and normalised by the size of the tree. For a more thorough evaluation of the table, it would be best to estimate the distribution of C by sampling repeatedly from the ensemble, which would given an indication of how common the observed value of C_{emp} is for the ensemble.

Table 5.3.1 Complexity values for the Housing Act 2004, prepared as in the main text. The different columns base the calculation on: edge weights based on vectorised text, mean edge weights based on estimated parameters, and unit edge weights, respectively. The second row shows C/n where $n = 2359$ is the number of nodes in the tree.

	C_{emp}	C_{MF}	C_{diff}
	22220.0	15712.0	22243.0
normalised	9.42	6.66	9.43

The numbers reported in tab. 5.3.1 depend on the target node chosen, but other choices lead to similar results. Our target is Part 7, Section 239, Paragraph (10) [114]:

> *A person authorised for the purposes of this section must, if required to do so, produce his authorisation for inspection by the owner or any occupier of the premises or anyone acting on his behalf.*

Tab. 5.3.1 seems to suggest that for the given target within the Housing Act 2004, the complexity C_{emp} is only marginally lower than for completely uninformative text C_{diff}, compared to the ensemble average C_{MF}. However, this result is likely to be a consequence of the shortcomings of the keyword extraction method described in the previous section. Moreover, we find that C_{MF} is very sensitive to varying the estimators reported in tab. 5.2.1 within one standard deviation. The fact that we can show tab. 5.3.1 at all demonstrates that we can build a full pipeline, taking an Act of Parliament as an input, and returning a numerical value for its complexity. Yet, for this value to be meaningful, the components of the pipeline must be optimised to the use case of legal corpora.

5.4 Conclusions and outlook

This chapter shows an exemplary workflow implementing the results of chapter 4. The aim of this undertaking is a proof of concept, elucidating the most necessary steps. Our minimalist setup shows how to prepare the input data from an Act of Parliament given in a suitably parsed form (this being an undertaking of its own, cf. [115]) and how to interact with the most important functionalities of the code developed for chapter 4.

We have pointed out some potential improvements throughout the chapter. To summarise, the basic modules – each with its own potential – of the workflow are: preprocessing the text into tokens (e.g. lemmatisation), creating a glossary taking into account the importance of each keyword for each Part, vectorising all text Items, estimating the model parameters from chapter 4 and calculating the complexities defined therein based on the results of the previous steps.

The vision of the present thesis always was that any insight into legal systems obtained through social physics, may also be useful to improve them. The workflow described above is a significant step towards designing a tool not only to navigate legal text (a graph-based open source solution being presented in [115]), but also to support policy makers in the drafting process. Every element of the workflow used in this chapter is available in the supplementary material to this thesis, as well as free to be downloaded, modified and improved upon.

Appendix

Appendix 5.A Error propagation for parameter estimates

Here, we report our calculations producing the uncertainties in tab. 5.2.1, obtained via error propagation. Error propagation is a standard tool in physics and engineering to estimate the uncertainty of a function evaluated on a measured value – we refer the reader to [121] or most introductory texts on physics laboratory courses, measurements and error analysis. We will make the simplifying assumption that all estimators are independent, from which follows the well known formula for the variance of a sufficiently differentiable function f

$$\sigma_f^2 \approx \sum_{i=1}^{k} \left(\frac{\partial f}{\partial x_i} (\widehat{x}_1 \dots \widehat{x}_k) \right)^2 \sigma_i^2 \,, \tag{5.6}$$

evaluated at a number of estimators $\widehat{x}_1, \dots \widehat{x}_k$ with variances $\sigma_1^2, \dots, \sigma_k^2$.

Our estimate for a_l (a_h) was the the mean "activation" rate within the off-topic (topic-specific) keywords of a given Part, see eq. (5.1). If in a very rough approximation we assume that all patterns are independent, it is well known that the variance of this estimator is given by

$$\sigma^2_{\widehat{a_j}} = \frac{\widehat{a_j}(1-\widehat{a_j})}{n} \,, \tag{5.7}$$

with j being either h or l. Similarly, we estimated p_l as the mean of nearest-neighbour pattern-distances, divided by the number of keywords involved, eq. (5.3). In our model, pattern-distances between nearest neighbours are indeed independent due to the Markov property of the pattern mutation process (cf. section 4.2). Therefore, we again have

$$\sigma^2_{p_l} = \frac{p_l(1-p_l)}{t} \,, \tag{5.8}$$

in which t is the number of neighbour-pairs considered.

By virtue of eqs. (5.2) and (5.6), we obtain for the variance of $\widehat{\Gamma}$

$$\sigma^2_{\widehat{\Gamma}} = \frac{1}{4\widehat{a_l}^2(1-\widehat{a_l})^2}\left[\sigma^2_{p_l} + p^2\frac{(1-2\widehat{a_l})^2}{\widehat{a_l}^2(1-\widehat{a_l})^2}\sigma^2_{\widehat{a_l}}\right] \,. \tag{5.9}$$

Furthermore, employing eq. (5.4) for the variance of $\widehat{a}$, we have,

$$\sigma^2_{\widehat{a}} = \sigma^2_{\widehat{a_l}} + \sigma^2_{\widehat{a_h}} \,, \tag{5.10}$$

and using eq. (5.5) for the variance of $\widehat{\Gamma'}$

$$\sigma^2_{\Gamma'} = \frac{1}{4(1-\widehat{a})^2}\left[\sigma^2_{p'} + \sigma^2_{\widehat{a_l}} + \sigma^2_{\widehat{a}}\frac{1}{(1-\widehat{a})^2}\right] \,. \tag{5.11}$$

The uncertainties reported in the main text in tab. 5.2.1 are the square-roots of the variances above.

Chapter 6

Conclusions

We set off in chapters 2 and 3 with a technical exposition on mean first-passage times (MFPTs) for general random walkers on networks. For walkers navigating on networks of a certain class, we derived a new result which led the way into MFPT approximations for a much larger family of networks. We presented analytical and numerical studies of our results.

We proceeded to establish a new point of view onto the old problem of complexity in the law. The model presented in chapter 4 is the first to consider a reader's behaviour in conjunction with generative properties of the text on which they wander. This opens the door to making testable statements about the complexity of an Act of Parliament manifested in the interaction between its textual properties and a user of that law. In chapter 5, we showed in a simple prototype how one can construct an algorithm to determine the complexity of different Acts in practice.

Many other expressions of complexity have been identified over the decades. To name only two: firstly, paraphrasing [43], a legal text may be ambiguous or it may not be clear if a given Paragraph is the best answer available to a given question – we have so far considered the situation where the answer to the question becomes immediately apparent upon encounter. Secondly, the minimum complexity layout of an Act from the point of view of the reader may be suboptimal from a maintenance perspective: as amendments are enacted in an effort to adapt existing laws to real world events, one must carefully consider the interdependencies with other laws – this was for instance pointed out by [37] in analogy to software systems.

Our results of chapter 4 are driven by the main theorem of chapter 2, but for the full picture, extensions are required: we completely disregard cross-references, which break the tree structure of the hierarchical organisation, adding detours and shortcuts to legal corpora [110, 35]. Similarly, the network of judicial precedents is not a tree but merely devoid of directed cycles [122]. In common law systems like the UK, this network is a highly relevant source of information, which is however not tractable with the methods from chapters 2 and 3.

While the probabilistic rules defined in chapter 4 seem intuitive, they forego a deeper analysis of the processes involved in searching for information. Attention patterns studied via eye-movements suggest that readers engaged in a search do focus on keywords initially, before reading any paragraph in detail [107]. Other cognitive phenomena relevant to the search process are associations between keywords, misinterpretation, and the fact that the reader may already be familiar with a law or parts of it. These phenomena are distinct from *information seeking*, in that the latter describes the methods and procedures used by individuals to find information based on a current information need. Information seeking behaviour has been studied for a variety of professions, e.g. for lawyers [104]. We refer to [42] for a collection for further references. These studies highlight the importance of informal sources of information (such as asking knowledgeable colleagues) and searchable databases which increasingly supersede “traditional” index-based text searches. We focus on the legal amateur who may have only partial or no access to any of these sources, or might be unable to translate their information need into a suitable query - an issue that professionals experience as well, see e.g. [123]. In our case, we make the assumption that the glossary of a law is provided, is unambiguous to the reader, and that the reader is not familiar with the text. Research on how individuals search their own mental or conceptual space suggest that the mental search process resembles the spatial search process of foraging animals [124, 125]. These results may be used to build evidence for a model of text search based on similarity and distances as discussed in chapter 4.

There is hope that our research will be taken up by the legal community for further theoretical studies. Especially via the aspect of information seeking, our work is very much related to the references [42, 41], which study how topical similarity correlates with the network of citations in legal opinions.

We conclude by taking up a general criticism on social physics issued in [2]: the scarcity of empirical verification. Chapter 4 makes a number of assumptions and statements about the behaviour of human readers and the keyword-related properties of structured corpora. The properties of the keyword-distribution may be tested with the help of state-of-the-art keyword-extraction algorithms – these in turn are usually tested by comparing the results to keyword labels written by humans (see [128] and references therein). Direct experiments on readers’ behaviour on an online mask are conceivably close, but are at danger of conflating comprehension time and stepping time units. A reader’s movements on a corpus can be tracked if they retrieve the text on a device. The platform developed in [115] with involvement of the author of this thesis, or other commercially available databases for legal research, should be able to accommodate experiments involving readers. In such an experiment it would be preferable to measure the number of steps taken rather than the *time* spent searching, as the latter relates the cognitive task of comprehending the written information, too. Our model may then be compared to a suitable null-model to separate the influence imposed by policy design from cognitive processes on the side of the reader. Specifically, a human reader can be expected to use memory to avoid parts of the text already visited, unless their search elsewhere turned out to be unsuccessful, too. In a null-model, this could be represented by a random walk with self-avoiding properties (see e.g. [112]). For instance, the walker may have a memory of a certain length in which the most recent steps are kept, and the walker will not repeat these unless there is no other option. Additionally, [100] suggests *stochastic resetting* as a natural component of search processes. Stochastic resets allow the walker to be reset to the initial (or a random) node with a fixed probability at each step.

With further developments following the direction of the present thesis, the future holds methods and tools for policy-makers to evaluate their drafts in terms of various aspects of complexity. In the face of modern information retrieval tools, such "analogue" structural layouts seem outdated. However, we still argue that there is an element of importance to the "analogue" design. Firstly, as described in [123], even professionals occasionally struggle formulating the right query for their information need, preventing them from using full-text databases efficiently. Secondly, studies like [126, 127] have found that readers without prior knowledge in the subject of a text navigate it more efficiently if provided with a hierarchical (hypertext) layout. In fact, statistical models such as structural topic models, presented in [108], can capitalise on the correlates represented by the relationships encoded in structured documents.

References

[1] C. Schulze, D. Stauffer, and S. Wichmann, “Birth, survival and death of languages by Monte Carlo simulation,” *Commun. Comput. Phys.*, **3**, pp. 271–294, 2008.

[2] C. Castellano, S. Fortunato, and V. Loreto, “Statistical physics of social dynamics,” *Rev. Mod. Phys.*, **81**, pp. 591–646, 2009.

[3] M. Jusup, P. Holme, K. Kanazawa, M. Takayasu, I. Romić, Z. Wang, S. Geček, T. Lipić, B. Podobnik, L. Wang, W. Luo, T. Klanjšček, J. Fan, S. Boccaletti, and M. Perc, “Social physics,” *Phys. Rep.*, **948**, pp. 1–148, 2022.

[4] S. Galam, Y. Gefen, and Y. Shapir, “Sociophysics: A new approach of sociological collective behaviour. I. mean-behaviour description of a strike,” *J. Math. Sociol.*, **9**, pp. 1–13, 1982.

[5] P. Clifford and A. Sudbury, “A model for spatial conflict,” *Biometrika*, **60**, pp. 581–588, 1973.

[6] V. Sood, T. Antal, and S. Redner, “Voter models on heterogeneous networks,” *Phys. Rev. E*, **77**, 041121 (13pp), 2008.

[7] L. Gamberi, P. Vivo, Y.-P. Förster, E. Tzanis, and A. Annibale, “Rationalizing systematic discrepancies between election outcomes and opinion polls,” *J. Stat. Mech.*, 123403 (23pp), 2022.

[8] D. M. Abrams and S. H. Strogatz, “Modelling the dynamics of language death,” *Nature*, **424**, p. 900, 2003.

[9] L. Steels, “A self-organizing spatial vocabulary,” *Artif. Life*, **2**, pp. 319–332, 1995.

[10] A. Baronchelli, M. Felici, V. Loreto, E. Caglioti, and L. Steels, “Sharp transition towards shared vocabularies in multi-agent systems,” *J. Stat. Mech.*, P06014 (12pp) 2006.

[11] A. M. Collins and M. R. Quillian, “Retrieval time from semantic memory,” *J. Verbal Learning Verbal Behav.*, **8**, pp. 240–247, 1969.

[12] R. Solé, B. Corominas-Murtra, S. Valverde, and L. Steels, “Language networks: Their structure, function, and evolution,” *Complexity*, **15**, pp. 20–26, 2010.

[13] M. Stella, N. M. Beckage, M. Brede, and M. De Domenico, “Multiplex model of mental lexicon reveals explosive learning in humans,” *Sci. Rep.*, **8**, 2259 (11pp), 2018.

[14] C. S. Siew, D. U. Wulff, N. M. Beckage, and Y. N. Kenett, “Cognitive network science: A review of research on cognition through the lens of network representations, processes, and dynamics,” *Complexity*, **2019**, 2108423 (24pp), 2019.

[15] J. Cong and H. Liu, “Approaching human language with complex networks,” *Phys. Life Rev.*, **11**, pp. 598–618, 2014.

[16] D. Brockmann, L. Hufnagel, and T. Geisel, “The scaling laws of human travel,” *Nature*, **439**, pp. 462–465, 2006.

[17] M. C. González, C. A. Hidalgo, and A. L. Barabási, “Understanding individual human mobility patterns,” *Nature*, **453**, pp. 779–782, 2008.

[18] H. Barbosa, M. Barthelemy, G. Ghoshal, C. R. James, M. Lenormand, T. Louail, R. Menezes, J. J. Ramasco, F. Simini, and M. Tomasini, “Human mobility: Models and applications,” *Phys. Rep.*, **734**, pp. 1–74, 2018.

[19] R. Kutner, M. Ausloos, D. Grech, T. Di Matteo, C. Schinckus, and H. Eugene Stanley, “Econophysics and sociophysics: Their milestones & challenges,” *Physica A*, **516**, pp. 240–253, 2019.

[20] F. Caccioli, P. Barucca, and T. Kobayashi, “Network models of financial systemic risk: a review,” *J. Comput. Soc. Sci.*, **1**, pp. 81–114, 2018.

[21] D. Adam, “Modeling the pandemic - The simulations driving the world’s response to COVID-19,” *Nature*, **580**, pp. 316–318, 2020.

[22] W. O. Kermack and A. G. Mckendrick, “A contribution to the mathematical theory of epidemics,” *Proc. Roy. Soc. London*, **115**, pp. 700–721, 1927.

[23] H. Abbey, “An examination of the Reed-Frost theory of epidemics,” *Hum. Biol.*, **24**, pp. 201–233, 1952.

[24] R. Pastor-Satorras and A. Vespignani, “Epidemic spreading in scale-free networks,” *Phys. Rev. Lett.*, **86**, pp. 3200–3203, 2001.

[25] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, “A survey of statistical network models,” *Found. Trends Mach. Learn.*, **2**, pp. 129–233, 2009.

[26] P. Erdős and A. Rényi, “On the evolution of random graphs,” *Publ. Math. Inst. Hung. Acad. Sci*, **5**, pp. 17–60, 1960.

[27] A. Lesne, “Complex networks: From graph theory to biology,” *Lett. Math. Phys.*, **78**, pp. 235–262, 2006.

[28] A.-L. Barabási and R. Albert, “Emergence of Scaling in Random Networks,” Science, **286**, pp. 509–511, 1999.

[29] L. Gamberi, Y.-P. Förster, E. Tzanis, A. Annibale, and P. Vivo, “Maximal modularity and the optimal size of parliaments,” *Sci. Rep.*, **11**, 14452 (15pp), 2021.

[30] D. J. Watts and S. H. Strogatz, “Collective dynamics of “smal-world” networks,” *Nature*, **393**, pp. 440–442, 1998.

[31] M. J. Bommarito, D. M. Katz, J. L. Zelner, and J. H. Fowler, "Distance measures for dynamic citation networks," *Physica A*, **389**, pp. 4201–4208, 2010.

[32] M. J. Bommarito, D. M. Katz, and J. Zelner, "Law as a seamless web? Comparison of various network representations of the United States Supreme Court corpus (1791-2005)," in *Proc. 12th ICAIL*, pp. 234–235, 2009.

[33] D. M. Katz and M. J. Bommarito, "Measuring the complexity of the law: the United States Code," *Artif. Intell. Law*, **22**, pp. 337–374, 2014.

[34] J. H. Fowler and S. Jeon, "The authority of Supreme Court precedent," *Soc. Networks*, **30**, pp. 16–30, 2008.

[35] D. M. Katz, C. Coupette, J. Beckedorf, and D. Hartung, "Complex societies and the growth of the law," *Sci. Rep.*, **10**, 18737 (14pp), 2020.

[36] B. Lee, K. M. Lee, and J. S. Yang, "Network structure reveals patterns of legal complexity in human society: The case of the constitutional legal network," *PLoS One*, **14**, e0209844 (15pp), 2019.

[37] W. P. Li, P. Azar, D. Larochelle, P. Hill, and A. W. Lo, "Law is code: A software engineering approach to analyzing the United States Code," *J. Bus. Tech. L.*, **10**, pp. 297–374, 2015.

[38] C. Coupette, D. Hartung, J. Beckedorf, M. Böther, and D. M. Katz, "Law smells: Defining and detecting problematic patterns in legal drafting," *Artif. Intell. Law*, 2022.

[39] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, **3**, pp. 993–1022, 2003.

[40] M. A. Livermore, A. B. Riddell, and D. N. Rockmore, "The Supreme Court and the judicial genre," *Ariz. Law Rev.*, **59**, pp. 837–901, 2010.

[41] G. Leibon, M. Livermore, R. Harder, A. Riddell, and D. Rockmore, "Bending the law: geometric tools for quantifying influence in the multinetwork of legal opinions," *Artif. Intell. Law*, **26**, pp. 145–167, 2018.

[42] F. Dadgostari, M. Guim, P. A. Beling, M. A. Livermore, D. N. Rockmore, "Modeling law search as prediction," *Artif. Intell. Law*, **29**, pp. 3–34, 2020.

[43] P. H. Schuck, "Legal complexity: Some causes, consequences, and cures," *Duke Law J.*, **42**, pp. 1–52, 1992.

[44] M. J. White, "Legal complexity and lawyers' benefit from litigation," *Int. Rev. Law Econ.*, **12**, pp. 381–395, 1992.

[45] A. D'Amato, "Legal uncertainty," *Calif. Law Rev.*, **71**, pp. 1–52, 1983.

[46] T. Sichelman, "Quantifying legal entropy," *Front. Phys.*, **9**, 665054 (14pp), 2021.

[47] D. Bourcier and P. Mazzega, "Toward measures of complexity in legal systems," in *Proc. Int. Conf. Artif. Intell. Law*, **704**, 2007, pp. 211–215.

[48] J. B. Ruhl, "Law's complexity: A primer," *Ga. State Univ. Law Rev.*, **24**, pp. 885–912, 2008.

[49] ——, "The fitness of law: Using complexity theory to describe the evolution of law and society and its practical meaning for democracy," *Vanderbilt Law Rev.*, **49**, pp. 1406–1490, 1996.

[50] Office of the Parliamentary Counsel, "When Laws Become Too Complex," 2013. [Online]. Available: https://www.gov.uk/government/publications/when-laws-become-too-complex/when-laws-become-too-complex, Retrieved: 15/10/2019

[51] N. E. Humphries, N. Queiroz, J. R. M. Dyer, N. G. Pade, M. K. Musyl, K. M. Schaefer, D. W. Fuller, J. M. Brunnschweiler, T. K. Doyle, J. D. R. Houghton, G. C. Hays, C. S. Jones, L. R. Noble, V. J. Wearmouth, E. J. Southall, and D. W. Sims, "Environmental context explains Lévy and Brownian movement patterns of marine predators," *Nature*, **465**, pp. 1066–1069, 2010.

[52] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. da Luz, E. P. Raposo, and E. Stanley, "Optimising the success of random searches," *Nature*, **401**, pp. 911–914, 1999.

[53] E. A. Codling, M. J. Plank, and S. Benhamou, "Random walk models in biology," *J. R. Soc. Interface*, **5**, pp. 813–834, 2008.

[54] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Comput. Networks ISDN Syst.*, **30**, pp. 107–117, 1998.

[55] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," 1998. [Online]. Available: http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf, Retrieved: 25/05/2021

[56] O. Bénichou and R. Voituriez, "From first-passage times of random walks in confinement to geometry-controlled kinetics," *Phys. Rep.*, **539**, pp. 225–284, 2014.

[57] S. Pandey and R. Kühn, "A random walk perspective on hide-and-seek games," *J. Phys. A*, **52**, 085001 (29pp), 2019.

[58] W. Feller, *An Introduction to Probability Theory and its Applications Vol. 1*, 3rd ed. New York: Wiley, 1968.

[59] C. R. Doering, K. V. Sargsyan, and L. M. Sander, "Mean extinction time for birth-death processes & failure of the Fokker-Planck approximation," in *AIP Conf. Proc.*, **800**, 2005, pp. 3–9.

[60] S. Redner, *A Guide to First-Passage Processes*. Cambridge: Cambridge University Press, 1951.

[61] D. Aldous, *Probability Approximations via the Poisson Clumping Heuristic*. Springer-Verlag, 1989.

[62] D. Aldous and J. A. Fill, *Reversible Markov Chains and Random Walks on Graphs*, 2002. [Online]. Available: http://stat-www.berkeley.edu/users/aldous/RWG/book.html, Retrieved: 03/07/2020

[63] L. Lovász, “Random walks on graphs: a survey,” *Comb. Paul Erdos is Eighty*, **2**, pp. 1–37, 1993.

[64] A. Kells, V. Koskin, E. Rosta, and A. Annibale, “Correlation functions, mean first passage times, and the Kemeny constant,” *J. Chem. Phys.*, **152**, 104108 (13pp), 2020.

[65] J. D. Noh and H. Rieger, “Random walks on complex networks,” *Phys. Rev. Lett.*, **92**, 118701 (5pp) 2004.

[66] A. Bassolas and V. Nicosia, “First-passage times to quantify and compare structural correlations and heterogeneity in complex systems,” *Commun. Phys.*, **4**, p. 1–14, 2021.

[67] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Princeton; London: Van Nostrad, 1960.

[68] C. D. Meyer, “Role of the group generalized inverse in the theory of finite Markov Chains.” *SIAM Rev.*, **17**, pp. 443–464, 1975.

[69] R. B. Bapat, “On the first passage time of a simple random walk on a tree,” *Stat. Probab. Lett.*, **81**, pp. 1552–1558, 2011.

[70] C. Van Den Broeck, “Waiting times for random walks on regular and fractal lattices,” *Phys. Rev. Lett.*, **62**, pp. 1421–1424, 1989.

[71] E. Agliari, “Exact mean first-passage time on the T-graph,” *Phys Rev E*, **77**, 011128 (6pp), 2008.

[72] V. Balakrishnan, E. Abad, T. Abil, and J. J. Kozak, “First-passage properties of mortal random walks: Ballistic behavior, effective reduction of dimensionality, and scaling functions for hierarchical graphs,” *Phys. Rev. E*, **99**, 062110 (16pp), 2019.

[73] A. Baronchelli, M. Catanzaro, and R. Pastor-Satorras, “Random walks on complex trees,” *Phys. Rev. E*, **78**, 011114 (9pp), 2008.

[74] A. Baronchelli and V. Loreto, “Ring structures and mean first passage time in networks,” *Phys. Rev. E*, **73**, 026103 (8pp) 2006.

[75] S. Bartolucci, F. Caccioli, F. Caravelli, and P. Vivo, “"Spectrally gapped" random walks on networks: A mean first passage time formula,” *SciPost Phys.*, **11**, pp. 1–19, 2021.

[76] O. C. Martin and P. Šulc, “Return probabilities and hitting times of random walks on sparse Erdös-Rényi graphs,” *Phys. Rev. E*, **81**, pp. 1–5, 2010.

[77] G. Frobenius, “Über Matrizen aus nicht negativen Elementen,” in *Sitzung der physikalisch-mathematischen Classe*, **1**, pp. 456–477, 1912.

[78] N. Masuda, M. A. Porter, and R. Lambiotte, “Random walks and diffusion on networks,” *Phys. Rep.*, **716-717**, pp. 1–58, 2017.

[79] G. Hummer and A. Szabo, “Optimal dimensionality reduction of multistate kinetic and Markov-State Models,” *J. Phys. Chem. B*, **119**, pp. 9029–9037, 2015.

[80] O. Matan and S. Havlin, "Mean first-passage time on loopless aggregates," *Phys. Rev. A*, **40**, p. 6573 (7pp), 1989.

[81] P. Chebotarev, "A graph theoretic interpretation of the mean first passage times," arXiv:math/0701359, 2007.

[82] J. Pitman and W. Tang, "Tree formulas, mean first passage times and Kemeny's constant of a Markov chain," *Bernoulli*, **24**, p. 1942–1972, 2018.

[83] A. Coulier, S. Hellander, and A. Hellander, "A multiscale compartment-based model of stochastic gene regulatory networks using hitting-time analysis," *J. Chem. Phys.*, **154**, 184105 (14pp), 2021.

[84] D. Kannan, D. J. Sharpe, T. D. Swinburne, and D. J. Wales, "Optimal dimensionality reduction of Markov chains using graph transformation," *J. Chem. Phys.*, **153**, 244108 (17pp), 2020.

[85] S. Condamin, O. Bénichou, and M. Moreau, "First-passage times for random walks in bounded domains," *Phys. Rev. Lett.*, **95**, 260601 (5pp), 2005.

[86] P. J. Schweitzer, "Perturbation theory and finite Markov chains," *J. Appl. Probab.*, **5**, pp. 401–413, 1968.

[87] C. A. O'Cinneide, "Entrywise perturbation theory and error analysis for Markov chains," *Numer. Math.*, **65**, pp. 109–120, 1993.

[88] G. E. Cho and C. D. Meyer, "Comparison of perturbation bounds for the stationary distribution of a Markov chain," *Linear Algebr. Appl.*, **335**, pp. 137–150, 2001.

[89] J. J. Hunter, "Stationary distributions and mean first passage times of perturbed Markov chains," *Linear Algebr. Appl.*, **410**, pp. 217–243, 2005.

[90] G. E. Cho and C. D. Meyer, "Markov chain sensitivity measured by mean first passage times," *Linear Algebr. Appl.*, **316**, pp. 21–28, 2000.

[91] A. Zeifman, V. Korolev, and Y. Satin, "Two approaches to the construction of perturbation bounds for continuous-time Markov chains," *Mathematics*, **8**, pp. 1–25, 2020.

[92] E. Deadman and S. D. Relton, "Taylor's theorem for matrix functions with applications to condition number estimation," *Linear Algebr. Appl.*, **504**, pp. 354–371, 2016.

[93] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Soc. Networks*, **5**, pp. 109–137, 1983.

[94] The National Archives, "Housing Act 1985." [Online]. Available: https://www.legislation.gov.uk/ukpga/1985/68/contents, Retrieved 06/12/2022

[95] M. G. Kohn and S. Shavell, "The theory of search," *J. Econ. Theory*, **9**, pp. 93–123, 1974.

[96] P. A. Diamond, "Aggregate demand management in search equilibrium," *J. Polit. Econ.*, **90**, pp. 881–894, 1982.

[97] G. Ramos-Fernández, J. L. Mateos, O. Miramontes, G. Cocho, H. Larralde, and B. Ayala-Orozco, "Lévy walk patterns in the foraging movements of spider monkeys (Ateles geoffroyi)," *Behav. Ecol. Sociobiol.*, **55**, pp. 223–230, 2004.

[98] D. Boyer, O. Miramontes, G. Ramos-Fernández, J. L. Mateos, and G. Cocho, "Modeling the searching behavior of social monkeys," *Physica A*, **342**, pp. 329–335, 2004.

[99] K. Carlson, F. Dadgostari, M. A. Livermore, and D. Rockmore, "Structure and content in the United States Code," *SSRN Electron. J.*, Available: https://ssrn.com/abstract=3690508, Retrieved: 28/11/2020 2020.

[100] M. R. Evans, S. N. Majumdar, and G. Schehr, "Stochastic resetting and applications," *J. Phys. A*, **53**, 193001 (67pp), 2020.

[101] D. M. Blei and J. D. Lafferty, "A correlated topic model of Science," *Ann. Appl. Stat.*, **1**, pp. 17–35, 2007.

[102] D. Blei, "Probabilistic topic models," *Commun. ACM*, **55**, pp. 77–84, 2012.

[103] T. D. Wilson, "On sser studies and information needs," *J. Doc.*, **37**, pp. 3–15, 1981.

[104] M. A. Wilkinson, "Information sources used by lawyers in problem solving: An empirical exploration," *Libr. Inf. Sci. Res.*, **23**, pp. 257–276, 2001.

[105] C. Courtright, "Context in information behavior research," *Annu. Rev. Inf. Sci. Technol.*, **41**, pp. 273–306, 2007.

[106] D. O. Case, *Looking for Information: A Survey of Research on Information Seeking*, 2nd ed., B. Boyce, Ed. London: Elsevier Academic Press, 2007.

[107] Y. Liu, C. Wang, K. Zhou, J. Y. Nie, M. Zhang, and S. Ma, "From skimming to reading: A two-stage examination model for web search," in *CIKM*, 2014, pp. 849–858.

[108] M. E. Roberts, B. M. Stewart, D. Tingley, and E. M. Airoldi, "The structural topic model and applied social science," *NIPS 2013 Work. Top. Model.*, pp. 2–5, 2013.

[109] F. Crestani, L. M. De Campos, J. M. Fernández-Luna, and J. F. Huete, "Ranking structured documents using utility theory in the Bayesian Network Retrieval Model," in *Int. Symp. String Process. Inf. Retr.*, 2003, pp. 168–182.

[110] M. Koniaris, I. Anagnostopoulos, and Y. Vassiliou, "Network analysis in the legal domain: A complex model for European Union legal sources," *J. Complex Networks*, **6**, pp. 243–268, 2018.

[111] C. Coupette, J. Beckedorf, D. Hartung, M. Bommarito, and D. M. Katz, "Measuring law over time: A network analytical framework with an application to statutes and regulations in the United States and Germany," *Front. Phys.*, **9**, pp. 1–23, 2021.

[112] V. M. López Millán, V. Cholvi, L. López, and A. Fernández Anta, "A model of self-avoiding random walks for searching complex networks," *Networks*, **60**, pp. 71–85, 2012.

[113] M. T. Chao and W. E. Strawderman, "Negative moments of positive random variables," *J. Am. Stat. Assoc.*, **67**, pp. 429–431, 1972.

[114] The National Archives, "Housing Act 2004." [Online]. Available: https://www.legislation.gov.uk/ukpga/2004/34, Retrieved: 22/11/2022

[115] E. Tzanis, P. Vivo, Y.-P. Förster, L. Gamberi, and A. Annibale, "Graphie: A network-based visual interface for UK's primary legislation," arXiv:2210.02165, 2022.

[116] Quantlaw, "Graphie: Housing Act 2004." [Online]. Available: https://graphie.quantlaw.co.uk/tree.json, Retrieved: 11/10/2022

[117] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in *Proc. first Instr. Conf. Mach. Learn.*, **242**, 2003, pp. 29–48.

[118] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," *Text Min. Appl. Theory*, pp. 1–20, 2010.

[119] The National Archives, "Localism Act 2011." [Online]. Available: https://www.legislation.gov.uk/ukpga/2011/20/contents, Retrieved: 27/11/2022

[120] S. Beliga, "Keyword extraction: a review of methods and approaches," *Univ. Rijeka, Dep. Informatics, Rijeka*, pp. 1–9, 2014.

[121] H. Ku, "Notes on the use of propagation of error formulas," *J. Res. Natl. Bur. Stand. Sect. C Eng. Instrum.*, **70C**, p. 263, 1966.

[122] R. Whalen, "Legal networks: The promises and challenges of legal network analysis," *Michigan State Law Rev.*, **2016**, p. 539, 2016.

[123] K. Davies, "The information-seeking behaviour of doctors: A review of the evidence," *Health Info. Libr. J.*, **24**, pp. 78-–94, 2007.

[124] T. T. Hills, P. M. Todd, and R. L. Goldstone, "Search in external and internal spaces: Evidence for generalized cognitive search processes," *Psychol. Sci.*, **19**, pp. 802-–808, 2008.

[125] T. T. Hills, P. M. Todd, and M. N. Jones, "Optimal Foraging in Semantic Memory Publication Date Optimal Foraging in Semantic Memory," *Proc. Annu. Meet. Cogn. Sci. Soc.*, **31**, pp. 620–625, 2009.

[126] F. Amadieu, A. Tricot, and C. Mariné, "Interaction between prior knowledge and concept-map structure on hypertext comprehension, coherence of reading orders and disorientation," *Interact. Comput.*, **22**, pp. 88-–97, 2010.

[127] F. Calisir and Z. Gurel, "Influence of text structure and prior knowledge of the learner on reading comprehension, browsing and perceived control," *Comput. Human Behav.*, **19**, pp. 135—145, 2003.

[128] S. Siddiqi and A. Sharan, "Keyword and keyphrase extraction techniques: A literature review," *Int. J. Comput. Appl.*, **109**, pp. 18–23, 2015.

[129] Y.-P. Förster, L. Gamberi, E. Tzanis, P. Vivo, and A. Annibale, "Exact and approximate mean first passage times on trees and other necklace structures: a local equilibrium approach," *J. Phys. A*, **55**, 115001 (33pp), 2022.

[130] Y.-P. Förster, A. Annibale, L. Gamberi, E. Tzanis, and P. Vivo, “Information retrieval and structural complexity of legal trees,” *J. Phys. Complex.*, **3**, 035008 (28pp), 2022.

Appendix A

List of Publications

The author of the present thesis was published the articles listed in tab. A.0.1 in the course of the project. Those published as leading author are [129] and [130], which largely correspond to the chapters 2 and 4, respectively.

Table A.0.1 Publications and submitted works prepared during the course of the project. *as leading author

[29]	L. Gamberi, YPF, E. Tzanis, A. Annibale, and P. Vivo, "Maximal modularity and the optimal size of parliaments," *Sci. Rep.*, **11**, 14452 (15pp), 2021.
[129]*	YPF, L. Gamberi, E. Tzanis, P. Vivo, and A. Annibale, "Exact and approximate mean first passage times on trees and other necklace structures: a local equilibrium approach," *J. Phys. A Math. Theor.*, **55**, 115001 (33pp), 2022.
[130]*	YPF, A. Annibale, L. Gamberi, E. Tzanis, and P. Vivo, "Information retrieval and structural complexity of legal trees," *J. Phys. Complex.*, **3**, 035008 (28pp), 2022.
[7]	L. Gamberi, P. Vivo, YPF, E. Tzanis, and A. Annibale, L. Gamberi, P. Vivo, YPF, E. Tzanis, and A. Annibale, "Rationalizing systematic discrepancies between election outcomes and opinion polls," *J. Stat. Mech.*, 123403 (23pp), 2022.
[115]	E. Tzanis, P. Vivo, YPF, L. Gamberi, and A. Annibale, "Graphie: A network-based visual interface for UK's Primary Legislation," arXiv:2210.02165, 2022.

Appendix B

Supplementary Material

With this thesis comes the code used for chapters 4 and 5. Alongside the `pattern_walker` module, we provide a three-minute animation of the walker, generated using the notebook “animation-example.ipynb” in the directory “examples” of the module.

Below we include a snapshot of a notebook performing the steps as described chapter 5. This notebook can be found at the location “examples/inference_example.ipynb”.

In [17]:

```
import numpy as np
import utils
import matplotlib.pyplot as plt

import pattern_walker as pw
from pattern_walker.mean_field import MF_patternWalker_general,MF_mfpt_cary_tree
import networkx as nx

from copy import deepcopy

import warnings
warnings.filterwarnings('ignore') # suppress warning for presentability
```

In [2]:

```
source = 'HousingAct.json'
```

In [3]:

```
data = utils.load_data(source)
```

In [4]:

```
# dictionary where all documents within the same part are stored in one "flat" list
parts_flat = utils.flat_parts_json(data)
```

In [5]:

```
# keyword parameters for preprocessing
instructions = {'analyzer':'words', 'ngram_range':(1,4),'norm':'l1','sublinear_tf':True}
vocabs = utils.get_vocabs(parts_flat,num_top_terms=100,**instructions)
```

In [6]:

```
estimators = {} #hold the parameters passed to patternWalker class later
#can also be solved with utils.get_mask_length once on a tree

estimators['pattern_len'] = len(vocabs[0]) # L
c = len(vocabs[2]) #number of parts
#estimate Delta
estimators['overlap'] = np.max([ (sum(p)-len(vocabs[0])/c)/2 for p in vocabs[2] ])
print(estimators)
```

```
{'pattern_len': 650, 'overlap': 3.5714285714285694}
```

In [7]:

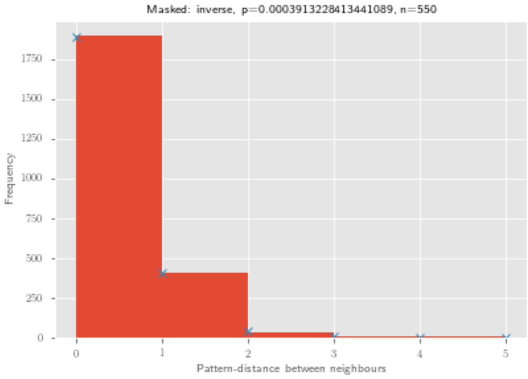
```
# data dict with all text items replace by patterns under key "name"
vec_data = utils.vectorise_json(vocabs[0],data,binary=True)
# add binary vector to part nodes indicating topic-specific and generic keywords
for ndx,part in enumerate(vec_data['_children']):
    part['mask'] = vocabs[-1][ndx]
```

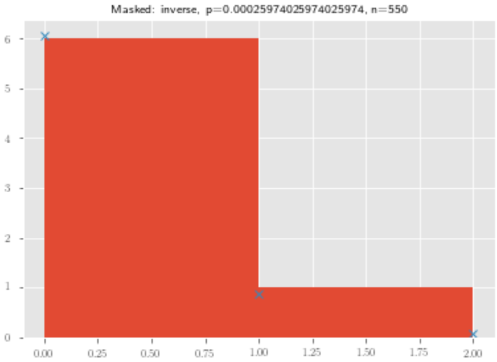
In [8]:

```
other_params={} #for standard deviations and other estimators not to be fed
#into patternWalker class

#make a networkx.DiGraph from the dict
tree = utils.build_tree(vec_data,int_ids=True)
other_params['number of nodes'] = len(tree)
root=vec_data['node_id']

estimators['root']=root

# for convenience, add to each node a hint to which part it belongs
utils.add_part_attribute(vec_data,tree)
```

```python
# list of nodes on part level...
part_nodes = [node['node_id'] for node in vec_data['_children'] ]
# ...and below part level
non_part_nodes = list(set(tree.nodes)-set(part_nodes)-set([root]))

# list of lists indicating part membership
parts = [
            [ node for node in non_part_nodes
              if tree.nodes[node]['part']==part ]
         for part in part_nodes]

# estimate a_h from topic-specific keyword presence (masked=True)...
temp_h = np.array(list(utils.estimate_a(tree,tree.nodes,masked=True)[1].values()))
# ... and a_l from generic keyword presence (masked='inverse')
temp_l = np.array(list(utils.estimate_a(tree,tree.nodes,masked='inverse')[1].values()))

# a_h is the mean of the above list
a_h = np.mean(temp_h)
sigma_h = np.sqrt(a_h*(1-a_h)/len(temp_h))
a_l = np.mean(temp_l)
sigma_l = np.sqrt(a_l*(1-a_l)/len(temp_l))

# estimate a_root as the difference between a_h and a_l
a_root,sigma_root = a_h-a_l, np.sqrt(abs(sigma_h**2+sigma_l**2))

# get next-neighbour distances in the tree, excluding part and root nodes.
#'inverse' has most equidistributed bits
distances = utils.get_distances_to_parent(tree,masked='inverse',nodes=non_part_nodes)
n=utils.get_mask_length(tree,nodes=non_part_nodes,masked='inverse')

p=np.mean([ distances[node]/n[node] for node in distances.keys() ])
sigma_p = p*(1-p)/len(distances)


# plot next-neighbour distances as histogram and compare to binomial
hist_data=utils.get_distance_histogram(tree,masked='inverse',nodes=non_part_nodes)
plt.xlabel('Pattern-distance between neighbours')
plt.ylabel('Frequency')

# same as above between part-level and root nodes
root_part_distances = utils.get_distances_to_parent(tree,masked='inverse',
                                                        nodes=part_nodes)
root_hist_data=utils.get_distance_histogram(tree,masked='inverse',nodes=part_nodes)

n_part = utils.get_mask_length(tree,part_nodes,'inverse')

p_p =  np.mean([ root_part_distances[node]/n_part[node]
                for node in root_part_distances.keys() ] )
sigma_p_p = p_p*(1-p_p)/len(root_part_distances)


# save estimated parameters
estimators['a_high'] = a_h
estimators['a_low'] = a_l
estimators['a_root'] = a_root
estimators['Gamma']=p/(2*a_l*(1-a_l))
estimators['Gamma_root'] = (p_p-a_root+a_l)/(2*(1-a_root))

other_params['p']=p,sigma_p
other_params['p_p']=p_p,sigma_p_p

other_params['sigma_h'] = sigma_h
other_params['sigma_l'] = sigma_l
other_params['sigma_root']= sigma_root
other_params['sigma_Gamma']=\
            1/(2*a_l*(1-a_l))*np.sqrt(sigma_p**2+p**2*sigma_l**2*\
                                          (1-2*a_l)**2/(a_l**2*(1-a_l)**2))
other_params['sigma_Gamma_root'] =\
            2/(1-a_root)*np.sqrt(sigma_p_p**2+sigma_l**2+sigma_root**2/(1-a_root)**2)
```

```
Power_divergenceResult(statistic=206.60692931734678, pvalue=3.557208439393673e-05)
Power_divergenceResult(statistic=0.08601682890853525, pvalue=1.0)
```





In [9]:

```
estimators,other_params
```

Out[9]:

```
({'pattern_len': 650,
  'overlap': 3.5714285714285694,
  'root': 0,
  'a_high': 0.0019097130823901027,
  'a_low': 0.0012321690184285603,
  'a_root': 0.0006775440639615424,
  'Gamma': 0.1574520636295045,
  'Gamma_root': 0.0003982386721382268},
 {'number of nodes': 2359,
  'p': (0.00038753700918737067, 1.6477533996336911e-07),
  'p_p': (0.00024131274131274132, 3.446492992480306e-05),
  'sigma_h': 0.0008990773893079629,
  'sigma_l': 0.0007224295716393701,
  'sigma_root': 0.0011533623185902451,
  'sigma_Gamma': 0.09220141508608323,
  'sigma_Gamma_root': 0.002725918432365252})
```

In [10]:

```
#print top-5 terms
```

```
for part in vocabs[1]:
    print(part[:5])
print('-----------')
```

```
['publish', 'premise landlord', 'tenant premise', 'reference premise', 'specified period d
ay']
['exemption notice serve respect', 'date temporary exemption notice', 'temporary exemption
notice serve', 'designation authority', 'person require pay']
['reference matter circumstance', 'financial penalty person respect', 'relevant award', 't
ime review operation designation', 'penalty alternative prosecution']
['planning include', 'letting', 'legal owner premise purpose', 'house right', 'local autho
rity deal']
['supplementary', 'act estate agent', 'duty person act estate', 'person act estate agent',
'home information pack']
['relation agreement arrangement day', 'time deposit hold accordance', 'tenancy time depos
it hold', 'tenancy require deposit consist', 'mobile home c. particular']
['apply converted block', 'household regulation', 'wale order connection', 'consent', 'pre
mise effectively secure trespasser']
-----------
```

In [11]:

```
estimators['target'] = 226 # for reproducibility
print(estimators)
```

```
{'pattern_len': 650, 'overlap': 3.5714285714285694, 'root': 0, 'a_high': 0.001909713082390
1027, 'a_low': 0.0012321690184285603, 'a_root': 0.00067754406396l5424, 'Gamma': 0.15745206
36295045, 'Gamma_root': 0.0003982386721382268, 'target': 226}
```

In [12]:

```
#store vectors under new name 'pattern' in tree...
pattern_tree = deepcopy(tree)
pattern_dict=nx.get_node_attributes(pattern_tree,'name')
for node,vector in pattern_dict.items():
    pattern_dict[node]=np.squeeze(vector)
nx.set_node_attributes(pattern_tree,pattern_dict,'pattern')
PW_emp = pw.empiricalPatternWalker(pattern_tree,root=estimators['root'],\
                                   target=estimators['target'])
PW_MF= MF_patternWalker_general(pattern_tree,**estimators)
```

In [13]:

```
#make a diffusive version too, in which all patterns are just 0
PW_diffusive = MF_patternWalker_general(pattern_tree,estimators['root'],\
                                        estimators['pattern_len'],0.,0.,0.,0.,0.,0.,\
                                        target=estimators['target']
                                       )
```

In [14]:

```
m_MF=pw.utils.mfpt(PW_MF,[(PW_MF.root,PW_MF.target_node)],weight_str='mean_weight',\
                   method='grounded_Laplacian')
print(m_MF,m_MF/len(PW_MF))
```

```
15712.192504210238 6.660530947100567
```

In [15]:

```
m_emp=pw.utils.mfpt(PW_emp,[(PW_emp.root,PW_emp.target_node)],weight_str='weight',\
                    method='grounded_Laplacian')
print(m_emp,m_emp/len(PW_emp))
```

```
22219.661904768953 9.419102121563778
```

In [16]:

```
m_diff=pw.utils.mfpt(PW_diffusive,[(PW_diffusive.root,
          PW_diffusive.target_node)],weight_str='weight',\
                     method='grounded_Laplacian')
print(m_diff,m_diff/len(PW_diffusive))
```

```
22243.000000010714 9.428995337011749
```