SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

Utrecht University

When Memory Pays: Discord in Hidden Markov Models

Author:
Emma Lathouwers

Supervisors:
Prof. Dr. John Bechhoefer
Dr. Laura Filion

Master's thesis

July 8, 2016

Abstract

When does keeping a memory of observations pay off? In this thesis, we use hidden Markov models to look at phase transitions that emerge when comparing state estimates in systems with discrete states and noisy observations. We infer the underlying state of the hidden Markov models from the observations in two ways: through naive observations, which take into account only the current observation, and a state estimate found through Bayesian filtering, which takes the history of observations into account. We compare the state estimates by calculating the discord parameter and study the behaviour of the discord. We then explore a range of different hidden Markov models to investigate how general the behaviour of the discord is and, in particular the phase transition to non-zero discord. We varied the symmetry of the system, number of symbols, number of states, and parameters of the system. We find quantitatively similar behaviour in all the considered systems, and we can calculate the critical point where keeping a memory of observations starts to pay off. We also explored a mapping from hidden Markov models to Ising models to get more insight in the emergence of the phase transitions. We only find the phase transitions when comparing the two state estimates, not when we compare the true, hidden state with a state estimate.

Contents

Acknowledgement

I would like to sincerely thank my supervisors Laura Filion and John Bechhoefer for their support in my crazy Canadian adventure right from the start. Thank you, Laura, for helping me, answering all of my questions, and making me believe everything would work out. Thank you, John, for welcoming me into the lab, and for all of your time and energy. I had an amazing time coming to Vancouver.

I also would like to thank Laura Geissler for helping me find my way at SFU and around Vancouver. Thank you, Momčilo, Robert, Aidan, and many others, for the great time in the lab and outside of it, with coffee, lunch, hikes, games, and movies. Thanks to everyone in the lab for group meetings and questions, it was so helpful for organizing my thoughts. David, thank you for all the walks around SFU: It was a great help in keeping my sanity when I was a little stressed out. And, lastly, many thanks to Felix for proofreading my thesis when he could have been out enjoying the summer.

Chapter 1

Introduction

1.1 Inference problems

Inference is the act of deriving of logical conclusions from information known (or assumed) to be true. In statistical inference, properties of an underlying distribution are deduced from the analysis of data. The data is usually only a sample of a larger population, and statistical methods are used to infer properties of the total population. Nowadays, inference problems appear everywhere: in neuroscience [1], signal processing [2], artificial intelligence [3], and many other areas.

There are several established schools or paradigms of statistical inference [4, Chapt. 17]. We will restrict ourselves to Bayesian inference in this thesis. Bayesian inference is a method of statistical inference in which probabilities are updated using Bayes' theorem as more information becomes available. Bayes' theorem [5] for an event A, given an event B, is

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{1.1}$$

It relates the posterior probability of an event A given an event B, $P(A|B)$, to the prior probability, $P(A)$, and the likelihood $P(B|A)/P(A)$ which expresses the impact of the event B on the probability of A. The prior probability (prior to getting more information) is updated with the likelihood to get the posterior probability.

Some examples of problems that can be tackled by Bayesian inference include decoding problems for error-correcting codes, the inferring of clusters from data, interpolation through noisy data, and the classifying of patterns given labelled examples [6, Part IV]. In exact inference, the conditional probability distribution of a variable of interest is calculated analytically. In problems that cannot be solved exactly, approximation techniques are implemented, for example when the exact model parameters are not known in addition to the variables that one wants to infer.

The conditional dependance structure between random variables can be represented by graphs. Each node of a graph is associated with a probability function that depends on the parent variables: nodes from which you can directly get to the node under consideration. The nodes to which you can get in a single step are called the children of that node. Bayesian networks, or directed acyclic graphs [4, Chapt. 10], are a common graphical representation. The probability in such graphs can only 'flow' in one direction (an arrow of time), and loops are not allowed. An example of a directed acyclic graph is shown in Figure 1.1. The joint probability distribution functions of directed acyclic graphs can be factored, which makes inference possible in a straightforward way. We will look at Markov chains and hidden Markov models, which are types of directed acyclic graphs. Markov chains are a special case of directed acyclic graphs, where each node has only one parent and one child node. In Chapter 2, we will present the theory of these systems.

There are three main types of inference for Bayesian networks: inferring unobserved variables, parameter learning, and structure learning [4, Chapt. 10]. The first type assumes that the model parameters are

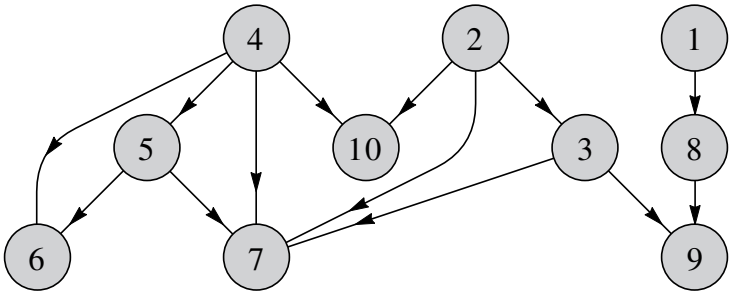

Figure 1.1: Randomly generated directed acyclic graph.

known, and the hidden state of the network is inferred from observations. The second uses the realization of a network and the knowledge of its structure to infer the model's parameters. In the last case, the realization of a network is used to infer the structure of the model. We will be concerned with issues that fall into the first class of problems: inferring the unobserved variables of a model, assuming that the correct structure and parameters of the model are known. This is a problem that we are faced with daily, whenever we have to make decisions based on incomplete or uncertain data. For example, deciding whether or not to take an umbrella based on the weather forecast, determining your next move in a game without knowing your opponent's strategy, or tracking objects with a noisy sensor.

Two important, more general, scientific questions about inference are: The question of sufficient information, Under what conditions can the variables of interest be recovered sufficiently well? And the question of computational efficiency, Can the inference problem be solved algorithmically efficiently? And, Can an optimal algorithm be formulated? A particular research interest is defining when an inference problem is easy, hard or impossible, and developing efficient algorithms to infer the variables with as little information as possible [5, 7]. These boundaries are still unknown for general inference problems.

1.1.1 Decision making

In many applications of Bayesian inference, simply calculating the probability of the system being in a certain state is not enough. Often, one needs to make a decision based on the inference outcome. For example, one might want to act the system based on observations: decide whether or not to bring an umbrella based the forecast or applying a control signal to a system.

These type of problems are called *statistical decision problems* and fall in the general area of control (feedback) theory. They can be thought of as playing a game against nature; this is similar to playing a game against a strategic player, as is usually done in game theory [8]. In either case, the objective is to win the game, maximize winnings, or minimize losses. Only, in the case of playing against nature, the other player is not actively trying to optimize something.

This paragraph is based on [9, Chapt. 14]. The performance of a strategy (inference + decision making) can be measured by a cost function. It measures the compatibility of the actions that make up possible strategies with the variable to be inferred. In order to get the best possible result, one tries to find the optimal procedure for each possible situation: minimizing the expected loss. In situations where stochastisticity is important, the expectation value of the loss function is also known as the risk function. In problems where the state of a system is estimated from observations, a commonly used cost function is 0-1 loss. The loss is 0 when the state coincides with the state estimate from observations, and 1 when they do not coincide. The strategy that minimizes the expected loss is the posterior mode or maximum a posteriori (MAP) estimate in this case. We will come back to the MAP estimator later. When inferring model parameters, the mean-square error is commonly used as risk function. In physics, it is often used to fit a curve to data. The mean value of a parameter can be found from the minimum of the least-square cost function. Similarly, the median value of a parameter is found using the absolute value cost function.

Adding a decision-making component to a hidden Markov model results in a partially observed Markov

decision processes (POMDP) [4, Chapt. 10]. POMDPs are generally not exactly solvable, but approximate algorithms have been developed [10]. Some interesting applications are aircraft collision avoidance [8], and the conservation of the critically endangered and difficult to detect Sumatran tigers [11].

1.1.2 Statistical mechanics and inference problems

So far, we have talked about inference from as an information-theoretical problem. However, it turns out that many inference problems are essentially the same as some physical systems. Shannon introduced an entropy as a measure of difference in information content. Though this entropy has the same form as the traditional statistical mechanical entropy, he did not make a connection between the two. Later, Jaynes suggested that statistical physics could just be a special case of a more general theory of Bayesian inference [5], but this is heavily debated [12].

The joint values of random variables in a statistical inference problem can be interpreted as possible microstates of an imaginary physical system. In neural networks, Ising models and Potts models are commonly seen in inference problems. The inferring of variables, depending on what exactly one is inferring and how, can be the same problem as the minimalization of a Hamiltonian. MAP estimation can be seen as the ground state of a physical system; the minimum of the Hamiltonian of that system. The problem of graph coloring turns out to be equivalent to studying the zero temperature behaviour of the anti-ferromagnetic Potts model [13]. Also, there are connections between spin glasses and error-correcting codes [14]. And, the statistical mechanics of learning with a neural networks is also similar to that of spin glasses [15–17].

1.2 Phase transitions

A remarkable feature of statistical physics is the possibility to have phase transitions between states of matter. The term *phase transition* is often used to indicate transitions between states of matter: liquid, solid, and gas. However, there are many different kinds of phase transitions, such as the ferro- to paramagnetic transitions in magnetic materials [18], superconducter to superfluid transitions [19], and even symmetry breaking transitions in the laws of physics as the early universe cooled [20].

We distinguish between first- and second-order phase transitions [5]: first-order transitions have discontinuities in the first derivative of the free energy with respect to some thermodynamic variable, whereas second-order transitions are continuous in the first derivative of the free energy but have discontinuities in their second derivatives. An order parameter measures the degree of order across phase transitions. In liquid-gas transitions, the order parameter is the difference in densities, and in magnetic systems, it is the net magnetization. The different phases of a system and the phase transitions can be visualized in a *phase diagram*.

1.2.1 Phase transitions in inference problems

We have seen that certain inference problems are equivalent to standard physical systems. When a phase transition is found in a physical system, it makes sense that it also occurs in the analoguous inference problem. They can be found in parameter estimation problems [21], in state estimation problems in networks [22], in state estimation in hidden Markov models [23], and in a Maxwell demon system that is very similar to a hidden Markov model [24]. As we will focus on phase transitions in hidden Markov models that are relevant to the understanding of similar transitions studied in Maxwell demons, we present this application in more detail; Maxwell made up a demon to demonstrate the limitations of the second law of thermodynamics [25]. Consider a chamber filled with gas, divided into two equal sized chambers, and connected through a trapdoor. An intelligent creature such as a Maxwell demon could observe the speed of the gas molecules, and open the trapdoor whenever a fast molecule approaches the door from the left and when a slow molecule approaches the door from the right. The demon sorts the molecules based on speed, creating a hot and a cold reservoir. The temperature difference between the

reservoirs could be used to do useful work. However, this violates Clausius's formulation of the second law; it is not possible to transfer heat from a colder body to a hotter one, on average, without doing any work. For decades, the increase in entropy was attributed to the acquisition of information, but we know now that the erasure of information, not the acquisition, is the source of the entropy increase [26].

A Maxwell demon that can experimentally be realized to convert information into work is a particle in a double-well potential [24]. Phase transitions are found as the observation error probability is varied [27], and when the strategies of using memory and not using memory are compared [24]. This phase transition is found in a 'discord' order parameter D as the observation error probability is increased. These transitions also turn up in symmetric two-state, two-symbol hidden Markov models, which we will discuss in more detail in Chapters 2 and 4. Interestingly, these transitions are only seen when comparing the state estimates based on filtering and naive observations, not when comparing the inferred and underlying state [23].

1.3 Outline

In this thesis, we will explore the phase transitions as seen in hidden Markov models where the states have to be inferred from observations. We will explore the properties of the phase transitions. In particular, we will study a range of hidden Markov models to see how general these transitions are and how robust they are to the details of the model. We will formulate more detailed questions and aims after we introduce the theory necessary to study these systems.

This thesis is organized as follows: First, we discuss the theoretical background of the systems we study and start looking at symmetric two-state, two-symbol HMMs, systems in which phase transitions have been observed [24], in Chapter 2. In Chapter 3, the simulation methods used in generating HMMs and measuring their properties are discussed. Then, in Chapters 4-6, we study various HMMs and their phase transitions. Finally, the results will be discussed and the conclusions will be presented in Chapter 7.

Chapter 2

Theory

In this chapter, we will build up the theory needed to study discord in hidden Markov models. We will start with the basics of Markov chains and hidden Markov models. Then, we discuss state estimation, in particular Bayesian filtering, and we introduce the discord parameter as a measure to compare two state estimates. Finally, all of this is applied to one of the simplest non-trivial hidden Markov models: the symmetric two-state, two-symbol hidden Markov model.

2.1 Hidden Markov models

2.1.1 Markov chains

One of the most widely used discrete state-space systems is a Markov chain. Its relatively straightforward mathematical description has applications to a wide range of topics, from physics [28, 29] to economics [30, 31] to biology [32, 33] to information science [34, 35]. Discrete-time Markov chains describe the evolution of a system in discrete time t, with discrete states x_t. What makes them special is the Markov property, which refers to the memorylessness of the process: The next state x_{t+1} of a system with Markovian dynamics depends only on the current state x_t, and not on past of states,

$$P(x_{t+1}|x_t, x_{t-1}, \ldots, x_2, x_1) = P(x_{t+1}|x_t). \tag{2.1}$$

A graphical presentation of a Markov chain in time is shown in Figure 2.1. It is a simple directed acyclic graph; each node has only one parent and one child node. We will consider only *time-homogeneous* Markov chains, which have time-independent transition probabilities [36]. A common example of a Markov chain is a simple weather model (Figure 2.2). The state of this system can be either 'Sunny' or 'Rainy', and the current state determines the probabilities to transition to another state. For a more realistic (and more complicated) model, we might want to take into account seasonal changes, a greater variety of states, or dependence on more past states. Physical systems are often coarse grained when modelled with Markov chains [37] — a continuum of states is made discrete or the number of states is decreased. Consider, for example, an ion channel in the cell membrane [38]. Markov chains can be used to describe this system by coarse graining over appropriate timescales. An ion channel can be open or closed, but it can also be

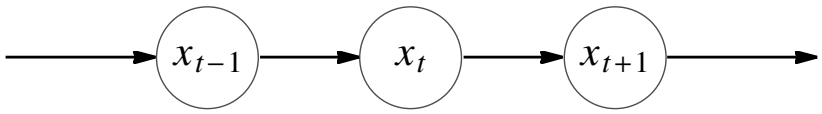

Figure 2.1: Graphical presentation of a Markov chain. State x_{t+1} depends only on state x_t.

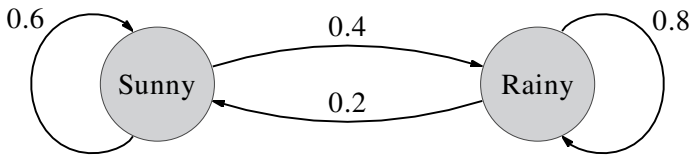

Figure 2.2: Example of a Markov chain, a simple weather model with two states.

partly open. When the channel is studied at timescales where opening and closing of channels is fast and the time spent in an intermediate state is neglible, the approximation of two states is fairly accurate. The coarse-graining timescale should be longer than the correlation time, so that the system can be accurately described as a Markov process, but it should be short enough to capture all the details of the dynamics.

The dynamics of an n-state Markov chain can be described using an $n \times n$ transition matrix $\mathbf{A}$ with elements $\mathbf{A}_{ij} = P(x_{t+1} = i|x_t = j)$ [24]. All entries should satisfy $0 \leq \mathbf{A}_{ij} \leq 1$, because they are normalized probabilities. We use left stochastic matrices, and consequently the columns of the matrix should sum to 1, $\sum_i \mathbf{A}_{ij} = 1$, to ensure proper normalization in a closed system. In words, the probability of being in any of the states, given the previous state, is one. The steady-state distribution $\boldsymbol{\pi}$ of $\mathbf{A}$ is defined as $\boldsymbol{\pi} = \mathbf{A}\boldsymbol{\pi}$, if the components of $\boldsymbol{\pi}$ are positive and their sum is finite. The matrix needs to satisfy certain conditions to make sure a unique stationary state exists. A discrete-time Markov process has a unique stationary distribution if it is time homogeneous, irreducible and aperiodic. A Markov process is *irreducible* if every state can be reached from any state, and *aperiodic* if the states can be visited at irregular times. Most of the time, we will study matrices that have only non-zero elements, in which case all of these conditions are met automatically. If the steady-state distribution exists, then $\boldsymbol{\pi}$ is also the limiting distribution for $t \to \infty$, and the process is *ergodic*, since the proportion of time spent in each state converges to $\boldsymbol{\pi}$ [36].

In physics, we often demand that a system obey detailed balance. Detailed balance is met when it is impossible to distinguish between a system going forward or backward in time. Billiard balls or atoms colliding are systems where this is a natural condition. When a Markov chain satisfies the *detailed balance* condition,

$$\mathbf{A}_{ij}\boldsymbol{\pi}_j = \mathbf{A}_{ji}\boldsymbol{\pi}_i, \tag{2.2}$$

for all components of the vector $\boldsymbol{\pi}$ and these components sum to unity, the Markov chain is reversible. If this vector exists, it is the steady-state distribution of the process [36].

2.1.2 Hidden Markov models

Markov chains are convenient for modelling a range of physical processes. However, the process of interest is often not directly observable. If we are lucky, we can observe a variable that correlates with the desired quantity. In which case, we can use that variable to infer the state of the hidden process. We can describe this, the hidden state and the correlated observabed symbol, with a hidden Markov model (HMM). The structure of such a model is shown in Figure 2.3. To clarify, let us consider the ion channel once again. An ion channel is so small that we cannot directly observe it, and want to measure the state of the channel, open or closed. It turns out, measurements of the electrical conductance of ion channels can be used to infer the state of the channels [39]. When a channel is open, ions can move through the channel, and there is a flow of charge. When it is closed, there is no flow of ions; when it is partly open, there is some flow, but less than for an open channel. The measurements will have errors; there may be natural fluctuations in the system and measurements are usually not perfect. Here, imperfect measurements that correlate with hidden states make the model a hidden Markov model.

The evolution of the hidden state is described by a Markov chain, as we have seen before, $\mathbf{A}_{ij} = P(x_{t+1} = i|x_t = j)$. The observation of an emitted symbol y_t is described by a time-homogeneous $m \times n$

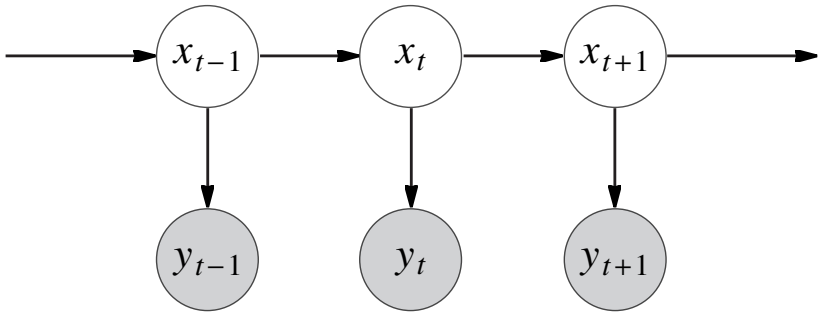

Figure 2.3: Graphical presentation of the dependence structure of a HMM. The variables x are hidden states with Markovian dependence; the variables y are observations that correlate with the underlying, hidden, states.

observation matrix $\mathbf{B}$, with elements $\mathbf{B}_{ij} = P(y_t = i|x_t = j)$. The emission of symbols is described by a Markov chain that depends on the hidden state. The observations have no memory and depend only on the current state of the system x_t. The number of states n and symbols m does not have to be equal; m can either be bigger or smaller than n [24]. Usually, one considers $m \geq n$, so that the observations can represent the entire state space. We will refer to an m-state, n-symbol HMM as an $m \times n$ HMM. The simple weather model, in Figure 2.2, has two states and two symbols. The discussed ion-channel model is slightly more complicated, it has two states and infinitely many symbols (a continuum of possible observations). The behaviour of a hidden Markov model is governed completely by its transition matrix $\mathbf{A}$ and observation matrix $\mathbf{B}$.

The observations as a function of time $P(y_{t+1}|y_t)$ are generally not Markovian: The probability of an observation conditioned on previous observations $P(y_{t+1}|y_t, y_{t-1}, \ldots, y_1)$ does not depend on only y_t. However, the combined process of $P(x_t, y_t)$ can be described as a Markov chain: A Markov chain on the product space of states and symbols $\{X_t \times Y_t\}$ [39]. This becomes clear when we rewrite the joint probability, $P(x_t, y_t) = P(y_t|x_t)P(x_t)$. Since both x_t and y_t are determined entirely by x_{t-1}, (x_t, y_t) can be written as functions of (x_{t-1}, y_{t-1}). Hence, we can use the properties of Markov chains for hidden Markov models as well. We just have to be careful to apply them to the right process. In a HMM where the possible states and emitted symbols are isomorphic, the emitted symbols can be interpreted as noisy observations of the state.

2.2 State estimation

Two common problems associated with HMMs are inference of the model parameters and inference of the sequence of hidden states. For simplicity, we assume that we have perfect knowledge of our model parameters, and we will focus on state estimation. State estimation or optimal filters are traditionally used in telecommunications [40]. A signal (sequence of states) is transmitted and the receiver wants to minimize the signal's distortion due to noise. Algorithms are available to infer the most likely state at each time or the most likely sequence of states, among other inference tools.

There are a number of algorithms that try to infer the output sequence of a HMM. Which algorithm is the most suitable for a particular problem depends on the quantity you want to infer and the information you have available. In this section, we will review a few of these algorithms. The focus of this work is on the Bayesian filtering algorithm, which will be discussed in more detail in Section 2.2.1.

- *Filtering*, $P(x_t|y^t)$, estimates the probability of being in state x_t based on current and past observations $y^t \equiv \{y_1, y_2, \ldots, y_t\}$. It is interesting for real-time applications, such as feedback systems. The modelling of a system from noisy data and/or imperfect knowledge of the dynamics can be formulated as an optimal filtering problem [40].

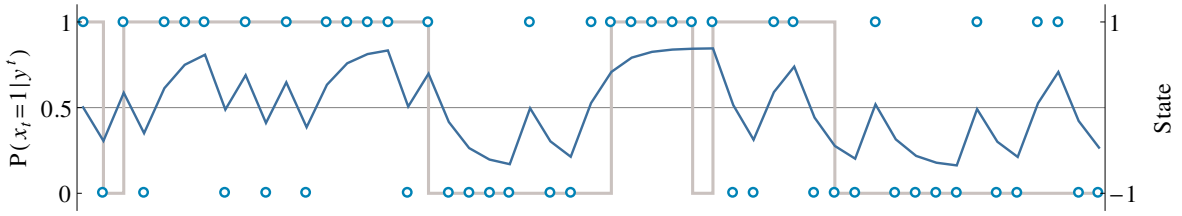

Figure 2.4: Realization of a two-state, two-symbol HMM for 50 timesteps. The hidden state is shown in gray, the emitted symbols in blue open circles and the probability from the Bayesian filtering equations in blue. Note that the hidden state is actually discrete but is shown as a line for visibility.

- *Smoothing*, $P(x_t|y^N)$, similar to filtering, estimates the probability of being in state x_t based on past, current and future observations $y^N \equiv \{y_1, y_2, \ldots, y_t, \ldots, y_N\}$. It is generally more accurate than filtering; however, it can only be used for post processing. Bayesian smoothing can be used in any situation where Bayesian filtering is appropriate, excluding real-time applications.

- *Maximum a posteriori (MAP) estimation*, can be used to find the most likely path given a sequence of observations $P(x^N|y^N)$. Unfortunately, for long sequences this problem is very complex, since all possible paths x^N have to be considered. A commonly used algorithm to find the most likely path is the Viterbi algorithm [41]. It is used, for example, in speech recognition [42], where a sequence of events, rather than the individual symbols, contains important information. Also, MAP sequence estimation has the interesting property that, when mapping a symmetric 2×2 HMM to an Ising model, it is equivalent to minimizing the Hamiltonian $\mathcal{H}(y, x) = -\log(P(y|x)P(x))$ [43]. We will come back to the mapping of HMMs onto Ising models in Sections 2.4.4 and 5.4.

- *Kalman filtering*, for linear-Gaussian state-space models is analogous to Bayesian filtering for HMMs. A state-space model is just like a HMM, except that the state space is continuous. A linear-Gaussian state-space model is a special case where the transition and observation models are linear, and the noise is Gaussian [4]. Applications include the tracking of airplanes from noisy measurements, the control of robotic motion, and the reconstruction of images from noisy sensors in brain imaging methods [40].

Algorithms that estimate the probability to be in a particular state at a given time are never certain for finite transition and observation probabilities. Even after a long string of identical symbols, there is always a (small) probability that the state changed and we observe a wrong symbol. Figure 2.4 shows a realization of a 2×2 HMM: the hidden state, the observations and the probability $P(x_t = 1|y^t)$ resulting from the application of the Bayesian filtering algorithm. The states and symbols can take on the values '± 1' and $P(x_t = 1|y^t) = 1 - P(x_t = -1|y^t)$ in a 2×2 system. Also, in a real system, the state would be unknown.

We see that the probability levels off after a long string of identical observations; the state estimate never exceeds a certain maximum confidence level. The maximum confidence level p_i^* is the probability that the system is in state i given a long sequence of identical observations, $y^t = i$,

$$p_i^* = P(x_t = i|y^t = i). \tag{2.3}$$

For a specific HMM, we can calculate p_i^* as a function of the model's parameters. We write it out further

using Bayes' theorem and the Markov property,

$$\begin{aligned} p_i^* &= P(x_t = i | y^t = i) \\ &= P(x_t = i | y_t = i, y^{t-1} = i) \\ &= \frac{P(y_t = i | x_t = i, \cancel{y^{t-1} = i}) P(x_t = i | y^{t-1} = i)}{P(y_t = i | y^{t-1} = i)} \\ &= \frac{1}{Z_{t,i}} P(y_t = i | x_t = i) P(x_t = i | y^{t-1} = i). \end{aligned} \tag{2.4}$$

Here, $y^{t-1} = i$ is crossed out because once we know the current state x_t, the history of observations does not add any information to our knowledge of y_t. The normalization factor in the denominator is written as $Z_{t,i}$. We use similar tricks and marginalization to get an expression in terms of transition probabilities, observation probabilities and p_i^*:

$$\begin{aligned} p_i^* &= \frac{1}{Z_{t,i}} P(y_t = i | x_t = i) \sum_{x_{t-1}} P(x_t = i, x_{t-1} | y^{t-1} = i) \\ &= \frac{1}{Z_{t,i}} P(y_t = i | x_t = i) \sum_{x_{t-1}} P(x_t = i | x_{t-1}, \cancel{y^{t-1} = i}) P(x_{t-1} | y^{t-1} = i) \\ &= \frac{1}{Z_{t,i}} P(y_t = i | x_t = i) \sum_{x_{t-1}} P(x_t = i | x_{t-1}) P(x_{t-1} | y^{t-1} = i). \end{aligned} \tag{2.5}$$

The last term $P(x_{t-1}|y^{t-1} = i)$ is simply p_i^* when $x_{t-1} = i$, but can be complicated for $x_t \neq 1$ in HMMs that are not symmetric or have more states. We will come back to the maximum confidence level in Section 2.4 and in the analysis of different HMMs.

An interesting property of the filtering equations applied to a HMM is 'forgetting of the initial condition'. It is a result of the Markovian dynamics of Markov chains and hidden Markov models. Markov chains with a unique steady-state distribution go to this distribution in the long-time limit, regardless of the initial condition. The observations as a function of time are not Markovian, but they inherit some forgetting properties from the hidden chain [39, Chapt. 4.3]. Since the filter depends on the observations, it consequently inherits these forgetting properties, too. This property might break down when the filter is not optimal, i.e., when inaccurate model parameters ($\mathbf{A}$ and $\mathbf{B}$) are used to calculate the filtering probability [44]. This will not be a problem here, as we will use the filter only with exact model parameters. In short, the initial condition does not matter when averaging over sufficiently long times.

2.2.1 Bayesian filtering equations

Bayesian filtering is of interest to us for its real-time applications, such as tracking and control systems. Also, compared to MAP estimation we get quite a lot of information about our system from filtering; Through MAP estimation we can calculate the most likely path, but other paths, that are only slightly less likely than the MAP estimate, could be interesting as well.

The Bayesian filtering equations recursively calculate the probability for the system to be in a state x_t given the whole history of observations y^t [24]. It is calculated in two steps: the prediction step $P(x_{t+1}|y^t)$, and then the update step $P(x_{t+1}|y^{t+1})$. The prediction step can be worked out using marginalization,

the definition of conditional probability, and the Markov property,

$$\begin{aligned} P(x_{t+1}|y^t) &= \sum_{x_t} P(x_{t+1}, x_t|y^t) \\ &= \sum_{x_t} P(x_{t+1}|x_t, \cancel{y^t}) P(x_t|y^t) \\ &= \sum_{x_t} P(x_{t+1}|x_t) P(x_t|y^t). \end{aligned} \tag{2.6}$$

The transition matrix and the previous filter estimate are necessary to predict the next state. To write out the update step, we use the same tricks as we used to write out p_i^* in Equation 2.4,

$$P(x_{t+1}|y^{t+1}) = \frac{1}{Z_{t+1}} P(y_{t+1}|x_{t+1}) P(x_{t+1}|y^t). \tag{2.7}$$

Here, we have again written the normalization factor as Z_{t+1}. It is the same factor as before; the symbols y_{t+1} and y^t are just not specified explicitly. The normalization factor can be interpreted as a kind of partition function, since it is a sum over all possible states. We need the observation matrix and the prediction step to update the probability.

The complete set of Bayesian filtering equations is given by

$$P(x_{t+1}|y^t) = \sum_{x_t} P(x_{t+1}|x_t) P(x_t|y^t) \tag{2.8a}$$

$$P(x_{t+1}|y^{t+1}) = \frac{1}{Z_{t+1}} P(y_{t+1}|x_{t+1}) P(x_{t+1}|y^t), \tag{2.8b}$$

with the normalization factor

$$\begin{aligned} Z_{t+1} &= P(y_{t+1}|y^t) \\ &= \sum_{x_{t+1}} P(y_{t+1}|x_{t+1}) P(x_{t+1}|y^t). \end{aligned} \tag{2.9}$$

The probability can be calculated recursively from the transition matrix, the observation matrix and the initial condition $P(x_1|y^0) = P(x_1)$. We use the steady-state distribution of the transition matrix as the initial condition. Equation 2.8a is the discrete version of the Chapman-Kolmogorov equation [40, Chapt. 4.2], which describes the evolution of continuous-time Markov chains.

2.3 Comparing strategies

In the previous section, we introduced an algorithm that uses a memory of observations to, hopefully, improve on the naive observations of a HMM. In this thesis, we focus on *comparing* the strategies of keeping and not keeping a memory. In particular, we will ask when keeping a memory of past observations is useful. We will introduce a parameter that quantifies how similar state estimates of two strategies are.

We start by defining our state estimates In essence, we reduce the information $P(x_t)$ to a single number. This kind of condensation is useful in, for example, feedback applications, where one applies a control signal u_t to the system as a function of the (estimated) state. The estimated state based on the filtering algorithm, we define as

$$\hat{x}_t^f \equiv \arg\max_{x_t}(P(x_t|y^t)). \tag{2.10}$$

It is the most likely state, given the history of observations, at a time t. Similarly, we define the state estimate based on the current (naive) observations,

$$\hat{x}_t^o \equiv \arg\max_{x_t}(P(x_t|y_t)). \tag{2.11}$$

Now, to compare the information in sequences of state estimates, there are several measures one could consider. Some options are: the Kullback-Leibler divergence, relative entropy, or mutual information [45]. The discord order parameter is particularly useful to see when the different strategies give different results, it is defined as

$$D \equiv 1 - \langle \hat{x}_t^o \hat{x}_t^f \rangle, \tag{2.12}$$

for a 2×2 HMM, where $\langle \dots \rangle$ indicates an ensemble average. The state estimates can take on all values in the state space of the HMM: '1' and '-1'. The discord parameter is zero when the state estimates agree at all times. In such a case, there is no value in keeping a memory of observations. When the state estimates differ, $D > 0$ and there is value in keeping a memory. At an observation probability $b = 0.5$, the observations are uncorrelated to the actual state, and we find $D = 1$.

Note that the discord, as defined here, compares the strategies of keeping a memory and not keeping a memory. It does not necessarily reflect how similar a state estimate and the hidden state are. However, one can calculate the discord between the filter $\hat{x}_t^f$ and the actual hidden state x_t, or the naive observations $\hat{x}_t^0$ and the hidden state x_t. The latter can even be calculated analytically, since we know $\mathbf{B}$, which relates x_t and y_t.

2.4 Symmetric two-state two-symbol hidden Markov models

Now, we take a look at symmetric two-state, two-symbol hidden Markov models. The transition and observation probabilities are the same for each state, reducing the number of independent parameters, and simplifying the probability matrices. The transition and observation matrix are now doubly stochastic matrices, given by

$$\mathbf{A} = \begin{pmatrix} 1-a & a \\ a & 1-a \end{pmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{pmatrix} 1-b & b \\ b & 1-b \end{pmatrix}, \tag{2.13}$$

respectively. The transition matrix depends on a, the probability of transitioning from one state to another, and the observation matrix depends on b, the probability of observing a symbol i given that the system is in some other state j. To ensure proper normalization of the probabilities, $0 \leq a, b \leq 1$. In practice, we will consider $0 \leq a, b \leq \frac{1}{2}$, for reasons that will become clear in following sections. The two parameters a and b govern the dynamics of the hidden Markov model. In this section, we will consider a HMM that can be in hidden states $x_t = \pm 1$ and the emit symbols $y_t = \pm 1$.

The steady-state probability of the hidden state $\boldsymbol{p}_x$ can be found from $\mathbf{A}\,\boldsymbol{p}_x = \boldsymbol{p}_x$, where $\boldsymbol{p}_x$ is a vector with elements $P(x = i)$. The steady state of the emitted symbols $\boldsymbol{p}_y$ can be found using marginalization, the definition of conditional probability, and the steady state of the hidden state,

$$P(y) = \sum_x P(x, y) = \sum_x P(y|x)P(x). \tag{2.14}$$

We find

$$\boldsymbol{p}_x = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{p}_y = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \tag{2.15}$$

for any symmetric 2×2 HMMs.

The theory in this section is largely based on [24]; the calculations and simulations were repeated. We will calculate and discuss the maximum confidence level for a symmetric 2×2 HMM in the following section. Then, the behaviour of the discord parameter for this system will be discussed in Sections 2.4.2 and 2.4.3. And finally, we introduce another way of looking at this HMM, by considering a mapping to an Ising model in Section 2.4.4.

2.4.1 Maximum confidence level

The states of a symmetric HMM are all equivalent; there is no way we can distinguish between them based on the dynamics of the system. As a result, the maximum confidence levels p_i^* are all the same. And,

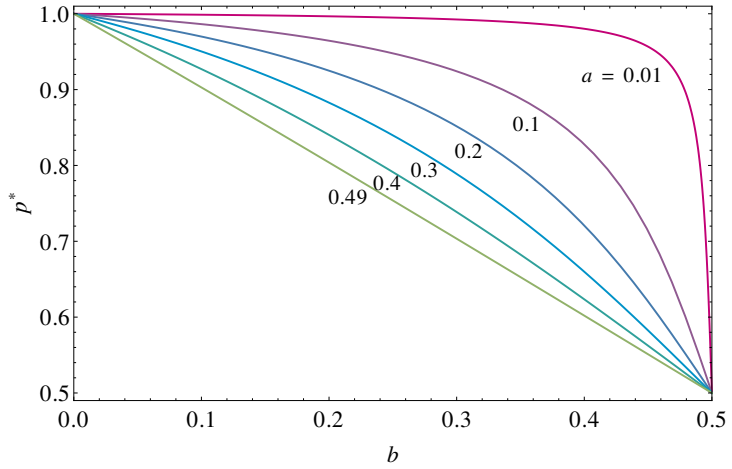

Figure 2.5: Maximum confidence level for a 2×2 symmetric HMM, for several transition probabilities a, as a function of the observation error probability b.

hence, we need to calculate only one maximum confidence level p^*. We start from the expression for p_i^* in Equation 2.5. For a 2×2 HMM, the term $P(x_{t-1}|y^{t-1})$ reduces to either p^* or $1 - p^*$, depending on the combination of states and symbols. Using this, together with the transition and observation probabilities, we find

$$p^* = \frac{(1-b)\left[(1-a)p^* + a(1-p^*)\right]}{(1-b)\left[(1-a)p^* + a(1-p^*)\right] + b\left[ap^* + (1-a)(1-p^*)\right]}. \tag{2.16}$$

Solving for p^*, we get

$$p^* = \frac{1 - 2b + a(4b-3) + \sqrt{a^2 - 2a(1-2b)^2 + (1-2b)^2}}{2(2a-1)(2b-1)}. \tag{2.17}$$

This result is plotted in Figure 2.5, for several values of the transition probability a, as a function of the symbol error probability b. The filter becomes less confident for higher symbol error probabilities and for higher transition probabilities. Intuitively, this makes sense, since the filter uses the transition and observation probabilities as input. When there are more errors, the filter expects this and it is less confident. When the state transitions are more frequent, it is more likely that there are transitions close together rather than just some wrongly emitted symbols, and the filter has a hard time distinguishing between the two.

Perfect information $b = 0$ corresponds to all emitted symbols' being the same as the underlying states; the maximum confidence level equals one here, regardless of the transition rate. The observations and the filter agree at each time, because the filter knows the observation error probability. When $b = 0.5$, $p^* = 0.5$ for every transition probability a. At this point, the emitted symbol is wrong 50% of the time: The observation is no better than a coin flip. The filter cannot decide between the two states and is 'stuck' at a probability of 0.5. For low transition rates, the maximum confidence is slowly decreasing until, at high b, the curve drops rapidly to 0.5. The transitions are rare, and only a sufficiently high error rate can make the filter significantly less certain. For a transition rate of 0.5 on the other hand, the decrease in the maximum confidence is constant. At each time step, a transition is just as likely as no transition, and p^* is a linear function of b.

2.4.2 Discord parameter

Now we compare the strategies of state estimation through filtering and naive observations by looking at the discord parameter, which we defined in Equation 2.12. The discord for symmetric 2×2 HMMs,

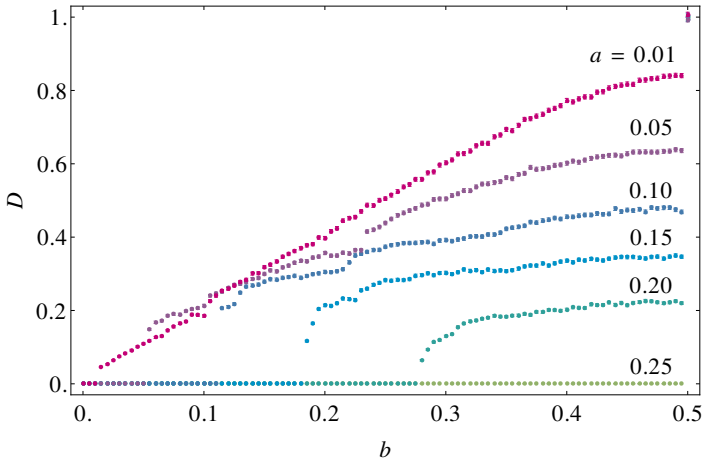

Figure 2.6: Discord parameter for 2×2 symmetric HMMs as a function of the symbol error probability b, for several transition probabilities a. Each point is an average over 30,000 timesteps; the error bars show the standard deviation of the average.

found from simulations, is shown in Figure 2.6. The simulations will be discussed in Chapter 3, and the relevant code can be found in Appendix A. When the filter's state estimate perfectly matches the emitted symbols, $b = 0$, the discord is zero for all transition probabilities. The strategies lead to the same result, and there is no advantage in keeping a memory of observations. When the emitted symbols are completely uncorrelated with the system's state, $b = 0.5$, the discord is one. This is exactly as expected from the definition of the discord. We will not consider $b > 0.5$, since, we could improve our naive observations simply by switching all observed symbols $-1 \leftrightarrow 1$ and work with a observation error that is lower than 0.5.

The discord parameter increases for increasing error probability, which is to be expected. In general, the discord increases for decreasing transition probabilities and fixed error probabilities. In other words, the discord increases when the dwell time increases and the error rate is fixed. At low transition rates, long chains of identical symbols are expected, and more wrong symbols are required to 'fool' the filter. Hence, the discord is higher at lower transition probabilities. At sufficiently high a, the discord parameter is zero everywhere, except at $b = 0.5$, where it jumps to one. The filter again cannot improve upon the naive observations in this case. Note that b is the error rate in the state estimate based on naive observations $\hat{x}_t^o$ with respect to the state x_t. So, to perfectly recreate x_t with the filter's state estimate $\hat{x}_t^f$, we need $D = 2b$. However, when we find $D = 2b$, it is not guaranteed that the filter finds the original sequence of states.

There are several features of the discord's behaviour that are puzzling at a first glance. For instance, the discord does not increase monotonously or even continuously everywhere as a function of b. Can we understand why it behaves this way? More specifically, some questions that we will address are

- At what observation probability b does the discord become non-zero?
- When is this transition to non-zero discord (dis)continuous?
- What causes the small jumps and kinks in the curve at intermediate values of b?
- Why is there a jump at $b = 0.5$ to $D = 1$? Can we calculate the discord's limit as b approaches 0.5?

We will answer the first question in the following section; the remaining questions will be addressed in Chapter 4. Also, a mapping of the 2×2 HMM onto the Ising model is considered in Sections 2.4.4 and 5.4 to investigate the transitions in the discord parameter at various observation probabilities.

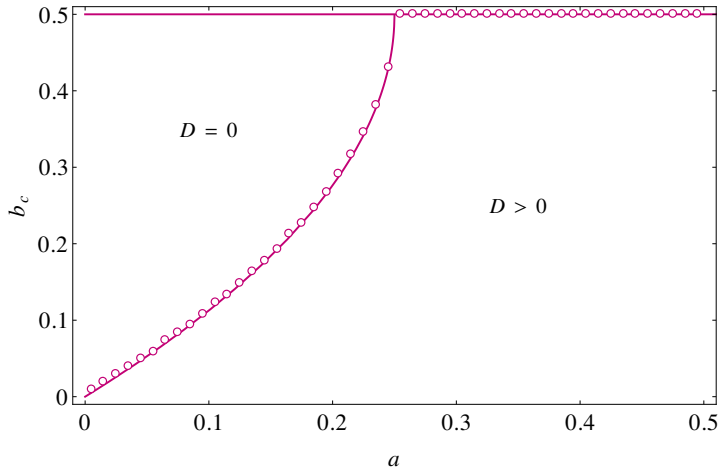

Figure 2.7: Phase diagram of a symmetric 2×2 HMM. The critical symbol error probability b_c is shown as a function of the transition probability a. The lines are the analytical solution from Equation 2.22 and the circles are results found in simulations. The critical error probability separates the 'phases' $D = 0$ and $D > 0$.

2.4.3 Critical observation probability

We want to calculate the lowest observation error probability at which the state estimates from observations $\hat{x}_t^o$ and the filtering algorithm $\hat{x}_t^f$ are not the same, $D > 0$. We call this value of b the critical symbol error probability b_c. The condition defining the critical error probability for a 2×2 HMM is

$$P(x_{t+1} = 1 | y_{t+1} = -1, y^t = 1) = \frac{1}{2}. \tag{2.18}$$

In words, given a long string of identical observations followed by a single different observation, When is the probability that the last observation is (in)correct equal to $\frac{1}{2}$? When this probability is lower than $\frac{1}{2}$, the state estimate always agrees with the observations and the discord will be zero. Hence, the transition is exactly where the probability equals $\frac{1}{2}$. And when the probability is greater than $\frac{1}{2}$, the discord is greater than zero. Note that the states 1 and -1 could be switched to find the same result because of the system's symmetry.

This condition can be solved for b_c in terms of a. We start by writing out the right-hand side as before, using Bayes's theorem, the Markov property, and marginalization:

$$\begin{aligned} P(x_{t+1} = 1 | y_{t+1} = -1, y^t = 1) &= \frac{P(y_{t+1} = -1 | x_{t+1} = 1, \cancel{y^t = 1}) P(x_{t+1} = 1 | y^t = 1)}{P(y_{t+1} = -1 | y^t = 1)} \\ &= \frac{P(y_{t+1} = -1 | x_{t+1} = 1) P(x_{t+1} = 1 | y^t = 1)}{\sum_{x_{t+1}} P(y_{t+1} = -1 | x_{t+1}) P(x_{t+1} | y^t = 1)}. \end{aligned} \tag{2.19}$$

We now recognize elements of the observation matrix and can write out the other terms further,

$$P(x_{t+1} | y^t = 1) = \sum_{x_t} P(x_{t+1} | x_t) P(x_t | y^t = 1). \tag{2.20}$$

Now, we can write the condition in terms of known variables,

$$\frac{b_c[(1-a)p^* + a(1-p^*)]}{b_c[(1-a)p^* + a(1-p^*)] + (1-b_c)[ap^* + (1-a)(1-p^*)]} = \frac{1}{2}. \tag{2.21}$$

We plug in p^* from Equation 2.17, simplify and solve for b_c

$$\frac{b_c\left(1-a-2b_c+\sqrt{a^2+(1-2a)(1-2b_c)^2}\right)}{(1-2b_c)\left(1+a-\sqrt{a^2+(1-2a)(1-2b_c)^2}\right)}=\frac{1}{2} \tag{2.22}$$

$$(1-2b_c)(b_c^2-b_c+a)=0$$

$$b_c=\frac{1}{2} \text{ and } b_c=\frac{1}{2}(1\pm\sqrt{1-4a})$$

The critical error probability is plotted in Figure 2.7, together with the results of simulations. The simulations and analytical calculations agree perfectly. The area under the curve is the phase space where the discord is zero, and above the curve, the discord is greater than zero.

2.4.4 Ising model

So far, we have looked at the discord parameter of a symmetric 2×2 HMM and its behaviour. Although the model seems simple, the behaviour of the discord is complex and not yet fully understood. Specifically, the transition to non-zero discord, and the smaller kinks and jumps at intermediate b are still unexplained. The transition to $D>0$ is continuous for some a, whereas it is discontinuous for others. The 'phase transition' is first or second order, depending on the value of the transition probability. This presence of first- and second-order transitions reminds us of phase transitions in statistical physics. The Ising model is a relatively simple model in which phase transitions can be found. As it turns out, one can map the symmetric 2×2 HMM onto a one-dimensional Ising model [43,46]. Since Ising models have been studied a great deal, we might be able to use the available literature to shed some light on the behaviour of the discord and the presence of phase transitions.

We start by defining a mapping, a change of variables, to go from a HMM to an Ising model. Then we work out the Hamiltonian of our Ising model. The transition and observation probabilities are transformed by

$$P(x_{t+1}|x_t)=\frac{\exp(Jx_{t+1}x_t)}{2\cosh(J)}, \qquad J=\frac{1}{2}\log\left(\frac{1-a}{a}\right), \tag{2.23}$$

$$P(y_t|x_t)=\frac{\exp(hy_tx_t)}{2\cosh(h)}, \qquad h=\frac{1}{2}\log\left(\frac{1-b}{b}\right).$$

Using these transformations, we can work out the Hamiltonian,

$$\mathcal{H}=-\log\left(P(x^N,y^N)\right). \tag{2.24}$$

We write out the argument of the logarithm using the definition of conditional probability and Markov property of the transition and observation probabilities,

$$\begin{aligned} P(x^N,y^N)&=P(y^N|x^N)P(x^N) \\ &=\prod_{t=1}^{N}P(y_t|x^N)P(x^N) \\ &=\prod_{t=1}^{N}P(y_t|x_t)P(x^N) \\ &=\prod_{t=1}^{N}P(y_t|x_t)\prod_{s=1}^{N-1}P(x_{s+1}|x_s)P(x_1). \end{aligned} \tag{2.25}$$

Recall that y_t is determined completely by x_t. Plugging this into Equation 2.24, we get

$$\begin{aligned}\mathcal{H} &= -\log\left(\prod_{t=1}^{N} P(y_t|x_t)\prod_{s=1}^{N-1} P(x_{s+1}|x_s)P(x_1)\right) \\ &= -\sum_{t=1}^{N}\log\left(P(y_t|x_t)\right) - \sum_{s=1}^{N-1}\log\left(P(x_{s+1}|x_s)\right) - \log\left(P(x_1)\right)\end{aligned} \tag{2.26}$$

The last term can be neglected since it is constant. The other terms can be rewritten using the change of variables introduced in Equation 2.23,

$$\begin{aligned}\mathcal{H} &= -\sum_{t=1}^{N}\log\left(P(y_t|x_t)\right) - \sum_{s=1}^{N-1}\log\left(P(x_{s+1}|x_s)\right) \\ &= -\sum_{t=1}^{N}\log\left(\frac{\exp(hy_tx_t)}{2\cosh(h)}\right) - \sum_{s=1}^{N-1}\log\left(\frac{\exp(Jx_{s+1}x_s)}{2\cosh(J)}\right) \\ &= -\sum_{t=1}^{N}\left[hy_tx_t - \log(2\cosh(h))\right] - \sum_{s=1}^{N-1}\left[Jx_{s+1}x_s - \log(2\cosh(J))\right].\end{aligned} \tag{2.27}$$

The logarithmic terms can be neglected since they are constant. In the case of periodic boundary conditions ($x_N = x_1$), both sums are over $N-1$ terms. For large N, we can approximate that the sums are both over N, neglecting boundary terms, regardless of the boundary conditions. The resulting Hamiltonian is

$$\mathcal{H}(y,x) = -J\sum_{s=1}^{N} x_{s+1}x_s - h\sum_{t=1}^{N} y_tx_t,$$

which is a one-dimensional Ising model with random local external fields. The transition probability a of the HMM determines the strength of the coupling constant J between spins in the Ising model. Transition probabilities $0 \le a \le \frac{1}{2}$, correspond to a positive coupling constant, $J \ge 0$, giving rise to ferromagnetic interactions between neighbouring spins. Likewise, the observation probability b determines the coupling between local external fields and the spins. A local, quenched field of strength hy_t tries to align its local spin along the direction of y_t. When $b = 0$, the strength of the external fields $h \to \infty$: The spins are forced to align with the local field. In context of the HMMs, the observations are the same as the hidden states when the error probability is zero. When $b = \frac{1}{2}$, the strength of the external fields $h = 0$: The external fields do not couple to the spins. In this case, the observations tell nothing about the underlying, hidden states.

We can perform another change of variables, to $z_t = y_tx_t$ and $\tau_t = y_ty_{t+1}$. We rewrite the first term using $y_t^2 = 1$,

$$x_tx_{t+1} = \frac{z_tz_{t+1}}{y_ty_{t+1}} = y_ty_{t+1}z_tz_{t+1} = \tau_tz_tz_{t+1}. \tag{2.28}$$

Rewriting the second term is straightforward, and we find

$$\mathcal{H}(\tau,z) = -J\sum_{t}\tau_tz_tz_{t+1} - h\sum_{t} z_t. \tag{2.29}$$

This Hamiltonian describes a random-bond Ising model in a uniform external field h [43]. Here, the hidden variables correspond to z_t and the observations to τ_t. The ground state of this Hamiltonian at zero temperature has an infinite number of transitions at $h = 2J/m$ for $m = 1, 2, \ldots, \infty$ [47]. For field strengths above the first transition $h > 2J$, i.e. sufficiently weak noise, the MAP estimation is the same as the observed chain [43]. As we discussed before, MAP sequence estimation is equivalent to minimizing

the system's Hamiltonian. The MAP estimate is not the same as the filter's state estimate, but perhaps a similar explanation can be found for the transitions in the discord from Figure 2.6 [24].

In this chapter, we have built up the theory and tools needed to study hidden Markov models. We then started to explore one of the simplest non-trivial HMMs: symmetric two-state, two-symbol HMMs. We compared two situations: one where the system's state is estimated based on naive observations, and one where we kept a memory of the past observations and used the filtering algorithm to find a state estimate based on the sequence of observations. We introduced a discord order parameter to compare two strategies. Despite the simplicity of the model, the behaviour of the discord parameter is not straightforward. We were able to calculate the transition from zero to non-zero discord, but this is just a piece of the puzzle. We considered a mapping to the Ising model to get more insight in this transition, which is sometimes continuous and sometimes not, and others.

We would first like to understand the behaviour of the discord parameter more completely. Therefore, we will address further aspects of the 2×2 symmetric HMM in Chapter 4. Then we will generalize the concepts introduced here and study the discord in more complicated systems. In Chapter 5, more general 2×2 HMMs will be considered. We also generalize the mapping to an Ising model such that it is valid for a 2×2 asymmetric HMM. In Chapter 6, we will look at two different symmetric $n \times n$ symmetric HMMs. The first is a straightforward generalization of the 2×2 symmetric HMM and the second is a model of a diffusing particle, which has more obvious real-world applications. The last type of HMMs that we will consider is two-state, n-symbol symmetric HMMs, also in Chapter 6. First, we will discuss the methods used in simulating and studying these HMMs in the next chapter.

Chapter 3

Methods

In this chapter, we will discuss how to simulate trajectories of hidden Markov models. All simulations were done in Matlab R2015a, and the visualization of simulations and calculations was done in Mathematica 9. In general, the code for each system consists of an initialization part and a measurement part. The first part sets up the HMM, generating the hidden state, the emitted symbols, and applying the filtering algorithm. The second part measures or calculates properties of the HMM, such as the discord parameter or the critical error probability.

3.1 n-state n-symbol hidden Markov models

The initialization of a HMM uses a simple Monte Carlo method to sample the probability distributions of the states and observations. It generates random trajectories from state to state and from state to symbol according to the model's probabilities. The probability distributions are determined by the transition and observation matrices. The only difference between the symmetric and asymmetric HMMs is in these matrices. We will number the states and symbols of an $n \times n$ HMM with the integers 1 through n and denote the number of timesteps N. We also need an initial condition, the starting state of Markov chain, to start the simulation. We set the starting state to 1 in every simulation, since a sufficiently long sequence of states 'forgets' the initial condition (Section 2.2) and we are averaging over a large number of timesteps. Before the chains can be generated, we first transform the probability matrices into cumulative probability matrices [48, ch. 16.3]. The first row still contains the probabilities to be in state 1, given the previous state. The second row contains the probabilities to be in either state 1 or 2, given the previous state. And so on, until the nth row, which should contain only 1's, because the probabilities to be in any state should add up to one, regardless of the previous state.

Now everything is set up to simulate a hidden Markov model. Starting from the initial condition and cumulative transition probability matrix, we generate the hidden chain. A random number between zero and one is drawn for each time step. The random number is compared to a column of the cumulative probability matrix, depending on the state at the previous time step (or initial condition), to determine the current state. The generation of the observations is done similarly. A random number is drawn to compare to the cumulative observation probability matrix, depending on the hidden state at the same time step, to determine the emitted symbol. The function chain() from Appendix A.1 performs all these steps and outputs sequences of the hidden states and the observations.

Then, the filtering algorithm (Equation 2.8) is applied recursively to the realized sequence of observations y^t. It needs the transition and observation probabilities as input parameters and an initial condition $P(x_1)$. The steady-state distribution of the transition matrix is used as an initial condition. The algorithm yields the probability of being in a state given the observations up to and including the current time. The function filtering(...) from Appendix A.1 calculates this probability for a sequence of observations and returns an $n \times N$ array with probabilities $P(x_t|y^t)$.

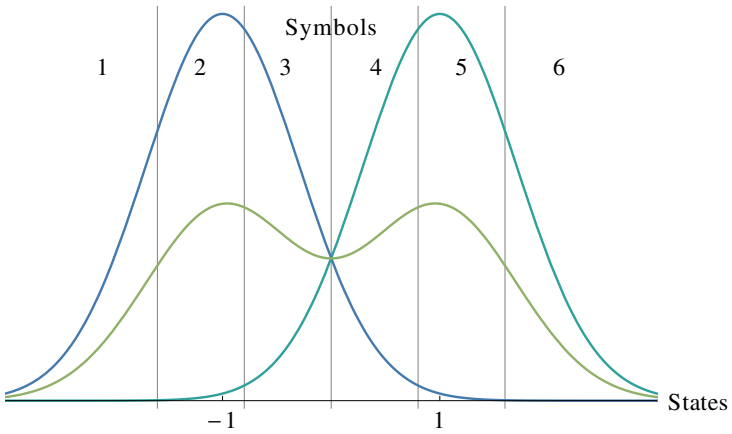

Figure 3.1: Visualization a symmetric 2×6 HMM. The errors in observing the two states 1 and -1 are Gaussian distributed (blue curves). There are six possible symbols to observe; the left three symbols correspond to state -1 and the right three correspond to state 1. The green line shows the sum of the two observation distributions, which is used to determine the bin boundaries. The bins are shown in gray (not exact).

The discord parameter, introduced in Section 2.3, can now be calculated from the observations and filter probabilities. The most likely state according to the filtering algorithm $\hat{x}_t^f$ is determined first. Next, the state estimate according to observations $\hat{x}_t^o$ and the filter's state estimate are compared. If they are the same, the contribution to the discord is $-\frac{1}{N}$; otherwise, the contribution is $+\frac{1}{N}$. We will use HMMs of length $N = 30,000$ in the discord calculations. Summing up all contributions results in a discord between zero and two. Usually, our results will be between zero and one, as $D > 1$ implies a negative correlation. For the size of the error bars, we use the standard error of the mean: the standard deviation over the square root of N. One discord curve is calculated for a fixed a and increasing b. Typically, we calculate the discord for around one hundred values of b between 0 and 0.5: $N_{\text{step}} = 100$. The function discord(...) calculates the discord and the standard deviation of the discord for one HMM (fixed $\mathbf{A}$ and $\mathbf{B}$). It is the calculation of one point in a plot of the discord (Figure 2.6).

To find the lowest observation probability for which the discord is non-zero, we can do a very naive calculation. Start at an observation probability of zero and some (fixed) transition probability, generate a trajectory of a HMM, calculate the discord and repeat for increasing b until we find a discord greater than zero. However, there is a faster way: We use our knowledge of b_c, so that we do not need search the entire plane. Instead of starting at $b = 0$ for every transition probability a, we start from $b = \max(0, b_c - 5 \cdot \text{stepsize})$, where b_c the critical error probability found for the previous a, and 'stepsize' is the size of the steps in b used in the calculation. The stepsize is taken to be $0.5/N_{\text{step}}$. For higher resolution, the increments in a and b can be made smaller. We use a threshold of $2/N$ to determine when the discord is greater than zero. This is done to rule out a non-zero discord as a consequence of the choice of the initial condition, causing non-zero discord in the first time step. The discord goes to zero in the limit of an infinitely long HMM in such a case. The code to find b_c and to make a phase diagram for an $n \times n$ HMM can be found in Appendix A.2.

3.2 Symmetric two-state n-symbol hidden Markov models

Simulating $2 \times n$ HMMs is very similar to simulating $n \times n$ HMMs. However, there are some differences in the initialization of the system and the calculation of the discord that we will discuss here. The difference in the initialization of the system is in setting up the observation matrix. Before, we changed the observation probability b to look at different systems. Now, we have to be careful to define observation

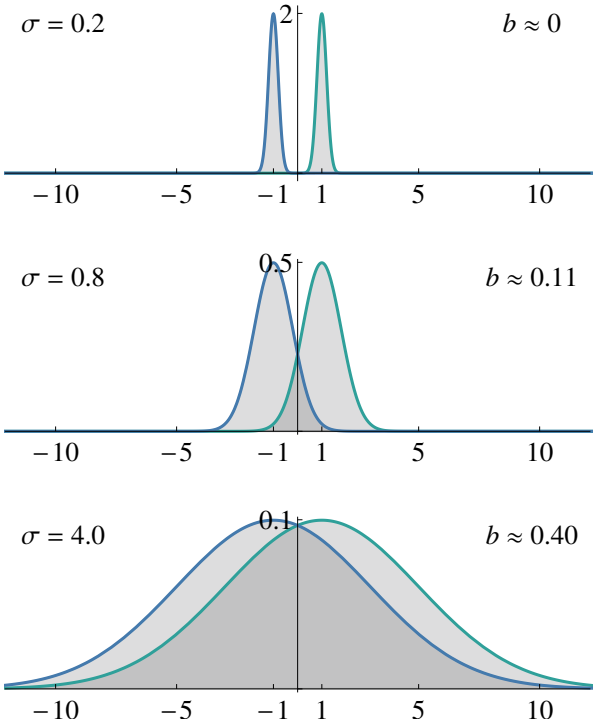

Figure 3.2: Visualization of the observation error probability in a HMM with two states and Gaussian distributed observation errors. The observation error probability increases with increasing standard deviation σ of the Gaussians.

probabilities in such a way that we can still tune a single parameter b. We assume that the observation probabilities are determined by Gaussian distribution centered on the two states -1 and 1; see Figure 3.1. The variance of the distributions sets the width of the Gaussians and, consequently, the observation probabilities of each symbol. Note that observing symbol 1, 2, or 3 corresponds to a state estimate $\hat{x}_t^o = -1$, and likewise, observing symbol 4, 5, or 6 corresponds to $\hat{x}_t^o = 1$. So, a 'correct' observation of state 1 is made if a symbol 4, 5, or 6 is observed. Simply by writing out the definition of the observation probability b for this system, we can relate it to the variance of the Gaussian distributions,

$$\begin{aligned} b &= P(y_t = 1, 2, \ldots, \text{ or } \frac{n}{2} | x_t = 1) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{0} \exp\left(\frac{-(x-1)^2}{2\sigma^2}\right) \mathrm{d}x \\ &= \frac{1}{2}\left[1 - \operatorname{erf}\left(\frac{1}{\sqrt{2}\sigma}\right)\right]. \end{aligned} \tag{3.1}$$

This is illustrated in Figure 3.2; the observation error probability increases as the Gaussians become wider.

Taking similar integrals over each Gaussian with the appropriate boundaries yields the observation probabilities for each state and symbol. We write down the probability of observing a symbol i, given

the state is 1,

$$\begin{aligned} b_{i1} &= P(y_t = i | x_t = 1) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \int_{\ell_{i-1}}^{\ell_i} \exp\left(\frac{-(x-1)^2}{2\sigma^2}\right) \mathrm{d}x \\ &= \frac{1}{2}\left[\operatorname{erf}\left(\frac{\ell_i - 1}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{\ell_{i-1} - 1}{\sqrt{2}\sigma}\right)\right]. \end{aligned} \tag{3.2}$$

Symmetry dictates that the probability of observing a symbol i, given the state is -1, equals $b_{(n-i+1)1}$. To determine the appropriate boundaries for the integrals, we consider the sum of the Gaussian distributions, the green curve in Figure 3.1. The distribution is divided into bins of equal probability of $1/n$ and we can calculate the boundaries, by solving

$$\begin{aligned} \frac{1}{2\sqrt{2\pi}\sigma} \int_{-\infty}^{\ell_i} \exp\left(\frac{-(x-1)^2}{2\sigma^2}\right) + \exp\left(\frac{-(x+1)^2}{2\sigma^2}\right) \mathrm{d}x &= \frac{i}{n} \\ \frac{1}{4}\left[2 + \operatorname{erf}\left(\frac{\ell_i - 1}{\sqrt{2}\sigma}\right) + \operatorname{erf}\left(\frac{\ell_i + 1}{\sqrt{2}\sigma}\right)\right] &= \frac{i}{n} \end{aligned} \tag{3.3}$$

for ℓ_i, where i is an integer between 0 to n. The symmetry of the problem reduces the number of equations we need to solve: We know that $\ell_0 = -\infty$, $\ell_n = \infty$, $\ell_{n/2} = 0$, and $\ell_{(n/2)-j} = -\ell_{(n/2)+j}$ for integers j between 1 and $n/2 - 1$ (all for even n).

Now that we have defined an observation matrix, the procedure for initializing the HMM and applying the filtering algorithm is the same. The only change in the calculation of the discord is that now several symbols correspond to one state. The critical observation error can be found in the same way as before with the adjusted discord calculation.

3.2.1 A continuum of observations

In the limiting case of infinitely many observable symbols (i.e. a continuum of observations), the methods need to be adjusted slightly. We no longer define a number of bins in which the observations fall and do not calculate the boundaries of these bins. The definition of the observation error probability in Equation 3.1 is unchanged. We no longer define an observation matrix of $2 \times n$ elements, but we use the probability distribution functions of the possible observations: Gaussians with mean -1 or 1 and standard deviation σ, see Figure 3.2.

The main methods of the simulations used in this thesis were discussed in this chapter. If at any point, we use a different or adjusted method it will be mentioned. The main code can be found in Appendix A and the complete code is available upon request.

Chapter 4

Symmetric two-state, two-symbol hidden Markov models

In Chapter 2, we introduced a symmetric 2×2 HMM. After looking at the discord parameter for this system, we were left with several questions. One question was answered in Section 2.4.3; the others we will address in this chapter. We will look at the continuity of the discord parameter at the transition to non-zero discord, the small kinks and jumps at intermediate error probabilities, and the transition at the observation probability where observations are uncorrelated with the underlying state.

4.1 Continuity of the discord at the critical observation probability

In Sections 2.4.2 and 2.4.3, the discord (Figure 2.6) and the associated critical-error probability (Figure 2.7) were studied of a symmetric 2×2 HMM. We found that the transition to non-zero discord looks discontinuous for some transition probabilities, because the discord discontinuity is much larger than the stepsize in b. We will determine the size of this discontinuity more accurately here. We determined the position of this transition in Figure 2.7, but the plot contains no information on the continuity of the transition. We will use our knowledge of the positions of the critical points to determine the size of the discontinuities.

For a given transition probability a, we know the critical-error probability. Suppose the critical observation error was found on the jth iteration, at b_j. We consider the interval $[b_{j-1}, b_j]$. We pick a new observation probability b_k exactly in the middle of this interval and calculate the discord $D(b_k)$. We determine which interval $[b_j, b_k]$ or $[b_k, b_{j-1}]$ has the greater increase in D. Then we pick a new observation probability in the middle of this smaller interval and repeat the process. After n iterations, the final interval in b is of length stepsize $\times\, 2^{-n}$. We run this algorithm until $\Delta D = |D(b_k) - D(b_{j/j-1})| < \frac{2}{N}$, until it has converged. This method is called the method of bisection [48, Chapt. 9]. The resulting jump size in the discord, ΔD, as a function of a is shown in Figure 4.1.

The data is collected from HMMs of length $N = 30,000$. The errors (not shown) are on the order of the size of the plotmarkers; they are the same as the error bars on the corresponding points in Figure 2.6. At $a = 0$, the discord as a function of b is a straight line, and hence there is no jump: $\Delta D = 0$. The jump size slowly increases until $a = 0.10$, and then decreases again. At these values of a, the jump in the discord is discontinuous. Around $a = 0.23$, the jump becomes very small, and it looks like a continuous transition up to $a = 0.25$. At this point, the discontinuity's size jumps to 1. Clearly, the jump in discord at $b = \frac{1}{2}$ to $D = 1$ is discontinuous, at least for $a \geq 0.25$. There are some small dips in ΔD. It is not clear what the significance of these dips is or where they come from, but they are reproducible.

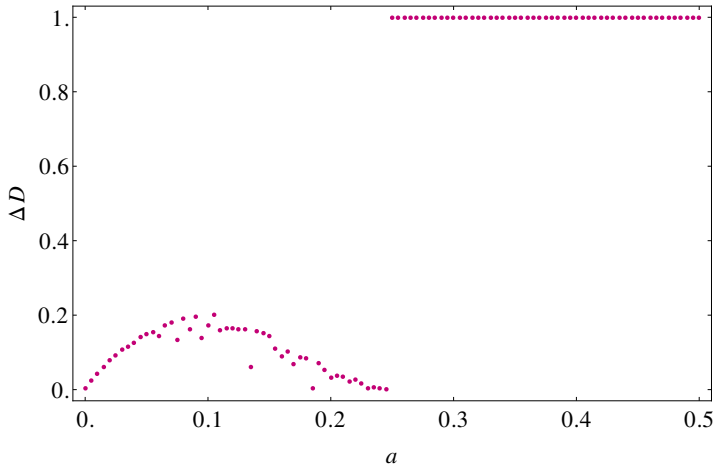

Figure 4.1: Size of the discontinuity in discord at transition to non-zero discord, as a function of the transition probability a.

4.2 Higher-order discord transitions

As we pointed out before, the discord is not a smooth function of the observation probability b. We have seen discontinuities for a range of transition and observation probabilities, not just at the critical-error probability $b = b_c$, or $b = 0.5$. We will use a calculation similar to that of the critical-error probability to explain the jumps at intermediate values of the observation error. First, we recall how b_c is calculated; then we extend the method to other jumps.

The critical-error probability, b_c, is the lowest b at which the state estimate as found from the filtering algorithm differs from the state estimate based on naive observations. The first place where the filter's state estimate will start to differ from the observations' is after long sequences of identical observations. Consider, for example $y^t = \{1, 1, \ldots, 1\}$, the probability to be in state 1 at time t reaches the maximum confidence level p^*, and the high probability of observing $x_t = 1$ given y_t is the most difficult place to get agreement between $\hat{x}_t^o$ and $\hat{x}_t^f$. For sufficiently low b, an observation $y_{t+1} = -1$ will lower the filter's probability $P(x_t = 1|y^t)$ below 0.5. The filter thinks it more likely that system has transitioned to another state, than that there was a wrongly emitted symbol; the filter agrees with the observation. Above $b = b_c$, the filter's probability will be lowered below 0.5, resulting in a non-zero discord. To summarize, after a long sequence of identical observations, the filter reaches the maximum confidence level. We then ask when a single discordant observation lowers the probability $P(x_t|y^t)$ to 0.5.

We now consider other sequences of observations, all starting with a chain of identical observations long enough that the filter's probabilities reaches the maximum confidence level. We will apply the filtering algorithm to those observations and find the b that lowers the filter's probability $P(x_t = 1|y^t)$ to 0.5. Consider the chain $y^t = \{1, 1, \ldots, 1, -1, 1, -1\}$, for example. We know that the filter's probability will be lower than p^* after the first $t - 1$ observations. It is easier for the filter to follow the observation y_t (compared to starting from a probability of p^*) in this case. A higher error rate is needed to make the filter doubt the observation and lower the probability below 0.5. One can think about it as a higher-order transition in the discord. We find that the probability $P(x_t = 1|y^t)$ is exactly 0.5 at $b = 0.104$, for a transition probability $a = 0.05$, and this particular y_t.

We repeat this procedure for several parameters and sequences of observations. We are in fact calculating where in the discord curve contributions from certain sequences of observations can be found. Sequences that are more common have a greater contribution to the discord. A jump in the discord can be caused by very common sequences or when contributions of several sequences are found close together. In order to explain the entire discord curve, one would have to 'cut' the sequences at places where the

observations y^t	$a = 0.05$	$a = 0.15$	$a = 0.20$	$a = 0.25$
$\{1, \ldots, 1, -1\}$	$b = 0.053\,(b_1)$	$b = 0.184$	$b = 0.276$	$b = 0.500$
$\{1, \ldots, 1, -1, 1, 1, -1\}$	$0.056\,(b_1)$	0.226	0.399	0.500
$\{1, \ldots, 1, -1, 1, -1\}$	$0.104\,(b_2)$	0.354	0.500	0.500
$\{1, \ldots, 1, -1, -1\}$	$0.230\,(b_3)$	0.500	0.500	0.500
$\{1, \ldots, 1, -1, -1, -1\}$	0.346	0.500	0.500	0.500

Table 4.1: Higher-order critical observation probabilities, for several transition probabilities.

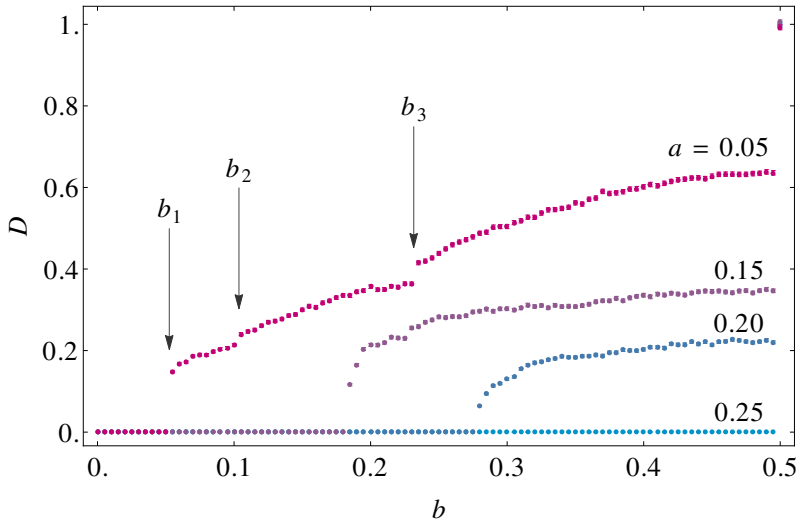

Figure 4.2: Discord parameter for a symmetric 2×2 HMM. The arrows indicate some higher-order transitions in the curve. The first four sequences in Table 4.1 correspond to these jumps.

filter probability reaches the maximum confidence level, account for all the different sequence and the frequency with which they occur. We do not have access to this information, and most sequences will have very small contributions. Therefore, we restrict our calculations to some short chains, some of which we expect to be quite common, to account for some visible jumps.

Table 4.1 shows these resulting higher-order critical error probabilities, for several transition probabilities. The discord for these transition probabilities is shown in Figure 4.2 for comparison. The arrows point to discontinuities in the discord for $a = 0.05$. Some sequences in Table 4.1 correspond to these jumps. The first jump, labeled b_1, corresponds to the position of the critical observation error of the first two sequences in the table. The position b of the first sequence is also the critical-error probability b_c that we have seen before. The second sequence has some additional observations, $\{1, 1, -1\}$, compared to the first. At low transition probabilities, the additional two 1 observation increase the filter's probability to almost p^*, and hence the contribution is close to b_c for $a = 0.05$. We see similar behaviour for the third sequence (corresponds to jump b_2) The only difference with the previous sequence is one less 1 between the two -1 observations. The contribution is found at higher b, since the one observation 1 does not raise the filter's probability as much as the two 1 observations. Also, at lower transition probabilities, the filter expects longer chains of identical observations. In which case, the second sequence is more likely to occur than the third, and the contribution to the discord of the third sequence should less. We do see a smaller jump at b_2 than at b_1, but it is not possible to separate the contributions of all sequences, and therefore we cannot directly compare the size of the contributions of different sequences. Another way of looking at these calculations is asking: How high should the error probability be in order for the filter to think

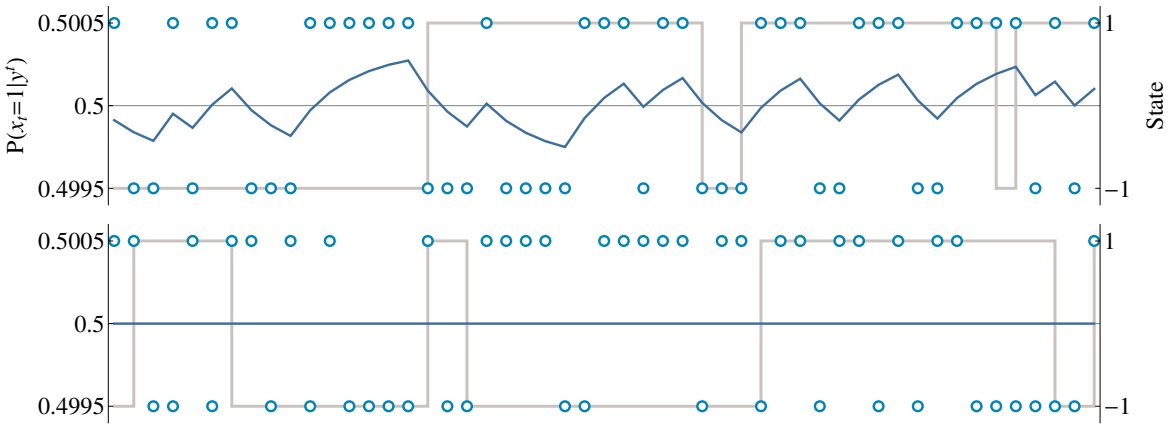

Figure 4.3: Realizations of two HMMs with $a = 0.15$ to illustrate the filter's behaviour as $b \to \frac{1}{2}$. When $b < \frac{1}{2}$, the filter's probability can change from one timestep to the next. However, at $b = \frac{1}{2}$ the filter's probabilities are constant everywhere.

that the last observation in the sequence is a wrong observation rather than a transition? Evidently, the fourth sequence, $y^t = \{1, \ldots, 1, -1, -1\}$, requires a higher b than the first sequence to make the filter think this. The contribution is found at the arrow labeled b_3. Similarly, the fifth sequence, which has an additional -1 observation compared to the previous one, requires an even higher b to make it likely that the last -1 is a wrong observation. Indeed, we find $b = 0.346$, but we cannot identify a jump at this point. The sequence is not likely enough te create a discontinuity. As we consider the contributions of longer chains, the probability of encountering a specific sequence decreases; there are simply more possible sequences to consider. Only taking into account the length of the chains, the second, third and last sequence from Table 4.1 should contribute less than the first and fourth sequence. Based on this very crude estimate, we would expect the jumps at b_1 and b_3 to be the largest.

The discord at the higher transition probabilities looks smoother than the $a = 0.05$ curve. This can be explained partly by the contributions of the sequences that shift towards higher b for increasing a. Once they reach $b = 0.5$, they stay there. This might explain why the jump at $b = 0.5$ is larger for the highest transition probabilities; the contributions of more and more sequences are located here. In the curve for $a = 0.15$, we again recognize a jump at $b = b_c$ and a small jump one around $a = 0.226$. These correspond to the first two sequences from the table. Another cause of smoother discord at higher a might be that we need longer (and less probable) sequences to accurately describe the curves. When the HMM has a higher transition rate, there will be fewer and shorter sequences of identical observations. As a result, the maximum confidence level will be reached less often, and thus we need longer sequences to describe the discord.

We have made plausible that some common sequences of observations can explain the higher-order transitions in the discord curve. We can calculate the position of the contributions of these sequences in the discord curve but, unfortunately, we are unable to calculate the size of these contributions. The smoother discord for higher transition probabilities and the increasing size of the discontinuity at $b = 0.5$, as a function of a can both be explained in this framework.

4.3 Transition to uncorrelated observations

In this section, we investigate the transition in the discord parameter at $b = \frac{1}{2}$, where D jumps to 1. We have looked at this transition for several values of a in Figures 2.6 and 4.2. Also, in Section 4.1, we have seen that the transition is discontinuous, for at least $a \geq 0.25$. The transitions at lower a look discontinuous, too, because there are no jumps to intermediate values of D between the seemingly

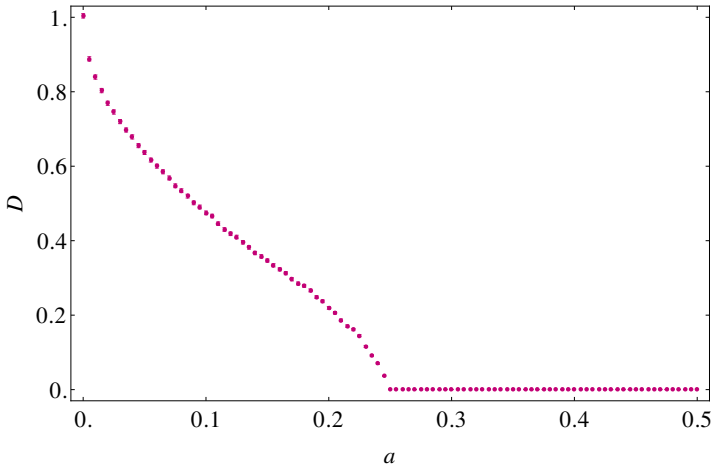

Figure 4.4: Discord parameter for a symmetric 2×2 HMM as a function of the transition probability a, in the limit of $b \to \frac{1}{2}$: Observations uncorrelated with state ($b = \frac{1}{2}$), however the filtering algorithm uses the parameter $b = 0.4999$.

continuous curves, as b approaches 0.5, and the points at $D = 1$.

The value of D at $b = \frac{1}{2}$ can be explained by the behaviour of the maximum confidence level and its effect on the filtering algorithm. The state space that the filter can explore is limited to the interval $[1 - p^*, p^*]$ for a symmetric 2×2 HMM. When the observations are not correlated with the underlying state, the maximum confidence level is 0.5, regardless of the transition probability and, hence, the state space is reduced to a single point; $P(x_t|y^t) = 0.5$ for all y^t, x_t, and t. Consequently, the filter's state estimate $\hat{x}_t^f$ is the same for all times and all possible realizations of the HMM. Recall that the observations are still 1 or -1 with a 50% chance. So, half the time the filter and the observations agree, and half the time they disagree. The resulting discord is exactly 1, for any transition probability.

We understand why there is a transition to $D = 1$ at $b = \frac{1}{2}$. Now, can we understand why this transition is discontinuous? The state space that the filter can explore is smaller for increasing b, until it is reduced to a single point at $b = \frac{1}{2}$. The shrinking of the phase space is continuous, because p^* is a continuous function of b. For b very close to, but still below $\frac{1}{2}$, the filter's state estimate can be either state 1 or -1, and it changes with the observations that are made. The difference between HMMs at $b = \frac{1}{2}$ and a b just below $\frac{1}{2}$ is illustrated in Figure 4.3. The 'amplitude' of the filter's probability is enlarged compared to the scale of the underlying state and observations. When b is just below $\frac{1}{2}$, the filter moves up and down depending on the chain observations and the state estimate changes accordingly. Whereas, at $b = \frac{1}{2}$ these are constant.

We investigate the discord in limit of $b \to \frac{1}{2}$, but where $b = \frac{1}{2}$ is not allowed as input for the filtering algorithm. To do so, we generate a HMM with $b = \frac{1}{2}$ and apply the filtering algorithm with a slightly different b: $b = 0.4999$. The generated chain is virtually indistinguishable from one with $b = 0.4999$, but there is a clear difference in the behaviour of the filter (Figure 4.3). The discord for this limiting case is shown in Figure 4.4.

This result confirms our suspicion of a discontinuous transition, since the limiting values of D are the same as the values at $b = 0.495$ in Figures 2.6 and 4.2. The discontinuity makes it likely that the jump is a result of the decision making to go from probabilities to state estimates. The size of the jump depends on the parameters of the HMM, it varies continuously as a function of a.

In this chapter, we have looked at the behaviour of the discord parameter of a symmetric 2×2 HMM in more detail. All features of the discord of this system has been discussed now. The transition to non-zero discord has been quantified; we have calculated the critical-error probability where this transition is located and the size of the discontinuity. The transition to a discord of one is explained, and the

discontinuity and corresponding jump size are resolved by considering the limit of $b \to \frac{1}{2}$. We are still unable to calculate the discord analytically, and as a result we cannot verify the behaviour at intermediate observation probabilities to a high precision. Focussing on visible kinks and jumps in the discord curve, we have tried to make our explanation of the behaviour plausible at least. Some simple calculations support the reasoning, but this is not a guarantee.

Chapter 5

Symmetry breaking in two-state, two-symbol hidden Markov models

Now that we have a fairly good understanding of the behaviour of the discord in 2×2 symmetric HMMss, we are ready to consider more complicated systems. In this chapter, we will look at 2×2 HMMs where the symmetry of the transition and observation matrices is broken. We distinguish between three cases of asymmetric HMMs: HMMs with asymmetric transition matrices, asymmetric observation matrices, and both asymmetric transition and observation matrices. We will again look at the discord parameter; the focus will be on the differences compared to the results from Chapter 4.

In contrast with the symmetric matrices, the asymmetric transition and observation matrices have two independent parameters. We parametrize the probabilities such that only one independent parameter is left when the transition and observation matrices are symmetric. They given by

$$\mathbf{A} = \begin{pmatrix} 1 - \bar{a} + \frac{1}{2}\Delta a & \bar{a} + \frac{1}{2}\Delta a \\ \bar{a} - \frac{1}{2}\Delta a & 1 - \bar{a} - \frac{1}{2}\Delta a \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 1 - \bar{b} + \frac{1}{2}\Delta b & \bar{b} + \frac{1}{2}\Delta b \\ \bar{b} - \frac{1}{2}\Delta b & 1 - \bar{b} - \frac{1}{2}\Delta b \end{pmatrix}. \tag{5.1}$$

The transition matrix depends on the mean transition probability $\bar{a} = \frac{1}{2}(\mathbf{A}_{21} + \mathbf{A}_{12})$ and the difference in transition probabilities $\Delta a = \mathbf{A}_{21} - \mathbf{A}_{12}$. When the difference between the transition probabilities is zero ($\Delta a = 0$), the transition matrix is symmetric. Similarly, the observation matrix depends on the mean observation probability $\bar{b} = \frac{1}{2}(\mathbf{B}_{21} + \mathbf{B}_{12})$ and the difference in observation probabilities $\Delta b = \mathbf{B}_{21} - \mathbf{B}_{12}$. To ensure that each probability is at least zero and at most one, the parameters have to obey the conditions

$$\begin{aligned} 0 \leq \bar{a} \leq 1, \qquad & \max(-2\bar{a}, 2\bar{a} - 2) \leq \Delta a \leq \min(2\bar{a}, 2 - 2\bar{a}), \\ 0 \leq \bar{b} \leq 1, \qquad & \max(-2\bar{b}, 2\bar{b} - 2) \leq \Delta b \leq \min(2\bar{b}, 2 - 2\bar{b}). \end{aligned} \tag{5.2}$$

Note that Δa and Δb are allowed to be negative to cover the entire parameter space. In practise, we want to make sure that the off-diagonal elements in $\mathbf{A}$ and $\mathbf{B}$ are smaller than 0.5, just as we did in the symmetric 2×2 HMM. To illustrate, if we had an observation probability $P(y_t|x_t) > 0.5$, we would simply use $1 - P(y_t|x_t)$ to improve our state estimate. Also, we expect more accurate state estimates for transition probabilities below 0.5, because the filter needs sequences of identical symbols to get to higher confidence levels.

The steady-state probabilities of the states and observations can be calculated in the same way as we did in Section 2.4 for the symmetric matrices. The resulting steady states of the hidden states and observations are generally not symmetric,

$$\boldsymbol{p}_x = \frac{1}{2}\begin{pmatrix} 1 + \frac{\Delta a}{2\bar{a}} \\ 1 - \frac{\Delta a}{2\bar{a}} \end{pmatrix} \quad \text{and} \quad \boldsymbol{p}_y = \frac{1}{2}\begin{pmatrix} 1 + \Delta b + \frac{\Delta a}{\bar{a}}\left(\frac{1}{2} - \bar{b}\right) \\ 1 - \Delta b - \frac{\Delta a}{\bar{a}}\left(\frac{1}{2} - \bar{b}\right) \end{pmatrix}. \tag{5.3}$$

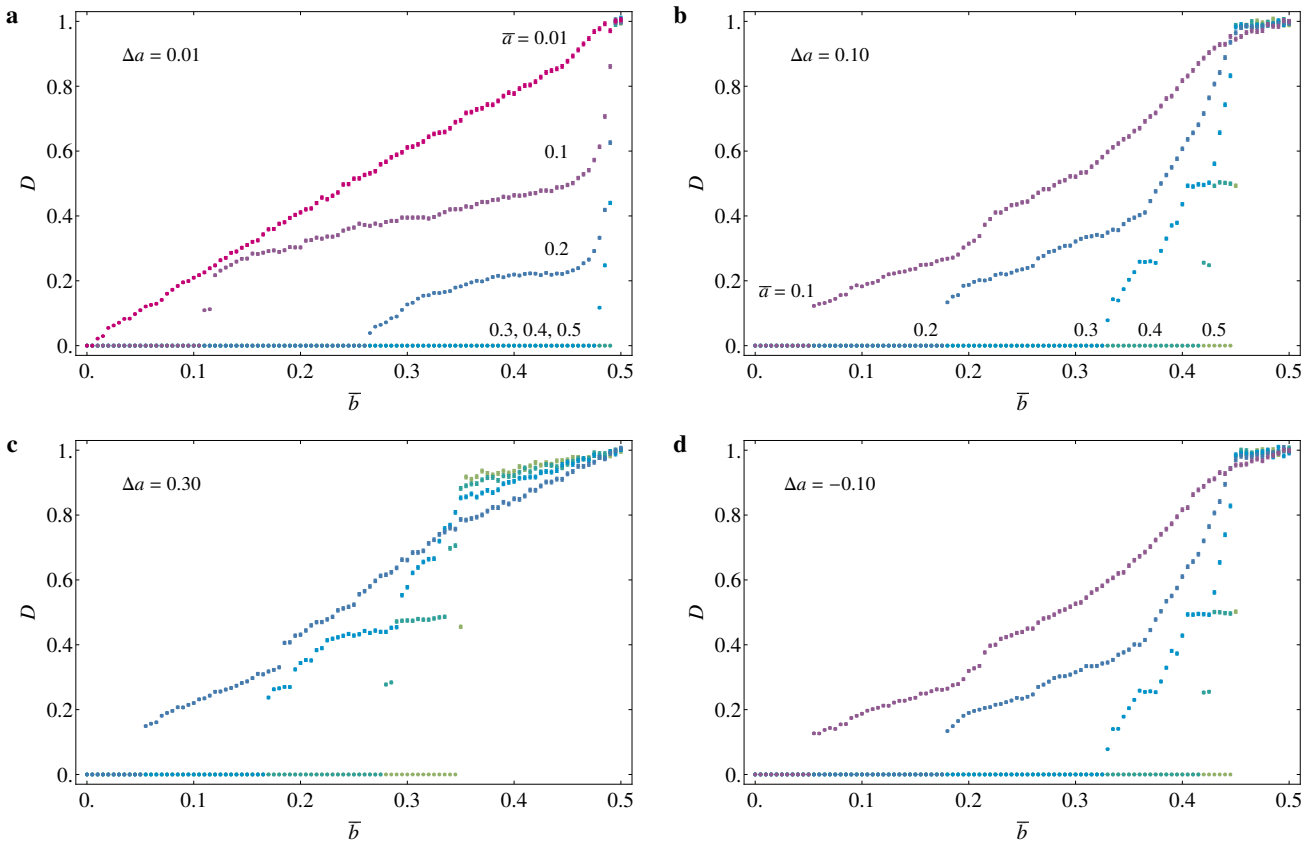

Figure 5.1: Discord parameter of HMMs with asymmetric transition matrices, as a function of the mean observation probability $\bar{b}$. Each curve is for a different mean transition probability $\bar{a}$. **a-d**, diagrams for different Δa.

5.1 Asymmetric transition probabilities

In this section, we will consider HMMs with asymmetric transition matrices and symmetric observation matrices: $\Delta a \neq 0$, $\Delta b = 0$. The discord parameter for several values of Δa is shown in Figure 5.1. More details about the simulations can be found in Section 3.1 and Appendix A.1.

Note that some curves are not visible; some are not plotted because they would result in negative transition probabilities, and some are obscured by other curves. In Figure 5.1a, a small asymmetry is introduced in the HMM: $\Delta a = 0.01$. The discord of this system is still fairly similar to those of the symmetric system (Figure 2.6). The most obvious difference is the transition to $D = 1$; it is no longer discontinuous at $b = 0.5$. Also, the position of the transition where the discord becomes non-zero and of the higher-order transitions shifted. Looking at a somewhat larger asymmetry, $\Delta a = 0.10$, in Figure 5.1b, these differences are amplified. We will not discuss the higher-order transitions. For observation probabilities approaching 0.5, the discord is slowly increasing to 1. These 'plateaus' are preceded by a continuous transition to a discord close to 1. This behaviour is even clearer in Figure 5.1c; the 'plateaus' look like linear functions of $\bar{b}$. The results in Figures 5.1b and d are the same (apart from errors due to the stochasticity of the system), because we still have symmetric observation probabilities. It does not matter for the discord which of the two states has the higher (lower) transition probability, when the observation matrix is symmetric.

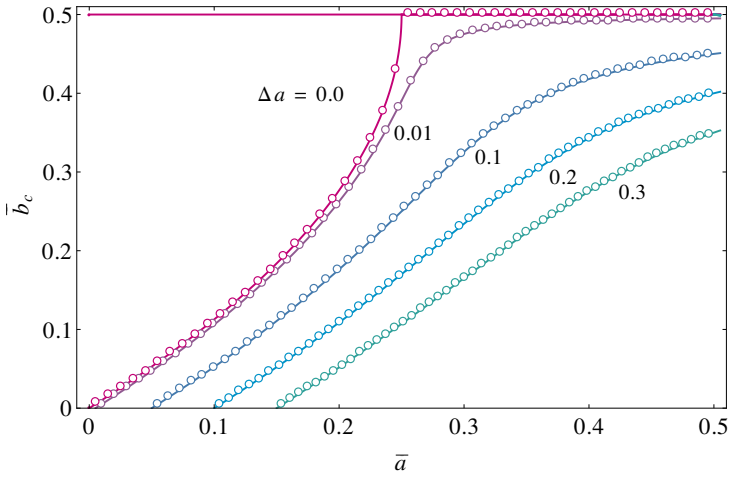

Figure 5.2: Mean critical observation probability of HMMs with asymmetric transition matrices and symmetric observation matrices ($\Delta b = 0$), as a function of $\bar{a}$. Simulated results are shown as circles and the analytical solutions are solid lines.

5.1.1 Critical observation probability

The condition for calculating the critical observation error is the same for both symmetric and asymmetric 2×2 HMMs; see Equation 2.18. The resulting equation is more complicated though, because of the extra variables Δa and Δb in the transition and observation probabilities. We solve for the mean critical error probability $\bar{b}_c$ and plot it as a function of the mean transition probability $\bar{a}$; see Figure 5.2. Appendix B.1 has more details about the calculation, it ends with the explicit equation that need to be solve to find $\bar{b}_c$. The complete solutions are omitted because they are so long and tedious to write out. The solutions agree with simulations, which are shown as circles in the same diagram. The solutions are the same when we look at 'negative' asymmetries, $\bar{b}_c(\Delta a) = \bar{b}_c(-\Delta a)$. We see that the curves for $\Delta a > 0$ never reach $\bar{b}_c = 0.5$, where we found discontinuities in the discord in the symmetric HMMs. Each curve starts at $\bar{b}_c = 0$ where $\bar{a} - \Delta a/2 = 0$, i.e. the probability of transitioning from one state is zero. The discord becomes non-zero at lower mean transition probabilities for larger asymmetries.

5.1.2 Linear discord

We can explain the linear regime in the discord by considering the maximum confidence levels of the states. In asymmetric HMMs, the maximum confidence levels of each state (p_1^* and p_{-1}^*) are generally not the same. The state space that the filter can explore is given by $1 - p_{-1}^* \leq P(x_t = 1|y^t) \leq p_1^*$. Once either $p_1^* < 0.5$ or $p_{-1}^* < 0.5$, this interval is restricted to probabilities either below or above 0.5, and consequently the state estimate $\hat{x}_t^f$ will be constant in time. To clarify, we look at the behaviour of the filter for three different values of $\bar{b}$ in Figure 5.3. In the first diagram, with $\bar{b} = 0.20$, we can see a clear difference in the maximum confidence levels of the two states. With increasing $\bar{b}$, the state space that the filter explores narrows gradually. In the lower diagram, the filter is 'stuck' in state 1: The filter's state estimate is independent of the naive state estimate (observations). This causes the linear regime in the discord diagrams, since the discord parameter now effectively measures how often the observations are in the same state as $\hat{x}_t^f$. The gradual narrowing of the allowed range of probabilities also explains the continuous transition before the linear regime. We calculate the onset of the linear regime, the $\bar{b}$ that lowers one of the maximum confidence levels to 0.5. For a general asymmetric two-state, two-symbol

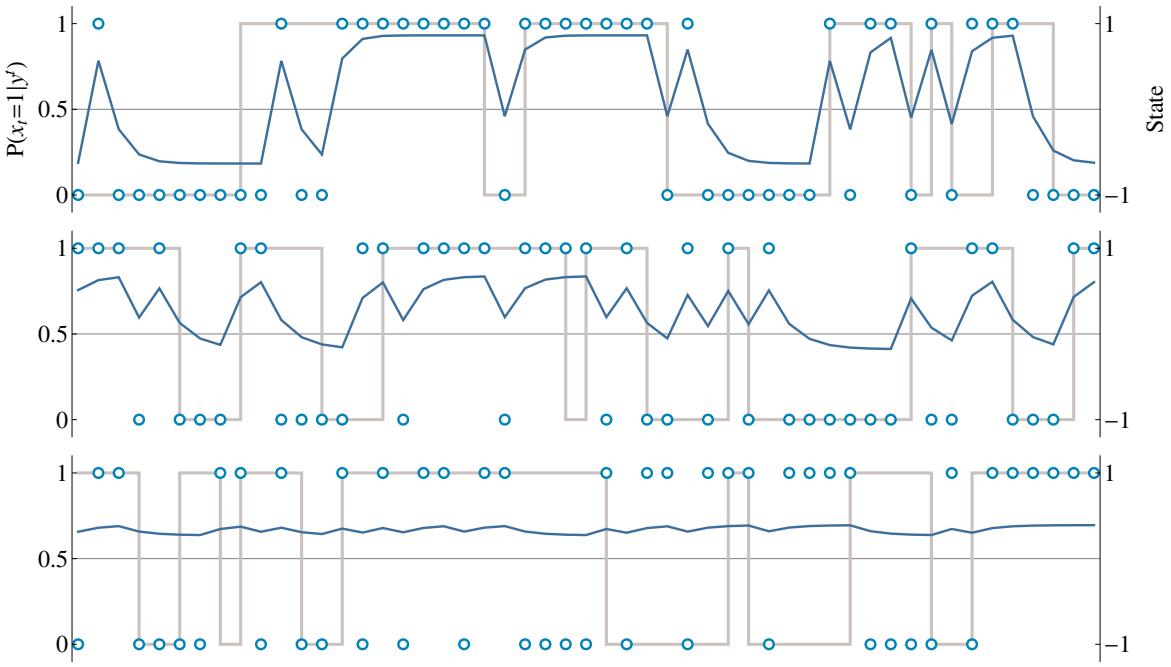

Figure 5.3: Realizations of HMMs with asymmetric transition matrices. The parameters used are $\Delta b = 0$, $\Delta a = 0.2$, $\bar{a} = 0.3$, and from top to bottom $\bar{b} = 0.20, 0.35, 0.48$. We see that the state estimate $\hat{x}_t^f$ becomes constant in the bottom diagram.

HMM, we find

$$\bar{b} = \min\left(\frac{1}{2}(1 + \Delta a \Delta b \pm \Delta a)\right). \tag{5.4}$$

This agrees with the onset of the linear regime in Figure 5.1. We can calculate the value of the discord in this regime too. We assume the filter is 'stuck' in state ± 1, and we use the steady-state probability of the observations from Equation 5.3,

$$\begin{aligned} D &= 1 - \langle \hat{x}_t^o \hat{x}_t^f \rangle \\ &= 1 - \sum_{\hat{x}_t^o, \hat{x}_t^f} P(\hat{x}_t^o, \hat{x}_t^f) \hat{x}_t^o \hat{x}_t^f \\ &= 1 - \sum_{\hat{x}_t^o} \pm P(\hat{x}_t^o) \hat{x}_t^o \\ &= 1 \mp \left(\Delta b + \frac{\Delta a}{2\bar{a}}\left(1 - 2\bar{b}\right)\right). \end{aligned} \tag{5.5}$$

We find a linear expression for the discord as a function of the mean error probability and the slope agrees with the slope seen in the discord figures. The upper and lower signs distinguish between a filter's state estimate that is stuck in state 1 and -1, respectively.

We see that the behaviour of the discord parameter changes when introducing asymmetries in the transition probabilities. We can still identify similar transitions as before, especially at small Δa, and we can explain the changes in the behaviour.

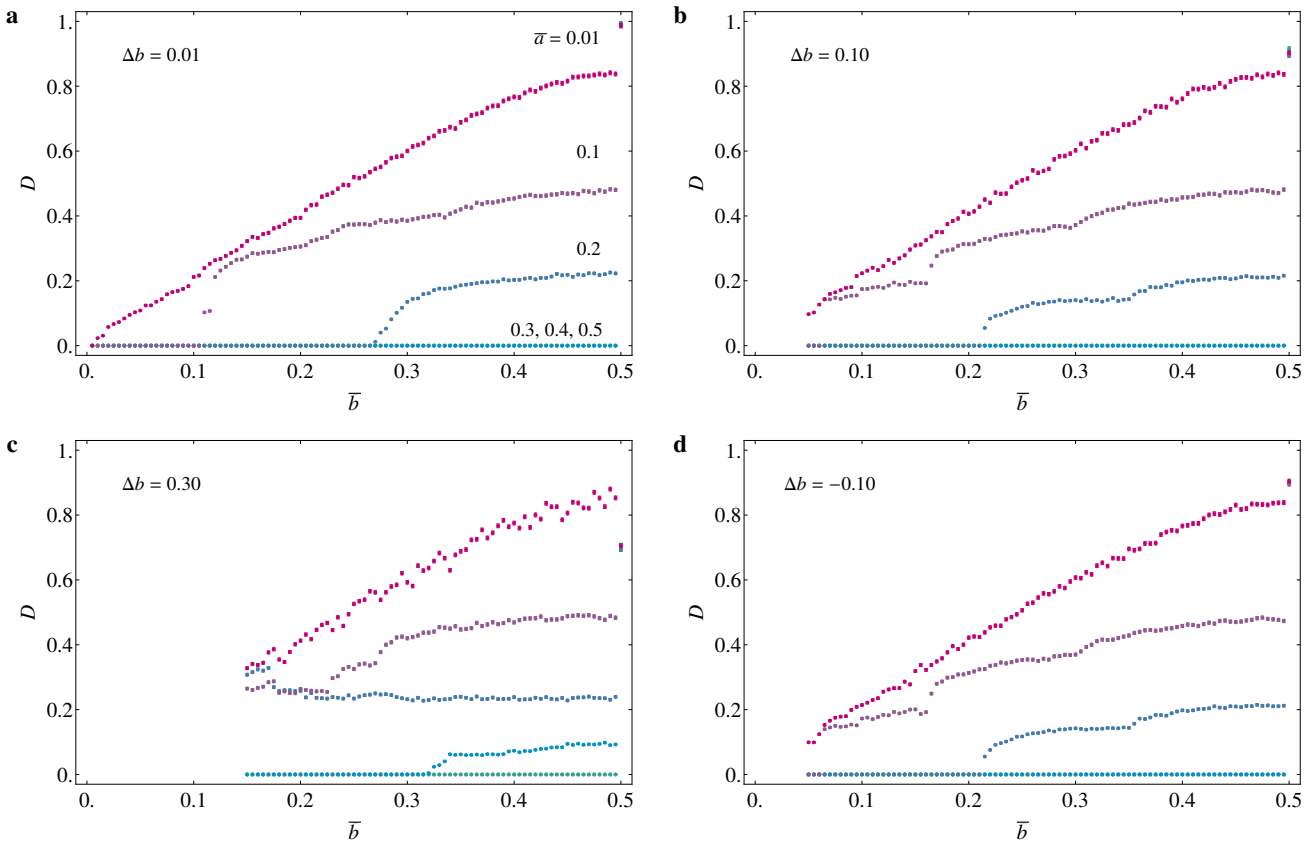

Figure 5.4: Discord parameter for HMMs with asymmetric observation matrices, as a function of the mean error probability $\bar{b}$ for several mean transition probabilities $\bar{a}$. **a-d**, diagrams for various Δb.

5.2 Asymmetric observation probabilities

In this section, we will consider 2×2 HMMs with symmetric transition matrices and asymmetric observation matrices; $\Delta a = 0$ and $\Delta b \neq 0$. We will discuss the main differences between these and the symmetric HMMs. The discord parameter plots for several Δb are shown in Figure 5.4. There are no points plotted for $\bar{b} < |\Delta b|/2$, to avoid negative observation probabilities. The results for a small asymmetry, $\Delta b = 0.01$, in Figure 5.4a are almost indistinguishable from the results of the symmetric HMMs. Naturally, for greater asymmetries, the differences are greater. The transition to non-zero discord and the higher-order transitions change. Also, the discontinuous transition at $\bar{b} = 0.5$ is no longer to $D = 1$, but to $D = 1 - \Delta b$. We find this value from the derivation in Equation 5.5, keeping in mind that $\Delta a = 0$. Since the transition probabilities are symmetric, the states are effectively interchangable, and the results in Figures 5.4b and d are the same. Interestingly, at fixed $\bar{a}$, the discord curves look 'noisier' for higher Δb. This is particularly clear when comparing the pink curves, $\Delta a = 0.01$.

Once again, we calculate the mean critical error probability as a function of the mean transition probability. The resulting phase diagram is shown in Figure 5.5, together with the results of simulations. The simulations and analytical solutions agree very well. We avoided plotting negative observation probabilities, however for some values of $\bar{a}$ the effective observation probability of one state $(\bar{b} + \Delta b/2)$ exceeds 0.5. This is a situation we would in practise avoid by considering $1 - (\bar{b} + \Delta b/2)$ to improve our naive state estimate. Curves with negative Δb would be the same as the curves with positive Δb, $\bar{b}_c(-\Delta b) = \bar{b}_c(\Delta b)$. The mean critical error probability curve moves to the right by 0.025 in $\bar{a}$ for every

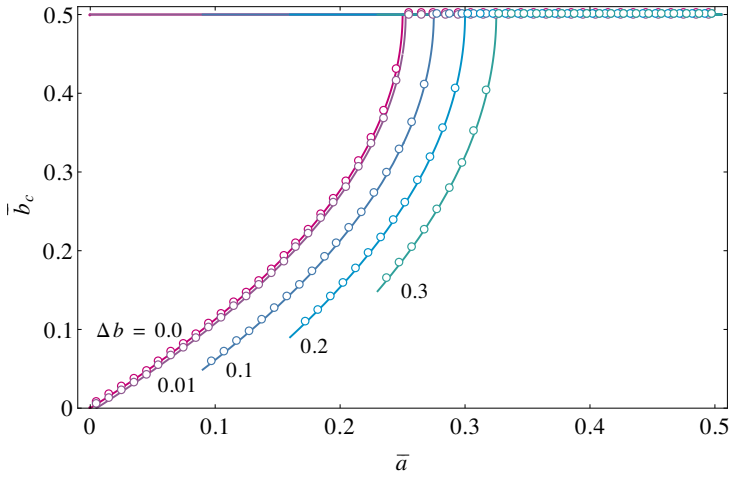

Figure 5.5: Mean critical observation probability of HMMs with symmetric transition matrices ($\Delta a = 0$) and asymmetric observation matrices, as a function of the mean transition probability $\bar{a}$. Simulated results are shown as circles and the analytical solutions are solid lines.

0.10 increase in Δb.

The changes in behaviour with asymmetric observation matrices are not as clear as with asymmetric transition matrices, but they are there. The slight changes in the transitions we can explain with similar calculations that we used before.

5.3 Asymmetric transition and observation probabilities

In this section, we will combine the asymmetric transition and asymmetric observation matrices in 2×2 HMMs, and study the behaviour of the discord parameter. The discord for some values of Δa and Δb is shown in Figure 5.6.

We recognize behaviour of the discord parameter from the previous sections. The transition at high Δb is no longer discontinuous and we see a linear regime, just like we had in Section 5.1. The linear discord is in agreement with the calculations we did in that section too. In the top two figures, the discord is most similar to that in Section 5.2, which makes sense because the asymmetry in Δa is small. The bottom four figures resemble the discord from Section 5.1 more closely. For the bottom two figures, this makes sense, because the asymmetry in the observation probability is small. However, in the middle figures, the asymmetries are of the same order, and still the behaviour of the asymmetric transition matrix dominates.

The mean critical error probability for several asymmetric HMMs is shown in Figure 5.7. The phase diagram for small Δa, Figure 5.7a, is quite different from the phase diagram for $\Delta a = 0$, Figure 5.5. At high $\bar{b}_c$, the curves bend back, which we have not seen before in any phase diagram. Studying the discord for parameters in this area we find that the discord goes back to zero at sufficiently high $\bar{b}$ and then at $\bar{b} = 0.5$ it jumps to 1. The analytical solutions and simulations agree very well once again. To find the points on the backwards bending curves, the simulations were adapted to find a transition from non-zero to zero discord instead. For slightly larger Δa, Figure 5.7b, the behaviour is similar and the backwards bending curves can be found at even lower $\bar{b}$. This behaviour is only seen in sufficiently asymmetric systems and at high $\bar{b}$; when $\bar{b} + \Delta b/2 > 0.5$. So, in practise, we would not encounter this behaviour. We do not understand this behaviour completely; Why would the state estimates become equal again at high $\bar{b}$? Looking at realizations of HMMs with various asymmetries (not shown), we notice some interesting behaviour. When the discord becomes non-zero at $\bar{b}_c$, all of the observations that the filter 'corrects' are

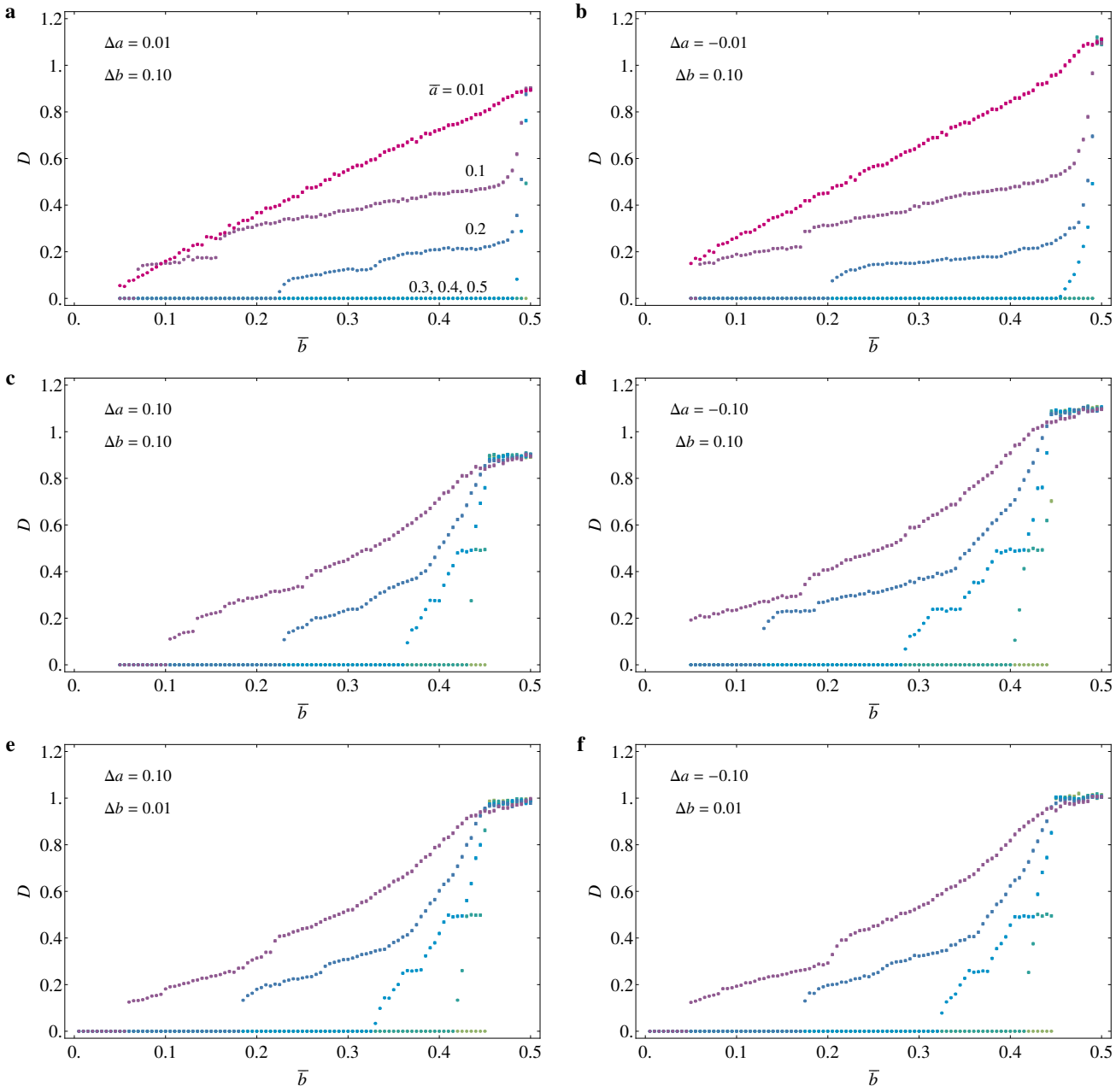

Figure 5.6: Discord parameter for HMMs with asymmetric transition and observation matrices, as a function of the mean error probability $\bar{b}$ for several mean transition probabilities $\bar{a}$.

in one particular state. For example, $\hat{x}_t^o = \hat{x}_t^f$ when $\hat{x}_t^o = 1$, but not when $\hat{x}_t^o = -1$. In these systems we are also dealing with an asymmetric transition probability, which causes the filter's state estimate to be constant above a certain mean observation error (Equation 5.4). This is reflected in the linear discord curves at high $\bar{b}$, and it requires the discord to be non-zero in this regime. The discord going back to zero before this regime could be caused by the combination of the filter expecting more errors in one state and then becoming 'stuck' in the other state. This happens while the filter's state space shrinks continuously. Maybe, at some point, the maximum confidence level of the state where errors are being

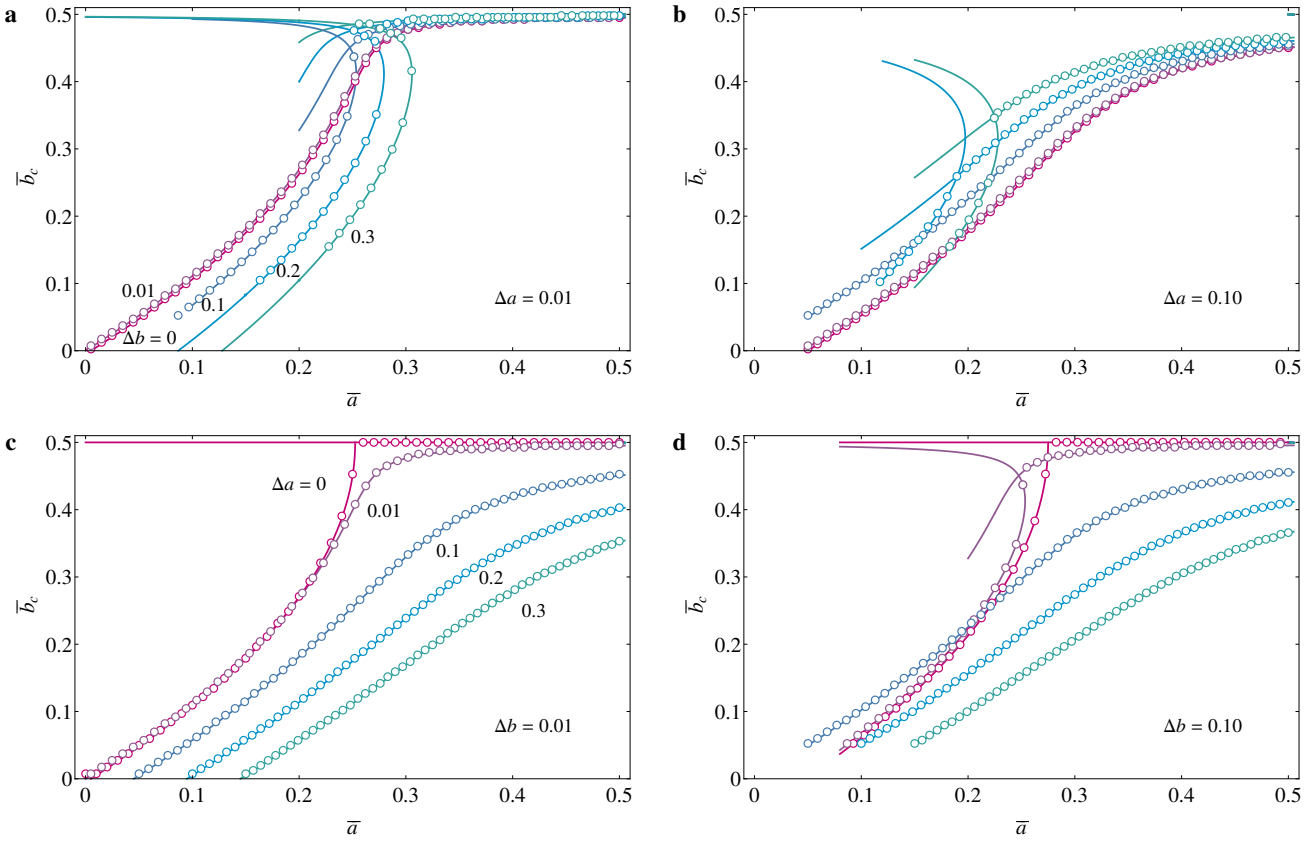

Figure 5.7: Phase diagram of HMMs with asymmetric transition and observation matrices.

corrected gets so close to 0.5, that the filter's state estimate just follows the observations again. Then, if we continue to look at higher $\overline{b}$, we will enter the linear discord regime. Note also, that the backwards bending behaviour in the analytical solutions is a result of two solutions of $\overline{b}_c$ intersecting.

Figures 5.7c and d show the phase diagrams for small asymmetries in the observation probabilities: $\Delta b = 0.01$ and 0.1, respectively. These diagrams are quite similar to phase diagram for symmetric observation probabilities; see Figure 5.2. The similarity is expected, since the discord parameter itself did not change much for small asymmetries in the observation matrix.

5.4 Mapping to Ising models

In Section 2.4.4, we examined a mapping from a symmetric 2×2 HMM to an Ising model as a way to understand the 'phase transitions' in the discord parameter. Now, we generalize the mapping so that it is valid for asymmetric 2×2 HMMs, as well. Since the derivation of the appropriate Hamiltonian is similar to the derivation discussed before, we will skip some algebraic steps. We start by defining the mapping from the transition and observation probabilities to the spin-spin coupling and the spin-field

coupling constants,

$$P(x_{t+1}|x_t) = \frac{\exp(J(x_t)x_{t+1}x_t)}{2\cosh(J(x_t))},\ J(x_t) = \begin{cases} J_+ = \frac{1}{2}\log\left(\frac{1-\bar{a}+\Delta a/2}{\bar{a}-\Delta a/2}\right), & \text{if } x_t = 1 \\ J_- = \frac{1}{2}\log\left(\frac{1-\bar{a}-\Delta a/2}{\bar{a}+\Delta a/2}\right), & \text{if } x_t = -1 \end{cases} \tag{5.6}$$

$$P(y_t|x_t) = \frac{\exp(h(x_t)y_tx_t)}{2\cosh(h(x_t))},\ h(x_t) = \begin{cases} h_+ = \frac{1}{2}\log\left(\frac{1-\bar{b}+\Delta b/2}{\bar{b}-\Delta b/2}\right), & \text{if } x_t = 1 \\ h_- = \frac{1}{2}\log\left(\frac{1-\bar{b}-\Delta b/2}{\bar{b}+\Delta b/2}\right), & \text{if } x_t = -1 \end{cases}.$$

Now, we work out the Hamiltonian $\mathcal{H} = -\log(P(x^N, y^N))$. We start from Equation 2.27 and apply the mapping from Equation 5.6,

$$\begin{aligned} \mathcal{H} &= -\sum_{t=1}^{N}\log\left(P(y_t|x_t)\right) - \sum_{s=1}^{N-1}\log\left(P(x_{s+1}|x_s)\right) \\ &= -\sum_{t=1}^{N}\log\left(\frac{\exp(h(x_t)y_tx_t)}{2\cosh(h(x_t))}\right) - \sum_{s=1}^{N-1}\log\left(\frac{\exp(J(x_s)x_{s+1}x_s)}{2\cosh(J(x_s))}\right) \\ &= -\sum_{t=1}^{N}\left[h(x_t)y_tx_t - \log(2\cosh(h(x_t)))\right] - \sum_{s=1}^{N-1}\left[J(x_s)x_{s+1}x_s - \log(2\cosh(J(x_s)))\right]. \end{aligned} \tag{5.7}$$

Next, we rewrite $h(x_t)$ and $J(x_t)$ in a convenient way:

$$\begin{aligned} h(x_t) &= \bar{h} + \Delta h x_t, \qquad \text{where } \bar{h} = \frac{1}{2}(h_+ + h_-) \text{ and } \Delta h = \frac{1}{2}(h_+ - h_-), \\ J(x_t) &= \bar{J} + \Delta J x_t, \qquad \text{where } \bar{J} = \frac{1}{2}(J_+ + J_-) \text{ and } \Delta J = \frac{1}{2}(J_+ - J_-), \end{aligned} \tag{5.8}$$

When Δa is zero, we have $J_+ = J_-$ and consequently, $\Delta J = 0$ and $\bar{J} = J$, where J is the coupling constant we found in the derivation of the mapping from of symmetric 2×2 HMMs. The same happens with the h-terms when $\Delta b = 0$. The terms consisting of a logarithm with a hyperbolic cosine are also rewritten by taking the mean value of the possible terms and a deviation from that mean value. We also use that $x_t^2 = 1$ for each state and we get,

$$\begin{aligned} \mathcal{H} &= -\sum_{t=1}^{N}\left[\bar{h}y_tx_t + \Delta h y_t - \log(2\cosh(h(x_t)))\right] - \sum_{s=1}^{N-1}\left[\bar{J}x_{s+1}x_s + \Delta J x_{s+1} - \log(2\cosh(J(x_s)))\right] \\ &= -\sum_{t=1}^{N}\left[\bar{h}y_tx_t + \Delta h y_t - \frac{1}{2}\log(4\cosh(h_+)\cosh(h_-)) - \frac{1}{2}x_t\log\left(\frac{\cosh(h_+)}{\cosh(h_-)}\right)\right] \\ &\quad -\sum_{s=1}^{N-1}\left[\bar{J}x_{s+1}x_s + \Delta J x_{s+1} - \frac{1}{2}\log(4\cosh(J_+)\cosh(J_-)) - \frac{1}{2}x_s\log\left(\frac{\cosh(J_+)}{\cosh(J_-)}\right)\right]. \end{aligned} \tag{5.9}$$

The terms without any x_t or y_t can be neglected since they provide a constant shift in the energy. The terms that depend on a factor y_t only, also cause a shift in energy and can be neglected. Higher-order

terms that depend on a product of these factors still contribute. The full Hamiltonian is then given by

$$\mathcal{H} = -\sum_{t=1}^{N}\left[\bar{h}y_t x_t + \Delta h y_t - \frac{1}{2}x_t \log\left(\frac{\cosh(h_+)}{\cosh(h_-)}\right)\right] - \sum_{s=1}^{N-1}\left[\bar{J}x_{s+1}x_s + \Delta J x_{s+1} - \frac{1}{2}x_s \log\left(\frac{\cosh(J_+)}{\cosh(J_-)}\right)\right] \tag{5.10}$$

$$= -\sum_{t=1}^{N}\left[\bar{h}y_t x_t - \frac{1}{2}x_t \log\left(\frac{\cosh(\bar{h}+\Delta h)}{\cosh(\bar{h}-\Delta h)}\right)\right] - \sum_{s=1}^{N-1}\left[\bar{J}x_{s+1}x_s + \Delta J x_{s+1} - \frac{1}{2}x_s \log\left(\frac{\cosh(\bar{J}+\Delta J)}{\cosh(\bar{J}-\Delta J)}\right)\right].$$

For large N, we can approximate that the sums are both over N, neglecting boundary terms. In this approximation, we can write $\Delta J x_{s+1} = \Delta J x_s$, because we are summing over the entire chain. Then we can rearrange the Hamiltonian such that one sum represents the nearest neighbour interactions and the other terms an external field.

$$\begin{aligned}\mathcal{H} &= -\sum_t \bar{J}x_{t+1}x_t - \sum_s\left[\bar{h}y_s x_s - \frac{1}{2}x_s \log\left(\frac{\cosh(\bar{h}+\Delta h)}{\cosh(\bar{h}-\Delta h)}\right) - \frac{1}{2}x_s \log\left(\frac{\cosh(\bar{J}+\Delta J)}{\cosh(\bar{J}-\Delta J)}\right) + \Delta J x_{s+1}\right] \\ &= -\sum_t \bar{J}x_{t+1}x_t - \sum_s\left[\bar{h}y_s - \frac{1}{2}\log\left(\frac{\cosh(\bar{h}+\Delta h)}{\cosh(\bar{h}-\Delta h)}\right) - \frac{1}{2}\log\left(\frac{\cosh(\bar{J}+\Delta J)}{\cosh(\bar{J}-\Delta J)}\right) + \Delta J\right]x_s \\ &= -\sum_t \bar{J}x_{t+1}x_t - \sum_s \widetilde{h}_j(y_s)x_s,\end{aligned} \tag{5.11}$$

where the external field $\widetilde{h}_j(y_s)$ consists of a fluctuating term that depends on y_s and a constant term,

$$\widetilde{h}_j(y_s) = \bar{h}y_s - \frac{1}{2}\log\left(\frac{\cosh(\bar{h}+\Delta h)}{\cosh(\bar{h}-\Delta h)}\right) - \frac{1}{2}\log\left(\frac{\cosh(\bar{J}+\Delta J)}{\cosh(\bar{J}-\Delta J)}\right) + \Delta J. \tag{5.12}$$

In the last line of Equation 5.11, it is clear that the Hamiltonian is really just the Hamiltonian of our familiar Ising model. There is a constant spin-spin coupling term, the strength of which is determined by the transition probabilities $\bar{a}$ and Δ. Then, there is the fluctuating term of the local external fields, the magnitude is constant and determined by the observation probabilities, but the direction is assigned randomly through y_s. And lastly, there is a constant term in the external fields that depends on both the transition and observation probabilities. The change of variables that we used in Section 2.4.4 to find the Hamiltonian of a random-bond Ising model with a constant external field does not help us here. The mapping would give a random-bond Ising model with an external field that has both a constant and a random fluctuating part.

We will now write the Hamiltonian in terms of the transition and observation probabilities again, to get a better picture of its behaviour in the context of the HMMs,

$$\begin{aligned}\mathcal{H} = &-\sum_t \frac{1}{4}\log\left(\frac{4\left(\bar{a}-1\right)^2 - \Delta a^2}{4\bar{a}^2 - \Delta a^2}\right)x_{t+1}x_t \\ &-\sum_s\left[\frac{1}{4}\log\left(\frac{4\left(\bar{b}-1\right)^2 - \Delta b^2}{4\bar{b}^2 - \Delta b^2}\right)y_s - \frac{1}{4}\log\left(\frac{(2\bar{a}+\Delta a-2)(2\bar{a}+\Delta a)}{(2\bar{a}-\Delta a-2)(2\bar{a}-\Delta a)}\right)\right. \\ &\left.-\frac{1}{4}\log\left(\frac{\left(2\bar{b}+\Delta b-2\right)\left(2\bar{b}+\Delta b\right)}{\left(2\bar{b}-\Delta b-2\right)\left(2\bar{b}-\Delta b\right)}\right) + \frac{1}{4}\log\left(\frac{(-2\bar{a}+\Delta a+2)(2\bar{a}+\Delta a)}{(-2\bar{a}-\Delta a+2)(2\bar{a}-\Delta a)}\right)\right]x_s.\end{aligned} \tag{5.13}$$

When Δa and Δb are zero, the known Hamiltonian for the symmetric case is recovered. To make this

more explicit, we Taylor expand for small Δa and Δb,

$$\begin{aligned}\mathcal{H} =& -\sum_t \frac{1}{4}\log\left(\frac{(1-\bar{a})^2}{\bar{a}^2}\right) x_{t+1}x_t \\ &-\sum_s \left[\frac{1}{4}\log\left(\frac{(1-\bar{b})^2}{\bar{b}^2}\right) y_s + \frac{(2\bar{a}-1)}{4\bar{a}(1-\bar{a})}\Delta a + \frac{(2\bar{b}-1)}{4\bar{b}(1-\bar{b})}\Delta b + \frac{\Delta a}{4\bar{a}(1-\bar{a})}\right] x_s + \mathcal{O}(\Delta a^2, \Delta b^2) \\ =& -\sum_t \frac{1}{2}\log\left(\frac{1-\bar{a}}{\bar{a}}\right) x_{t+1}x_t - \sum_s \frac{1}{2}\left[\log\left(\frac{1-\bar{b}}{\bar{b}}\right) y_s + \frac{\Delta a}{1-\bar{a}} + \frac{(2\bar{b}-1)}{2\bar{b}(1-\bar{b})}\Delta b\right] x_s + \mathcal{O}(\Delta a^2, \Delta b^2).\end{aligned} \tag{5.14}$$

We have seen that MAP estimation of the hidden state of the two-state, two-symbol HMM is essentially the same as the minimalization of the appropiate Ising model Hamiltonian. But, how does Bayesian filtering fit in? The phase transitions found in MAP estimation [23] do not coincide with those found through Bayesian filtering in symmetric 2×2 HMMs. The equivalent of filtering in the context of an Ising model would be something like this: Imagine a random string of spin ups and spin downs, the probability of switching states from one spin to another is still governed by the transition matrix. The observation of the spins is randomly influenced by the external fields, just as before. However, the question we ask is different now. Instead of finding the most likely sequence of state, we are inferring just one. Given a string of observed spins, add one spin at the end randomly, and observe it, What is the most likely state of the spin that was just added? Repeat this for every added spin. It seems like an odd question to ask in the context of an Ising model. However, maybe similar methods could be used when asking the right question.

All in all, we have a reasonable understanding of the behaviour of the discord parameter in 2×2 HMMs. In this chapter, we have seen that the transitions in the discord exist not only in 2×2 symmetric HMMs but also when we introduce asymmetries in the transition and/or observation matrices. We can calculate the mean critical error probability for asymmetric systems as well, and they agree with the results of simulations. A new feature is a linear regime in the discord at sufficiently high mean observation probabilities. We can explain and quantify the onset and the value of the discord in this regime. Lastly, we considered the mapping of the 2×2 HMM to an Ising model again. We extended the derivation for symmetric HMMs from Section 2.4.4 to more general asymmetric HMMs.

We acknowledge helpful suggestions from Malcolm Kennett. Particularly, for pointing out that some nasty terms in the Hamiltonian can be ignored because they are, in fact, constants.

Chapter 6

Symmetric m-state, n-symbol hidden Markov models

Up to now, we have restricted ourselves to two-state, two-symbol HMMs. We have seen that the transitions in the discord parameter exist in both symmetric and asymmetric 2×2 HMMs. Now, we are interested to see what happens in a model with more states and symbols. In this chapter, we will generalize the symmetric 2×2 hidden Markov model to a symmetric HMM with an arbitrary number of states: a symmetric $n \times n$ HMM. Then, we will look at another symmetric $n \times n$ HMM, a random walk on a lattice with constant background noise. We will also consider $2 \times n$ HMMs, which we have to approach in a slightly different way since multiple symbols correspond to the same hidden state. We investigate whether the transitions that we have encountered in the discord parameter exist in these systems and, in particular, we look at the limiting cases where $n \to \infty$.

6.1 Symmetric n-state n-symbol hidden Markov models

In general, an $n \times n$ HMM can depend on up to $2n(n-1)$ independent variables. To keep the number of parameters manageable, we will consider symmetric HMMs with doubly stochastic matrices and only two classes of states: An observation is either correct and the symbol is the 'same' as the underlying state, or an observation is incorrect and the system sends out an 'other' symbol). The HMM matrices are

$$\mathbf{A} = \begin{pmatrix} 1-a & \frac{a}{n-1} & \cdots & \frac{a}{n-1} \\ \frac{a}{n-1} & 1-a & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{a}{n-1} \\ \frac{a}{n-1} & \cdots & \frac{a}{n-1} & 1-a \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} 1-b & \frac{b}{n-1} & \cdots & \frac{b}{n-1} \\ \frac{b}{n-1} & 1-b & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{b}{n-1} \\ \frac{b}{n-1} & \cdots & \frac{b}{n-1} & 1-b \end{pmatrix}. \tag{6.1}$$

This system depends on only two parameters for a given number of states n: the transition probability a, and the observation error probability b. It is a straightforward generalization of the symmetric 2×2 HMM that we studied in Chapters 2 and 4. Before, we labeled the states 1 and -1; now we label the states $1, 2, \ldots, n$. This transition matrix describes a system that has a probability $1-a$ to stay in the same state and equal probabilities to transition to any other state, $\frac{a}{n-1}$. The observation matrix describes a measurement with constant background noise; there is a certain probability of observing the correct symbol $1-b$ and equal probabilities of observing a different symbol, $\frac{b}{n-1}$. Note that the probability of observing a correct or incorrect symbol does not change when we increase the number of states n; only the probability of observing a specific incorrect symbol decreases.

The steady-state probabilities of these systems are

$$\boldsymbol{p}_x = \frac{1}{n}\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix} \quad \text{and} \quad \boldsymbol{p}_y = \frac{1}{n}\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}, \tag{6.2}$$

which makes sense in a system where all states and symbols are equivalent.

One can define a signal-to-noise ratio based on the observation probabilities, the probability of observing a correct symbol and the probability of observing another symbol:

$$SNR = \frac{1-b}{b/(n-1)} - 1 = \frac{n-1-bn}{b}. \tag{6.3}$$

When $b = 0$ (perfect information), the signal-to-noise ratio goes to infinity. At $b = \frac{n-1}{n}$, we have $SNR = 0$: The signal is completely obscured by noise. Hence, we restrict ourselves to the range of parameters: $0 \leq a, b \leq \frac{n-1}{n}$, for an integer number of states, $n \geq 2$.

We will take a look at the maximum confidence level and the discord parameter for $n = 2, 3, 4, 10$ and discuss the limit of $n \to \infty$. Also, we will consider the discord between the actual (hidden) state and the filter for a direct measure of the filter's performance.

6.1.1 Maximum confidence level

We start with the calculation of the maximum confidence level, p^*. As in a symmetric 2×2 HMM, all states of a symmetric $n \times n$ HMM are equivalent, and thus the maximum confidence levels are the same. We calculate the maximum confidence for an arbitrary state i, starting from the last line of Equation 2.5,

$$p^* = \frac{1}{Z_{t,i}} P(y_t = i|x_t = i) \sum_{x_{t-1}} P(x_t = i|x_{t-1}) P(x_{t-1}|y^{t-1} = i). \tag{6.4}$$

The first two terms in the numerator are known from the transition and observation matrix of the HMM. The last term is p^* if $x_{t-1} = i$. For $x_{t-1} \neq i$, we can calculate it by demanding a normalized probability,

$$\begin{aligned} \sum_{x_{t-1}} P(x_{t-1}|y^{t-1} = i) &= 1 \\ p^* + (n-1)P(x_{t-1} = j|y^{t-1} = i) &= 1 \\ P(x_{t-1} = j|y^{t-1} = i) &= \frac{1-p^*}{n-1}. \end{aligned} \tag{6.5}$$

Plugging all of the terms into Equation 6.4, we are left the equation

$$p^* = \frac{(1-b)\left[(1-a)p^* + a\frac{1-p^*}{n-1}\right]}{(1-b)\left[(1-a)p^* + a\frac{1-p^*}{n-1}\right] + \frac{b}{n-1}\left[ap^* + \left(1-a+(n-2)\frac{a}{n-1}\right)(1-p^*)\right]}. \tag{6.6}$$

This expression can be solved for p^* in terms of a, b, and n; the solution is given by Equation B.12 in Appendix B.2, and it is plotted in Figure 6.1 as a function of b, for $a = 0.2$ and several n. The behaviour of the maximum confidence level as a function of a is qualitatively the same for the various n when we consider $0 \leq b \leq \frac{n-1}{n}$ and $\frac{1}{n} \leq p^* \leq 1$. The pink dashed line indicates the minimum p^* necessary for the filter to be useful. At p^*'s lower than this, the state estimate based on the filter, $\hat{x}_t^f$, will be negatively correlated to the observations. We see that at $b = \frac{n-1}{n}$ the maximum confidence levels fall below this line, see red dots,exactly when the signal-to-noise ratio falls below zero. The filter cannot say anything useful when it is impossible to distinguish between the signal and the noise.

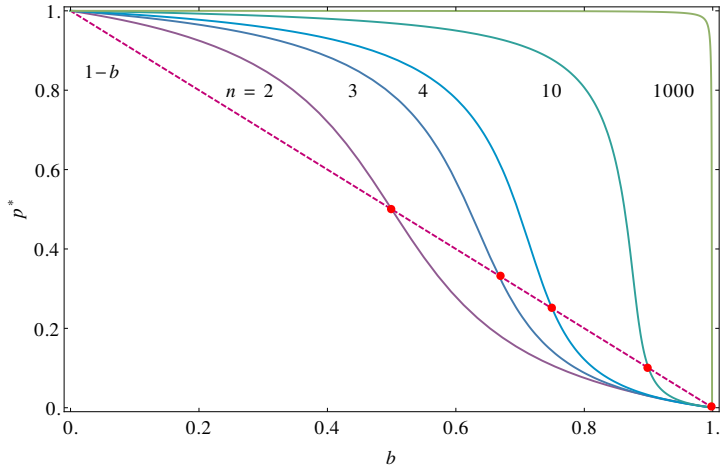

Figure 6.1: Maximum confidence level p^* for symmetric $n \times n$ HMMs. The curves are all for a system with transition probability $a = 0.2$ and as function of the observation error probability b. The curve '$1 - b$' indicates the maximum b and minimum p^*, where it intersects the maximum confidence levels (red dots), that we will study.

Lastly, we consider the limit of an infinite number of states and symbols ($n \to \infty$). The maximum confidence level goes to 1 for all $0 \leq a, b \leq 1$. The probability to transition to any other specific state becomes vanishingly small, while the probability to stay in the same state is unchanged. The filter can, theoretically, be 100% certain that the system is in a particular state for a long chain of identical observations. Depending on the parameters a and b, it might be very rare or even impossible for the filter to achieve this level of confidence.

6.1.2 Discord parameter

The definition of the discord parameter as introduced in Section 2.3 has to be generalized to accommodate more than two states and symbols. We will define the discord so that it is still 0 when the filter's state estimate and the naive state estimate are the same; $\hat{x}_t^f = \hat{x}_t^o$ for all t, 2 when $\hat{x}_t^f \neq \hat{x}_t^o$ everywhere, and 1 when they are uncorrelated. The discord is given by

$$D = 1 - \frac{1}{N} \sum_{t=1}^{N} f(\hat{x}_t^o, \hat{x}_t^f), \tag{6.7}$$

where f is a function that depends on the most likely states according to the naive observations and the filter algorithm;

$$f(\hat{x}_t^o, \hat{x}_t^f) = \begin{cases} 1, & \text{if } \hat{x}_t^o = \hat{x}_t^f \\ -1, & \text{if } \hat{x}_t^o \neq \hat{x}_t^f \end{cases}. \tag{6.8}$$

When $n = 2$, this definition reduces to that used for 2×2 HMMs from Section 2.3. The discord parameter for n states and symbols is shown in Figure 6.2, for $0 \leq b \leq \frac{n-1}{n}$ ($\infty \geq SNR \geq 0$). Both the discord and the observation probability are scaled to emphasize the similarities between the plots:

$$D' = D \cdot \frac{n}{2(n-1)}, \quad b' = b \cdot \frac{n}{2(n-1)}. \tag{6.9}$$

The colours are the same for every subfigure, but some curves overlay each other. The stepsize $\delta b = 0.005$ is constant across the subfigures.

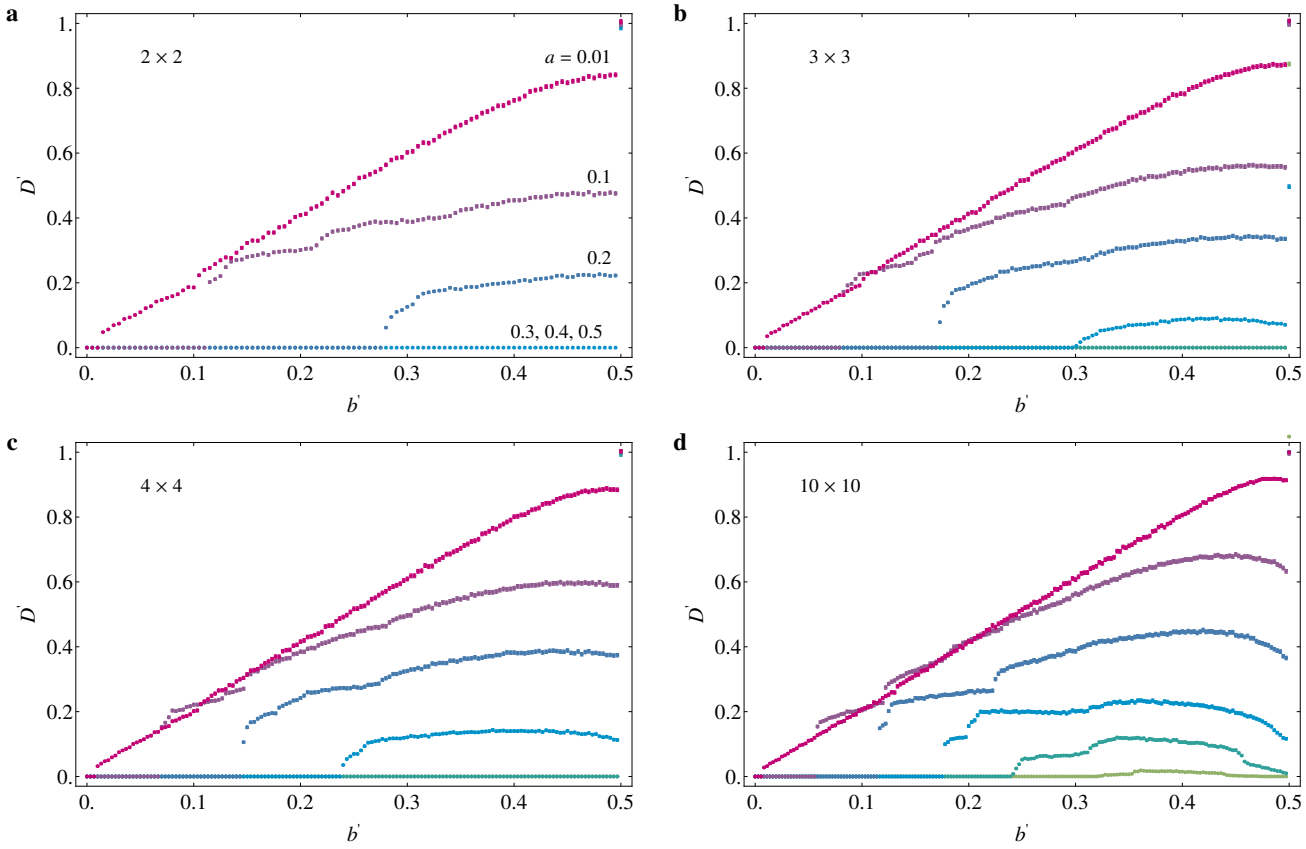

Figure 6.2: Scaled discord parameter for symmetric $n \times n$ HMMs, as a function of the scaled observation error probability b' for several transition probabilities a.

Qualitatively, the behaviour of the discord stays the same for an increasing number of states and symbols. Every curve starts at a discord of zero, becomes non-zero at a certain observation probability, increases for increasing $b < 0.5$ and has a jump at $b' = \frac{1}{2}$ ($b = \frac{n-1}{n}$). At this point, the filter algorithm returns a constant probability for all states and the filter is useless. Effectively, the filter now measures how often we observe a certain symbol, because the filter estimates the same state over and over again, whereas the observations are negatively correlated to the state. In the limit of an infinite chain, the system will spend an equal fraction of time in each state and observe each symbol for an equal fraction of time. We will observe each symbol for a time $T = 1/n$ on average. This corresponds to the jump to $D = 1 - \frac{1-(n-1)}{n} = 2\frac{n-1}{n}$ or $D' = 1$ that can been seen in the figures. At $b = 0.5$, observations are still uncorrelated with the underlying state. This point is shifted for the scaled variables to $b' = 0.375, 0.333$ and 0.278 for $n = 3, 4$ and 10, respectively. At higher b, the observations are negatively correlated to the underlying state, and we can see the discord *decrease* for increasing b. Interestingly, for $n > 2$, the filter seems to be useful even for $b > 0.5$ in some cases. Both the observations and the filter are getting less accurate, but they are inaccurate in the same way. The filter recognizes that it cannot do (much) better than the naive observations for certain parameters, and the observations predominantly determine the filter's behaviour. We have found no other possible explanation for the decreasing discord.

We have seen that there are discord transitions between the filter and naive state estimate, but are there discord transitions between the filtered state and the hidden, true, state? And, can we get a better measure of the filter's performance with this discord? We consider the discord D_{TF} between the hidden (true) state x_t and the filter's state estimate $\hat{x}_t^f$ to look at the filter's performance. The behaviour of D_{TF}

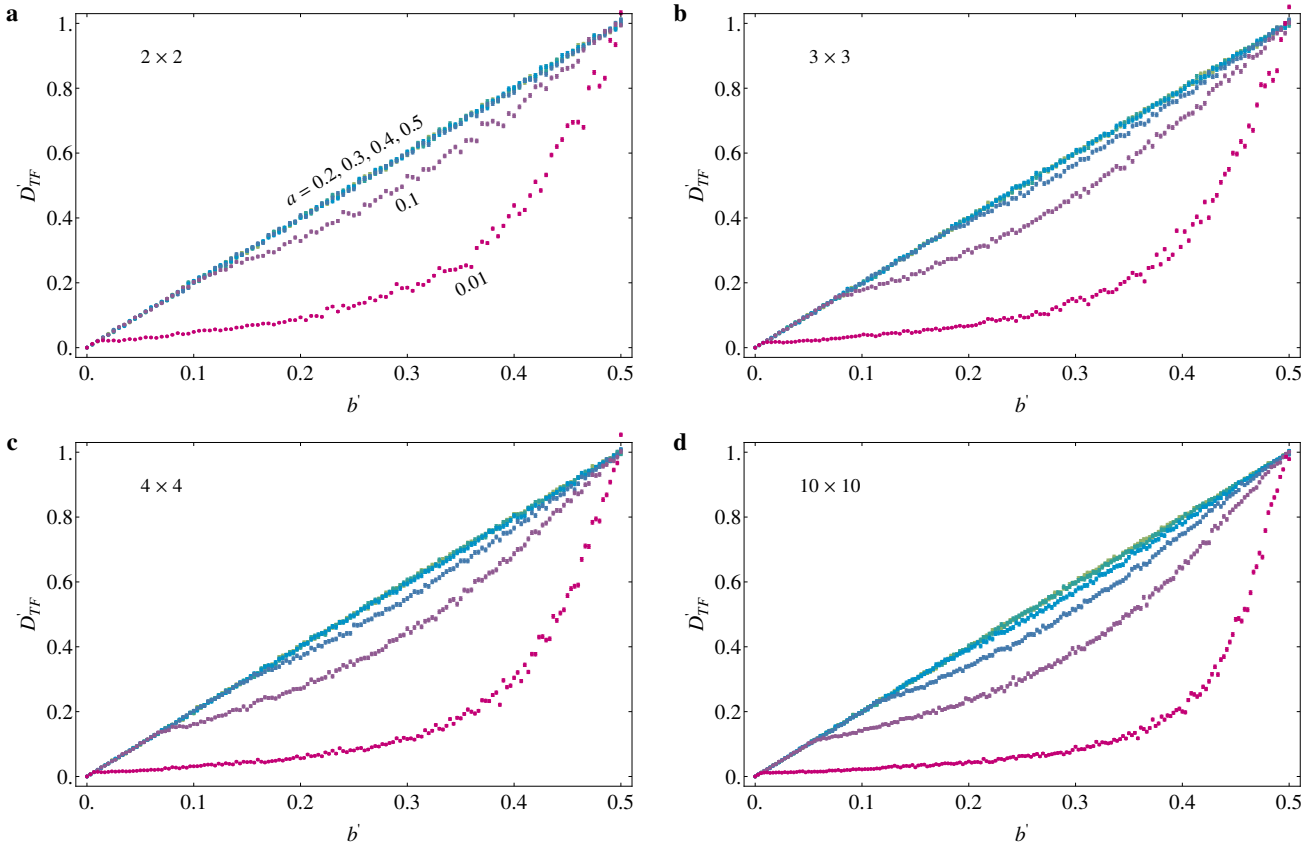

Figure 6.3: Scaled discord parameter between the hidden state and the filter's state estimate for symmetric $n \times n$ HMMs, as a function of the scaled observation error probability b' for several transition probabilities a.

(Figure 6.3) also stays qualitatively the same for different numbers of states. The colours indicating the various transition probabilities are the same as in the previous figure. When $D_{TF} = 0$, the filter perfectly follows the hidden state, and when $D_{TF} = 1$, the filter and the state are uncorrelated. The discord never exceeds the line $D_{OT} = 2b$ ($D'_{OT} = 2b'$), which is the discord between the observations and the hidden states for symmetric HMMs. Hence, the filter never performs worse than the naive observations. The filter clearly performs best for low transition probabilities and worst for high transition probabilities. Keeping a and b fixed, we see that the filter is indeed more accurate for higher n. More symbols means a greater variety in wrong observations, which makes it easier for the filter to distinguish between a sequence of wrong observations and transitions. At low error probabilities, the discord is a linear function of b, which corresponds to the parts in Figure 6.2 where the discord is zero.

Interestingly, we see no evidence of phase transitions such as the ones in the discord between the observations and the filter's state estimate. This discord looks like a continuous function of b apart from fluctuations due to noise. The decreasing discord in Figure 6.2 at high b has no clear equivalent in Figure 6.3 either.

6.1.3 Critical observation probability

The critical observation error, the b where the discord becomes non-zero, decreases with increasing number of states. In other words, the observations and filter disagree when there are fewer errors in the system. This is another indication that the filter performs better for higher n. We now quantify the behaviour of

the critical observation error, as seen in Figure 6.2. We start by defining the threshold where the discord becomes non-zero for a symmetric $n \times n$ HMM:

$$P(x_{t+1} = i | y_{t+1} = j, y^t = i) = P(x_{t+1} = j | y_{t+1} = j, y^t = i) \tag{6.10a}$$

$$P(x_{t+1} = i | y_{t+1} = j, y^t = i) \geq P(x_{t+1} = m | y_{t+1} = j, y^t = i). \tag{6.10b}$$

Equation 6.10 has to hold for all states $i \in \{1, 2, \ldots, n\}$, $j \neq i$, and $m \neq i, j$. In words, after observing a long sequence of symbols i, followed by an observation j, When is the probability of being in state i equal to the probability of being in state j? Additionally, the probability of being in any other state has to be smaller than the probability of being in state i. Again, we are just looking for the lowest b at which the filter's state estimate starts to disagree with the naive state estimate. In general, we need to obey these $n(n-1)(n-2)$ equations for $n \times n$ HMMs. However, for the symmetric HMM under consideration, all states are equivalent and the set of equations simplifies greatly. We need only consider one i and one j; for example, take $i = 1$ and $j = 2$. Then there are still $n - 1$ equations left that need to be satisfied. Fortunately, the inequalities in Equation 6.10b are automatically satisfied, at least in the symmetric case. Now we are left with a single equation,

$$P(x_{t+1} = 1 | y_{t+1} = 2, y^t = 1) = P(x_{t+1} = 2 | y_{t+1} = 2, y^t = 1). \tag{6.11}$$

For a symmetric 2×2 system, we can use the normalization of probabilities to write $P(x_{t+1} = 2 | y_{t+1} = 2, y^t = 1) = 1 - P(x_{t+1} = 1 | y_{t+1} = 2, y^t = 1)$, and the condition reduces Equation 2.18. The complete derivation of the critical error probabilities for symmetric $n \times n$ HMMs can be found in Appendix B.2. The solutions we find are

$$b_c = \frac{1}{2(n-1)} \left((n-1) + (n-2)a - \sqrt{(n-2)^2 a^2 - 2n(n-1)a + (n-1)^2} \right), \quad b_c = \frac{n-1}{n}. \tag{6.12}$$

This result is plotted in Figure 6.4 for $n = 2, 3, 4$ and 10. The figure shows both the analytical results and the simulated results, which agree quite well, especially for the lower n. The results for $n = 2$ are the same results as shown in Section 2.4. The $n = 10$ curve deviates from the simulations a little bit at higher a; it is not clear to us why this happens. The area under the curves indicates the parameter regime where $D = 0$; where the state estimates with and without memory are in agreement. Above the critical error probability, the two state estimates differ. There are no discontuities as $b_c \to \frac{n-1}{n}$; the curves are simply very steep and the simulations are done using a constant stepsize in a.

At small transition probabilities a, the curves are almost indistinguishable. There are few transitions, and the probability of staying in the same state is equal for all the different HMMs. This means that the dynamics change very little when the number of states changes, and hence the critical error probabilities are similar. Eventually, at higher a, each curve reaches a constant value $b_c = \frac{n-1}{n}$. This 'plateau' corresponds to the jump from $D = 0$ to $D = 2\frac{1-n}{n}$, as seen in Figure 6.2.

Lastly, we consider the limit of an infinite number of states, $n \to \infty$. The probability of transitioning into another state is still a, but the probability of transitioning to a particular state becomes vanishingly small. The limit of the critical error probability for an infinite number of states is given by

$$\lim_{n \to \infty} b_c = a. \tag{6.13}$$

This limit is plotted together with the b_c's for finite systems in Figure 6.4. As expected, the critical error probability in the limiting case is similar to other b_c's at lower a. This result implies that even in the limit of an infinite number of states, the discord has a transition from $D = 0$ to $D > 0$.

6.2 Diffusing particle

In this section, we look at another special case of an $n \times n$ hidden Markov model, one with more obvious real-world applications: A one-dimensional diffusing particle on a lattice of size n with periodic boundary

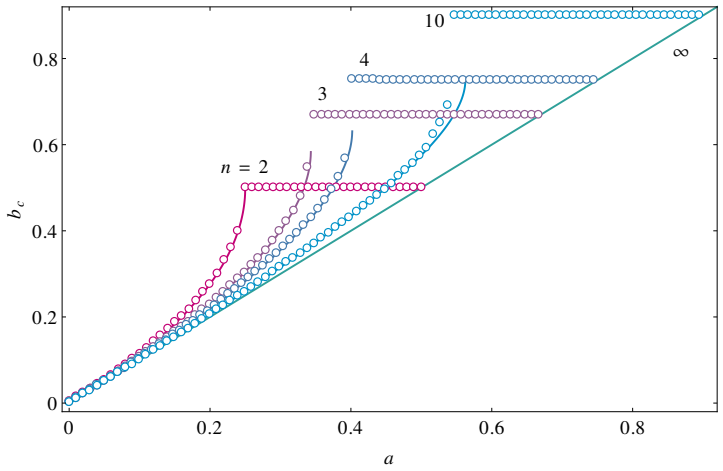

Figure 6.4: Phase diagram of symmetric HMMs with n states and symbols. The critical error probability is shown as a function of the transition probability a. The lines are analytical solutions; the circles are the results of simulations.

conditions and constant background noise. It is a particular case of a symmetric $n \times n$ HMM. The particle can stay at the same position with probability $1 - a$, or it can move either one step to the left or to the right with equal probabilities $a/2$. The transition matrix has non-zero elements on the diagonal only, the off-diagonals above and below it and the corners of the matrix.

$$\mathbf{A} = \begin{pmatrix} 1-a & \frac{a}{2} & 0 & \dots & \frac{a}{2} \\ \frac{a}{2} & 1-a & \frac{a}{2} & \ddots & 0 \\ 0 & \frac{a}{2} & 1-a & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{a}{2} \\ \frac{a}{2} & 0 & \dots & \frac{a}{2} & 1-a \end{pmatrix}. \tag{6.14}$$

The observation matrix is as defined at the beginning of the chapter (see Equation 6.1). Note that both matrices are symmetric and doubly stochastic, just as before. The steady-state probabilities are the same, too: The probabilities of being in a particular state and the probabilities of observing a specific symbol are all equal to $1/n$ in the long run.

We start by looking at the scaled discord parameter in Figure 6.5. The discord parameter and the observations are scaled in the same way as in the previous section (Equation 6.9). Figures 6.5a and b show results for a particle hopping over four lattice sites for different transition probabilities; Figures 6.5c and d are for a particle hopping over ten lattice sites. For $a = \frac{1}{2}$, the particle is equally likely to stay put or hop; for $a = \frac{2}{3}$, it is equally likely to stay, hop to the left or hop to the right; and at $a = 1$, the particle equally likely to hop left or right (no chance of staying put). We still see transitions in the discord as a function of the observation error probability. The curves seem smoother for an increasing number of lattice sites. Also, above a certain observation probability, the discord *decreases* with increasing b. From the unscaled data, it is clear that this happens only when $b > 0.5$, that is when the probability of a wrong observation is greater than the probability of an accurate observation.

The critical observation error for this system can be calculated from the same equation as b_c for the $n \times n$ symmetric system. It is zero for most parameters, especially for systems with more lattice sites. An exception is the curve for $a = 0.6$ in Figure 6.5b, where the transition to non-zero discord is found at $b' = \frac{1}{2}$. Intuitively, we can understand that the discord becomes non-zero at lower b in systems with more

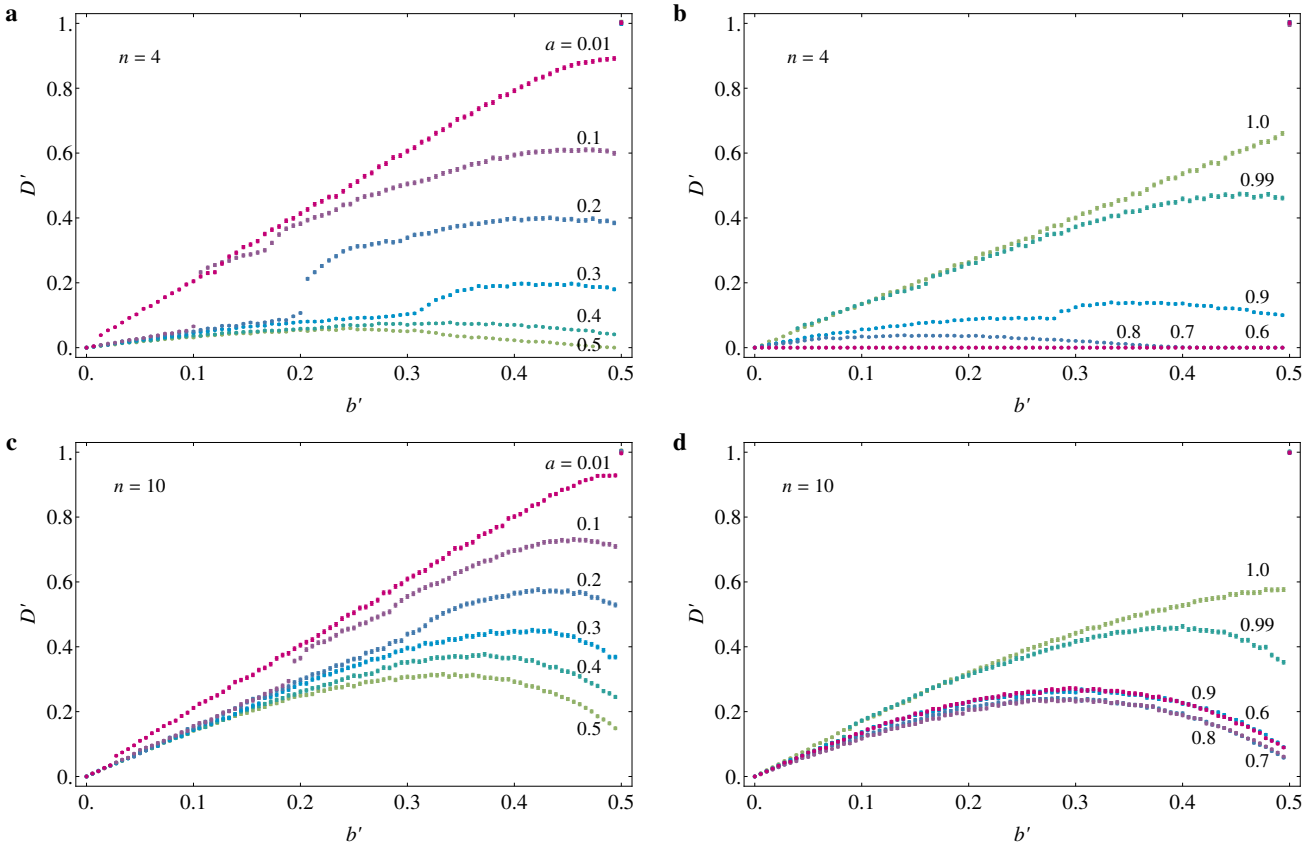

Figure 6.5: Scaled discord parameter for a particle performing a random walk on a lattice in one dimension, with constant background noise and periodic boundary conditions. **a**-**b**, four lattice sites. **c**-**d**, ten lattice sites.

lattice sites. Let us consider systems at fixed transition and observation probabilities; the probabilities of not transitioning and of observing a correct symbol are all equal. However, when there are more lattice sites, the transitions are relatively restricted; the fraction of allowed transitions decreases with an increasing number of lattice sites. For example, in a system with four lattice sites, at any given time, three out of four lattice sites can be reached with a single transition. By contrast, in a system with 10 lattice sites, only three out of ten sites can be reached. Thus, for a higher number of lattice sites, the filter will be able to distinguish between right and wrong symbols more accurately. This is reflected in the discord curves being closer to a straight line at low b' (remember, the discord between the true state and the observations is a straight line $D = 2b$). We cannot compare Figures 6.5b and d directly to previous results, because we have not looked at such high transition probabilities before. These high transition probabilities do make sense in this system, where $a = 1$ corresponds to the particle being forced to go either left or right at every timestep. At most high a, we see that the filter does not improve much over the naive observations. As $a \to 1$, interestingly, the filter's performance seems to improve a little. However, from these plots we cannot be sure, because we are not comparing with the hidden state.

The system with only four lattice sites is quite similar to the 4×4 symmetric HMM from Chapter 6. The discord parameters of these systems are also quite similar; see Figures 6.5a and 6.2c. The main difference is the non-zero discord at higher transition probabilities in the current system. Comparing the discord of the system with ten lattice sites with the 10×10 symmetric HMM (Figures 6.5c and 6.2d), the differences are clearer. The non-zero discord at higher transition probabilities and the decrease in

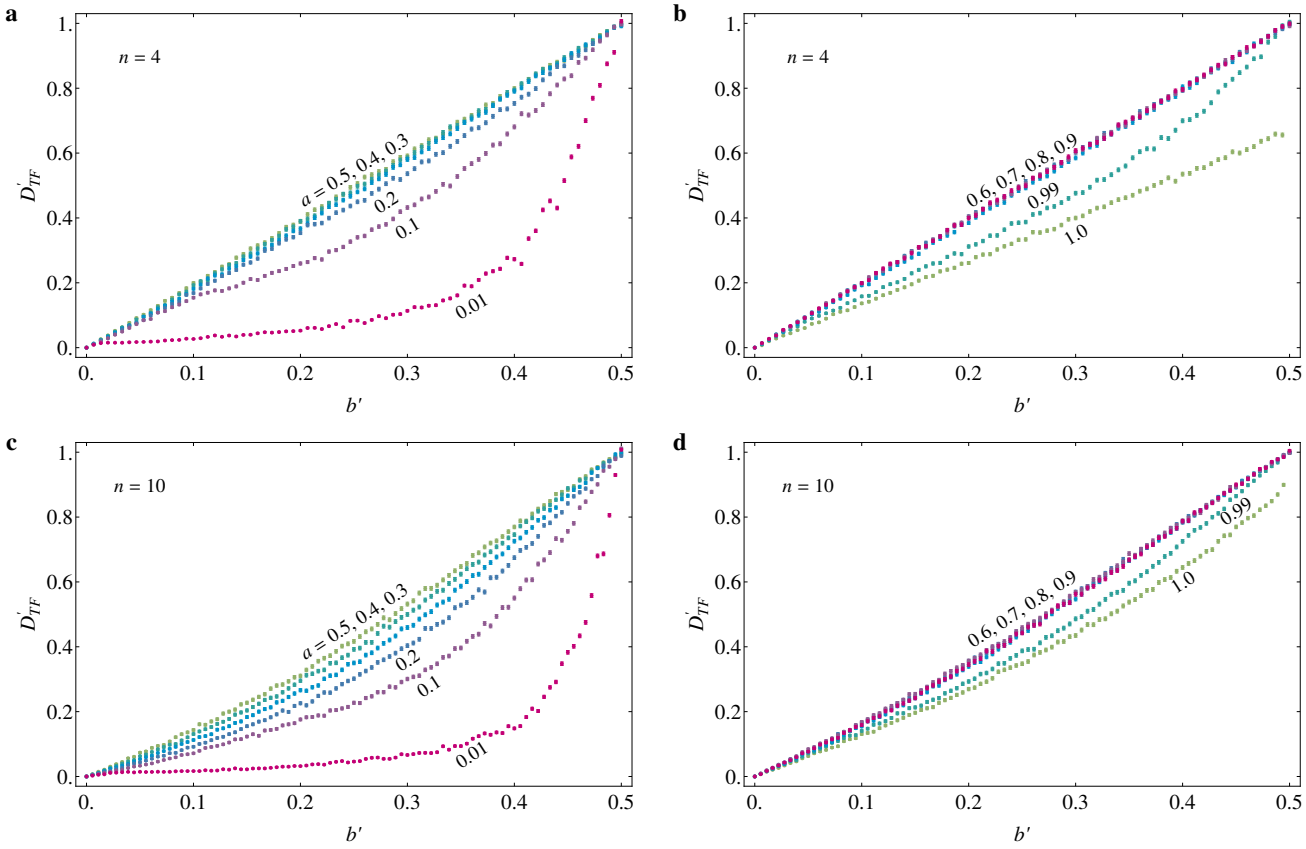

Figure 6.6: Scaled discord between the 'true' state and the filter's state estimate for a particle performing a random walk on a lattice, in one dimension, with constant background noise and periodic boundary conditions. **a-b**, four lattice sites. **c-d**, ten lattice sites.

discord are more dramatic.

We look at the discord between the hidden (true) state of the system and the filter's state estimate; see Figure 6.6. Again, these results (Figures 6.6a and c) are strikingly similar to the results for symmetric $n \times n$ systems (Figures 6.3c and d). We see that the filter performs best for low transition probabilities. At high transition probabilities, the discord D'_{TF} approaches the line $D' = 2b'$. There is no sign of the transitions seen in the discord between the observations' and the filter's state estimate. The discord D'_{TF} curves for the diffusing particle are spread out more at the same parameters as those of the $n \times n$ symmetric HMM, and they do not approach $D' = 2b'$ as closely; they remain a little more bent. In short, the discord D'_{TF} of the diffusing particle is a little lower than that of the corresponding $n \times n$ HMMs at the same parameters. From this, we conclude that the filter performs better for the diffusing particle than the $n \times n$ HMMs, just as we argued before.

6.3 Symmetric two-state, n-symbol hidden Markov models

In this section, we consider a different kind of HMM, one with more symbols than states. In Section 2.1, we discussed the ion channel as a system that can be described by of a HMM with more symbols than states. It can be described as having two states: 'open' and 'closed'. Experimentally, the state cannot be observed directly and a correlated observable, the electrical conductance, is measured. The conductance depends on the state of the system and can take on a continuum of values. We look at systems with

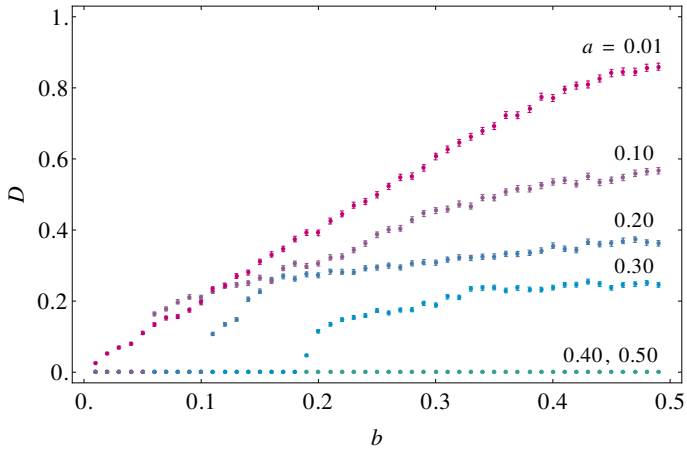

Figure 6.7: Discord parameter of a symmetric 2×4 HMM as a function of the observation probability b, for several transition probabilities a.

only two states and an even number of symbols, and systems with two states and an infinite number of symbols. The observation probability distributions for a two-state system with Gaussian errors is shown in Figure 3.2. The calculations and simulations for these systems are described in Section 3.2.

The result of simulations of the discord of a system with four symbols is shown in Figure 6.7. There are no points at $b = 0$ and $b = 0.5$, because that would require simulating infinitely narrow and infinitely wide Gaussian distributions, respectively. However, we know that for $b = 0$, the discord is equal to zero, because the filter's state estimate follows the observations exactly in the case of perfect information. And, for $b = 0.5$, the filter's probability is once again restricted to $P(x_t = 1|y^t) = 0.5$ for all t, and consequently, the discord should be one. The behaviour of the discord in this system is quite similar to that of symmetric 2×2 HMMs (Figure 2.6). The discord becomes non-zero at lower observation probabilities for fixed transition probabilities, which could indicate that the filter performs a little better with the extra information that is available here. For example, in the 2×2 system, the discord is zero for $a = 0.30$ and $b < 0.50$, but in this 2×4 case, the discord is non-zero for $b > 0.18$. The critical error probability as a function of the number of symbols, for a fixed transition probability $a = 0.30$, is shown on a log-log scale in Figure 6.8. The line is a fit of the data to a function $b_c(n) = a/n$, where a is a constant. The resulting curve is $b_c(n) = 0.923/n$, which gives a good asymptotic fit to the data. The behaviour is similar for other transition probabilities, and it suggests that only in the limit of infinitely many symbols, the (discontinuous) transitions to non-zero discord disappear.

In the $n \times n$ symmetric HMMs, Figure 6.2, we saw similar behaviour: The critical error probability decreases as a function of the number of symbols (and states). However, if we look at the limit of $n \to \infty$ of the critical error probability of these HMMs (Figure 6.4), the critical error probability does not go to zero.

Now, we will study the limiting case of infinitely many symbols in the two-state, n-symbol HMM. For the generalization to a continuous state space of observations, the filtering algorithm has to be adjusted slightly, since we cannot generate an infinitely large matrix. Instead of first generating an observation matrix and using the elements in the filtering algorithm, one uses the probability distribution function of the appropriate normal distributions. The discord for a HMM with two states and an infinite number of symbols is shown in Figure 6.9. The behaviour is qualitatively the same as the discord in systems with two states and more symbols, the similarities are, of course, stronger when looking at a larger number of symbols. We see that the transition to non-zero discord shifted to lower transition probabilities in this diagram (except for the curve with $a = 0.50$, which is zero everywhere). The curves level off more quickly compared to the 2×4 HMM, especially at high a. This is behaviour we expect for the discord when we consider systems with a finite number of symbols. The discord at $a = 0.01$ is almost linear, only at higher

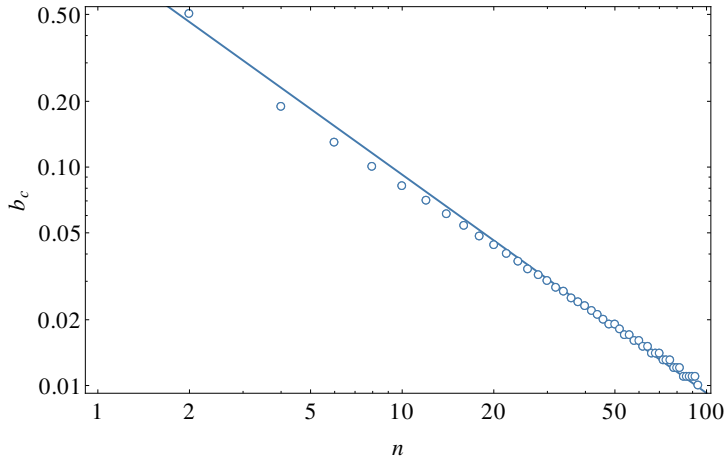

Figure 6.8: Log-log plot of the critical observation probability b_c of symmetric $2 \times n$ HMMs, for a fixed transition probability $a = 0.30$, as a function of the number of symbols n. The line is a fit of the form $b_c(n) = a/n$, where a is a constant.

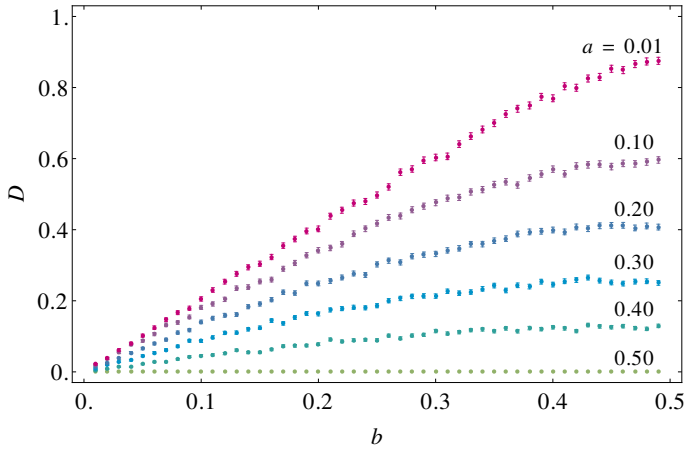

Figure 6.9: Discord parameter of a HMM with two states and infinitely many symbols as a function of the observation probability b, for several transition probabilities a.

observation probabilities does it start to deviate, and therefore, it is quite close to the discord between the hidden state and the filter's state estimate. This system has no clear higher-order transitions, except the transition at $b = 0.5$,

We have seen that several features in this chapter's various $m \times n$ HMMs that are analogous to those observed in the discord parameter of 2×2 and $n \times n$ HMMs: the critical error probability, increasing discord, intermediate transitions and a jump at $b = \frac{n-1}{n}$. We were even able to quantify the transitions from $D = 0$ to $D > 0$ and the transition at $D = 2\frac{n-1}{n}$ for $n \times n$ HMMs. These transitions were found in all systems that were considered; they are not exclusive to the dimensionality or symmetry of the HMM. Also, we explored the performance of the filter as a function of observation error probability by looking at the discord between the hidden state and the filter's state estimate. We find that keeping a memory of observations is most useful at intermediate error probabilities and low transition probabilities. The same conclusion was reached for symmetric 2×2 HMMs by Bechhoefer [24]. Moreover, one is never worse off

keeping a memory of observations compared to not keeping a memory, and the filter performs better for a higher number of states. No transitions equivalent to those in the discord between the observations' and filter's state estimates were seen.

Chapter 7

Conclusion

In this thesis, we have investigated when keeping a memory of observations pays off. We used hidden Markov models to look at a relatively simple system with discrete states and noisy observations. We infer the underlying state of the HMM from the observations in two ways: through naive observations, which only take into account the current observation, and a state estimate found through Bayesian filtering (and decision making), which take the history of observations into account. We compared the state estimates by calculating the discord, D, between the two.

We started by studying the simple symmetric two-state, two-symbol HMM. Our results, simulations and calculations, agree with those in [24]. We were able to explain some of the behaviour of the discord parameter, and we confirmed this with calculations. This was done for the critical observation probability and the transition to $D = 1$. We suggested a mechanism for the emergence of higher-order transitions and to explain the jump size at the transition to non-zero discord.

We looked at several other HMMs to investigate how general the behaviour of the discord is, and, in particular, if the phase transition to $D > 0$ exists in more complicated systems. We looked at asymmetric 2×2 HMMs, some symmetric $n \times n$ HMMs, and symmetric $2 \times n$ HMMs. In all these system we found a phase transition to non-zero discord, except in the $2 \times n$ in the limit of infinitely many symbols, $n \to \infty$. The general features of the discord stayed the same in all these systems: it starts at $D = 0$ for $b = 0$; it becomes non-zero at some critical error probability; and it increases for increasing error probability.

Some new features emerged in the asymmetric HMMs: the transition at $b = 0.5$ is no longer discontinuous for asymmetric transition probabilities; there is a regime where the discord is linear as a function of the observation probability; and the transition at $b = 0.5$ does not always go to 1. At certain values of a and b, we find that the critical error probability curves backward. We are not sure what causes this behaviour, but we suggested a mechanism that uses decreasing maximum confidence levels and the asymmetric steady-state probabilities. We also have extended the mapping of symmetric 2×2 HMMs onto Ising models to include asymmetric 2×2 HMMs. The resulting Hamiltonian is of an Ising model with a field that has both constant and randomly fluctuating components. Although much is known about solving Ising models, the questions we ask are odd in the context of Ising models. For example, filtering corresponds to asking the probabilities of the last spin in a chain at zero temperature, as a function of field. Thus, in mapping to an Ising model, we are trading off asking natural questions in a less-familiar context (HMMs) or asking strange questions of more familiar (Ising) models. It is not clear whether the latter will lead to a significant payoff.

We considered two different $n \times n$ HMMs: a straightforward generalization of the symmetric 2×2 HMM and a HMM describing a diffusing particle. The behaviour is very similar to the 2×2 symmetric HMM, especially when introducing a scaling factor that depends on n. The limit of $n \to \infty$, shows a transition at finite b, for the generalized $n \times n$ HMM. We also looked at the discord between the hidden state and the filter's state estimate, D_{TF}, to see if similar phase transitions can be found, and as a measure for the performance of the filter. The filter clearly performs best for low transition probabilities and low observation probabilities. Moreover, we find that keeping a memory of observations is never

worse than using the naive observations. We observed no phase transitions in D_{TF}, similar to those in D. A similar result is found in [23], where the Hamming distance between the true and estimated states is considered, which is very similar to the discord D_{TF}. An interesting result was found for the critical observation probability of $2 \times n$ HMMs was found: b_c, goes to zero for a large number of symbols and a fixed transition probability.

7.1 Outlook

We are left with several open questions. It is not yet clear why we find a phase transition when looking at the discord between naive observations and a state estimate based on filtering, but not in the discord between the true state and the filter's state estimate. Both parameters depend on the filter's state estimate and have the same decision making component. Also, there is a yet unexplained discrepancy in the critical observation probability at higher n in the $n \times n$ HMMs. Is this a problem with the numerics or the definition of the threshold used in the calculation? Another point we would like to understand better is the disappearance of the phase transition in the $2 \times n$ HMMs as the number of symbols goes to infinity.

Some other threads that could be worth pursuing are to

- Calculate the discord between the naive observations or the true state and the MAP state estimate, and verify the transitions seen in these inference problems [23]. Do we see the same difference in behaviour between D and D_{TF}?
- Try to describe Bayesian filtering from a statistical mechanical approach, perhaps coming back to an Ising or Potts model.
- Look at HMMs that describe more realistic systems, such as symmetric $n \times n$ HMM with Gaussian errors, and continuous state-space models. A linear dynamical systems is a continuous state-space model with linear dynamics and Gaussian noise. Kalman filtering can be used to infer the hidden state of the system in continuous state-space models [4, Chapt. 18].

We hope that this work motivates further inquiry, both theoretically and experimentally.

Bibliography

[1] D George and J Hawkins. A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. *IEEE International Joint Conference on Neural Networks*, 3:1812–1817, 2005.

[2] DG Tzikas, AC Likas, and NP Galatsanos. The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146, 2008.

[3] GF Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artif. intell.*, 42(2-3):393–405, 1990.

[4] KP Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[5] L Zdeborová and F Krzakala. Statistical physics of inference: Thresholds and algorithms. *arXiv:1511.02476*, 2015.

[6] DJC MacKay. *Information theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[7] AL Yuille and JM Coughlan. Fundamental limits of Bayesian inference: order parameters and phase transitions for road tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):160–173, 2000.

[8] MJ Kochenderfer and HJD Reynolds. *Decision Making under Uncertainty: Theory and Application*. MIT press, 2015.

[9] W von der Linden, V Dose, and U Von Toussaint. *Bayesian Probability Theory: Applications in the Physical Sciences*. Cambridge University Press, 2014.

[10] M Hauskrecht. Value-function approximations for partially observable Markov decision processes. *J. Artif. Intell. Res.*, 13:33–94, 2000.

[11] I Chadès, E McDonald-Madden, MA McCarthy, B Wintle, M Linkie, and HP Possingham. When to stop managing or surveying cryptic threatened species. *PNAS*, 105(37):13936–13940, 2008.

[12] PMC Dias and A Shimony. A critique of Jaynes' maximum entropy principle. *Advances in Applied Mathematics*, 2(2):172–211, 1981.

[13] L Zdeborová and F Krzakala. Phase transitions in the coloring of random graphs. *Phys. Rev. E*, 76(3):031131, 2007.

[14] N Sourlas. Spin-glass models as error-correcting codes. *Nature*, 339(6227):693–695, 1989.

[15] TLH Watkin, A Rau, and M Biehl. The statistical mechanics of learning a rule. *Rev. Mod. Phys.*, 65(2):499, 1993.

[16] ACC Coolen, R Kühn, and P Sollich. *Theory of Neural Information Processing Systems*. OUP Oxford, 2005.

[17] DJ Amit. *Modeling Brain Function: The World of Attractor Neural Networks*. Cambridge University Press, 1992.

[18] FJ Dyson. Existence of a phase-transition in a one-dimensional Ising ferromagnet. *Commun. Math. Phys.*, 12(2):91–107, 1969.

[19] E Babaev, A Sudbø, and NW Ashcroft. A superconductor to superfluid phase transition in liquid metallic hydrogen. *Nature*, 431(7009):666–668, 2004.

[20] A Albrecht and Paul J Steinhardt. Cosmology for grand unified theories with radiatively induced symmetry breaking. *Phys. Rev. Lett.*, 48(17):1220, 1982.

[21] N Merhav. Threshold effects in parameter estimation as phase transitions in statistical mechanics. *IEEE Transactions on Information Theory*, 57(10):7000–7010, 2011.

[22] A Decelle, F Krzakala, C Moore, and L Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Phys. Rev. Lett.*, 107(6):065701, 2011.

[23] A Allahverdyan and A Galstyan. On maximum a posteriori estimation of hidden Markov processes. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 1–9, 2009.

[24] J Bechhoefer. Hidden Markov models for stochastic thermodynamics. *New J. Phys.*, 17(7), 2015.

[25] K Maruyama, F Nori, and V Vedral. Colloquium: The physics of Maxwell's demon and information. *Rev. Mod. Phys.*, 81(1):1, 2009.

[26] CH Bennett. The thermodynamics of computation — a review. *Int. J. Theor. Phys.*, 21(12):905–940, 1982.

[27] M Bauer, AC Barato, and U Seifert. Optimized finite-time information machine. *J. Stat. Mech. Theor. Exp.*, 2014(9):P09010, 2014.

[28] E Dieterich, J Camunas-Soler, M Ribezzi-Crivellari, U Seifert, and F Ritort. Single-molecule measurement of the effective temperature in non-equilibrium steady states. *Nature Phys.*, 11(11):971–977, 2015.

[29] MA Rotondi. To ski or not to ski: Estimating transition matrices to predict tomorrow's snowfall using real data. *J. Stat. Educ*, 18(3):1–14, 2010.

[30] JD Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, pages 357–384, 1989.

[31] CI Jones. On the evolution of the world income distribution. *SSRN*, 1997. Available at `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=59412`.

[32] D George and J Hawkins. Towards a mathematical theory of cortical micro-circuits. *PLoS Comput. Biol.*, 5(10):e1000532, 2009.

[33] MC Gibson, AB Patel, R Nagpal, and N Perrimon. The emergence of geometric order in proliferating metazoan epithelia. *Nature*, 442(7106):1038–1041, 2006.

[34] L Page, S Brin, R Motwani, and T Winograd. *The PageRank citation ranking: bringing order to the web.* Stanford InfoLab, 1999. Available at `http://ilpubs.stanford.edu:8090/422/`.

[35] EL Cohen and S Berkovits. Exponential distributions in Markov chain models for communication channels. *Inform. Control*, 13(2):134–139, 1968.

[36] FP Kelly. *Reversibility and Stochastic Processes*. Wiley NY, 1979.

[37] S Rahav and C Jarzynski. Fluctuation relations and coarse-graining. *J. Stat. Mech. Theor. Exp.*, 2007(09):P09012, 2007.

[38] L Venkataramanan and FJ Sigworth. Applying hidden Markov models to the analysis of single ion channel activity. *Biophys. J.*, 82(4):1930–1942, 2002.

[39] O Cappé, E Moulines, and T Rydén. *Inference in Hidden Markov Models*. Springer Science & Business Media, 2006.

[40] S Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[41] SM Ross. *Introduction to Probability Models*. Academic Press, 10th edition, 2010.

[42] S Godsill, A Doucet, and M West. Maximum a posteriori sequence estimation using Monte Carlo particle filters. *Ann. I. Stat. Math.*, 53(1):82–96, 2001.

[43] A Allahverdyan and A Galstyan. Active inference for binary symmetric hidden Markov models. *J. Stat. Phys*, 161(2):452–466, 2015.

[44] R Douc, G Fort, E Moulines, and P Priouret. Forgetting the initial distribution for hidden Markov models. *Stochastic Process. Appl.*, 119(4):1235–1256, 2009.

[45] JA Thomas and TM Cover. *Elements of Information Theory*. Wiley New York, 2nd edition, 2006.

[46] O Zuk, I Kanter, and E Domany. The entropy of a binary hidden Markov process. *J. Stat. Phys*, 121(3-4):343–360, 2005.

[47] B Derrida, J Vannimenus, and Y Pomeau. Simple frustrated systems: Chains, strips and squares. *J. Phys. C*, 11(23):4749, 1978.

[48] WH Press, SA Teukolsky, WT Vetterling, and BP Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007.

Appendix A

Matlab code

In this appendix, the main code used for the simulations in this thesis can be found. Some simulations require this code to be adapted, for example if one wants to simulate a two-state, n symbol system. A description of the methods can be found in Chapter 3. All simulations were done in Matlab, and visualization was done in Mathematica. In the first section, an $n \times n$ hidden Markov model is set up, the Bayesian filtering algorithm is applied, and the discord between the observations and filter's state estimate is calculated. In the second section, these functions are combined to calculate the critical observation probability of a HMM.

A.1 Discord of n-state, n-symbol HMMs

The function HMMdiscord( ) sets up a HMM with the desired parameters and calculates the system's discord as a function of the mean observation error probability $\bar{b}$. The parameters that one can change to simulation a different system are:

- M (n in the text), the number of states and symbols of the HMM.
- N, the length or number of timesteps of the generated HMM. This affects the accuracy of the calculated discord.
- Npoints, the number of points $\bar{b}$ simulated for one discord curve.
- da (Δa), the asymmetry in the transition probabilities.
- da (Δb), the asymmetry in the observation probabilities.
- alst, list of $\bar{a}$'s for which to calculate the discord (each is a different curve).

These can be found in lines 5-11 of the code below. The transition matrix A (line 16) and observation matrix B (line 22) are for 2×2 HMMs, but one can easily use different matrices.

If one wants to simulate two-state, n-symbol HMMs, functions chain(M, N, A, B), filtering(M, N, A, B, Ostate), and discord(M, N, Ostate, Fstate) need to be modified as well.

```
1  % function calculates the discord of an n-state, n-symbol HMM
2  function HMMdiscord()
3
4      % set parameters for simulation
5      M = 2;                % number of states and symbols
6      N = 30000;            % number of timesteps
7      Npoints = 101;        % number points per curve
```

```
8       da = 0.0;               % difference in transition prob's
9       db = 0.0;               % difference in observation prob's
10      alst = linspace(0, 0.5, 6); % list of mean transition prob's
11      blst = linspace(0, 0.5, Npoints); % list of mean observation prob's
12
13      for j = 1:length(alst) % loop over values of a
14          a = alst(j);
15          A = [1-a+da/2, a+da/2; a-da/2, 1-a-da/2]; % transition matrix
16
17          file = fopen( sprintf('Discord_M%d_N%d_a%1.2f_da%1.2f_db%1.2f.dat',
                M, N, a, da, db), 'w'); % save data in different file for every
                curve
18
19          for i = 1:Npoints % loop over values of b
20              b = blst(i);
21              B = [1-b+db/2, b+db/2; b-db/2, 1-b-db/2]; % observation matrix
22
23              [Rstate, Ostate] = chain(M, N, A, B); % generate HMM
24              [Fstate] = filtering(M, N, A, B, Ostate); % apply filtering
                    algorithm
25              [D, SD] = discord(M, N, Ostate, Fstate); % calculate discord
                    between observations and filter
26              fprintf(file, '%1.6f %1.6f %1.6f\n', b, D, SD); % save data
27          end
28
29          fclose(file);
30      end
31  end
32
33
34  % function to generate M-state, M-symbol HMM with N timesteps, transition
        matrix A and observation matrix B. function returns an array with the
        state of the HMM and an array with the observations.
35  function [Rstate, Ostate] = chain(M, N, A, B)
36
37      % allocate memory
38      PcumA = zeros(M,M);         % cumulative prob. matrix of A
39      PcumB = zeros(M,M);         % cumulative prob. matrix of B
40      Rstate = zeros(1,N);        % array of the state of the system
41      Ostate = zeros(1,N);        % array of observations
42
43      % fill matrices with cumulative probabilities
44      PcumA(1,:) = A(1,:);
45      for i = 2:M                 % loop over columns
46          PcumA(i,:) = A(i,:) + PcumA(i-1,:);
47      end
48
49      if abs(PcumA(M,:)-1.) > 0.001 % check that columns sum to 1
50          error('Columns of matrix A must sum to 1')
51      end
52
53      PcumB(1,:) = B(1,:);
```

```
54      for i = 2:M
55          PcumB(i,:) = B(i,:) + PcumB(i-1,:);
56      end
57
58      if abs(PcumB(M,:)-1.) > 0.001
59          error('Columns of matrix B must sum to 1')
60      end
61
62      % generate hidden markov model
63      Rstate(1) = 1; % initial condition
64
65      for i = 2:N % fill array with states as described by A
66          r = rand(); % random number between 0 and 1
67          j = M - 1;
68
69          while j > 0 % compare r to cumulative prob's, if state is found,
                assign state and break loop
70              if r > PcumA(j,Rstate(i-1))
71                  Rstate(i) = j+1;
72                  j = -1;
73              else % if state not found, try next
74                  j = j-1;
75              end
76          end
77          if j == 0
78              Rstate(i) = 1;
79          end
80      end
81
82      for i = 1:N % fill array with observations as described by B
83          r = rand();
84          j = M - 1;
85          while j > 0
86              if r > PcumB(j,Rstate(i)) % prob of observing symbol j at time
                    i depends on the state at time i (Rstate(i))
87                  Ostate(i) = j+1;
88                  j = -1;
89              else
90                  j = j-1;
91              end
92          end
93          if j == 0
94              Ostate(i) = 1;
95          end
96      end
97  end
98
99
100 % function to apply filtering algorithm to array of observations. function
        returns an M by N array of prob's to be in state M at time N
101 function [Fstate] = filtering(M, N, A, B, Ostate)
102
```

```
103     Fstate = zeros(M,N);          % array of P(x_t|y^t)
104     V = steady_state(A);          % initial condition, steady-state prob
105     Fstate(:,1) = V(:);
106
107     % filter calculation
108     for i = 2:N % loop over chain length
109         pxxy = zeros(1,M);        % P(x_i|y^{i-1})
110
111         for k = 1:M
112             pxxy(:) = pxxy(:) + A(:,k)*Fstate(k,i-1);
113         end
114
115         Zk(:) = B(Ostate(i),:).*pxxy;
116         Fstate(:,i) = Zk(:)/sum(Zk); % Fstate(j,i) = P(x_i = j|y^i)
117     end
118 end
119
120
121 % function calculates the discord between the state estimates based on
      observations and filtered observations. function returns the discord and
      the standard deviation of the mean of this discord
122 function [D, SD] = discord(M, N, Ostate, Fstate)
123
124     % find state estimate based on filter prob's
125     MLS = zeros(1,N);             % Most Likely State = \hat{x}^f_t
126     for i = 1:N
127         num = 1; j = 1;
128         maxP = Fstate(1,i);
129         while j < M
130             if Fstate(j+1,i) > maxP
131                 maxP = Fstate(j+1,i);
132                 num = j+1;
133             end
134             j = j+1;
135         end
136         MLS(i) = num;
137     end
138
139     D = 0; % calculate discord
140     SD = 0; % calculate standard deviation of the mean discord
141     for i = 1:N
142         if MLS(i) == Ostate(i)
143             D = D + 1;
144             SD = SD + (D*D);
145         else
146             D = D - 1;
147             SD = SD + ((D-2)*(D-2));
148         end
149     end
150     D = 1 - (D/N);
151     SD = sqrt(SD)/N;
152 end
```

```
153
154
155 % function returns steady state vector V of a prob matrix A
156 function [V] = steady_state(A)
157
158     opts.v0 = ones(length(A),1)/length(A);
159     [V,~] = eigs(A,1,'LM',opts); % find eigenvector corresponding to
        eigenvalue 1
160     V = V/sum(V);
161 end
```

A.2 Phase diagram of n-state, n-symbol HMMs

The function Phasediagram() makes a 'phase diagram' of an $n \times n$ HMM; it determines the critical error probability b_c as a function of the transition probability a. It uses all functions from the previous section, except the main HMMdiscord(). The parameters to change to simulate a different HMM are very similar to those in the previous section:

- N, the length or number of timesteps of the generated HMM. This affects the accuracy of the calculated discord and the critical error probability.
- M (n in the text), the number of states and symbols of the HMM.
- Npoints, the number of points $\bar{a}$ and $\bar{b}$ simulated. This affects the number of points at which b_c is calculated and the stepsize with which b_c is determined.
- da (Δa), the asymmetry in the transition probabilities.
- db (Δb), the asymmetry in the observation probabilities.

These variables can be found in lines 4-8. The matrices A and B can be changed too, in lines 30 and 33.

```
1  % function to calculate the critical error probability b_c as a function of
       a for fixed da and db.
2  function Phasediagram()
3
4      N = 30000;
5      M = 2;
6      Npoints = 101;
7      da = linspace(0,0.5,6);
8      db = 0;
9
10     for i = 1:length(da)
11
12         fileID = fopen( sprintf('bc_M%d_N%d_da%1.3f_db%1.3f.dat', M, N, da,
               db), 'w');
13         criticalb(N, M, Npoints, da(i), db, fileID); % critical points b_c
               in M dimensions at fixed da and db
14         fclose(fileID);
15     end
16 end
17
18
```

```
19 % function that calculates b_c as a function of a, it uses functions from
       Appendix A.1
20 function criticalb(N, M, Npoints, da, db, fileID)
21
22     a = linspace(abs(da/2), (M-1)/M, Npoints); % values of a where to
           calculate b_c(a)
23     Dlst = zeros(1, Npoints+1);          % list to save D's
24     Dlst(1) = 0;
25     b = abs(db/2);                       % starting value of b
26     step = ((M-1)/M - abs(db/2))/Npoints; % stepsize in b
27
28     for i = 1:Npoints    % loop over values of a
29         j = 2;           % starting values for while loop
30         A = [1-a(i)+da/2, a(i)+da/2; a(i)-da/2, 1-a(i)-da/2];
31
32         while(b <= 1)    % try values of b until the b_c is reached
33             B = [1-b+db/2, b+db/2; b-db/2, 1-b-db/2];
34             [~, Ostate] = chain(M, N, A, B); % generate HMM
35             [Fstate] = filtering(M, N, A, B, Ostate); % apply filter
36             [Dlst(j),~] = discord(M, N, Ostate, Fstate); % calculate
                   discord
37
38             if Dlst(j) - Dlst(j - 1) > 0.001
39                 fprintf(fileID, '%1.6f %1.6f\n', a(i), b);
40                 b = max(b - 2*step, db/2);
41                 break;
42             else
43                 b = b + step;
44                 j = j + 1;
45             end
46         end
47     end
48 end
```

Appendix B

Calculation of the critical error probability

We used Mathematica to calculate the critical error probability analytically. In this appendix, we show the calculation for asymmetric 2×2 HMMs and symmetric $n \times n$ HMMs.

B.1 Asymmetric two-state, two-symbol HMMs

We start off by calculating the maximum confidence levels $p_1^* = P(x_t = 1|y^t = 1)$ and $p_{-1}^* = P(x_t = -1|y^t = -1)$, which are distinct for asymmetric transition and/or observation probabilities. We start from Equation 2.5 to calculate p_1^*. The calculation of p_{-1}^* is essentially the same, and we will not do it explicitly.

$$\begin{aligned} p_1^* &= P(x_t = 1|y^t = 1) \\ &= \frac{1}{Z_{t,1}} P(y_t = 1|x_t = 1) \sum_{x_{t-1}} P(x_t = 1|x_{t-1}) P(x_{t-1}|y^{t-1} = 1). \end{aligned} \tag{B.1}$$

We work out the normalization factor,

$$\begin{aligned} Z_{t,1} =& P(y_t = 1|y^{t-1} = 1) \\ =& \sum_{x_t} P(y_t = 1|x_t) P(x_t|y^{t-1} = 1) \\ =& \sum_{x_t} P(y_t = 1|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}|y^{t-1} = 1) \\ =& \left(1 - \bar{b} + \frac{\Delta b}{2}\right) \left[\left(1 - \bar{a} + \frac{\Delta a}{2}\right) p_1^* + \left(\bar{a} + \frac{\Delta a}{2}\right)(1 - p_1^*)\right] \\ &+ \left(\bar{b} + \frac{\Delta b}{2}\right) \left[\left(\bar{a} - \frac{\Delta a}{2}\right) p_1^* + \left(1 - \bar{a} - \frac{\Delta a}{2}\right)(1 - p_1^*)\right], \end{aligned} \tag{B.2}$$

and write out the equation in terms of $\bar{a}$, $\bar{b}$, Δa, Δb and p_1^*:

$$p_1^* = \frac{\left(1 - \bar{b} + \frac{\Delta b}{2}\right)\left[\left(1 - \bar{a} + \frac{\Delta a}{2}\right) p_1^* + \left(\bar{a} + \frac{\Delta a}{2}\right)(1 - p_1^*)\right]}{\left(1 - \bar{b} + \frac{\Delta b}{2}\right)\left[\left(1 - \bar{a} + \frac{\Delta a}{2}\right) p_1^* + \left(\bar{a} + \frac{\Delta a}{2}\right)(1 - p_1^*)\right] + \left(\bar{b} + \frac{\Delta b}{2}\right)\left[\left(\bar{a} - \frac{\Delta a}{2}\right) p_1^* + \left(1 - \bar{a} - \frac{\Delta a}{2}\right)(1 - p_1^*)\right]} \tag{B.3}$$

Isolating p_1^*, we find

$$p_1^* = \frac{\sqrt{\left(\bar{a}\left(-8\bar{b}+2\Delta b+6\right)-2(\Delta a-2)\bar{b}+\Delta a-2\right)^2-4\left(2\bar{a}-1\right)\left(2\bar{b}-1\right)\left(2\bar{a}+\Delta a\right)\left(2\bar{b}-\Delta b-2\right)}}{4\left(2\bar{a}-1\right)\left(2\bar{b}-1\right)} + \frac{8\bar{a}\bar{b}-2\bar{a}\Delta b-6\bar{a}+2\bar{b}\Delta a-4\bar{b}-\Delta a+2}{4\left(2\bar{a}-1\right)\left(2\bar{b}-1\right)}. \tag{B.4}$$

The only difference between p_1^* and p_{-1}^* is the sign of Δa and Δb.

Now we can work out the critical error probability. The conditions defining the critical error probability are

$$P(x_{t+1}=1|y_{t+1}=-1,y^t=1)=P(x_{t+1}=-1|y_{t+1}=-1,y^t=1) \tag{B.5}$$
$$P(x_{t+1}=-1|y_{t+1}=1,y^t=-1)=P(x_{t+1}=1|y_{t+1}=1,y^t=-1). \tag{B.6}$$

We need the solutions to both equations to find a complete set of solutions that agree with our simulations. We will work out Equation B.5 further. The left- and right-hand sides can rewritten using Bayes' theorem, the Markov property and marginalization,

$$\begin{aligned} P(x_{t+1}=\pm1|y_{t+1}=-1,y^t=1) &= \frac{P(y_{t+1}=-1|x_{t+1}=\pm1,\cancel{y^t=1})P(x_{t+1}=\pm1|y^t=1)}{P(y_{t+1}=-1|y^t=1)} \\ &= \frac{P(y_{t+1}=-1|x_{t+1}=\pm1)P(x_{t+1}=\pm1|y^t=1)}{\sum_{x_{t+1}}P(y_{t+1}=-1|x_{t+1})P(x_{t+1}|y^t=1)}. \end{aligned} \tag{B.7}$$

Also, we write out the upper left terms and the denominator,

$$\begin{aligned} P(x_{t+1}=1|y^t=1) &= \sum_{x_t}P(x_{t+1}=1|x_t)P(x_t|y^t=1) \\ &= (1-\bar{a}+\frac{1}{2}\Delta a)p_1^*+(\bar{a}+\frac{1}{2}\Delta a)(1-p_1^*) \end{aligned} \tag{B.8}$$

$$\begin{aligned} P(x_{t+1}=-1|y^t=1) &= \sum_{x_t}P(x_{t+1}=-1|x_t)P(x_t|y^t=1), \\ &= (\bar{a}-\frac{1}{2}\Delta a)p_1^*+(1-\bar{a}-\frac{1}{2}\Delta a)(1-p_1^*), \end{aligned} \tag{B.9}$$

$$\begin{aligned} P(y_{t+1}=-1|y^t=1) &= \left(\bar{b}-\frac{1}{2}\Delta b\right)P(x_{t+1}=1|y^t=1)+\left(1-\bar{b}-\frac{1}{2}\Delta b\right)P(x_{t+1}=-1|y^t=1) \\ &= \left(\bar{b}-\frac{1}{2}\Delta b\right)\left[(1-\bar{a}+\frac{1}{2}\Delta a)p_1^*+(\bar{a}+\frac{1}{2}\Delta a)(1-p_1^*)\right] \\ &\quad+\left(1-\bar{b}-\frac{1}{2}\Delta b\right)\left[(\bar{a}-\frac{1}{2}\Delta a)p_1^*+(1-\bar{a}-\frac{1}{2}\Delta a)(1-p_1^*)\right]. \end{aligned} \tag{B.10}$$

Then, we plug all these terms into Equation B.5,

$$\frac{\left(\bar{b}-\frac{1}{2}\Delta b\right)\left[(1-\bar{a}+\frac{1}{2}\Delta a)p_1^*+(\bar{a}+\frac{1}{2}\Delta a)(1-p_1^*)\right]}{\left(\bar{b}-\frac{1}{2}\Delta b\right)\left[(1-\bar{a}+\frac{1}{2}\Delta a)p_1^*+(\bar{a}+\frac{1}{2}\Delta a)(1-p_1^*)\right]+\left(1-\bar{b}-\frac{1}{2}\Delta b\right)\left[(\bar{a}-\frac{1}{2}\Delta a)p_1^*+(1-\bar{a}-\frac{1}{2}\Delta a)(1-p_1^*)\right]}$$
$$=\frac{\left(1-\bar{b}-\frac{1}{2}\Delta b\right)\left[(\bar{a}-\frac{1}{2}\Delta a)p_1^*+(1-\bar{a}-\frac{1}{2}\Delta a)(1-p_1^*)\right]}{\left(\bar{b}-\frac{1}{2}\Delta b\right)\left[(1-\bar{a}+\frac{1}{2}\Delta a)p_1^*+(\bar{a}+\frac{1}{2}\Delta a)(1-p_1^*)\right]+\left(1-\bar{b}-\frac{1}{2}\Delta b\right)\left[(\bar{a}-\frac{1}{2}\Delta a)p_1^*+(1-\bar{a}-\frac{1}{2}\Delta a)(1-p_1^*)\right]}.$$

Bringing everything to the left-hand side of the equation, reduces the equality to

$$\frac{\frac{1}{2}\left(2(\Delta b-1)\left(2p_1^*-1\right)\bar{a}+2\bar{b}-\Delta a\Delta b+\Delta a+\Delta b-2p_1^*\Delta b+2p_1^*-2\right)}{\left(\bar{b}-\frac{1}{2}\Delta b\right)\left[(1-\bar{a}+\frac{1}{2}\Delta a)p_1^*+(\bar{a}+\frac{1}{2}\Delta a)(1-p_1^*)\right]+\left(1-\bar{b}-\frac{1}{2}\Delta b\right)\left[(\bar{a}-\frac{1}{2}\Delta a)p_1^*+(1-\bar{a}-\frac{1}{2}\Delta a)(1-p_1^*)\right]}=0.$$

Plugging in p_1^* from Equation B.4 leaves us with an equation for $\bar{b}=\bar{b}_c$ in terms of $\bar{a}$, Δa and Δb. The resulting equation can be solved for $\bar{b}$ in Mathematica. The resulting solutions are lengthy and we will not give them here explicitly. One can repeat the same calculations for p_{-1}^* and the condition from Equation B.6 to find the full set of solutions.

B.2 Symmetric n-state, n-symbol HMMs

The calculation of the critical observation probability for symmetric $n\times n$, follows the same steps as the previous calculation. We, again, start by calculating the maximum confidence level from Equation 6.6:

$$p^*=\frac{(1-b)\left[(1-a)p^*+a\frac{1-p^*}{n-1}\right]}{(1-b)\left[(1-a)p^*+a\frac{1-p^*}{n-1}\right]+\frac{b}{n-1}\left[ap^*+\left(1-a+(n-2)\frac{a}{n-1}\right)(1-p^*)\right]} \tag{B.11}$$

We rewrite this in the form of a polynomial in p^* of degree two on one side and zero on the other side,

$$\left(abn^2-an^2+an-bn^2+bn+n^2-2n+1\right)p^{*2}+\left(-abn^2+an^2-a+bn^2-bn-n^2+2n-1\right)p^*$$
$$+(abn-ab-an+a)=0$$

We can solve this for p^* in terms of a, b and n. This leaves us with two solutions, but we are only interested in the solutions that takes on positive values for $0\leq a,b\leq 1$ and $n>1$ an integer. We find

$$p^*=\frac{1}{2(an-n+1)(bn-n+1)}\Bigg((a-1)(b-1)n^2+a+(b-2)n+1 \tag{B.12}$$
$$+\sqrt{\left((n-1)(bn-n+1)-a\left((b-1)n^2+1\right)\right)^2-4a(b-1)(n-1)(an-n+1)(bn-n+1)}\Bigg)$$

Now we have everything we need to calculate the critical error probability. We start from Equation 6.11,

$$P(x_{t+1}=1|y_{t+1}=2,y^t=1)=P(x_{t+1}=2|y_{t+1}=2,y^t=1). \tag{B.13}$$

Writing out the left- and right-hand sides of the equation,

$$\begin{aligned}P(x_{t+1}=1|y_{t+1}=2,y^t=1)&=\frac{P(y_{t+1}=2|x_{t+1}=1,\cancel{y^t=1})P(x_{t+1}=1|y^t=1)}{P(y_{t+1}=2|y^t=1)} \\ &=\frac{1}{Z_{t+1}}P(y_{t+1}=2|x_{t+1}=1)P(x_{t+1}=1|y^t=1),\end{aligned} \tag{B.14}$$

$$\begin{aligned}P(x_{t+1}=2|y_{t+1}=2,y^t=1)&=\frac{P(y_{t+1}=2|x_{t+1}=2,\cancel{y^t=1})P(x_{t+1}=2|y^t=1)}{P(y_{t+1}=2|y^t=1)} \\ &=\frac{1}{Z_{t+1}}P(y_{t+1}=2|x_{t+1}=2)P(x_{t+1}=2|y^t=1).\end{aligned}$$

Now, we write out the individual terms of the equation.

$$\begin{aligned} P(x_{t+1}=1|y^t=1) &= \sum_{x_t} P(x_{t+1}=1|x_t)P(x_t|y^t=1) \\ &= (1-a)\,p^* + (n-1)\frac{a}{n-1}\frac{1-p^*}{n-1} \\ &= (1-a)\,p^* + a\frac{1-p^*}{n-1}, \end{aligned} \tag{B.15}$$

$$\begin{aligned} P(x_{t+1}=2|y^t=1) &= \sum_{x_t} P(x_{t+1}=2|x_t)P(x_t|y^t=1) \\ &= \frac{a}{n-1}p^* + (1-a)\frac{1-p^*}{n-1} + (n-2)\frac{a}{n-1}\frac{1-p}{n-1} \\ &= \frac{a}{n-1}p^* + \frac{1-p^*}{n-1}\left(1-\frac{a}{n-1}\right), \end{aligned} \tag{B.16}$$

$$\begin{aligned} Z_{t+1} &= P(y_{t+1}=2|y^t=1) \\ &= \sum_{x_{t+1}} P(y_{t+1}=2|x_{t+1})P(x_{t+1}|y^t=1) \\ &= \frac{b}{n-1}P(x_{t+1}=1|y^t=1) + (1-b)P(x_{t+1}=2|y^t=1) + (n-2)\frac{b}{n-1}P(x_{t+1}=2|y^t=1) \\ &= \frac{b}{n-1}P(x_{t+1}=1|y^t=1) + \left(1-\frac{b}{n-1}\right)P(x_{t+1}=2|y^t=1) \\ &= \frac{b}{n-1}\left[(1-a)\,p^* + a\frac{1-p^*}{n-1}\right] + \left(1-\frac{b}{n-1}\right)\left[\frac{a}{n-1}p^* + \frac{1-p^*}{n-1}\left(1-\frac{a}{n-1}\right)\right]. \end{aligned} \tag{B.17}$$

After plugging all these terms into Equation B.13 and taking some steps to simplify, we have

$$\begin{aligned} &\frac{\frac{b}{n-1}\left[(1-a)\,p^* + a\frac{1-p^*}{n-1}\right]}{\frac{b}{n-1}\left[(1-a)\,p^* + a\frac{1-p^*}{n-1}\right] + \left(1-\frac{b}{n-1}\right)\left[\frac{a}{n-1}p^* + \frac{1-p^*}{n-1}\left(1-\frac{a}{n-1}\right)\right]} \\ &= \frac{\left(1-\frac{b}{n-1}\right)\left[\frac{a}{n-1}p^* + \frac{1-p^*}{n-1}\left(1-\frac{a}{n-1}\right)\right]}{\frac{b}{n-1}\left[(1-a)\,p^* + a\frac{1-p^*}{n-1}\right] + \left(1-\frac{b}{n-1}\right)\left[\frac{a}{n-1}p^* + \frac{1-p^*}{n-1}\left(1-\frac{a}{n-1}\right)\right]} \\ &\frac{\frac{bp^*+p^*-1}{n-1} - \frac{a(1+b)(np^*-1)}{(n-1)^2}}{\frac{b}{n-1}\left[(1-a)\,p^* + a\frac{1-p^*}{n-1}\right] + \left(1-\frac{b}{n-1}\right)\left[\frac{a}{n-1}p^* + \frac{1-p^*}{n-1}\left(1-\frac{a}{n-1}\right)\right]} = 0 \\ &\frac{(bp^*+p^*-1)(n-1) - a(1+b)(np^*-1)}{b\left[(1-a)\,(n-1)p^* + a(1-p^*)\right] + (n-1-b)\left[ap^* + (1-p^*)\left(1-\frac{a}{n-1}\right)\right]} = 0. \end{aligned} \tag{B.18}$$

We need to plug in the p^* that we just found (Equation B.12) and solve for $b = b_c$. This is quite hard to do for a general n. So, instead of directly finding the solutions for general n, we first calculated them for

$n = 2, 3, 4$ and 10,

$$\begin{aligned} b_c &= \frac{1}{2}\left(1 - \sqrt{1-4a}\right), & b_c &= \frac{1}{2}, \\ b_c &= \frac{1}{4}\left(2 + a - \sqrt{a^2 - 12a + 4}\right), & b_c &= \frac{2}{3}, \\ b_c &= \frac{1}{6}\left(3 + 2a - \sqrt{4a^2 - 24a + 9}\right), & b_c &= \frac{3}{4}, \\ b_c &= \frac{1}{18}\left(9 + 8a - \sqrt{64a^2 - 180a + 81}\right), & b_c &= \frac{9}{10}, \end{aligned} \tag{B.19}$$

respectively. We then wrote down the solutions for general n based on these solutions, and checked that it is indeed a solution of Equation B.18. In the end, we find the critical error probability b_c as a function of a and n,

$$b_c = \frac{1}{2(n-1)}\left((n-1) + (n-2)a - \sqrt{(n-2)^2a^2 - 2n(n-1)a + (n-1)^2}\right), \quad b_c = \frac{n-1}{n} \tag{B.20}$$