# Information-Theoretic Models of Social Interaction

**Christoph Salge**

A thesis submitted in partial fulfilment of the
requirements of the University of Hertfordshire
for the degree of

**Doctor of Philosophy**

The programme of research was carried out in the School of Computer Science,
University of Hertfordshire.

That's why it's always worth having a few philosophers around the place. One minute it's all Is Truth Beauty and Is Beauty Truth, and Does A Falling Tree in the Forest Make A Sound if There's No one There to Hear It, and then just when you think they're going to start dribbling one of 'em says, Incidentally, putting a thirty-foot parabolic reflector on a high place to shoot the rays of the sun at an enemy's ships would be a very interesting demonstration of optical principles.

**Terry Pratchett**, *Small Gods*

# Abstract

This dissertation demonstrates, in a non-semantic information-theoretic framework, how the principles of "maximisation of relevant information" and "information parsimony" can guide the adaptation of an agent towards agent-agent interaction. Central to this thesis is the concept of *digested information*; I argue that an agent is intrinsically motivated to a.) process the *relevant information* in its environment and b.) display this information in its own actions. From the perspective of similar agents, who require similar information, this differentiates other agents from the rest of the environment, by virtue of the information they provide. This provides an informational incentive to observe other agents and integrate their information into one's own decision making process.

This process is formalized in the framework of information theory, which allows for a quantitative treatment of the resulting effects, specifically how the digested information of an agent is influenced by several factors, such as the agent's performance and the integrated information of other agents.

Two specific phenomena based on information maximisation arise in this thesis. One is flocking behaviour similar to boids that results when agents are searching for a location in a girdworld and integrated the information in other agent's actions via Bayes' Theorem. The other is an effect where integrating information from too many agents becomes detrimental to an agent's performance, for which several explanations are provided.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

In nature there are numerous organisms that interact with others of their own kind, displaying a list of behaviours and abilities to specifically facilitate this interaction. This list includes diverse phenomena, such as imitation, learning, cooperation and coordination. Humans are no exception; arguably having the most complex and best developed forms of social interaction, including language, writing, mass media, etc.

The development of those phenomena has evidently been a long process of gradual change (Darwin 1859). In this dissertation I want to investigate if the perspective of information theory can offer new insights into how this development was motivated, i.e., what gradient may have guided the evolution of social interaction?

For several of the aforementioned abilities the benefits for an organism seem obvious. Social learning and imitation lead to faster acquisition of skills, writing allows us to transfer information through time and space, and coordination allows joint efforts that achieve what a single organism could not. But most of these "high level" concepts also include a number of "lesser" interaction abilities, such as

- the ability to differentiate other agents from the environment,
- directed attention towards other agents,
- understanding of one's own actions and consequences,
- understanding of other agents' actions and consequences.

This makes an evolutionary argumentation for the abilities that we observe at the end of this gradual adaptation process susceptible to the counter argument of irreducible complexity. The argument being that these complex abilities, those necessary to enable high level social interaction, could not have been the result of a single mutation, which then spread based on its fitness. To counter this argument it would be ideal if one could not only demonstrate the benefits of the final social interaction abilities, but also identify a gradient of step-wise development leading there, where each of the developmental steps is shown to be beneficial by itself. Therefore, this thesis focuses on early stages of social interaction and asks how the earlier steps towards the development of social interaction can be motivated?

In this thesis I want to look at this question from the information theoretic perspective, taking the stance that one of the fundamental properties of life is information processing. In itself, this is more a change of perspective than an insight into what a living system is. If we were to consider information processing as a process that causes two different random variables to be correlated, for example the sensor input of an animal and its actions, then the idea that life processes information is true, but trivially so. Numerous non-living processes could make this claim, and so the criterion would fail to exclude any non-living systems.

It is possible, though, to further refine the hypothesis of life as information processing. For this purpose, I adopt a hypothesis that has been brought forward already in early cybernetics and has been revived due to new evidence (Barlow 1959, Barlow 2001, Touchette and Lloyd 2000, Touchette and Lloyd 2004, Attneave 1954, Laughlin 2001, Bialek, Nemenman and Tishby 2001, Polani 2009) namely that organisms attempt to optimize their information processing; more precisely, organisms attempt to maximize the information attained relevant to their goals under the constraints of their particular sensorimotor (and neural) equipment. Regarding the evolutionary perspective, this includes the idea that organism do not necessarily adapt to solve a specific problem, but are optimized regarding relatively generic information theoretic principles, which then in turn enable them to perform well in different, concrete situations. The resulting organism then would not be "hard wired" with strategies to deal with concrete situations, but rather be able to "intelligently" adapt to different problems by acting or adapting according to more general principles. This should also make organisms more adaptable in general, as the informational efficiency of the organism provides an immediate (rather than delayed) gradient for the effectiveness of the organism's behaviour, prior to any evolutionary feedback from an external long-term pay-off.

In this dissertation I will focus on two candidate principles, which are discussed in more detail in (Polani 2009), namely, *maximisation of relevant information* and *information parsimony*. Both principles are taken as pragmatic assumptions in this thesis, and the focus will be to investigate where theses principles lead. While there is some evidence, which will be discussed in the related work section, that those principles are reasonable assumptions for the development or real biological organism, proving or disproving these claims is beyond the scope of this thesis. The only aim related to those claims is to demonstrate that some of the behaviour that results from these principles is similar to behaviour observed in biological organism, thereby supporting the idea that such ideas can indeed lead to more complex, life-like behaviour.

**Information Parsimony** Information acquisition and processing are found to be very expensive in terms of metabolic costs. Therefore it is sensible from an evolutionary standpoint to process the needed information with the least amount of effort, using only the resources necessary. As a corollary, it is evolutionary sensible to assume that a given organism would process as much relevant information (information needed to achieve certain life goals) as possible given a specific organismic sensorimotor equipment. If the organism could do with less relevant information, its information processing equipment can be expected to be selected against during evolution.

This is related to, but not the same concept as information limitation, where due to some physical constraint it is impossible to obtain or process more information. Of course, all agents are also subject to a form of information limitation, and this might lead to specific behaviours to cope with the limitation. But information parsimony is slightly stronger, suggesting that even within the bounds of limited information, obtaining information is costly; if the same pay-off can be achieved with less information, then the agent will adapt to use even less information.

**Maximisation of Relevant Information** In general, a specific amount of information about the environment is necessary for an organism to select the best available action. Therefore, it is generally a good strategy to develop the ability to a.) determine where the relevant information is located and b.) to process this information so it can positively influence an agent's actions. Note here, that his principle uses the notion of "Relevant Information" as defined in Chap. 3. The definition of relevant information as the minimal mutual information over all optimal strategies implies that there is indeed an upper bound for the amount of relevant information. If all relevant information has been obtained, then all additional information is either redundant

or not relevant for choosing an action. So maximisation of relevant information is different from just maximising information intake for an agent.

There are indications that immediate sensorimotor efficiency already provides powerful local gradients for adaptation and evolution (Klyubin, Polani and Nehaniv 2005b, Klyubin, Polani and Nehaniv 2007, Der, Steinmetz and Pasemann 1999, Ay, Bertschinger, Der, Güttler and Olbrich 2008, Sporns and Lungarella 2006, Prokopenko, Gerasimov and Tanev 2006), where those simple, information theoretic principles already generate behaviour similar to those in simple, biological agents. In (Polani 2009) this is further developed into a hypothesis that these, and some other principles described in the paper, are not just descriptive of the properties that organism have acquired through the process of evolution, but that some organisms have adapted as to actively improve their information processing in line with these principles. This way, information theoretic principles could act as a stepping stone in the evolutionary process. Rather than adapting to solve a very specific goal agents could adapt to deal "better" with information, and thereby become more proficient at dealing with and adapting to the world in general.

In this context, and to further support this hypothesis, this dissertation investigates whether and how the previously mentioned principles of information parsimony and maximisation of relevant information can lead to agent-agent interaction. Furthermore, I want to inquire what social phenomena similar to those of biological agents can possible arise from said principles?

To address this, I will assume a slightly more specific criterion, namely that the agents are interested in maximizing the relevant information about a life goal (e.g. the location of food). For this, the agents will have the possibility to detect the food directly or to observe the behaviour of other agents. Our study will investigate whether and how, under these circumstances, social interaction can emerge simply from the immediate drive to maximize relevant information.

To implement a quantitative and consistently informational model, I will use an approach based on Shannon's Information Theory, Bayesian Modelling and Causal Bayesian Networks. Within this information theoretic framework, our agents will build their behaviours from the starting point of quite restricted assumptions; in particular, no *a priori* social dynamics will be assumed. Based on the information theoretic framework and using a few assumptions about the world the agents live in I will then introduce the concept of Digested Information; this will serve as an argument for why the actions produced by one agent might be of particular interest for another similar agent, even if those actions have no direct consequences for the other agent, meaning there are no joint pay-off matrices and the agent's actions do not affect other agents' performance.

I will support these conceptual arguments by presenting simulations that support the plausibility of the previous argument by providing quantitative data in line with the argument's predictions. In addition, this will also demonstrate that the information-theoretic framework allows us to quantify the concrete benefit of observing another agent.

## 1.2 Research Questions

The general direction of this dissertation can be summarized by the following two research questions:

1. Does the optimization of information processing lead to agent-agent interaction?
2. What insights can the analytical framework of information theory provide into agent-agent interaction?

While both questions are closely linked they demarcate nicely what one can take away from this dissertation depending on one's own research interests. The first question is more relevant to the artificial life community, which tries to understand life-like systems, and ideally wants to replicate them. A major focus in this field is the creation of complex behaviours or structures from simple principles or rules. So, while a lot of social behaviours can much easier be produced by some dedicated development towards this behaviour, there is an interest to create a whole range of such behaviours from the same simple principles, building agents from the "bottom up".

The second question is more related to the natural sciences, where actual social behaviour is studied. This thesis includes the development of several analytical tools, which can also be applied to real world biological systems, and could therefore be helpful to understand actual biological life better.

## 1.3 Overview

The dissertation will be organized as follows:

**Chapter 2** introduces the notation used in the dissertation and specifies the information theoretic model this thesis operates in. It also contains a review of the literature related to this work, specifically in the areas of "Information and Cognition", "Social Bayesian Learning" and "Game Theory".

**Chapter 3** introduces Polani et.al.'s concept of *Relevant Information*(RI). In this chapter I derive and discuss some of the essential properties of RI and demonstrate how the RI trade-off curve of a specific scenario can be obtained by an adaptive process. Furthermore, I introduce a distinction between general relevant information, and the relevant information in an agent's sensors. This leads to the introduction of unique relevant information, which is a formalism that allows us to quantitatively measure how much relevant information is contained in a specific part of the sensor input.

**Chapter 4** contains the *Digested Information* argument, where I explain why any agent that has to process information from the environment in order to perform well also has to display this information in its actions. I also introduce measurements to quantify different kinds of digested information, and discuss several factors that influence how much relevant information is encoded in an agent's action. This chapter also contains the analysis of two different simulations that demonstrate the digested information's effects, such as how the performance of an agent increases the relevant information in an agent's actions.

**Chapter 5** contains several different quantitative analyses that study what happens when an agent incorporates the digested information of other agents into its own internal model by using Bayes' Theorem. Even if only one agent uses the information of others, it is possible that this agent's selective observation of other agents close to it introduces a conditional dependence between different observed actions, which in turn violates one of the basic assumptions of the employed *Naive* Bayesian Update. Furthermore, too much information can destroy the information gradient used by the infotaxis search. The simulations where all agents observe each other also show evidence of an information cascade, where possibly misleading information is propagated through the agent population. Finally this chapter also demonstrates how incorporating the relevant information of others is another factor that changes the relevant information an agent provides itself.

**Chapter 6** demonstrates how boids-like flocking behaviour can result from the principle of information maximisation. Infotaxis search, combined with incorporating the digested information of other agents leads to agent flocking in the previously discussed grid-world scenario. This illustrates how a different agent-agent interaction phenomenon that is also present in nature can arise from the same information-theoretic principles as used in the previous chapter.

**Chapter 7** concludes the dissertation, and connects the conclusions of the different chapters into an overall argument, outlining why the basic principles of information maximisation and information parsimony can indeed create a gradient for gradual adaptation towards social interaction.

## 1.4 Contributions of the Thesis

- The development of an approximation strategy for the relevant information function for the dual constraint (relevant information / information parsimony) optimizing agent based on a genetic algorithm and a neural network which allows us to approximate relevant information in more complex scenarios. Application of said approximation to different complex scenarios, including an analysis relating the relevant information trade-off graph to some essential properties of the scenario it was derived from.

- Extension of the relevant information framework with a formalism for *unique relevant information*. This makes it possible for the agent to determine how much relevant information is contained in a subset of said agent's sensor input. Compared to the overall bandwidth it also provides a notion of "information density" which is helpful to guide adaptation based on information parsimony.

- Development of the Digested Information Concept, an argument why agents that need to obtain information from the environment in order to perform well have to display that information in their actions. This includes the development of measures necessary to quantify this effect, and an analysis of different factors, such as performance, that influence the provision of digested information. In general, this provides an argument why agents can act as pre-processors of relevant information for other agent with similar goals, which in turn provides an argument for the existence of an informational gradient for the adaptation of basic social interaction, such as attention to other agents.

- Implementation and analysis of two multi-agent models to evaluate the Digested Information Concept. This includes the adaptation of the continuous infotaxis formalism to a discrete grid world, and an extension of infotaxis to incorporate different temporal horizons for expected information gain.

- Application of the single-symbol information gain formalism to Social Bayesian learning in a grid world search task. Includes the detection of an effect where the incorpo-

ration of two different sources of information (other agents and environment) changes the internal prior systematically, so that one source make the agent less certain on average.

- Analysis of information cascade behaviour in regard to how it affects the provided digested information. Specifically, I demonstrate how the transition of the agent population into an information cascade moves their strategies away from the trade-off curve between performance and efficient information processing.
- Generation of boids-like flocking behaviour based on the principle of maximising relevant information, demonstrating how the optimization of information processing can lead to coordinated behaviour.

# Chapter 2

# Background and Related Work

The following chapter will provide the necessary background for the later chapters, and it will also give a general overview of the related work. This chapter includes definitions for the key terms used in this thesis, outlining what I understand by *information*, *information theoretic model* and *social interaction.* Furthermore, while reviewing the existing related work, I will argue why it is advantageous to employ the information theoretic perspective, presenting the key benefits of this approach.

## 2.1 Information

The central concept of this thesis is *information*, which will be formally defined as some variant of mutual information in classical information theory. I will give a short historical overview of the development of information theory to illuminate some of the implicit assumptions in regard to sender and receiver. I will then introduce the formal basis for information theory, including the notation used in this thesis. Based on the information theoretic formalism I will then define the term information as used in this thesis.

There are several common concepts using the term "Information"; a hierarchical overview based on their different properties can be found in an overview by Floridi (2011). To avoid confusing the reader I will differentiate the definition of information used in this thesis against other definitions by discussing some of its basic properties, namely observer independence and being non-semantic in nature. This will also help to argue against the notion that the absence of sender and receiver poses a problem in applying information theory to natural systems, which was raised by Gibson (1986).

### 2.1.1 Development of Information Theory

Information Theory has undergone several stages of development in which its scope and applications have changed considerably. The core elements of Information Theory were developed by Claude Shannon to deal with the limitations of transatlantic communication (Shannon 1948). His initial work, and the context it is applied to is well characterized by the title "A Mathematical Theory of Communication" (MTC).

> "Information theory answers two fundamental questions in communication theory: What is the ultimate data compression? (answer: the entropy $H$), and what is the ultimate transmission rate of communication? (answer: the channel capacity $C$)"(Cover and Thomas 1991)

But Thomas and Cover, and others, go on to argue that MTC has applications beyond the standard problems of communication theory.

The mathematical theory of communication formalises the fundamental limitations of any kind of communication channel. If the minimal encoding of a message has more bits than the available amount of transmission bandwidth, or the amount of storage (as storing information is transmitting a message through time), then perfect communication is not possible. But the general mathematical formulation of information theory based on random variables allows the application of those upper and lower bound considerations to more than just humans sending messages to each other. Shannon already argued that other natural processes, such as music or speech (Shannon 1951), have a certain irreducible complexity, a property he named *entropy*, or later, *self-information*.

Once the basic parameters of a given system are formalized in random variables, MTC can be used to illustrate the fundamental limitations of a diversity of systems. A common application is an argument that something is impossible to do, because the amount of necessary information that would need to be transferred to achieve a specific objective exceeds the channel capacity of the channel used for this transfer. This general idea leads to a wider application of MTC, where the information theoretic limitations of different systems were studied (Touchette and Lloyd 2000). One motivation was to evaluate how well a technical solution would approximate the theoretically achievable optimum. But it became clear that the same idea could also be applied to the study of natural systems, such as the replication process of genetic code (Prokopenko, Polani and Chadwick 2009), or the efficiency of animal communication (McCowan, Hanser, Doyle et al. 2004).

### 2.1.2 Formalism

Information theory (Shannon 1948, Cover and Thomas 1991) in a formal sense can be applied to any set of random variables. I denote random variables with capital letters, and the states they can assume with lower case letters. Let $X$ be a random variable that can assume the states $x$, where each state $x$ is an member of the alphabet $\mathcal{X}$. Then $P(X)$ is the probability distribution of $X$, and $P(X = x)$ is the probability that $X$ assumes the value $x$. This will also be denoted as $p(x)$.

With this notation information theory defines the *entropy* of a random variable $X$ as

$$H(X) = -\sum_{x \in \mathcal{X}} P(X = x) \log P(X = x). \tag{2.1}$$

This is often described as the uncertainty about the outcome of $X$, the average expected surprise, or else the average information gained if one was to observe the state of $X$, without having prior knowledge about $X$. The entropy has a number of important properties. Among others, the *a priori* uncertainty (i.e. entropy) is larger if the outcomes are more evenly distributed than if the outcomes are more concentrated on a particular value - in other words - concentrated values are easier to predict than uniformly spread ones.

Consider two jointly distributed random variables, $X$ and $Y$; then we can calculate the *conditional entropy* of $X$ given a particular outcome $Y = y$ as

$$H(X|Y = y) = -\sum_{x \in \mathcal{X}} P(X = x|Y = y) \log P(X = x|Y = y). \tag{2.2}$$

This can be averaged over all states of $Y$, resulting in the conditional entropy of $X$ given $Y$,

$$H(X|Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \sum_{x \in \mathcal{X}} P(X = x|Y = y) \log(P(X = x|Y = y)). \tag{2.3}$$

This is the entropy of $X$ that remains, on average, if $Y$ is known. So $H(X)$ and $H(X|Y)$ are the entropy of $X$ before and after we learn the state of $Y$. Thus, their difference is the amount of information we can learn, on average, about $X$ by knowing $Y$. Subtracting one from the other, we get a value called *mutual information*,

$$I(X;Y) = H(X) - H(XY). \tag{2.4}$$

The mutual information is symmetrical (Cover and Thomas 1991) and measures the

amount of information one random variable contains about another (and vice versa, by symmetry). Also, note that I use the binary logarithm for all log(.) operations, so all information measurements are in *bits*.

### 2.1.3 Definition of Information

Based on the formalism introduced in the last section, *Information* will be defined as the mutual information between two random variables. If I say that one variable contains information about another, I mean that the mutual information between those two variables is larger than zero. Furthermore, if one variable $X$ is said to contain a certain amount of information, this then refers to the mutual information with itself, $I(X;X)$. This is also often called self-information, and is numerically identical to the entropy of $X$, since

$$I(X;X) = H(X) - H(X|X) = H(X) - 0 = H(X). \tag{2.5}$$

#### Properties of Information

Some confusion regarding the properties of information, as defined in this thesis, results from the communication model presented in the original paper, and the implicit assumptions it introduced. Shannon defined a communication system as essentially consisting of the following five parts:

1. Information Source
2. Transmitter
3. Channel
4. Receiver
5. Destination

Communication is considered successful if the destination can reconstruct the state of the information source. The channel is defined by a distribution of the output states for every possible input state, a conditional distribution. The transmitter and receiver are also conditional distributions that map the information source to the channel input, and the channel output to the destination variable. It is assumed here that those mappings can be changed in order to optimize the use of the channel.

### Observer Independence

In the classical interpretation this usually carries an implicit assumption about involved agents. Either a sender and receiver who both know the channel, and agreed on an encoding and decoding scheme to use the channel efficiently, or in the case of a more technical application, an external agent who knows the channel distribution and engineers a transmitter and receiver to optimally use the channel. These assumptions limit the generalized application of the mathematical theory of communication. Gibson (1986) for example is sceptical, and argues that there are in general no intentional senders and receivers in nature. But the formalism of information as mutual information does not necessarily need these assumptions, as a simple example illustrates.

It is generally assumed that the number of tree rings correspond to the age of a tree. While there might sometimes be deviations from this rule, I think it is safe to say that there is a high correlation between the number of tree rings and the age of a tree. It follows that there is mutual information between the number of tree rings, and the age of the tree, hence one contains information about the other. For an agent to use this information, i.e. to determine the age of a tree, the agent would have to know about this relationship. The agent would have to understand the conditional distribution of the channel. But, even if no agent would know, even if there were no humans, the tree rings would still contain this information.

So we see that the term information defined as mutual information is *observer independent*, meaning that the value of information is not dependent on a specific observer, nor is it measured from the perspective of a specific observer. The idea of measurable information does imply that there is some model or other way to conceptualize the world, and I will assume for my arguments that such a model exists, even if it is not necessarily accessible to the agent.

To clarify this it might be helpful to use the terminology used for signs, where each sign has a signifier, an object and an interpretant. For a proper sign a signifier has to be about an object, and refer to it, and an interpretant that understands the relation between the last two. Information, as used here, differs from this as it is fully determined by the relation of the two variables, but does not require an interpretant.

Information could in theory also be defined for an observer. It would be possible to assume that an agent knows only to a certain degree how two variables are related. In this case one would ask how much information does one variable give the agent about another variable. This will be studied in more detail later in this thesis, but it is not what is quantified by mutual information, and therefore is not included in the concept of information as used in this thesis. In this definition variables can contain information

about another regardless whether someone can access it or not. Therefore, the definition of information presented here should be applicable even if there are no intentional senders or receivers.

### Non Semantic

Information, as used in this thesis, is considered to be non-semantic. There have been attempts to extend information theory, or build upon it, in order to attach some semantics to Shannon information (Dretske 1981), but mutual information in itself does not have a semantic interpretation, nor does it requires any semantics to evaluate.

Without going fully into what exactly is meant by semantics, and meaning, (a more extensive account can be found in (Floridi 2011), some simple examples already demonstrate that mutual information lacks already the basic properties for semantics. For one thing, it is, as Dretske calls it, "an argument by amount". Mutual information only answers the question of "how much" information is present, but does not address what this information *means*, or what this information *is*. All that mutual information returns is a numerical value.

Furthermore, all basic properties of information theory (entropy, mutual information, channel capacity) only depend on the probability distributions of the random variables involved, and not on any of their actually assumed values. So, even if there was some meaning attached to the specific state one of those variables could assume, then information theory would not treat this state any differently because of it.

While the formalism of information theory is unable to deal with any form of semantics, it should also be noted, that it might still be possible to gain insights into those fields, by using the tools provided by information theory. This thesis deliberately does not venture into the rich field of philosophical discussion surrounding the concept of representation, but there is a certain proximity to the idea of biosemantics (Millikan 1989) in the later chapters of this work. One central question regarding representations is how they gain the property of "intentionality" or "aboutness" regarding the thing they represent. Millikan argues that representations are the result of functions that adapted through an evolutionary process. In this process certain producers developed functions that would produce representations which would both contain a fact about the world but also an implicit directive to action, while consumer mechanism adapted to use these representations to their benefit. The meaning of those representations then is identical to the functions they fulfil.

In the later parts of this thesis I will make a slightly different argument, namely that agent's actions contain information because the agent adapted to act according to its environment, and thereby also adapted to encode specific valuable information about that

environment; not because it was interested in passing this information on, but because it has to display this information in its actions in order to act correctly. The adaptation of observers on the other hand invokes similar arguments to those used to argue for the adaptation of consumer mechanism, but does not require a co-evolution, as the development of the information display in the actions is self motivated.

In any case, the concept of information does not contain any semantics by itself.

## 2.2 Information Theoretic Model

### 2.2.1 Causal Bayesian Models

To model the causal structure connecting the random variables Causal Bayesian Networks (CBN) are used (Pearl 2000). A CBN is a directed, acyclic graph, in which the nodes represent random variables. The directed edges represent conditional probability distributions.

A CBN has the following property. Let $G = (V, E)$ be a directed, acyclic graph, and $X$ is a set of random variables indexed by $V$, and $x_{pa(v)}$ are defined as the states of the parent nodes of $x_v$. Then the probability for the overall system to assume the state $x$ is

$$p(x) = \prod_{v \in V} p(x_v | x_{pa(v)}). \tag{2.6}$$

From this follows the so called "causal Markov property", formalized as

$$X_v \perp\!\!\!\perp X_{V \setminus de(v)} | X_{pa(v)}, \tag{2.7}$$

where $de(v)$ indexes all those nodes that are descendants of $X_v$, and $V \setminus de(v)$ are all those nodes that are not descendants of $X_v$. This means any variable in a CBN is statistically independent of all its non-descendants if conditioned on its parents. Or, more informally, knowing the states of a variable's parents tells us all there is to know about that node; there is nothing else influencing it in the graph. All the descendants can be considered to assume their state "later" than $X_V$, and therefore have no influence at all.

Pearl describes how a CBN can be constructed for a set of random variables, given either a joint probability distribution, or sufficient statistics to construct such a distribution. The resulting CBN might not be unique, though. But if it is possible to intervene at any random variable at will, then a unique CBN can be constructed; one that, as Pearl argues, models the causal structure of the variables.

For specific computer simulations it is rarely necessary to reconstruct the CBN from

data, since looking at the implementation usually reveals which parameters influence what other parameters. On the other hand, if one wants to apply a CBN model to a real-world scenario, then it is necessary to reconstruct from statistics. Intervention can still be avoided in many cases if additional context can be used, such as the fact that a later event cannot causally influence an earlier event.

For the arguments in this thesis it is also secondary how the CBN is determined, it only matters that the system in question can be modelled by a CBN. Keeping in mind that even this, in general, is contested (Spohn 2000), I want to make clear that for the models we are looking at it is assumed that their relevant properties can be modelled with a CBN.

### 2.2.2 Perception Action Loop

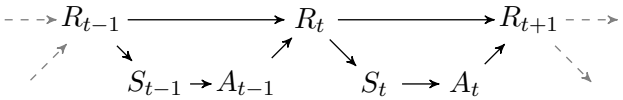
A possible way to model an agent's interaction with the world is the perception action loop (PAL). The PAL has been used as a model in various previous work, and all the models in this thesis can be formalized as PALs.

A simple PAL is a CBN consisting of three random variables, or sets of random variables, which will be labelled as $A$ (actuators), $S$ (sensors) and $R$ (rest of the world). Fig 2.1(a) shows this loop unrolled in time. The arrows make it clear that the sensors get influenced by the rest of the world, which in turn then influence the agent's actions. The next step of the environment then depends on the previous environment and the action of the agent. This clearly separates how information can get in and out of the agents. Influence from the agent on the environment has to go through $A$, and information from the environment to the agent has to go through $S$.

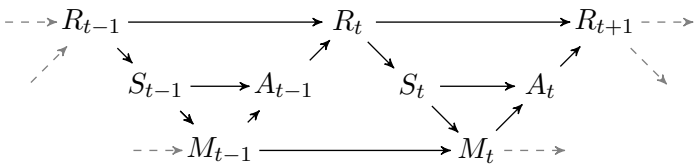
The agent's strategy, or decision making, is represented by the ability of the agent to change the conditional probability $P(A|S)$. If there is some dependence between $A$ and $S$, i.e. $I(A;S) > 0$, then I will call the agent *reactive*. The agent in this case processes information from its sensors and acts accordingly.

Fig. 2.1(b) shows a modification of this model by adding another random variable, called $M$, for memory. $M$ influences the agent's actions and is in turn influenced by previous states of the memory and by the sensors. This allows the agent to react to information from an earlier point in time or to aggregate information. An agent without such a variable is called *memoryless*, and can only react to the current sensor input.

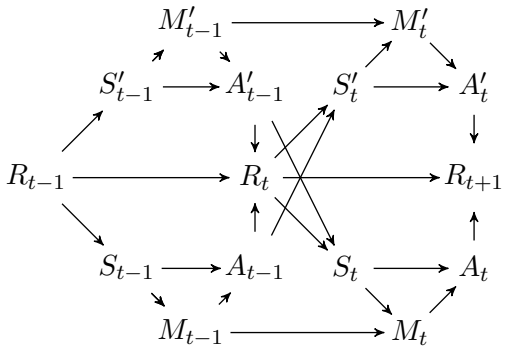
In case the agent has an internal memory, the agent's control over its behaviour then extends to how its internal state is influenced by its sensors, and in turns influences the agent's actuators. The agent, in general, has no control over how its actions affect the rest of the world, or how the world affects its sensors, i.e. the agent cannot change the




(a) Memoryless Perception Action Loop




(b) Perception Action Loop with Memory




(c) Two-Agent Perception Action Loop

Figure 2.1: Causal Bayesian network of the perception-action loop, unrolled in time, showing (a) a memoryless model, (b) a model including agent memmory, and (c) a model containing two agents.

conditional probabilities $P(R|A)$ and $P(S|R)$.

Similar models have been used in a variety of other scientific work (Capdepuy 2010, Klyubin et al. 2007, van Dijk, Polani and Nehaniv 2010). Most of the further related work either explicitly or implicitly assumes this model. The general arguments made in this thesis pertain to this model, and should therefore apply to those cases which can be captured by a PAL. The actual simulation models are more specific, but can be expressed in terms of a PAL with memory.

The perception action loop is closely related to the concept of Umwelt by von Uexküll

(1909). The Umwelt of an agent is all that the agent can interact with and perceive, those things that the agent causally interacts with. Through actions the agent shapes and changes its own Umwelt. Specifically, von Uexküll also introduces the idea of circular interaction with the Umwelt, where an agent effector would influence the Umwelt, which in turn would lead to different experiences for the agents receptor. More modern work (Capdepuy, Polani and Nehaniv 2007a) also relates the concept of Umwelt to information-theoretic studies of the perception action loop.

### 2.2.3 Agent Interaction

To deal with several agents in a perception action loop framework I will assume that the variable $R$ can be further decomposed into another agent and the remaining environment. The CBN seen in Fig. 2.1(c) captures this more closely. In general, this is still an agent interacting with the world, but the world now also contains another agent. This works well with our initial question of how to distinguish an agent from the world, because it sets up the random variables pertaining to an agent as just being part of the environment of another agent. So, initially, there is nothing special about the random variables belonging to another agent, compared to those belonging to the remaining environment.

Coming back to another major part of this dissertation, I need to formalize social interaction. Because of the non-semantic nature of the underlying information concept it is difficult to formalize social interaction as more than one agent reacting to the actions of other agents. Therefore, *Social Interaction* will be defined as statistical dependence between one agent's actions and another agent's actions. Or, more formally, as a non-zero mutual information between two agent's action variables. While this is not very helpful to differentiate between them, it at least captures most basic forms of social interaction, such as learning, coordination, cooperation and imitation. It does have the problem that it would also capture common-cause reasons for mutual information, where both agents act similarly because of similar observations in the past. It might be undesirable to classify this as social interaction, but in this case one could utilize some measure of information flow (Ay and Polani 2008) to differentiate further.

To summarize: the above described framework, consisting of random variables which form a perception-loop with the environment, constitutes the information-theoretic model in which I am going to ask how agent-agent interaction can be motivated through information theoretic constraints and optimization of specific information theoretic measures. The next section will review the related work regarding information and cognition, and discuss some of the existing formalisms for information theoretic behaviour generation.

## 2.3 Related Work

### 2.3.1 Information and Cognition

Information processing is a necessary requirement for life; an organism that wants to react to its environment has to first acquire this information. The law of "requisite variety" (Ashby 1956, Touchette and Lloyd 2000, Touchette and Lloyd 2004) formalizes this, and shows that the control of an organism is limited by the amount of information it has obtained. Already at that time is has also been suggested that information plays an important role in understanding cybernetic systems, and that information theory is a suitable way to gain quantitative insights (Barlow 1959, Attneave 1954).

In this context Polani's "Currency of Life" (Polani 2009) offered the hypothesis that the adaptive process that enables organisms to deal with the environment might not only be driven directly by the optimization of performance, but includes as an intermediate step the optimization of the agent and the agent's behaviour in regard to some information theoretic principles. This would increase the adaptability of the agent immensely, as it would not just rely on external reward functions, but could be supported by some agent-internal information gradient. The general idea here is that an agent would adapt to optimize its information processing, and this in turn would allow the agent to deal with a wide variety of problems. A good example here is Lizier's work (Lizier, Prokopenko and Zomaya 2008a, Lizier, Prokopenko, Tanev and Zomaya 2008b), where he analysed the control of a snakebot. Applying a genetic algorithm to optimize the snakebot in regard to its achieved forward momentum also increases the information transfer between the different actuators of the agent. The second value could be measured by the agent internally, and could then be optimized, which in turn might cause the snake bot to move faster.

To support the hypothesis that the adaptation of life is driven by informational principles it would be useful to demonstrate how real world biological phenomena can be reproduced from some of those principles. This artificial life approach (Adami 1998) would increase the plausibility of such principles. The two principles I want to focus on here specifically, are the maximisation of *relevant information* and *information parsimony*.

#### Information Maximisation

Following from the law of "requisite variety" (Ashby 1956) it becomes clear that a certain amount of information is necessary to perform a specific task. This leads to the question of how much information is necessary to perform optimally for an agent? This has been

formalized in the concept of *Relevant Information* (Polani, Martinetz and Kim 2001, Polani, Nehaniv, Martinetz and Kim 2006), which will be introduced in more detail later. In short, relevant information is the amount of information an agent needs to perform optimally, which is also the mutual information between the actions of an agent $A$ and the environment $R$.

Since it is necessary for an agent to obtain and process relevant information, this general idea motivates several optimizations regarding the agents information processing. For example, Linsker's "Infomax" (Linsker 1988) optimizes a multi-layer neural network so it preserves the maximum amount of mutual information between the layers. Other strategies include the maximisation of information intake. A biologically relevant example, which will be discussed in more detail later, is "infotaxis" (Vergassola, Villermaux and Shraiman 2007), where an agent chooses its actions to maximise the expected gain in entropy reduction in regard to some relevancy variable. Vergassola demonstrates that this leads to the reproduction of the idiosyncratic flight patterns used by moths trying to find mating partners.

#### Information Parsimony

The second principle is based on the physically motivated idea that processing information is work in itself and requires energy and resources expenditure (Polani 2009, Laughlin 2001). An organism therefore should only process information that is necessary and should optimize its information processing so that the necessary processing gets done with a minimum of informational cost. One exemplary biological inspiration for this idea is the sensor degradation observed in the eyes of animals that have no, or very little, exposure to light within their lifetime (Jeffery 2005).

A formalism which combines both principles is the information bottleneck approach (Tishby, Pereira and Bialek 1999) where a random variable $X$ is mapped to another variable $Y$, maximising the mutual information $I(Y;Z)$ to a relevancy variable $Z$, while at the same time minimizing the mutual information $I(X;Y)$. This both maximizes the relevant information $Y$ contains about $Z$, but at the same time it keeps the information processing from $X$ to $Y$ to a minimum.

### 2.3.2 Embodied and Situated Cognition

A major paradigm shift in artificial intelligence and studies of cognition has been brought about by the idea of embodiment and situated cognition (Varela, Thompson and Rosch 1992, Almeida e Costa and Rocha 2005). Capdepuy (2010) argues that the information

theoretic framework in general, and the perception action loop in particular, are well suited to model this paradigm. The CBN offers a natural decomposition where it is clear:

- which parts of the simulation are controlled by the agent's strategy (the mappings between the sensors and the actuators, and all mappings involving the memory)
- which mappings represent the embodiment of the agent and define how it interacts with the world (the mapping from the world to the sensors, and the mapping from the actuators to the world)

Also, since all information about the world has to pass through the sensor variable, it is easy to ensure that the agent can only act on information it obtained itself. One of the challenges associated with this shift is to figure out how an agent can make sense of its environment and act intelligently, when nothing is initially known about the world. A good example to illustrate this problem is the scenario by Bongard, Zykov and Lipson (2006), where an AI is placed inside an unknown robot body, and has to figure out how to control the body, and derive its basic configuration.

Recent research demonstrated that the information theoretic framework is well equipped to deal with this, especially in the area of sensor evolution and adaptation. Philipona, O'Regan and Nadal (2003) describes a scenario, where initially all the agent gets in terms of sensor input is a sequence of binary data. They demonstrate that it is possible to derive the dimensionality of the world the agent is situated in. Furthermore, Olsson demonstrates that, if separate sensor inputs can be identified, then it is possible to use the information distance between them to determine the configuration of a visual field (Olsson, Nehaniv and Polani 2004), or to derive the relationship between different actuators and sensors on a robot (Olsson, Nehaniv and Polani 2006). Furthermore, Salge and Polani (2009) demonstrated that hierarchical clustering based on the information distance, and subsequent bottleneck-like mapping of the clustered variables is able to extract salient features of the environment, such as dominant line structure, and regions of increased activity. All those applications can be done from the agent's internal perspective, and do not require meaning associated with the sensor input.

In terms of behaviour generation information theory has been successfully applied to an area called *guided self-organization* (Prokopenko 2009). The general idea here is to find generic, agent-internal principles that can be used to generate behaviour independent of specific agent goals. The information theoretic measure of empowerment (Klyubin et al. 2005b) for example measures the channel capacity between an agent's actions and its sensors. This is interpreted as a measure of how much reliable control an agent has over the world it can perceive. By choosing actions that increase empowerment an agent strives

to get to a position in the world where it has the largest effect on and most control of it. Unempowered states, such as death, are to be avoided. Even without a goal, this gives an agent an internally measurable utility function to guide is behaviour. Interestingly, the behaviour resulting from this one generic formalism corresponds to behaviour that seems reasonable in a lot of different scenarios. It causes agent to go to central points in a maze (Klyubin, Polani and Nehaniv 2005a), balances pendulums (Salge, Glackin and Polani 2012, Jung, Polani and Stone 2011) and even generates some form of collective behaviour (Capdepuy, Polani and Nehaniv 2007b) and coordination (Capdepuy, Polani and Nehaniv 2011). Similar successes have been achieved with "Predictive Information" (Ay et al. 2008) and "Homeokinesis" (Der et al. 1999), where other information-based measures were used to generate agent behaviour.

In summary, the idea that basic cognition can be understood in terms of information processing, and that basic adaptation can be guided by informational principles is well established. In this dissertation I want to build on this work, and explore whether the principles applied here are sufficient to generate agent-agent interaction.

### 2.3.3 Information and Social Interaction in Nature

The idea that the interaction between biological agents is related to the processing of information (the term information being used in a more general, commons sense way), has been well established. A classical example is the work of Ward and Zahavi (1973), which details for birds how communal roosting is beneficial, as other birds or aggregation of birds can provide important information regarding food, predators, etc. Even more closely related is Danchin's idea of inadvertent social information (Danchin, Giraldeau, Valone and Wagner 2004), in which he stipulates that agents are encoding information into their actions without a specific intend to communicate. This is then later supported by empirical studies of the effects of inadvertent social information in different animals (Parejo, Danchin, Silva, White, Dreiss and Avilés 2008, Baude, Dajoz and Danchin 2008).

There is in fact, as pointed out by Call and Carpenter (2002), a long list of research that uses information or related ideas to study social learning in animals. This research, and the terminology used there is varied, so that Call and Carpenter (2002) argue that the question of what information can be gained from social learning should be ordered in three categories: actions, results and goals. The work in this thesis focusses mainly on the information in actions.

Another problem, pointed out by Stephens (1993), is that a lot of models have agents that act as if they know where the relevant information is located in the actions of another agent. In this thesis, I also aim do demonstrate how an agent could determine this.

### 2.3.4 Game Theory

Another theory that is both well developed and often applied to the formal analysis of social interaction is game theory. In general, the theory deals with the question of how a rational economic agent should act (make a decision between a set of mutually exclusive options) to maximise its own pay-off in an interaction with other agents who are also assumed to act in a way to maximise their own pay-off. Interestingly, this can lead to outcomes that are neither preferred, nor intended by any of the agents (Ross 2011).

#### Economic Rationality

*Rationality*, or *economic rationality* in this context is understood as:

1. knowing which actions (probabilistically) lead to which outcomes,
2. having a consistent preference with regard to all outcomes, which defines a pay-off or utility for each outcome,
3. acting accordingly as to maximise the expectation of ones own outcome.

Formally, this can be easily applied to the information theoretic model we outlined earlier. The action variable $A$'s alphabet is exactly the set the agent has to chose an action from, and the underlying Causal Bayesian Network is exactly what the agent needs to know to understand how its actions potentially lead to different outcomes. Given a utility function for all outcomes, the agent would be faced with a simple decision making problem, where each action could be associated with a corresponding expected utility. The problem in an agent-agent interaction, and the defining problem for game theory, is how to make this decision if your decision also depends on another agent who rationally tries to maximise its own pay-off itself taking into account you decisions, etc.

To illustrate, imagine you and a friend, who both like cake, are offered three different cakes. Each of you has to separately pick one, and if you both choose different cakes then you get the cake you have chosen, respectively. If you both chose the same cake, then there will be no cake for either of you. Now, the cakes are of slightly different size, and both of you would prefer the biggest cake. Which cake should you pick? You might want to take the biggest cake, but then your friend could use the same reasoning, and then there would be no cake. Similarly, if you decide to settle for the medium cake to avoid a collision, your friend could do the same. Even if you go for the smallest cake, the one least likely to be picked by your friend, you could not exclude that your friend might reason the exact same way, again creating a situation with no cake.

It should be obvious that this self-referential and circular reasoning has the potential to create all kinds of problematic and paradoxical situations. Game theory now offers a framework to understand how an agent would determine its own optimal decision.

### Classical Game Theory

The field originated with the Minimax Theorem from von Neumann's paper "Zur Theorie der Gesellschaftspiele" (von Neumann 1928) which dealt with two player games with the properties of:

- full-information: All player know everything there is to know about the game up till now, its complete current state, and the rules of the game.
- zero-sum: For each outcome state the associated outcomes for all players sum to zero. So, if one player get a positive outcome of $V$, then the outcome of the other player is -$V$. In general, one player's gain is another player's loss.

Chess or Checkers are examples of such a games, as they have two players, both players know all there is to know about the rules and the state of the game (position of pieces on the board), and a better outcome for one player equates to a worse outcome for the other player.

For all two player, zero-sum, full-information games von Neumann proved the existence of a mixed strategy that will guarantee the two players a pay-off of *at least* $V$ or $-V$, respectively. If both players know these strategies, this essentially solves the game, and the outcome is predetermined. Deviating from the optimal minimax strategy allows for the deviating player to be exploited by its opponent. The resulting strategies are therefore called stable, or strategic equilibria.

### Prisoners Dilemma

Building upon this, game theory has also been extended for multiple player games (Von Neumann and Morgenstern 1944), and to games with a more general pay-off distribution than zero-sum games. A prominent example here is the Prisoner's Dilemma (PD)(Rapoport and Chammah 1965). The story to illustrate this dilemma is that of two criminals who are caught by the police and interrogated separately. Both are offered the same options: they can either confess to the crime (Defecting, in regard to their fellow prisoner), or be silent (Cooperating with their fellow prisoner).

If both stay silent the police can only incarcerate them for minor charges (1 year), but if one confesses he will go free, while the other will be put away for 15 years. If both

| | P2 cooperates | P2 defects |
|---|---|---|
| P1 cooperates | -1 / -1 | -15 / 0 |
| P1 defects | 0 / -15 | -10 / -10 |

Table 2.1: This table shows the different pay-offs for a two-player Prisoner Dilemma. Depending on the actions of both players, Player One (P1) will receive the first pay-off in a cell, and Player Two (P2) the second. The pay-offs are negative, so 0 is the largest and therefore the most desirable pay-off.

confess, then neither of them will receive mercy, and both will go to jail for 10 years. This can be visualized as a joint pay-off matrix seen in Table 2.1. Each player's pay-off depending on its own actions, and the actions of the other player(s).

PD is a particularly interesting example because a.) it has been related to a number of real world scenarios such as "Mutual Assured Destruction" (Darwen and Yao 2002) and b.) it is at first glance unclear why a rational agent would ever cooperate. If one looks at the pay-off matrix it becomes clear that no matter how the other player acts, defecting is always preferable. This of course leaves both player worse of than if they would both cooperate, but the are effectively stuck in the mutual defection position, since neither can change its own strategy unilaterally and receive a better pay-off. This state is then called a *Nash-Equilibrium* (Nash et al. 1950).

One possible solution has been proposed by Axelrod and Hamilton (1981) in "The Evolution of Cooperation". The key insight here is the idea of an iterated prisoner's dilemma. It would still be irrational for an agent to cooperate for a single encounter, but if the agent's would know that there was a possibility for future encounters then they would have to act with taking into account that their current actions could influence the willingness to cooperate of the other player.

Without going into much more detail, it should be mentioned that this kind of analysis has been applied to a number of real world social situations and phenomena, specifically when dealing with rational agents that are motivated by their own gain in a somewhat antagonistic scenario. To contrast the work in this thesis with the vast body of game theoretic analysis I would like to point out again that classical game theory not only presupposed the agent's own ability to make rational decision, but also the ability to determine the rational decisions of all other players. This require the agent to

- know that there is another agent,
- know the other agents preferences in outcome,
- know the available action options of the other agent,

- and know what the other agent's action would result in.

While this is already a problematic assumption in human agents, these aggressive assumptions go far beyond the intention of this thesis to look at the first steps towards agent-agent interaction, since these abilities would place a high cognitive burden on the agent.

### Evolutionary Approaches to Game Theory

A possible approach to shift the cognitive burden and make the models more biological plausible is the introduction of evolutionary processes. One classic example is the previously mentioned example of Axelrod's "Evolution of Cooperation". In (Axelrod 1997) he used and evolutionary algorithm, proposed by (Holland 1992) to search for good strategies to play iterated prisoners dilemma.

A genetic algorithm is, in essence, a heuristic to find a good solution to a high-dimensional optimization problem. The most simple version consists of the following steps:

1. **Initialization**: A random population of genomes is creates, each representing a solution for the problem. They are expressed in a language that is able to model all possible solutions or parameter combinations.

2. **Selection**: Based on a fitness function each genome gets a fitness value. A certain number of genomes is then selected to reproduce, while the selection favours the genomes with higher fitness values.

3. **Reproduction**: A new population is created, based on the selected genomes. Those new genomes are usually either mutated (changed slightly), or combinations of the selected genomes, or both.

4. **Termination**: The simulation then jumps back to the selection step, unless a termination criterion is reached. This is usually a certain number of time steps, a threshold fitness value, or the lack of fitness increase.

Several modifications and refinements have been introduced to make this process more efficient, but most versions still contain those four steps. All that is usually needed to apply a genetic algorithm is an appropriate representation of all possible solutions to form the genome, and some way to assign a fitness value. This makes it possible to apply this heuristic to find a good strategy for agents in a competitive scenario. For example, in (Salge and Mahlmann 2010), genetic algorithms were used to evolve several strategies to

play a turn-based strategy game. The fitness value then simply becomes the percentage of won games. Similarly, strategies for game theoretic scenarios can be evolved. For example, when Axelrod applied the optimization algorithm to a population of strategies that would play iterated prisoners dilemma against each other the population would over time evolve to contain only strategies that played "tit-for-tat", a strategy where you mirror the last move of your opponent, and start by cooperating. This led to mutual cooperation all around. But this result has to be treated with caution, since (Ashlock, Kim and Leahy 2006) demonstrated that the resulting population depends heavily on the representation chosen for the genome. He showed that it is possible to have stable populations full of defecting agents, or oscillating populations, depending on how the strategies are represented as genomes.

A very similar approach is the mathematical field of "Evolutionary Game Theory", which is based on the notion of differential reproduction. For a stable environment there are certain species with heritable features. If those features are beneficial, meaning they increase the expected number of offspring, then the next generation should have more agents with those heritable features.

Based on this it is also possible to define a game theoretic scenario, where the optimal strategy or feature set depends on other agents. A classical example is a world that contains two competing species, which fill the same niche, apart from the ability to digest two specific plants. If those two nutritious plants exist in the same quantity, then we can see that they only stable solution is a population where animals that can digest the first, and those that can digest the second, have the same number of specimens. If one species' population is larger, then the other species could always find slightly more food per specimen, and could reproduce faster. If both species have the same number, then switching from one strategy to the other would offer no benefit. This is called an evolutionarily stable strategy, a population consisting of different strategies or solutions in a relation so that any change away from this distribution would be harmful to an agent.

Without going into more detail, we can see that in those models the cognitive work of finding a solution if moved out of the agent, into the evolutionary process. Finding the right strategy becomes *evolving* the right strategy. Still, the models retain the basic property that stable solution are situations where it would be irrational for a single agent, or species to switch away from its current strategy.

### Difference between Game Theory and the Presented Work

While game theory offers a lot of insight into several social interaction phenomena, and can in theory be applied to similar agent-agent models (if a utility function is assumed),

it also differs in scope and in some underlying assumptions to the work presented in this thesis.

First of all, the classical game theoretic approach assumes not only rational behaviour in the agent itself, but also the ability to predict the rational behaviour of others. This requires the assumption of extensive cognitive abilities, including theory of mind-like abilities in regard to the other agents.

Even if we assume that the cognitive work of finding equilibrium strategies is not done by an individual agent in its own lifetime, but as an adaptive, evolutionary process in a population of agents, then game theory still requires a joint pay-off matrix to create any kind of social interaction. The basic assumption in all of game theory is that the decision of another agent can directly influence one's pay-off, and therefore has to be taken into account when deciding one's own action. If the joint pay-off matrix would show the same values regardless of what the other agent does, then game theory would not need to be applied at all. In this thesis, I want to focus on possible motivations for interaction in scenarios where an agent's action *does not* influence another agent's pay-off, which is not covered by game theory as such.

What will find application later on, however, is the basic idea that strategies can be subject to evolution, and the idea that a stable strategy has to be the result of an individuals agent's optimization. An overall stable population requires an equilibrium where no individual agent can change its own strategy without losing utility.

### 2.3.5 Social Bayesian Learning

Another related area of research is "Social Bayesian Learning", a field that deals with the integration of other agent's information via Bayes' Theorem. In the coming chapters I will demonstrate why it is reasonable to assume that another agent's actions contain information, and why it is likely that this information is relevant for other agents. Assuming that this is the case, an obvious incentive for social interaction is the acquisition of this information via learning. But if all agents acquire information from others it is possible that the influence of this information on their behaviour becomes stronger than the influence of their own private information. This can lead to a case where an agent acts based on the information from its observation of others, rather than based on its own observation of the world.

Consider the example by Easley and Kleinberg (2010), where one agent wants to choose between restaurant A and B, which are next door to each other. His own research suggests that restaurant A is better, but once he gets there, no one is eating in restaurant A, while restaurant B is filled with customers. Based on this information it is reasonable to infer

that several other agents have private information that caused them to choose B instead of A. By inferring this additional information it becomes rational to choose B instead of A, even if your own private information suggests otherwise.

The problem here is that others might make similar conclusions, and create a chain reaction of inferred private information that is based on no private information whatsoever. The first guest could just have been uninformed, had no preference for either A or B, and then has picked B at random. The second guest might have also been uninformed, and picked B because the observation of the first guest acted as a symmetry breaker. The third guest then already saw two guests, and this might have caused him to overrule is own private information for A. All following guest could have preferred A prior to arriving, and then all made the same rational inference to choose B. In the end nearly everyone had private information to go to A, but all ended up going to B, via a process of rational decisions.

This phenomenon has been called *herding* or an *information cascade* by Banerjee (1992). Similar concepts can be found in (Bikhchandani, Hirshleifer and Welch 1992), including examples of information cascades in the real world, and conceptual examples on how those can occur. Easley summarizes the general requirements for an information cascade as:

- There is a decision to be made from several choices
- Agents make decisions sequentially, and each agent can observe the choices of the other agents.
- Each agent has some private information to help it with its decision
- Agents can only observe the actions of their fellow agents, but not their private information

This phenomenon has been formalized in the framework of Social Bayesian Learning, where Bayes Theorem is used to integrate the information of others into an agent's own probabilistic model. Similar methods will be used in a later chapters, and will be introduced there in more detail. The work in (Bikhchandani et al. 1992, Banerjee 1992) shows that information cascades can be produced in the formal framework of Social Bayesian Learning as well. In those models several properties of information cascades became clear:

**Cascades can be wrong.** As seen in the previous restaurant example, it is possible for the population of agents to make choices that would not be rational given the overview of all private information available to the agents.

**Cascades can be based on very little information.** Similarly, it is possible, especially if little information is present, that some small initial preference for one choice gets amplified and then influences the whole system.

**Cascades can be fragile.** In the Bayesian Model it is quite possible to stop a cascade with a slight change of parameters. For example, if the prior for one restaurant was zero, then no Bayesian update could change that to anything else, and the agent would just make a choice not including this option.

This is somewhat in contrast to the argument presented in "The Wisdom of Crowds", where Surowiecki (2005) argues that agents that aggregate their information can produce very accurate results. But, as Easley and Kleinberg (2010) point out, this only applies if they are guessing independently. If they are influencing each other, then it is possible for the crowd to be rational and wrong at the same time.

This work has also been generalized to deal with different networks describing the agent observations. The previous examples all assume that all agents can observe each other. More recent work now asks what happens if agents are limited to observe only their neighbours in some form of network. Gale and Kariv (2003) show that the connectivity of the network plays an important role. Given similar parameters that would allow an information cascade in a full network they shows that synchronicity becomes likely ones the network connectivity reaches a certain percolation threshold.

This relates to the work in later chapters of this dissertation. Once the information maximising agents start using Social Bayesian Learning to use the information from others they become susceptible to information cascades. Assuming the agent could influence whether they observe others or not, they could actively influence the network structure of observations. From an information maximisation perspective this then raises the question if observing less information might actually lead to better information?

# Chapter 3

# Relevant Information

## 3.1 Chapter Overview

The purpose of this chapter is to introduce the *relevant information* formalism by Polani et al. (2001), and illustrate some of its properties. This is not directly relevant in regard to my research question; the main aim here is rather to familiarize the reader with this specific relevant information definition. This is crucial for the remaining thesis, because when I talk about optimization of information processing I mean maximisation of relevant information intake, with the technical meaning of relevant information as defined in this chapter.

First, I will state the general idea of relevant information, and reproduce Polani's formal definition. Some simple examples will be presented to both illustrate the formalism, and demonstrate some of its basic properties. Several of the derived properties are used in later chapters, or are helpful to understand the later chapters.

I will also define relevant information for sub-optimal strategies. This definition differs from Polani's existing one as it defines how much information is need for a given performance level, and not what performance level can be reached with a given amount of information.

Furthermore, I will then present an experiment to demonstrate:

- How a genetic algorithm can be used to approximate the relevant information of a given environment from an agent-centric perspective.
- That we can distinguish between three different world types, depending on how relevant information is related to agent performance.

These experiments have two purposes in this thesis. First, I want to illustrate how an adaptive process optimized in regard to performance and information parsimony will end up on the trade-off curve between performance and necessary information. This is unsurprising, but will be a helpful reminder for the later argument about how increased performance requires more relevant information. The second purpose is to introduce my idea that worlds can be classified by the shape of their relevant information trade-off curve, and demonstrate how agents can detect which kind of world they are in. In this context I will also argue why all the "interesting" cases that we will lock at in the remainder of the thesis are of a specific type, or should be assumed to be of a specific type.

Additionally, I will introduce the new concept of *partial relevant information*, both as a general idea and as a formal definition. Partial relevant information extends the relevant information formalism; instead of only measuring how much relevant information is present in the overall environment or sensor input, it also measures *where* that information resides.

## 3.2 Concept of Relevant Information

Relevant information is a concept introduced to tackle an often discussed limitation of information theory, its lack of semantics. While the general rejection of semantics in classical information theory offers the benefits of mathematical versatility, it also leads to problems when information theoretic methods are used by an agent to act intelligently in the world.

If we analyse a given signal it is possible to ask how much irreducible self-information, or entropy, is contained within the signal. This would also measure the maximum amount of information this message could contain about the world. The same principle applies to sensor input, and if an agent were to maximise its information about the world, it might be reasonable for the agent to adapt its sensors in a way that maximises the information intake. But the problem with this approach is that some of the information gained might be more relevant or useful than other information, and some information about the world might be completely useless for the agent. If we make the additional assumption that information processing requires some work that in itself incurs a cost to the agent, then taking in additional "useless" information might indeed be harmful to the agent's performance.

To address this problem Polani et al. (2001) suggest that the relevance of information could be determined by examining the actions resulting from information in regard to a utility function. In essence, information is relevant if it is necessary to increase the agent's performance. Relevant information is defined in (Polani et al. 2001) as the minimal amount

of information needed to choose an optimal strategy. In the next section I will give a formal definition close to and based on Polani's work.

## 3.3 Definition of Relevant Information

### 3.3.1 Relevant Information for Optimal Strategies

Assume that there is an agent that interacts with the environment by choosing an action in reaction to some form of sensor input. The environment $R$ is in the state $r$, and the agent chooses an action $a$ from a set of actions $A$. For simplicity, we assume for now that the agent can perceive the whole environment, so the sensor state is equal to the state of the environment. Furthermore, assume that the actions of the agent are connected to some unspecified form of utility function $U(a, r)$ (for example, survival probability, or fitness) which determines different pay-offs, depending on the agent's action $A = a$ and the state of the environment $R = r$. We also assume that the states of the world $R$ are distributed according to the probability distribution $p(r)$. In this case, for every state of the environment $r$, there exists a set $\mathcal{A}^{opt_r}$ of actions which result in the highest expected utility:

$$\mathcal{A}^{opt_r} = \arg\max_a \left(U(a, r)\right) \tag{3.1}$$

A *strategy* is defined as a conditional probability distribution $p(a|r)$, which defines for every state $r$ the probability of choosing the different actions $a$. We shall define an optimal strategy for the state $r$ as a distribution $p(a|r)$ which has the property such that:

$$\forall a : p(a|r) > 0 \Rightarrow a \in \mathcal{A}^{opt_r} \tag{3.2}$$

Meaning, that if an action $a$ has a non-zero probability of being chosen in reaction to state $r$, then this action must be one of the of optimal actions in $\mathcal{A}^{opt_r}$. This also allows us to define the set of all optimal strategies:

$$\pi^{opt} = \{p(a|r)|\forall a, r : p(A = a|R = r) > 0 \Rightarrow a \in \mathcal{A}^{opt_r}\} \tag{3.3}$$

Since we assumed an existing probability distribution for $p(r)$, we can calculate for every optimal strategy $p(a|r) \in \pi^{opt}$ the resulting probability for $a$ as $p(a)$:

$$p(a) = \sum_r p(a|r) \cdot p(r) \tag{3.4}$$

This makes it possible to compute, for every optimal strategy, the mutual information $I(A;R)$ between $A$ and $R$.

$$I(A;R) = H(A) - H(A|R) \tag{3.5}$$

Relevant Information (RI) is defined (Polani et al. 2001) as the minimal mutual information between the action random variable $A$ and the environment random variable $R$, over all possible optimal strategies.

$$RI = \min_{p(a|r)\in\pi^{opt}} I(A;R) \tag{3.6}$$

This can also be understood as:

- the minimal amount of information an agent has to acquire about the environment, in order to act optimally.
- the minimal amount of information an agent's actions have to contain about the environment, if the agent acts optimally.

The first interpretation is the standard interpretation present in Polani's work. The second interpretation is new, and follows from the symmetry principle of mutual information. It is the key insight that leads to the digested information argument in the later chapters. This new interpretation will be used later to argue why an agent's actions have to contain useful information for other agents. This different interpretation also leads to the different definition for sub-optimal relevant information, since I want to be able to measure how much relevant information is present in an agent's action at a specific performance level.

### Examples for Optimal Relevant Information

To illustrate the principle of relevant information, I will present a few simple examples. They are presented in the form of pay-off matrices where the columns denote the different states of the environment, and the rows denote the different actions of the agent. The values represent a positive pay-off for the agent for a specific state-action pair. They are the value of $U(a,r)$, the utility function, that result from the agent choosing action $a$ if the world is in the state $r$.

World 1 in Table 3.1 shows a scenario were each state of the environment has a different, corresponding optimal action. To choose the optimal action, the agent has to know the exact state of the world. Since the world has four possible states, this means the agent needs to acquire two bits of information, i.e., the agent would need at least two yes-no questions to determine the state of the world.

**World 1**

| Pay-Off | State 1 | State 2 | State 3 | State 4 |
|---|---|---|---|---|
| Action 1 | 1 | 0 | 0 | 0 |
| Action 2 | 0 | 1 | 0 | 0 |
| Action 3 | 0 | 0 | 1 | 0 |
| Action 4 | 0 | 0 | 0 | 1 |

Table 3.1: A pay-off matrix where each state of the environment has one, different corresponding optimal action (coloured in red)

**World 2**

| Pay-Off | State 1 | State 2 | State 3 | State 4 |
|---|---|---|---|---|
| Action 1 | 1 | 1 | 0 | 0 |
| Action 2 | 1 | 1 | 0 | 0 |
| Action 3 | 0 | 0 | 1 | 1 |
| Action 4 | 0 | 0 | 1 | 1 |

Table 3.2: A pay-off matrix where two groups of states have the same optimal actions (coloured in red)

Those two bits correspond to the amount of relevant information determined by the previously introduce formalism. If we assume that the states of the world are equally likely to occur, we can calculate the mutual information for the one possible optimal strategy as

$$I(A;R) = H(A) - H(A|R) = 2 - 0 = 2. \tag{3.7}$$

Here, the conditional entropy $H(A|R)$ of the actions given the state of the environment is zero, because the actions are fully determined by the environment, since there is only one optimal reaction to each state of the environment $R$. The entropy $H(A)$ of the actions itself is equal to the entropy of $H(R)$, and is therefore two bits.

The second example in Table 3.2 shows World 2, where the agent only needs to know if the world is either in the first two, or in the last two states. So the agent only needs to acquire one bit of information to act optimally. In this case several optimal strategies

**World 3**

| Pay-Off | State 1 | State 2 | State 3 | State 4 |
|---|---|---|---|---|
| Action 1 | 0 | 0 | 0 | 0 |
| Action 2 | 0 | 0 | 0 | 0 |
| Action 3 | 0 | 0 | 0 | 0 |
| Action 4 | 0 | 0 | 0 | 0 |

Table 3.3: A pay-off matrix where each action leads to the same pay-off regardless of the action the agent chooses

exist. For example, when the world is in State 1 the agent could always take Action 1, or always take Action 2, or any mixture of those two actions.

To determine one of the strategy with the minimal amount of mutual information, Polani et.al. suggest creating a strategy were every optimal reaction to a state of the environment is equally likely to occur:

$$p(a|r) = \begin{cases} 1/(|\mathcal{A}^{opt_r}|) & \text{if } a \in \mathcal{A}^{opt_r} \\ 0 & \text{if } a \notin \mathcal{A}^{opt_r} \end{cases} \tag{3.8}$$

The conditional entropy of $H(A|R)$ can then be calculated as:

$$H(A|R) = \sum_r p(r) \cdot H(A|R = r) = 4 \cdot \frac{1}{4} \cdot 1 = 1 \tag{3.9}$$

The entropy of $A$ is still two bits, because all reactions are still equally likely to occur, if they are summed over all states of the environment. All states of $R$ have equal probabilities, and also have the same resulting action state entropy, since every state of the environment has exactly two optimal actions, which results in one bit of entropy.

It follows that the mutual information for this specific optimal strategy is:

$$I(A; R) = H(A) - H(A|R) = 2 - 1 = 1, \tag{3.10}$$

which is also the minimal mutual information for any optimal strategy. For both cases we see that the formalism concurs with our intuition about how much information the agent needs to have about the environment.

In Table 3.3 we now see World 3, an example of a world with no relevant information.

**World 4**

| Pay-Off | State 1 | State 2 | State 3 | State 4 |
|---|---|---|---|---|
| Action 1 | 0 | 1 | 0 | 1 |
| Action 2 | 0 | 1 | 0 | 1 |
| Action 3 | 0 | 1 | 0 | 1 |
| Action 4 | 0 | 1 | 0 | 1 |

Table 3.4: A pay-off matrix that has different pay-off values but they only depend on the state of the environment, not on the action of the agent.

**World 5**

| Pay-Off | State 1 | State 2 | State 3 | State 4 |
|---|---|---|---|---|
| Action 1 | 0 | 0 | 0 | 0 |
| Action 2 | 1 | 1 | 1 | 1 |
| Action 3 | 0 | 0 | 0 | 0 |
| Action 4 | 0 | 0 | 0 | 0 |

Table 3.5: A pay-off matrix where the optimal action is always Action 2, no matter what the state of the environment is.

Every action in every state leads to the same result. Obviously, there is no information that could make the agent perform any better. Since all actions have the same utility we can minimize the mutual information by giving all states $r$ an equal probability for $p(a) = 1/4$. In this case the conditional entropy is equal to the unconditional entropy of $A$. This means the mutual information is zero,

$$I(A; R) = H(A) - H(A|R) = 2 - 2 = 0. \tag{3.11}$$

The next example, World 4 in Table 3.4, also contains no relevant information. There is the possibility that different pay-offs occur, but this only depends on the state of the environment, not on the action the agent chooses. Therefore, every strategy the agent could choose is equally good (or bad). The agent could in this case choose the random strategy, which has, as established in the the last example in Table 3.3, no relevant information.

**World 6**

| Pay-Off | State 1 | State 2 | State 3 | State 4 |
|---|---|---|---|---|
| Action 1 | 2 | 1 | 0 | 0 |
| Action 2 | 1 | 2 | 0 | 0 |
| Action 3 | 0 | 0 | 2 | 1 |
| Action 4 | 0 | 0 | 1 | 2 |

Table 3.6: A pay-off matrix with one, different optimal action for each state of the environment (coloured in red), and another suboptimal state for each action, that offer half the pay-off of the optimal state (coloured in yellow)

World 5 in Table 3.5 is different from World 3 or 4, but also contains no mutual information. Every state of $R$ results in the same optimal action $a$. So it seems the agent has to actually make a decision, rather than to act random, but it still has to acquire no information from the environment. An optimal strategy here would be to always chose the same action, thereby the entropy of $A$ is zero, $H(A) = 0$. Similarly, since the state of $R$ has no influence on the action, $H(A|R)$ is also zero. As a result:

$$I(A;R) = H(A) - H(A|R) = 0 - 0 = 0 \tag{3.12}$$

This can be generalized, since every strategy that does not depend on the state of the environment should have no mutual information between $R$ and $A$. If the distribution of $A$ does not depend on the state of $R$, then it follows that $p(a) = p(a|r)$ for all $a$, which leads to $H(A) = H(A|R)$. Since mutual information can be calculated as the difference of those two values, every strategy where $p(a)$ is equal to $p(a|r)$ for all $a$ has no mutual information. As a result, there are always several strategies (basically every possible distribution for $A$ independent of $R$) that have no mutual information. If any of those strategies that do not depend on the input states are optimal, then the relevant information is zero.

### 3.3.2 Relevant Information for Suboptimal Strategies

In the next example in Table 3.6 we are looking at different pay-off values. This illustrates a limitation of the original relevant information formalism (Polani et al. 2001), where only optimal actions were considered. The scenario seen here looks very similar the one in Table 3.1. The optimal strategy requires two bits of information, and there is an optimal action for every state of the environment. But if one were to settle for an average

pay-off of 1.5 then it would be possible to play a strategy that requires only one bit of information, as seen in Table 3.2. This makes this example a different scenario from World 1, but the original formalism does not account for this difference. This becomes even more problematic if we exaggerate the pay-off values. If the best pay-off was 1000, and the second best pay-off was 999, then the difference between the two strategies would be only 0.5, compared to the overall pay-off of 1000. But this marginal improvement would have to be bought by an increase of 100% in the required amount of information. If information processing has an associated cost this might be undesirable for the agent.

To account for this problem (Polani et al. 2006) extended the formalism to be able to answer the question,: "How much performance can the agent get for a given bit of information?" To formalise this, we first define the set $\pi^u$ as the set of all strategies that have the average pay-off level, or performance, of at least $u$ as

$$\pi^u = \left\{ p(a|r) \middle| \sum_a \sum_r U(a,r)p(a|r)p(r) \geq u \right\}. \tag{3.13}$$

Note that this raises the requirements for the pay-off function $U(a,r)$, which now has to return values that can be averaged. For the optimal relevant information an ordinal preference function that would have simply sorted all the possible outcomes according to the agent's preference would have been sufficient.

With this set of strategies it is now possible to define the relevant information for a certain performance level of $u$ as the minimal mutual information over all strategies that have at least the average pay-off of $u$ as

$$RI(u) = \min_{p(a|r)\in\pi^u} I(A;R). \tag{3.14}$$

This definition now allows us to formally address two conjugate questions:

- How much average pay-off can the agent achieve with a given amount of mutual information?
- How much information does the agent need to reach a certain performance level?

While Polani et al. (2006) focus mostly on the first question, I will put the focus on the second. This will become important later in the thesis, when it is crucial for the arguments to demonstrate how much information an agent actually has to process when it is acting on a certain performance level.

### 3.3.3 Properties of Relevant Information

So far this chapter has mostly restated the relevant information formalism, though with slight alterations to make it more applicable to the argument in this thesis. Before I will continue to expand upon this formalism, I will demonstrate some properties of relevant information. This will not only help to deepen our understanding for later arguments, but it will also illustrate how well the formalism is in line with our intuitions.

#### Upper Bounds

**Property 1.** *Relevant information is bound from above by the entropy of the environment $H(R)$:*

$$RI \leq H(R). \tag{3.15}$$

This follows directly from the definition of mutual information as a difference between the entropy and the conditional entropy:

$$I(A;R) = H(R) - H(R|A). \tag{3.16}$$

The value of $H(R|A)$, as a conditional entropy, is non-negative ($H(R|A) \geq 0$), which leads to the following inequality:

$$I(A;R) \leq H(R). \tag{3.17}$$

Since the mutual information of any strategy is smaller than $H(R)$, the minimal mutual information also is smaller than $H(R)$. The same argument also holds for any suboptimal relevant information. This agrees with the interpretation that relevant information is the amount of information the agent has to acquire from the environment to act optimally. Since the entropy of $R$ is all there is to know about the environment in terms of information, the agent cannot possibly acquire more information than $H(R)$. This also shows how relevant information is dependent on $p(r)$, our *a priori* assumption about the distribution of the states of $R$. $H(R)$ provides an upper bound for the mutual information of any possible strategy, and therefore is also an upper bound for the overall relevant information.

**Property 2.** *The relevant information is bound from above by the maximum entropy of $A$:*

$$RI \leq \max_{p(a)} H(A) = \log(|A|). \tag{3.18}$$

In analogy to the last property, mutual information can also be expressed as:

$$I(A;R) = H(A) - H(AR). \tag{3.19}$$

With the non-negativity of $H(A|R)$ we can again follow that:

$$I(A;R) \leq H(A). \tag{3.20}$$

Since the distribution $p(a)$ is not fixed, but dependent on the strategy $p(a|r)$, the relevant information is not bound by any actual entropy $H(A)$ for a specific $p(a|r)$, but is bound by the maximal entropy that $H(A)$ could achieve, which is the logarithm of the number of states of $A$:

$$RI \leq \max_{p(a)} H(A) = \log |A|. \tag{3.21}$$

This also agrees with the interpretation that $RI$ is the information needed to act optimally. When an agent has only two options to choose from, then the agent might acquire a lot of information, but ultimately at most one bit of information is relevant, the one that tells it which of the two options to chose.

### Relevant Information as Function

The formalism for sub-optimal relevant information $RI(u)$ in Eq. 3.14 defines a function that returns the amount of relevant information for every performance level that can be achieved by the agent. This can be used to construct a graph that illustrates the relationship between relevant information and performance in a specific scenario, similar to those graphs produced by Polani et al. (2006). I will use a scatter plot similar to these graphs to present the results of the next experiment. But before we do so, I would like to outline a few additional properties of the actual function approximated by these graphs.

**Property 3.** *The relevant information function is monotonically non-decreasing in regard to the performance level $u$. A higher performance $u$ always requires a larger, or equal, amount of relevant information than a lower performance $u'$.*

This follows directly from the definition. Compare two performance levels $u$ and $u'$, assuming that $u \geq u'$. We define the associated level-set of strategies that achieve at least the performance level of $u$ as:

$$\pi^u = \left\{ p(a|r) \middle| \left( \sum_a \sum_r U(a,r) \cdot p(a|r) \cdot p(r) \right) \geq u \right\}. \tag{3.22}$$

We can see that all the strategies in $\pi^u$ are also in $\pi^{u'}$ since everything that is larger than $u$ is also larger than $u'$. It follows that:

$$\pi^{u'} \supseteq \pi^u \tag{3.23}$$

If we then calculate a minimum over two sets, where one set is a subset of another, it is clear that the subset has a higher or equal minimum.

$$RI(u) = \min_{p(a|r)\in\pi^u} I(A;R) \geq \min_{p(a|r)\in\pi^{u'}} I(A;R) = RI(u') \tag{3.24}$$

In short we can state that for two performance levels $u$ and $u'$:

$$u \geq u' \Rightarrow RI(u) \geq RI(u') \tag{3.25}$$

This again is consistent with our intuition about how relevant information should behave. If an agent wants to do better it cannot do so with less information.

**Property 4.** *There is always a strategy and a performance level with no relevant information.*

As outlined before, if the agent chooses a strategy where $p(a) = p(a|r)$ for all $r$, then the conditional entropy of $H(A|R)$ and the entropy $H(A)$ become identical, and the mutual information becomes zero. The random strategy, defined as $p(a) = 1/|A|$, is one example for such a strategy. Since the mutual information cannot be less than zero it is certain that random is on the actual trade-off curve defined by $RI(u)$. And since its mutual information is zero, this means that there is at least one point on the trade-off curve where there is no relevant information.

Once should keep in mind that "random" is not the only strategy with zero mutual information. Any strategy, i.e. conditional distribution $p(a|r)$, where $A$ is independent of $R$, also leads to no mutual information, as discussed in the example of world 5.

**Property 5.** *The relevant information function is a property of the world and the agent's possible actions, it does not depend on any particular strategy.*

Relevant information depends on several variables. It is limited by the entropy of variable R, and by the logarithm of the number of actions states. It also depends on the utility function $U(a, r)$. But it is computed over all possible strategies, which should illustrate that no particular strategy can influence the function RI(u) *per se*.

If we would calculate, for all possible strategies, both the mutual information and the performance, we could then put a data point in a graph for each strategy. Those data points would all be on, or above RI(u). This means that for every strategy the amount of mutual information is larger, or equal, to the amount of relevant information needed for that performance level. The actual function runs along those data points that represent the minimal amount of mutual information for each performance level. Even if an agent does not utilize a certain strategy, this strategy would still define the relevant information. So, while an agent can chose how to act, the agent cannot influence the trade-off curve between performance and mutual information defined by RI(u).

## 3.4 Relevant Sensor Information

The next section contains another new extension of the original relevant information formalism. It highlights the difference between an agent's sensors and the environment, and asked what happens when the world is no fully accessible to the agent. I also proof that a limitation in sensor input can only lead to an increase in relevant information for a given performance level.

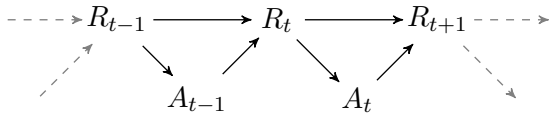
So far we assumed that the state of the environment $r \in R$ is identical to the sensor input $S$ of the agent, meaning that the world was fully accessible to the agent. In general, this cannot be assumed to be true, and we also have to deal with cases where the information about the world, and the subsequent choice of actions is limited by the sensor input. This is especially true if we want to maintain an agent-centric perspective regarding our sensor intake.

The Bayesian graph in Fig. 3.1 illustrates this extension to the model. The agent now only has access to the random variable $S$, instead of perceiving $R$ directly. $S$ is the output of a probabilistic function of $R$, which can be defined by the conditional probability $p(s|r)$. This limits the agent's possible choice of strategies. With full access to the environment an agent could choose any strategy $p(a|r)$. We will call the set of all those strategies $\mathcal{P} = \{p(a|r)\}$. The sensor-limited agent can only react to what it perceives in its sensor input $S$, which is created with a fixed $p(s|r)$, thereby all available strategies for that agent are in the set $\mathcal{P}_{p(s|r)}$, defined as

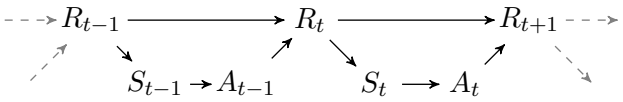
$$\mathcal{P}_{p(s|r)} := \{p(a|r) : p(a|r) = \sum_s p(a|s) \cdot p(s|r)\}. \tag{3.26}$$

Since $\mathcal{P}_{p(s|r)}$ has additional constraints, it is obviously a subset of $\mathcal{P}$.

This now allows the definition of *Relevant Sensor Information*, the minimal mutual




(a) Fully Accessible World




(b) Sensor Model

Figure 3.1: Causal Bayesian network of the perception-action loop, unrolled in time, showing (a) a fully accessible world model and (b) the case when the world access is limited through the sensor input.

information between the sensors and the actuators of an agent, over all optimal strategies available to the agent as:

$$RSI_{p(s|r)} = \min_{p(a|s)optimal} I(A;S). \tag{3.27}$$

Just as with the relevant information $RI$, it is also possible to define this for any suboptimal performance level $u$ as $RSI_{p(s|r)}(u)$, if there is actually a strategy that reaches performance level $u$.

$$RSI_{p(s|r)} = \min_{p(a|s)\in\pi^u} I(A;S). \tag{3.28}$$

In contrast, the previous definition for $RI$ measures the relevant information, not for a specific sensor set-up, but for all possible sensor configurations, including full world access. This distinction becomes clearer by looking at which variables are involved in the calculation of mutual information.

Relevant information looks at the mutual information between actions $A$ and environment $R$, while relevant sensor information replaces $R$ with the sensor input $S$. As a result relevant sensor information is not making a statement about the world and all possible strategies. It has the advantage, though, that it can be determined from the agents perspective, and it captures the limitations of the specific sensor configuration of the agent.

The question I now want to address concerns the relationship between $RI(u)$ and $RSI(u)$. Since the random variables $R_t, S_t, A_t$ form a Markov Chain,

$$R_t \rightarrow S_t \rightarrow A_t \tag{3.29}$$

it might appear that limiting the access to the world via $S$ might limit the mutual information between $R$ and $A$, and thereby reduce the relevant information, but the opposite is the case.

**Property 6.** *Let $u$ be a performance level, and $p(s|r)$ a given sensor configuration of the agent. If the agent, with that sensor configuration, can select a strategy $p(a|s)$ which on average achieves at least a performance level of $u$, then the Relevant Sensor Information for that level is larger than or equal to the Relevant Information for that level.*

$$RSI_{p(s|r)}(u) \geq RI(u) \tag{3.30}$$

*Proof.* This is a proof by contradiction. Assume that

$$RSI_{p(s|r)}(u) < RI(u), \tag{3.31}$$

with the actual amount of relevant information $RI(u) = k$. Since $RSI_{p(s|r)}(u) < k$, this would imply that there is a strategy $p(a|s)$ that would result in a mutual information between $S$ and $A$ that is smaller than $k$, i.e., $I(S;A) < k$. But this strategy, which would still achieve the pay-off level $u$, could then be used construct a conditional probability between $A$ and $R$ as:

$$p(a|r) = \sum_s p(a|s) \cdot p(s|r). \tag{3.32}$$

This conditional probability, which would be in $\mathcal{P}$ the set containing all conditional probabilities between $R$ and $A$, would also reach the performance level $u$. Following from the Markov Chain property in Eq. 3.29, we also know that the mutual information $I(A;R)$, based on this conditional probability would be less or equal to the mutual information of $I(S;A)$. So there would be a strategy with the following mutual information,

$$I(A;R) \leq I(S;A) < k. \tag{3.33}$$

As a result this means that there is a strategy that achieves $u$ and has less mutual information than $k$, so it cannot be that $k = RI(u)$. This contradicts out original assumption,

so the opposite must be true. □

While this result may appear counter-intuitive at first, it can be explained. When the agent's sensor input becomes limited, there are basically three options:

1. The agent cannot achieve the given performance level any more.
2. The agent can continue to use the same strategy $p(a|r)$, even though its sensor is now limited.
3. The agent has to use a different strategy that has a higher mutual information.

The third option always results in a strategy that is more "expensive", meaning that it has higher mutual information, since if there is a cheaper strategy, it would have been there before the sensor limitation, and that strategy would have been the one used to define the relevant information.

## 3.5 Experiment: Agent based Approximation of Relevant Information

### 3.5.1 Motivation

Now that I have defined relevant information and relevant sensor information and outlined some of its basic properties I want to introduce an experiment that will demonstrate how relevant information and relevant sensor information can be approximated with a genetic algorithm.

This is particularly relevant for my second research question, as it demonstrates how the relevant information function can be approximated for a wide range of scenarios. Previous work (Polani et al. 2001, Polani et al. 2006) showed how the trade-off function can be explicitly computed, and how a dynamic programming approach can be used to iteratively improve the strategies towards a convergence point. The approach presented here has the advantage that it can treat complex simulations as a black-box, as long as they provide some form of utility. This includes scenarios where the agent has to repeatedly make decision over an extended period, and only then acquires a result. This will be demonstrated, exemplary, with an agent playing a simple computer game.

Furthermore, I will then also demonstrate how scenarios can be differentiated based on their respective trade-off functions, and what properties this assigns to the world. This allows an agent or observer, who was able to determine the trade-off function, to derive certain properties about the world.

Furthermore, this experiment also addresses an important step towards answering the first research question, as it demonstrates where on the trade-off function an "actual" agent should be located. Assuming the agent is motivated to a.) improve its performance and b.) to achieve this with the lowest cost in information processing, then the agent would try to find a good trade-off between information parsimony and performance. Regardless of how information cost is weighted against performance, any optimal trade-off lies on the Pareto front that is defined by the RI function. Meaning, agents should only employ strategies that have the property that there is no other strategy that has a.) less mutual information but the same performance, or b.)more performance, but the same mutual information. So, no actual used strategy should be dominated by another existing strategy. Therefore, optimized strategies should be on the actual RI function, rather than above it. This would, in conjunction with Property 3 (the RI function being monotonically non-decreasing), indicate that an increase in performance would also likely lead to an increase in relevant information. This will later be used to argue that agent's act as an information preprocessor to other agents.

Of course, there would still be the possibility that the agent lives in a world where a performance increase is *not* accompanied by an increased level of relevant information (such as in Table 3.5). But the following experiment will also demonstrate that an approximation of the relevant information is the right tool to determine which of those different worlds an agent exists in.

### 3.5.2 Overview

Originally, the following experiment was designed to illustrate how relevant information corresponds to enjoyment-related factors in game design, and how information theory could offer a measurable, and quantifiable game play evaluation method. This work was done together with Tobias Mahlmann and published as Salge and Mahlmann (2010)[1].

The original paper has a more in depth discussion on how information theoretic properties relate to certain essential game play flaws, but the presentation here will focus on the approximation method, and the three main types of scenarios an agent can encounter. In relation to this thesis, it is interesting to note that the same game flaws that prevent fun, are also flaws that will make a game less likely to resemble a real life scenario. If

[1]T.Mahlmann's contribution was the implementation of the game simulation, the genetic algorithm and the discussion of the relation between game design and fun. I contributed the implementation of the information theoretic tools, designed the game rules and developed the theory of how to apply relevant information to games. The work presented in the following section is my own, apart from the implementation of the game itself, and the implementation of the genetic adaptation.

games are understood as practice for real life challenges, then this might also explain why such flaws make games less interesting, since they do not prepare the player for the kind of scenario the real world offers. This is briefly discussed in (Salge and Mahlmann 2010), but in this thesis I will present the argument in reverse. I will briefly introduce some of those flaws, which also hinder enjoyment, and then demonstrate why they are unlikely to be present in the actual challenges a biological agent has to face.

The three flaws I will talk about here (Inferior Choices, Dominant Strategies, Irrelevant Actions) are of special interest, because they do not only describe an undesirable scenario in a game, but their existence, in general, changes the overall nature of how the agent can interact with the world. They will be the later cornerstones to define the three different categories of worlds an agent can encounter, in terms of relevant information.

In short, I will argue that the scenarios which are most interesting for the player of a game are those with increasing relevant information, which, in turn, are also those most challenging to solve, and those where gaining information from other agents helps most.

### 3.5.3 Relevant Information and Player Satisfaction

This section describes how relevant information (RI) corresponds with game mechanical properties that foster or hinder enjoyment. Since it is questionable whether fun can be measured by some mathematical formalism, I am focusing on measurable factors that prevent or reduce fun in games and should therefore be avoided. Those factors are mainly taken from literature, such as (Koster 2005, Juul 2003), or are criteria which are self-evident. While some of them might be debatable, this is beyond the scope of the present exposition, as is a psychological or sociological evaluation of those factors and their relation to game play fun.

What I want to demonstrate instead is that RI offers some measurable values that relates to properties in game mechanics that should be avoided. The first data point I want to discuss in this context is the actual RI, the minimal amount of mutual information over the set of optimal strategies.

#### Inferior Choices

One possible game world design flaw is to offer the player a choice of actions where one action is an *inferior choice*, independent of the circumstances, since there is always another, better option. Game theory would call this a dominated option. As a result this action would never be played by an optimal strategy. According to Property 2, the relevant information has an upper bound of $\log(|A|)$. If we now eliminate one option for $A$, the

maximum entropy is reduced to $\log(|A|-1)$. So, for every inferior choice in $A$ the maximum of RI is reduced. Therefore, an increasing presence of inferior choices should be detectable by a decrease in the value of RI.

### Dominant Strategies

Even more limiting in terms of the reduction of possible actions is the existence of a dominant strategy. By *dominant strategy* I mean a strategy or action that is always better than all other options, independent of the circumstances (such as the actions of other players or changes in the environment). In those scenarios, an optimal agent's strategy will always choose the same action, regardless of the agent's sensor input. Such a scenario is also undesirable for a game, because once the player finds this strategy he is forced to play it continuously.

An example of this scenario is the world in Table 3.5. Here the player would always play Action 2. The amount of information one would need to acquire about the environment is 0, so the RI is also 0. If we only look at single actions this also follows mathematically from the argument in the last section. If the player only chooses the same action, no matter what the environment, then $H(A)$, the entropy of $A$ is 0, resulting in zero mutual information.

If the dominant strategy is a specific sequence of actions, its existence would not be immediately clear, but the same argument as for single actions can be applied. If the optimal strategy consists of some combination of actions that is played regardless of sensor input, then the conditional entropy of $H(A|S)$ and the marginal entropy $H(A)$ become identical, and the resulting mutual information is zero. So, in any case, the existence of a dominant strategy would result in a vanishing RI value.

But for the agent to detect this flaw, and differentiate it from the flaw described in the next section, we need to take an additional data point into account, the performance level of the random strategy. This strategy chooses its actions at random, with an equal chance for every action to be picked, disregarding any sensor input. This strategy's actions have obviously no mutual information with the environment, as outlined in Property 4. The performance level of this strategy indicates how much utility a player can get "for free", by acting without any thought or regard of the environment.

If there is a dominant strategy in the game, then the player can find this strategy, and we can observe a strategy that has the same amount of relevant information as random (none), but has a higher performance level than random. If this is not the case, then we are dealing with the next flaw.

### Irrelevant Actions

Another flaw is to design a game mechanic where the agent's effort has no impact on the outcome of the game. Apart from the question if this should be considered a game at all (Juul 2003), we postulate that this is not desirable. While it is unlikely that such a scenario would be designed by a human designer by choice, it is possible in a complex game world that such a pathological case arises.

The world in Table 3.3 describes the pay-off matrix of a scenario where neither the agent's action nor the states of the environment matter. All strategies have the same pay-off, and therefore, the RI is 0, because the strategy that plays randomly is also optimal.

To differentiate between this case and a dominant strategy we just have to consider whether there is an actual difference in the performance levels of the different strategies that is not explained by random noise, but due to different action choices. If the random strategy plays as well as all other strategies, than there seems to be nothing to do for the player, its actions are irrelevant. If there is actually a visible difference between bad and good strategies, but they both have zero mutual information then we are dealing with a dominant strategy.

### Desired Case

The desirable case in this context is a world where the previously discussed flaws are absent. Such a game would be designed such that:

- The player uses all possible options, in similar frequency
- The decision of the player have an impact on the world and on the utility of the player.
- The optimal decision depends on the different states of the environment

This would lead to a high degree of RI for the best strategy. Furthermore, the performance for the fully random strategy should be low, and the increase in performance should lead to an increase in RI.

In summary, this means there should be three distinct kind of scenarios. One where the agent has no influence on the world, where its action are irrelevant. Here all strategies should have roughly the same performance. The second kind of scenarios are those where there are strategies that do better, but they do not require any kind of sensor input. Here the RI should be zero for a wide range of different performances. Finally, the third kind of scenario should have an increase in necessary relevant information for higher performing strategies.

### 3.5.4 Experimental Model

To demonstrate the new approximation method we[2] implemented a simple, turn based tactics game where the player controls several groups of units and has to make the decision what actions those groups are taking. We will demonstrate how neural network-based AIs, adapted to the game via Genetic Algorithm (Holland 1992), were used to approximate the actual relation between RI and performance.

In the experiment we will approximate the relevant information function for three different scenarios (different rule sets for the world).

The hypothesis here is that a genetic algorithm can approximated the relevant information trade-off function, and that the shape of this function can be used to differentiate between different cases. Specifically, it should be possible to differentiate between a world with no player effect, a world where dominant strategies exist that are optimal, regardless of the sensor input, and the "desired" case, where the agent is forced to use the sensor input to achieve different performance levels.

I will now first introduce the general game mechanics, and describe how the approximation via genetic algorithm was performed. I will then separately describe each scenario, explain how an agent would play this scenario, what the actual RI function should look like and then discuss the scatter plot data for that scenario. At the end I will show a comparison between the different approximations.

#### Game Mechanics

The game used to demonstrate the approximation algorithm is a turn based, two player tactics game; a very simplified version of the battles in the "Heroes of Might & Magic" series. Both players start with three stacks containing three creatures each. The goal for both is to kill the opponent's stacks by attacking them with their own stacks. If only one player has remaining creatures, then that player wins.

The game is played in consecutive rounds, until one player wins. Each round lets the player act with their three stacks in alternating order. The player opposed to the one we study always gets to act first. One of its three stacks is chosen, and the player gets to decide what action to take. The four options are to either attack one of the three opposing stacks or to wait. The effect of that action is carried out. Then the other player gets to act with one of its stacks, also chosen at random. This is alternated, until both players have chosen an action for all their remaining stacks. Then the next round starts.

[2] the actual implementation of the software described here was a joint project between me and Tobias Mahlmann for (Salge and Mahlmann 2010).

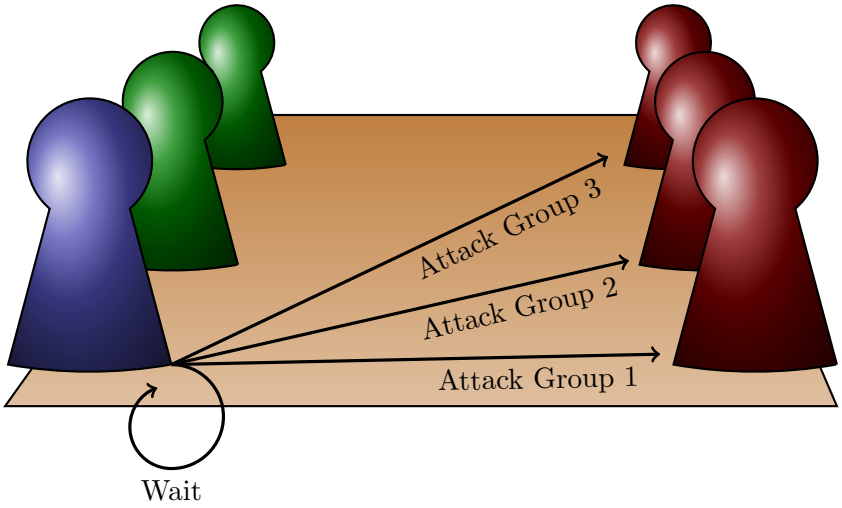



Figure 3.2: A diagram representing the player's options. Each pip represents a group of units. Whenever it is a groups turn to move it becomes active (here blue), and the owning player can decide to either attack one of the enemy groups (here red) or wait.

All creatures start with the same attributes for attack damage(1.5 to 4.0 points of damage per hit point) and hit points(3 per creature). To simplify, we removed the spatial component so stacks can attack each stack of the enemy, regardless of position. Every stack gets to act once per round, but the order is random. So, every round all six stacks, both those of the player and the opponent (or less if one of the armies is already destroyed) are able to take one action.

The players decision has to be made when one of their armies can make a move. The actions and consequences of previous armies have been fully resolved at that point.

When a stack attacks another, the damage dealt is calculated by multiplying the hit points of all remaining creatures in the stack with their attack damage. There is a random element in the attack damage, so while each creature has a certain damage range; the actual damage done varies. The damage is then subtracted from the hit points of the first enemy creature. If the hit points of that creature drop to zero the number of creatures in the stack is decreased by one, and the remaining damage is subtracted from the next creature. If the number of creatures in a stack reaches zero, the stack dies and is removed/ignored until the game ends. The game ends when one player has lost all its stacks.

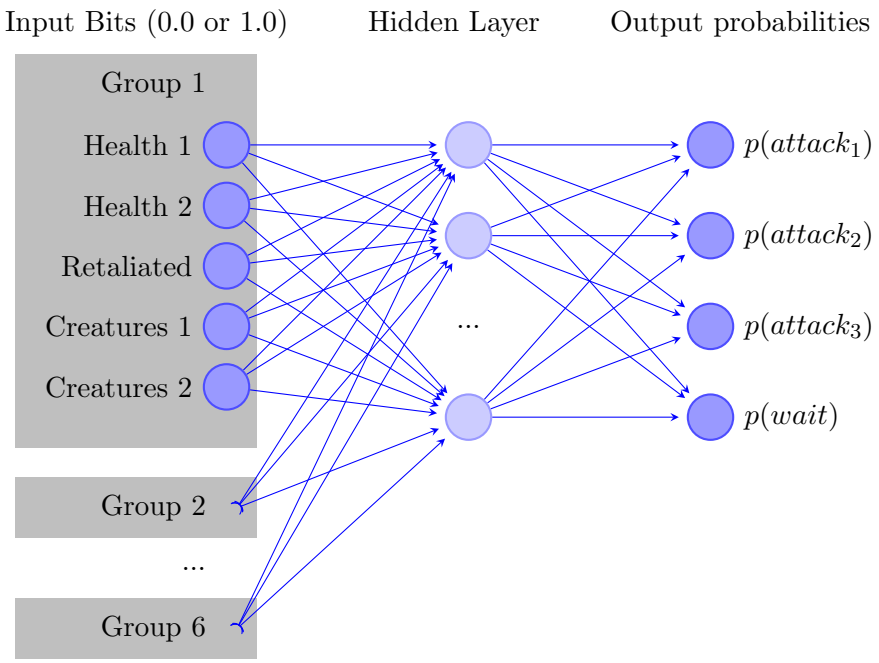



Figure 3.3: A neural network with one hidden layer is used to compute for each game state as input a resulting probability distribution of actions.

The only special ability given to some stacks is the ability to retaliate. If a stack is able to retaliate then it can make a counter attack on the stack that just attacked it and deal damage to the attacker after it has been damaged itself.

### Agent Control

The agent we implemented to test the game has the following functionality. When it is its turn to choose an action for one of its stacks it has to take in the current sensor state of the world and select an action from a list of choices. We deliberately designed the game in a way that it was not necessary to record or use any memory about the past of the game. All there is to know is currently visible, and so the agent controller only has to take in the current world state.

A further requirement for the agent controller was that it could model different strategies, meaning that different agent controllers should return different action distributions for the same sensor input. Furthermore, we also needed the agent to be able to express

probabilistic strategies and it should be possible to serialize the agent controller, so it could be subjected to a genetic algorithm.

We chose a simple, feed forward neural network with one hidden layer to realize the agent controller.

### Agent Sensor Input

The agent's input $S$ is a set of binary variables containing the following values:

- For each of the agent's own stacks:
  - two bits are used to encode the stack's topmost creature's health
  - two bits are used to encode the actual number of creatures in the stack
  - one bit to indicate if the stack can retaliate
- For each of the opponent's stacks:
  - two bits are used to encode the stack's topmost creature's health
  - two bits are used to encode the actual number of creatures in the stack
  - one bit to indicate if the stack can retaliate

Note that only the health of the top creature was given, as all remaining creatures have full health, since their health can only be reduced when they are the first creature in the stack. So, overall five bits were used to encode each stack's current state. Thus, the sensor input state for each stack has $2^5 = 32$ states.

Two players with three stacks each make six stacks in the game which makes the signature of each game state an array of thirty bits. So the overall number of different sensor states for $S$ is $2^{30} = 1,073,741,824$.

### Neural Network

The neural network, as seen in Fig. 3.3 has thirty input nodes, each input node receives one bit of the sensor data as a real value of either 1.0 or 0.0. The network has four output nodes, each associated with a different action the AI can take. In those scenarios where the wait action is not available the output of the wait node is ignored. There is also a hidden layer of 3 neurons.

Each of the 30 input neurons is now connected to each of the hidden neurons. Likewise, each of the hidden neurons is connected to each of the output neurons. Each of these

connections has an associated weight. The state of the input neurons is determined by the input they are receiving. The state of the hidden neurons results from a weighted linear combination of the connected input neurons which are used as the input of a sigmoid function, which returns a value between 0 and 1. This value is the resulting state of the neuron. Likewise, the output neurons state dependent on a weighted combination of the states of the connected hidden neurons.

Which action the AI decides to use is then determined at random, where the proportion of the value of a certain node would correspond to the probability of that action being chosen. For example, if node one had the value 1.0 and the three other nodes had the value of 0.5, then action one would be chosen with a probability of 40% and the other actions would be chosen with a probability of 20% each.

This probabilistic interpretation of the output was chosen deliberately over a "winner takes it all" interpretation, so it is possible for the neural network, which is by itself deterministic, to represent probabilistic mappings from input states to chosen actions. This allows the network to express a wider range of possibly strategies $p(a|s)$, including actual probability distributions, and not just those strategies that have a specific resulting action for each state.

### 3.5.5 Approximation via Genetic Algorithm

Calculating the actual RI for each performance level would make it necessary to look at all possible strategies, but this approach becomes quickly infeasible once the complexity of a game grows. An alternative option is to use a genetic algorithm to select a subsection of all strategies, those adapted to be of high performance and low mutual information. We then record the mutual information and performance of those strategies, and use those to approximate the actual RI.

Note that, since the computation of mutual information requires the joint probability of both variables, it is not sufficient to only look at a strategy $p(a|s)$ to compute $I(A;S)$; it is also necessary to get data about the distribution $p(s)$ of $S$. But this is not a problem in a game scenario, where the AI playing the game can be used to create an actual real distribution of $S$. The starting state of the game is defined as part of the game rules, and each subsequent game step is a result of the players' actions.

To adapt the controllers with any genetic algorithm we first have to define how a genome encodes the different controllers. In our case the genome is a list of all the weights associated with each connection in the neural network.

The next requirement is the definition of an objective or fitness function, a function that formalizes what should be optimized with the genetic algorithm. In our case we want

to evaluate the genomes with a fitness function that favours high performance and low mutual information, weighted with a variable weighting factor $\lambda$. Both values, performance and mutual information, are normalized to values between 1.0 and 0. For the performance we divide the number of victorious games $g_w$ by the number of played games $g_a$. For the mutual information we divide the results by the maximum entropy of the actions, in our case $\log(4) = 2$. The mutual information is then subtracted from 1.0, since we want to minimize it. The resulting fitness $U(p(a|s))$ function looks like this:

$$U(p(a|s)) = \lambda \left(1 - \frac{I(A;S)}{\log |A|}\right) + (1 - \lambda)\frac{g_w}{g_a} \tag{3.34}$$

We use the values of 0, 0.25, 0.5, 0.75 and 1.0 for $\lambda$, where $\lambda = 1.0$ means that only mutual information matters, and $\lambda = 0.0$ means that only performance is taken into account.

To determine the performance of a specific genome we created the associated neural networks that plays the game for 1000 games against an opponent that picks random actions. For each strategy, we measure both the performance, as the fraction of games won, and the mutual information for the recorded joint distribution of sensor states and actions. Note that each game consists of several pairs of sensor inputs and selected actions.

We then need to select a specific genetic algorithm to perform the optimization. In this experiment we used an particle swarm optimizer (Kennedy and Eberhart 1995, Shi and Eberhart 1998) as provided by the Computational Intelligence Library (Engelbrecht, Peer and Pampara 2010). For each scenario we ran a population of 20 genomes for 200 generations for varying through all 5 values of $\lambda$.

Note that this approach is not aiming to find the optimal solution, but aims to intelligently sample the overall search space to look mostly at those strategies that are close to the relevant information function.

### Scatter Plot

We measured the relation between performance and mutual information for all genomes in all generations and the result is a scatter plot as seen in Fig. 3.4. This means the points seen in the graph are not just the endpoints of evolutions, or the best results, but all strategies that have been tried during the evolutionary run. Every data point in the plot is a strategy; the values indicate its performance and its mutual information.

We combined all these values into one graph, because any additional data point can only improve the approximation. The actual function RI(u) would be a line that all data points are either on or above, since it is possible for a strategy to have higher mutual

information than the relevant information, but not lower. We also combined strategies for different values of $\lambda$ in the same graph, since they also are all subject to the same RI function, regardless of what fitness function was used to produce them. This relies on fact that the relevant information function is defined by the way how actions and states map to performance, and is independent of the actual approximation. So different values of $\lambda$ still approximate the same function.

Note, the two factors in our fitness function are used to evolve the strategies towards higher performance, and lower mutual information, thereby moving the resulting strategies closer to this actual function. Note that the function $RI(u)$ we want to determine is not an average of the strategies we are looking at, but a lower bound. Therefore, it is possible to take the results of several evolutionary runs and combine them all into the same graph. This can only improve the approximation. Also, since the mutual information is a function defined by the game mechanics, it is possible to vary $\lambda$ and evolve strategies that are more optimized towards performance or mutual information reduction, and still combine them in the same plot. Indeed, our experience suggests that this is advised to get a good selection of strategies that populate the whole Pareto front.

#### Approximated Lower Bound

In a comparison plot I will show graphs that approximate the actual RI function. To produce these I first select all strategies that are not dominated in terms of mutual information cost or performance. Meaning, I select all strategies for which there are no strategies that a.) have a higher performance and the same or less mutual information, or b.)have the same performance, but lower mutual information. I then draw a line through all these strategies; this line lies below all tested strategies. This line is our approximation of the relevant information function.

### 3.5.6 Problems

#### Deterministic Strategy

One problem in approximating the actual RI of a game is the use of deterministic strategies. A classical neural network usually picks one action based on its inputs, and normally it would always choose the same action for the same input. This automatically limits the strategies $p(a|r)$ to those where $H(A|R) = 0$ , since the action is determined by the sensor inputs. This leads to the mutual information being calculated as:

$$I(A;R) = H(A) - H(A|R) = H(A) - 0 \tag{3.35}$$

Since we are looking for the strategy with the least amount of mutual information, limiting us to deterministic strategies seems to hinder a good approximation. Strategies that take a random decision in those circumstances where it does not matter are a good candidate for a strategy that uses only the actual relevant information. Only searching in the subspace of deterministic strategies might result in the algorithm overlooking a good approximation candidate, and thereby will worsen our approximation. In any case, deterministic strategies are only a comparatively small subset of the overall available strategies, so excluding all other strategies would reduce the amount of available strategies that could offer a good solution significantly.

One solution to this problem is to modify the way the neural network chooses the actions. Instead of picking the actions whose nodes got the highest values, the algorithm associates the values of the end nodes with the probability for that action to be picked. This allows the neural network to realize random strategies; strategies that should be favoured if they have the same performance, but lower mutual information.

#### Large Input State Space

Treating the sensor input as one random variable quickly increases the state space. Every additional bit of information doubles the amount of theoretically possible sensor states. In our case, 30 bits of information already lead to 1,073,741,824 different states. Calculating properties such as the entropy $H(R)$, or the mutual information by summing up over all those possible states was already infeasible for the large number of computations we had to perform. This also stretches the plausibility of similar mechanism being used in biological systems. Even if we only argue that natural adaptation leads to a solution that has low relevant information and high performance, without actually computing it, it still raises the question how this is archived?

Fortunately, both in our example, and arguably in the real world as well, not all combination of inputs actually happen, so not all states of the overall input state space have to be considered. So, instead of using a data structure were the amount of occurrences for each state of the joint probability $p(a; s)$ is recorded, we used a data structure that records:

For every state $s = S$ that occurs at least once:

- Number of occurrences of $s$
- Number of occurrences of each state of $a \in A$, if the sensor has the state $s$

Combined with the overall number of state-action pairings it is possible to calculate the

mutual information with an alternative formula:

$$I(A;S) = \sum_{a \in A} \sum_{s \in S} p(a,s) \cdot \log \left( \frac{p(a,s)}{p(a)p(s)} \right) \tag{3.36}$$

Since $p(a,s)$ is zero for all $s$ that never occur we can neglect all terms that sum over a state $s$ that is not in our data structure. This reduces the calculation of $I(A;S)$ to summing over all existing states of $S$, thereby greatly decreasing the needed processing power.

### 3.5.7 Evaluation of Different Scenarios

In this sections i will now show the resulting scatter plots from three different scenarios. I will first outline the scenario, and explain how the relevant information function should look like. I will then show the scatter plot and discuss it. In the end of this section I will present a comparison plot that should illustrate how the different kind of scenarios can be differentiated by looking at the RI approximation.

#### Case 1, No Player Effect

The first case we look at is a scenario that is deliberately constructed so that the player's decision has no influence on the outcome of the game.

In this initial scenario both sides have the same creatures and there is no ability to retaliate. The player has the option to attack the stacks in position one, two or three, but does not have the option to wait. If the player tries to attack a stack that is dead, the game would redirect his attack to the next stack alive. In the case that an attack deals more damage than the current stack could take, the remaining damage is redirected to the next stack. The opponent here always chooses at random which stack to attack. As said before, the opponent always goes first.

In this case, the player's action has not real influence on the outcome of the game. When the player chooses what stack to attack we can see that all possible choices are equally good, as they have the same expected effect. Regardless of where the damage would be applied it would remove the same amount of hit points from the opposing team, thereby reducing the opposing teams ability to deal damage in the same way. All options to take bad decision are taken away. The player cannot wait, and if attacking an army where some of the damage would be wasted, this damage would then be redirected. This also meant that the opponent, which acted random, was not really doing anything wrong, and therefore, the ability to start each round should be a huge advantage.

The only way how a player could actually make a meaningful difference would be to attack those stacks that had not yet acted this round, so the dealt damage would reduce the hit points, and thereby the damage of creatures who have yet to attack. But this information was not available in the current sensor input, so there was no change for the AI to devise a strategy to use this information.

The scatter plot resulting from this should show little change in performance, since the player has no real influence on the outcome. All variations in outcome are due to the random elements of the game. Furthermore, I would expect that those strategies evolved to minimize their mutual information should end up using 0 mutual information, since a completely random strategy should be just as good as any other.

### Case 1, Results

Several evolution runs with different value for $\lambda$ of our adaptive AI yielded the results seen in Fig. 3.4.

Two effects can be observed here. Firstly, there is no real difference in performance levels between the different strategies, they all seem to be very close to zero. So this seems to confirm that the players actions have no real impact on the outcome of the game. The small variation in performance values is likely due to the random element in damage calculation that allows the player to win in rare cases. Note that the graphs performance axis is scaled to reach only from 0.0 to 0.025, otherwise, if the scale would go up to 1.0, the variations would be nearly invisible on the graph.

One property of the simulation that was only revealed through the analysis was the fact that the AI player nearly never wins, so the advantage of being able to go first seems to be very strong.

The second thing to observe here is that the RI for all the performance levels up to $\approx 0.011$ is zero, since there is always at least one strategy that does not use any information. Also, for the other strategies that go above that value the increase in RI is quite low.

Comparing that to our earlier theories we seem to be dealing with a case where the player's actions are irrelevant and a closer look at our current game mechanics supports this analysis. All actions are attacks, deal similar damage, always hit a valuable target, and even reduce the opponent's ability to deal damage in a similar way. So no matter what the player decides, the resulting action will have the same effect (a similar amount of damage to some enemy unit). Therefore, the strategic choices of the player do not change its game performance.

In summary, the resulting scatter plot supports out hypothesis, and the lack of player effect is visible in the lack of spread in performance levels.

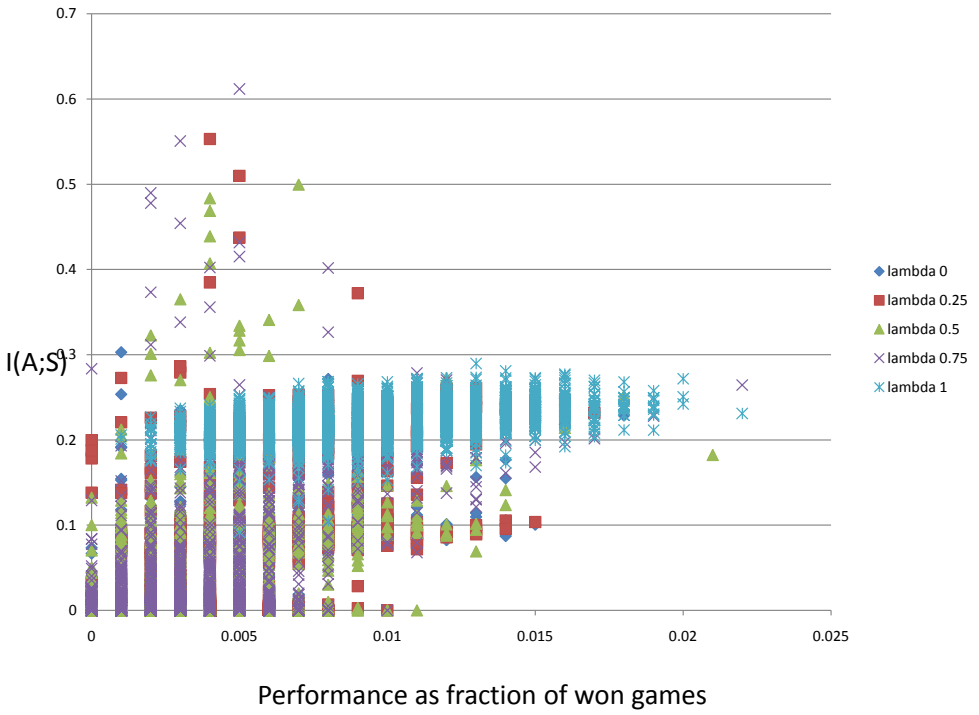



Figure 3.4: A scatter plot showing the relation between performance and mutual information for all evolved strategies for case 1. The plot includes the statistics for all genomes in every generation; the actual relevant information is a lower bound on all these data points. The different colours show how the strategies where optimized, corresponding to the different weighting factors between performance and low mutual information. Note that the scale for performance, as fraction of won games, only reaches from 0.0 to 0.025, otherwise all data points would appear in a line above 0.

### Case 2, dominant strategy

In this scenario the world is modified, so the player could do better, but in a way that required no sensor input. The game mechanics were modified so the player's actions have an impact on the game. We introduce the retaliation mechanics, and now each stack can retaliate once per game round. So, when a stack is attacked for the first time in a round it will retaliate, executing an attack on the attacking stack, after the damage of the original attack had been resolved. If the same unit would be attacked in the same round it would be unable to retaliate again. We also introduce the option for a stack to wait and do

nothing for one round.

Now the player has an impact on the game. A good strategy will have to avoid choosing to wait, as this actions deals no damage and is therefore a bad choice. Furthermore, it would also be good to attack an enemy stack that has already retaliated in order to minimize the damage received in return. The opponents still chooses actions uniformly random, but now this also means that the opponent can chose suboptimal actions such as wait.

While this might look like the agent would now be required to use its sensor input, there is also a simple strategy that is arguably optimal, which does not require any input regarding the current state of the environment. An example of such a strategy is to always attack the stack in position 1. This strategy avoids using wait, and it focuses all attacks on the same target to avoid retaliation (after the first attack). In case the first stack is dead, the attack will be forwarded to the next stack (still the same mechanics as in Case 1).

This is what we earlier identified as a dominant strategy, something that should be avoided in game design but cannot necessarily be seen as easily as here. As the player does not actually need to look at the game world to make a decision, the resulting RI function should have mutual information of zero for most of its performance levels. In contrast to case one there should now be strategies that improve well above the performance level of the random strategies but still keep a mutual information of zero.

### Case 2, Results

Looking at Fig. 3.5 we can see that there are several strategies with better performances than random, and we see several strategies that are able to win the game more than 60% of the time even though the game still lets the opponent start first. Even for those relatively high performance levels the amount of RI is zero, indicating that these strategies do not react to the agent's sensor input.

The graph in Fig. 3.5 also shows how the different weights in the fitness function push the different controllers along different paths in the two-dimensional projection (to low mutual information and high performance) of the solution space. The adaptation towards minimal mutual information ($\lambda = 0.0$) moves quickly towards the random strategies and then ends up in a cluster around zero performance and zero mutual information. The strategies that maximize performance ($\lambda = 1.0$) don't move towards the lower mutual information, but their cluster pushes to the right to explore strategies with higher performance. Finally, the strategies that balance both constraints ($\lambda = 0.5$) develop good strategies with a performance around 0.55 but also use no mutual information.

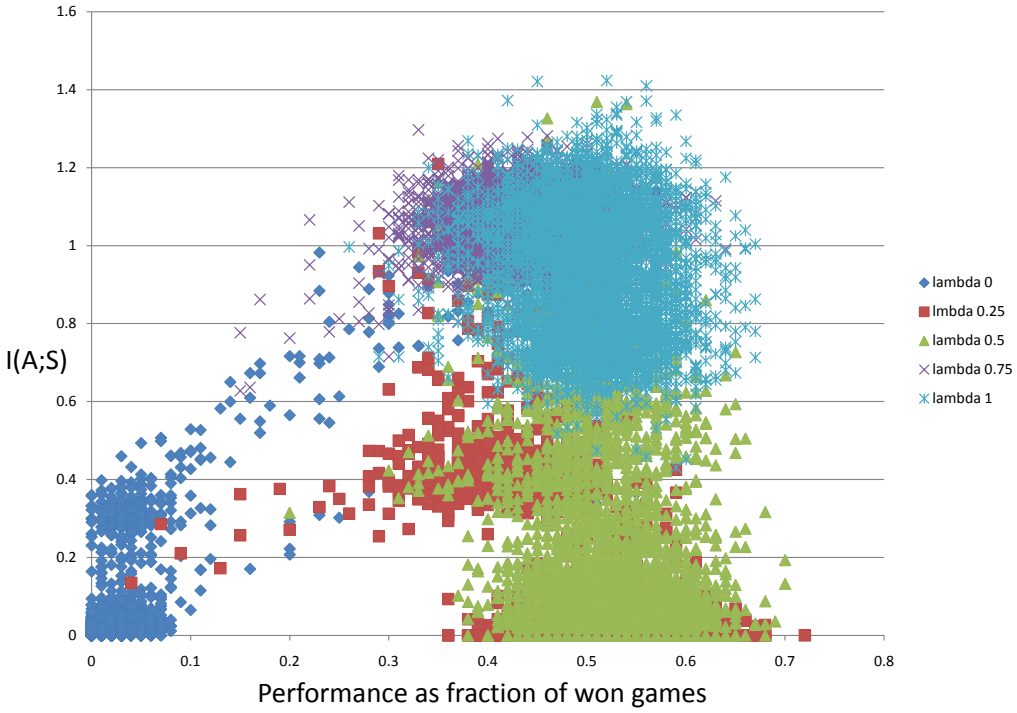



Figure 3.5: A scatter plot showing the relation between performance and mutual information for all evolved strategies for case 2. The plot includes the statistics for all genomes in every generation; the actual relevant information is a lower bound on all these data points. So, in this case the relevant information for all achieved performance levels is 0, as there is always at least one strategy that performs at least that well, and has no mutual information. The different colours show how the strategies where optimized, corresponding to the different weighting factors between performance and low mutual information. For example, the dark blue strategies are only optimized towards low mutual information, so they gravitate towards the bottom, but achieve little in terms of performance.

This shows how adding more simulations with varying $\lambda$ weights allows us to approximate the actual RI function in different places. Again, keep in mind that we are not interested in the average value of these strategies, but in approximating the lower bound of all possible strategies for each performance level. Also note, that since there are strategies with zero mutual information and a performance of more than 0.7 this means that for all performance levels below 0.7 the relevant information is also zero. Even though

there are no actual strategies with zero mutual information between 0.08 and 0.37, it still follows from the definition, as there are strategies with *at least* that performance level, which also have no mutual information.

So, in summary the approximated relevant information function is zero for all achievable performance levels. This supports our prediction as the resulting approximation here is in line with the existence of dominant, sensor-invariant strategies.

### Case 3, positive Relevant Information

We further modify the game so it is necessary for a good strategy to acquire information about the game world. Now retaliate is stronger, dealing three times the amount of damage than a regular attack. But retaliation will now only be activated if a stack has waited in the last turn. Since the AI chooses strategies at random this should lead to some opponent's stacks randomly being able to retaliate. These should be avoided, as their retaliation attack would be very negative for the attacking units.

A good strategy should be to avoid those stacks, which can be identified via the one bit of information that encodes if a stack is able to retaliate. Since it depends on the random actions of the opponent which stacks are able to retaliate it is now necessary to actually process the sensor information telling an agent which stacks can retaliate.

Furthermore, we also stop the forwarding of attack orders. So, if any player now attacks a stack that is dead, its attack will have no effect. Thus, the information of whether a stack has remaining creatures should become relevant.These modifications also make it harder for the random AI to successfully play the game, as it now has even more options to chose actions that are bad.

These modified game mechanics should now force the player to use sensor information to increase its performance. So the resulting RI function should show an increase in mutual information for higher performance levels.

### Case 3, Results

Looking at the graph in Fig. 3.6 we can see that our game play modifications have led to a measurable change in the scatter plot. It is still possible for the AI to actually develop good strategies, some winning in more than 70 % of the cases. But for all the strategies found by the AI adaptation that go beyond a performance of 10 % there seems to be at least a certain amount of information those strategies need to process. Up to a performance level of 0.08 there still seems to be a strategy which performs better than others, but uses no mutual information. For all higher performance levels the minimal mutual information for

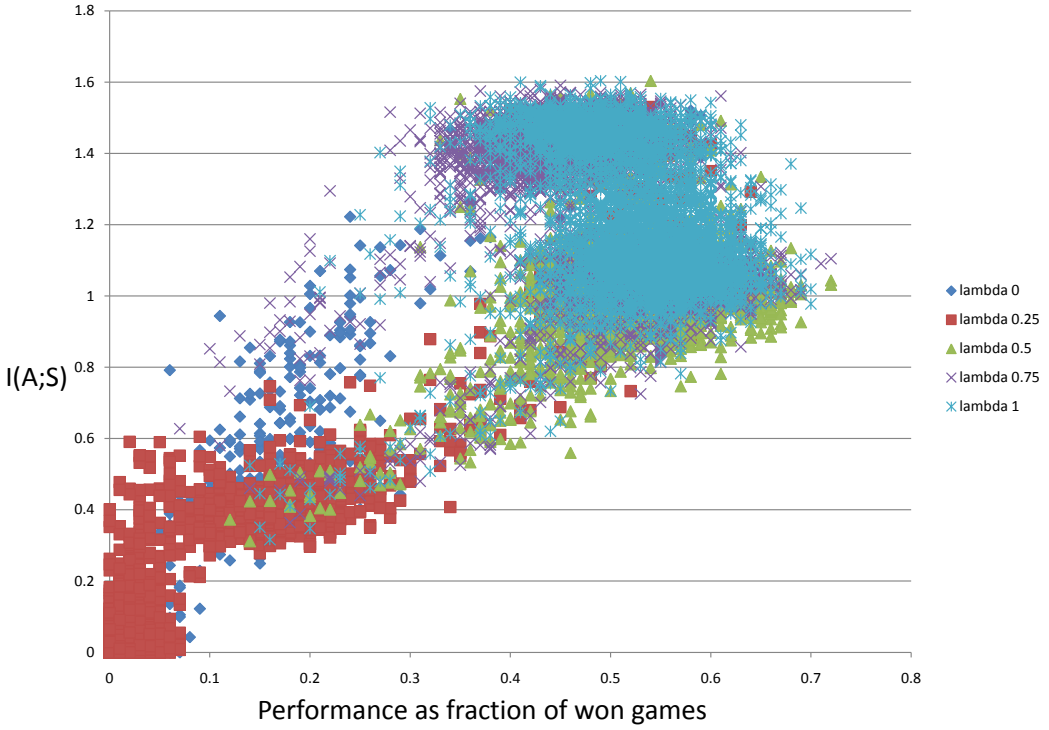



Figure 3.6: A scatter plot showing the relation between performance and mutual information for all evolved strategies for case 3. The plot includes the statistics for all genomes in every generation; the actual relevant information is a lower bound on all these data points. In this case the necessary mutual information seems to increase with performance. The different colours show how the strategies where optimized, corresponding to the different weighting factors between performance and low mutual information.

all realized strategies increases. For example, for all strategies above a performance of 0.5 there are no strategies with less than 0.7 bits of mutual information. This indicates that such performance level necessitates the processing of an average of 0.7 bits of information per decision.

This increase in necessary mutual information indicates that a higher performance level also needs a better analysis of the different factors of the game world, or that the strategies need to be more reactive towards those factors.

We can also see, again, how the different weighting factors push the strategies into

different areas in the solution space. For example, those strategies with where adapted with ($\lambda = 0.25$) are more optimized towards minimal mutual information, and therefore are more often found in the are of low or no mutual information with little performance. Strategies that are more optimized towards performance tend to explore solution that are better in terms of performance, but have a higher mutual information. But even in the area around the performance of 0.4, where strategies with different optimization parameters mix, the all seem to be lower bound by roughly the same function. As explained before, this is the case because the RI function bounds all possible strategies, regardless of what they are optimized towards.

In summary, as strategies get better the minimum of necessary mutual information increases. This also results in an RI function approximation which increases for higher performance values. This is also in line with out predictions, since the scenario was constructed in a way that it required the agent to actually pay attention to the environment to do well.

### 3.5.8 Comparison of Relevant Information Approximations

To compare the different results I created a graph of the approximated RI function, as described in section 3.5.5. These graphs basically draw a line under the different scatter plots, going from the point of zero performance and zero mutual information to the point that approximates the relevant information for the optimal strategies. They are drawn along all points that are not dominated, and no point in the scatter plot is below this graph. They are the actual approximations of the relevant information function.

The resulting graphs can be seen in Fig. 3.7. These approximations of the relevant information function behave as predicted earlier. The green graph, associated with case 3 shows how the minimal mutual information increases together with the performance level. The indicates that the scenario of case 3 is a world where an agent has to react to its sensor input in order to perform well. The graph for case 2, the blue line, shows that it is possible to reach higher performance levels without mutual information. This clearly marks the scenario as one where the sensor input does not matter. The red line, the graph for case 1 should have ideally been a dot. But in our simulation there was a strategy that had mutual information and performed slightly better. This is likely due to some strategy winning a few of the games simply by chance. All in all, there is still very little influence on the performance.

Comparing the three graphs we can see that they are very different, and their shape could be used to differentiate between the different scenarios. This should demonstrate that the approach presented here is indeed able to differentiate between the different

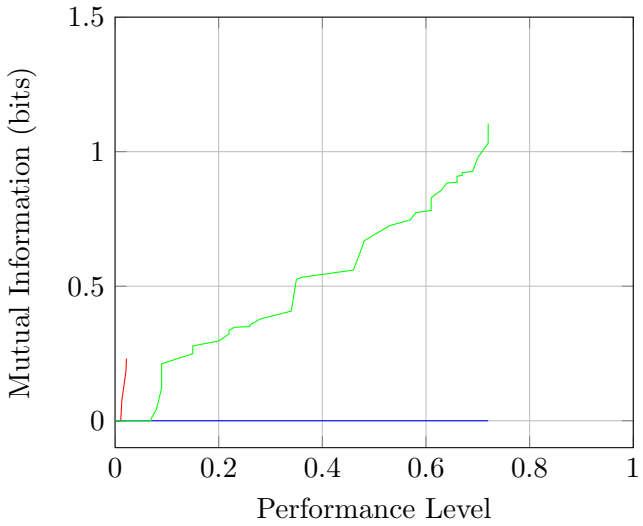



Figure 3.7: A comparison plot showing the approximations of the different relevant information function for the different scenario. The red line is case 1, where the agent had not influence on the world. The graph only has a very narrow range of performance levels. The blue line is the approximated relevant information graph for the second scenario. The agent here could do better or worse, but no performance level required any sensor input. This is reflected in the graph, as it spreads over a lot of performance levels, but always has a relevant information of zero. The green graph is associated with case 3, where the agent needed to react to the environment. The approximation of relevant information in case 3 suggests that more information is needed for higher performance levels.

scenarios by approximating their relevant information function.

### 3.5.9 Discussion of Relevant Information Approximation

While this approximation could serve as a possible analysis tool for game mechanics, and other designed scenarios, it is not necessarily a good tool for the agent itself. The first problem is that this kind of approximation requires an external feedback of the utility that the agent's strategy achieved, which might not be available to a specific agent. For example, if the utility is the reproductive fitness, then the agent itself might not be able to actually measure it and adapt accordingly. In such a case the approximation would only be possible for a population of agents, and the result would likely be inaccessible for any single agent in it. Also, even if the utility was available during the agent's lifetime for adaptation, then it would still be questionable if a random search through the strategy space would be the most efficient approach. But better ways of adaptation to find the optimal strategy are a whole field of study in itself, and not the focus of this dissertation.

More importantly for the following chapters is another observation. The two different parts of the fitness function are both based on assumptions we earlier made about living systems. One is information parsimony, the idea to only process as much information as needed, in order to save on the cost of information processing. This is realized as a constraint on the mutual information between input and output of the agent. The other is the utility of the strategy itself, possibly a measure of reproductive fitness, or a reward given to the agent. Since both seem to be reasonable assumptions it seems feasible to assume that either an agent that adapts during its own lifetime, or a population that adapts over several generations, would, just as the population in the examples, gravitate toward the actual relevant information vs. performance trade-off function. So, if we were to observe an optimized system of agents, they would likely be on, and not above the function.

### 3.5.10 Focus on Case 3 Scenarios

The other insight that was less evident in the experiment is the question which case we are likely going to be in? First, I shall argue that the three mentioned cases cover every possible category for relevant information functions. As discussed in the properties of RI, there are two specific points in every function, the point of *Optimal Relevant Information* (ORI) (associated with the mutual information needed to achieve optimality) and the point were the random strategy lies (zero mutual information). The ORI point always has a better or equal performance than random, and it always has a higher or equal amount of relevant information. Now there are only three configurations for the relation of those two points:

- Case 1: They are identical.
- Case 2: ORI has zero mutual information, but higher performance.
- Case 3: ORI has higher performance and higher mutual information.

If ORI was to only have higher mutual information and the same performance, then random would be a strategy that has the same performance and less mutual information. Therefore random would then be identical to the actual ORI.

Note that in reality most cases are very likely to be Case 3, just because it includes nearly all configurations but the two very specific cases of Case 1 and Case 2. But for the following argument let us assume that cases that are very close to a Case 1 or Case 2 are functional identical with them.

Returning to our original question, we can now ask what case an agent is likely to be in? I will argue that the interesting and likely case is Case 3, the one with the actual trade-off between relevant information and performance.

If the agent were to exist only in a Case 1 scenario, then its actions would not matter. So there would be no incentive to even develop the ability to act, let alone react to outside stimuli. Entities suited for this kind of world could hardly be called agents at all.

An agent that existed in a Case 2 scenario would act, but would not need to react. Its strategy would not depend on any state of the world, so processing sensor input would be a waste of resources. If we follow the idea of information parsimony we would then end up with an agent that has no sensor faculties, and a population of such agents would not be able to, or interested in, getting information from others.

The third case is the only one where actually having sensors and being able to perceive others to start with can be assumed as a result of development and adaptation. It is also the only case where getting information, both from the environment and from others, is useful.

This is somewhat mitigated by the possibility that an agent might only sometimes be in a Case 3 scenario, which would then be enough to develop the needed faculties to deal with them, and would still be available to it when it finds itself in a Case 1 or 2 scenario.

So, as a result, an agent that is reactive can assume that it and its fellow agents are likely in a Case 3 scenario (some of the time), and that the others have developed strategies on the actual trade-off curve, rather than above it. This is important for the later chapters, where several arguments assume either that the agent is in a Case 3 world, or that the agent is at least likely to be in a Case 3 world.

## 3.6 Unique Relevant Information

### 3.6.1 Motivation

Assuming that the previously introduced approximation works, then we are now able to determine what kind of world a specific agent lives in. This is useful for someone who is actively designing a world (i.e. a computer game designer) and wants to check what kind of relevant information functions describes this world. But as an analytical tool in general the provided insight is, at best, that the studied world is one where the agent needs more information to perform better. More interesting would be to know "where" the relevant information is located. Or more precisely: which part of the sensor input contains non-redundant relevant information, needed to determine the player's strategy?

This would be beneficial for adapting an agent in the following ways:

- Sensor Adaptation: What part of the world has to be made visible to the agent, so it can make an informed decision? Sensor input that only contains redundant or no relevant information is just a waste of resources, and should be removed for better information parsimony. If the sensor capacity is to be extended, additional new sensor input could be analysed for novel relevant information, and this could help to determine if a permanent extension of sensor capacity is beneficial.
- AI Input Reduction: If the sensor input is fixed, or is used across different scenarios, the same technique could be used to just reduce the amount of sensor input that is actually considered by the AI controlling the agent. Determining "where" the relevant information is located in the sensor input can then reduce the input state space and thereby enhance the AI's performance.

### 3.6.2 Definition

To determine the *partial relevant information* we first need to decompose the sensor input. In the case of our game the sensor input $S$ is both a random variable, but also a compound random variable composed of $n$ random variables, such as the health of a creature, the number of units in a group, their attack power, etc.

$$S = (S_1, \ldots, S_n) \tag{3.37}$$

We will define the partial relevant information of $S_1$ then as:

$$PRI(S_1) = \min_{p(a|s) \in \pi^{opt}} I(A; S_1) \tag{3.38}$$

But the problem with this measurement is that it does not consider *synergy*, nor *redundant information*.

#### Synergy

*Synergy* is the effect where two variables, $X$ and $Y$, together contain more information about a third variable $Z$ than the sum of what both of them individually contain about $Z$. In general, this can be expressed as:

$$I(X; Z) + I(Y; Z) < I(Z; X, Y) \tag{3.39}$$

A classical example for binary random variables is the XOR case where the state $Z$ is an XOR of the states of $X$ and $Y$. The mutual information $I(X;Z)$ and $I(Y;Z)$ is zero in both cases, but the mutual information of $I(Z;X,Y)$ with a vector containing both variables is 1 bit (assuming that the states of $X$ and $Y$ are distributed evenly).

If this effect would occur in the sensor input, the agent might look at the partial relevant information of each variable separately and would find that none of the variables contains any relevant information. This would be misleading, since the overall sensor input still contains relevant information. This is not only counter-intuitive, but also problematic, as it would cause the agent to discard variables that actually contain information if they were combined with other input variables.

### Redundancy

The other problem is *redundancy*, the case where the two variables $X$ and $Y$ contain the same information about $Z$. As a result, the sum of the amount of information each has about $Z$ is larger than the mutual information the joint variable $(X,Y)$ shares with $Z$.

$$I(X;Z) + I(Y;Z) > I(Z;(X,Y)) \tag{3.40}$$

This is the case when $X$ and $Y$ are highly correlated, and the effect is maximal when both variables have either always the same state, or are always in corresponding states. This is not as problematic as the first example. It would still be possible to identify which sensor inputs contain relevant information, but the AI might be fed with the same information several times.

### Unique Relevant Information

What we actually want is a measurement that can determine the amount of *unique relevant information* a certain sensor input contains, given the rest of the sensor input. We can address both problems by calculating the mutual information of A and $S_1$, conditioned on the rest of the sensor input. We shall define $S_{\setminus 1}$ as the random vector that contains all random variables in $S = (S_2, \ldots, S_n)$ but $S_1$. The resulting formula for the unique, partial relevant information then is

$$URI(S_1) = \min_{p(a|s) \in \pi^{opt}} I(A; S_1 | S_{\setminus 1}). \tag{3.41}$$

This can be expressed as the difference between the overall mutual information $I(A;S)$ and the partial mutual information of the sensor state that does not include $S_1$ as

$$I(A;S_1|S_{\setminus 1}) = I(A;S) - I(A;S_{\setminus 1}). \tag{3.42}$$

Not only is this often easier to compute, but additionally, this offers another good interpretation of the unique partial information. Since it is calculated as the difference between the overall relevant information, and the relevant information with all but one variable, it can be interpreted as the information an agent would lose about the environment when it would lose access to that part of its sensor input. Unique, partial relevant information for a specific variable $S_1$ thereby addresses both problems:

- redundancy: because an agent would not lose the information in $S_1$ if that information is also accessible through another variable in $S_{\setminus 1}$.
- synergy: If there is some information only available to the agent if it had access to both $S_1$ and $S_{\setminus 1}$, then loosing access to $S_1$, one of the two synergistic variables, would lead to the agent losing access to the synergetic information.

A more detailed mathematical treatment of Synergy and Redundancy, and a formalism for the decomposition of both into positive atomic units, can be found in (Williams and Beer 2010) and in (Harder, Salge and Polani 2013). The further decompositions are useful, but not necessary for this specific case, where they would needlessly complicate the formalism. Also, it should be noted that van Dijk's work (van Dijk et al. 2010) about goal oriented relevant information uses a very similar formalism, but applies it to a different problem.

Concluding, with the unique relevant information formalism it is now possible to measure how much relevant information is contained in a specific part of either the sensor input $S$ or the environment $R$. It can also be used to automatically determine which part of the sensor input is used by an AI game controller, and which part of the sensor input can be ignored to speed up computation or reduce the input state space.

### 3.6.3 Experiment: Unique Relevant Information Approximation

To demonstrate the functionally of the unique partial relevant information formalism I am going to revisit the strategy game experiment. Since Case 3 of the experiment had an actual increase of relevant information in regard to performance, we will use the Case 3 scenario to take a closer look at the distribution of the relevant information in the sensor input. This can help us understand what parts of the environment the neural network

AI is actually taking into account, and in extension, which elements contain information necessary to make good strategic decisions.

We assumed in the last section that our modified game mechanic forces the player to observe which of the opponent's creatures is able to retaliate, so the player can then attack a creature unable to do so. So, our hypothesis here is that the variables encoding the other player's stacks ability to retaliate should contain unique relevant information. To verify this, I shall now approximate the unique relevant information that is contained in the three one-bit random variables that encode, for each enemy creature if it is able to retaliate. For comparison, I also calculate the graph for the unique relevant information for the three one-bit random variables that encode the player's own ability to retaliate.

### Experimental Model

For this experiment we are using the same experimental model as described in the case 3 scenario in section 3.5.7. The difference in this experiment is how the mutual information for the fitness function is measures. Instead of measuring the overall mutual information we will use $I(A; S_1|S_{\setminus 1})$. $S_1$ is the retaliation indicator of the enemy creatures in the first case and the retaliation indicator of the player's own creatures in the second case.

The genetic optimization algorithm will then be used to optimize the fitness function, which is again a weighted combination, optimizing for high performance and low mutual information.

Due to the modification the resulting strategies should now be optimized to have low unique mutual information in the particular subset of their sensor input. In other words, the optimization should look for strategies that do well, but don't have to use information in that part of the sensor input. So, I am asking if it is possible to play the game well without looking at that part of the sensor input.

### Results

First, we again produce the scatter plots that includes data points for all strategies that have been tried out, resulting in the graphs in Fig. 3.8.

We can see that for low performance levels below 0.1, there are strategies that require not unique information, neither from the opponent's, nor from the player's own retaliation variables. This is consistent with our earlier observation for the case 3 scenario in Fig. 3.7. There we could see that there are strategies that needed no information from any of its sensor input to perform on that level. Consequently, there should also be a strategy for the same performance level, in the same game world, that requires no information from a

subset of its sensor input.

For higher performance level we can see that the lower bound on the mutual information is higher for the unique relevant information in the opponents variables, compared to the players own retaliation variables. This becomes even clearer in Fig. 3.9, which compares the two approximated unique relevant information functions. This indicates that the decision of which creature to attack depends more on the opponents ability to retaliate, then on the retaliation ability of the players own creatures.

Interestingly, there seems to be at least some information contained in the player's own retaliation variable, indicating that a high performing strategy needs to consider its own retaliation variables sometimes. One speculation here would be, that it would be good to wait with one stack of creatures, and thereby activate the ability to retaliate, if all other own stacks are also able to retaliate. This would leave the opponents with no good choices for an attack. But the exact reason for why there is information there is unknown.

This also illustrates the advantages and disadvantages of this information theory based approach. The quantitative analysis can reveal some information that is not necessarily visible form an analysis of the game. So, in this case, it indicates that at least some information needs to be processed from the agent's own retaliation variable. On the other hand, it does not necessarily reveal what this means. Why is this information relevant remains an unanswered question.

In general, the results support our hypothesis about the retaliation variable. At least some of the necessary information for a well performing strategy has to come from the other player's retaliation variable.

### 3.6.4 Discussion of Unique Relevant Information

The actual use of the unique relevant information analysis depends on which area of investigation it is applied to. Regarding the design of a game, a possible consequence of the unique relevant information analysis could be to either exclude the sensor inputs with low URI from the user interface, or make them less visible so they are not drawing the players attention away from important information.

If we were to design an AI to act in this world, we might also consider to exclude the parameters with no unique relevant information from the AI input, to reduce the amount of data processing.

If we relate this to the overall topic of adaptive agents then this analysis could also serve as the basis for some form of sensor adaptation. Assuming that we have an external adaptation process that selects and breeds those strategies that perform well and use little

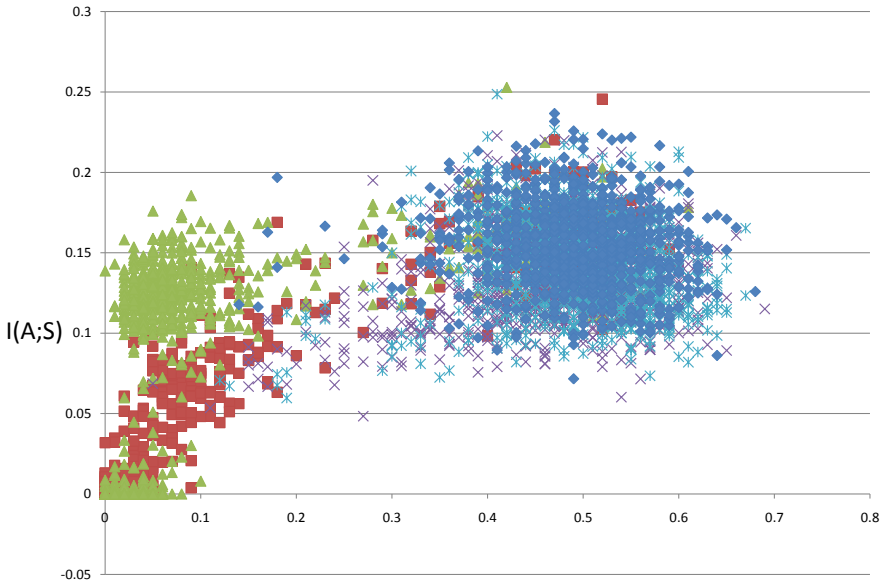
information, then the agent could be assume to be on the actual relevant information function.

The agents itself could then determine how much unique information is provided by a specific part of their sensor input, by calculating the unique mutual information. This contains one problem though, namely, that the agent was, in that case, not adapted towards minimizing the unique relevant information in that specific sensor input. So, it might be that there is a strategy that is cheaper in regard to that partial sensor input, but not cheaper in overall mutual information. In this case the agent might overestimate the information contained in that part of its sensor input.

## 3.7 Conclusion

A central assumption in the following chapters is that adapting agents apply the principles of information maximisation and information parsimony not only to information in general, but specifically to relevant information. I want to argue that agents adapt to obtain a maximum of *relevant* information, in the cheapest possible way. Furthermore, I want to assume that performing better requires a increasing amount of *relevant* information. To make this assumption plausible I aimed to convince the reader that we are likely living in a world that has non-zero relevant information that increases with agent performance (a case 3 world). In the following simulations, we will deliberately look at case 3 scenarios.

Furthermore, the formalism for unique relevant information allows an agent to determine how much relevant information is provided by different sensor inputs. It is even possible to approximate this from the agent's own perspective, given that we assume that some previous process (such as adaptation based on information parsimony and performance) has put the agent on the relevant information trade-off curve. In this case, the actual relevant information is similar to the mutual information between its inputs and outputs, and the unique relevant information for parts of the agent's sensor input can then be computed as well. This will form the basis for later arguments, as it enables us to quantify how some specific part of the sensor input is better than other parts, and thereby allows an adaptation during the lifetime of an agent that pays more attention to specific inputs.



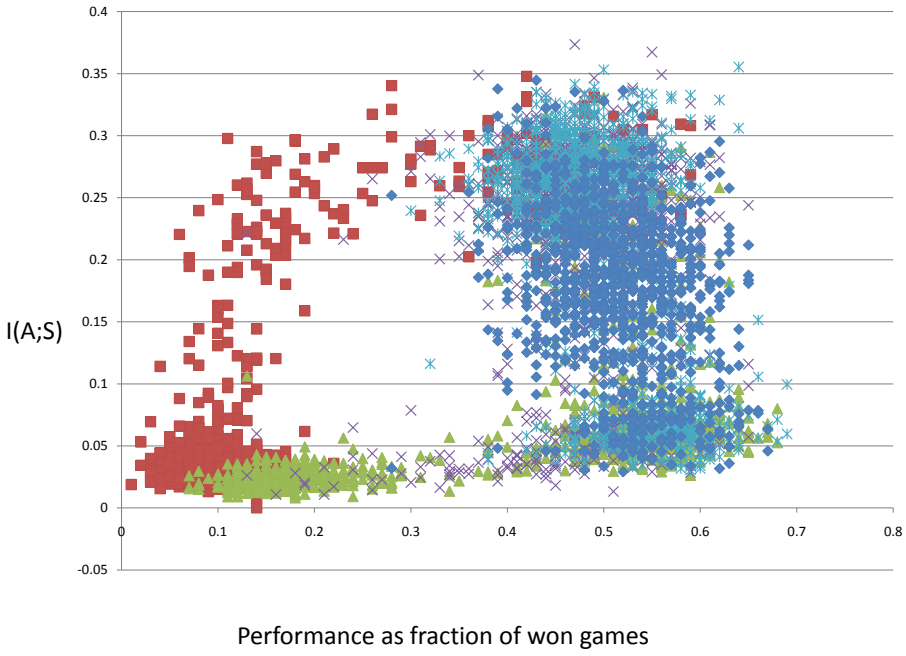






Figure 3.8: The first scatter plot shows data points for all tested strategies for the unique relevant information for the three one-bit variables that encode the enemy's ability to retaliate. It shows how all strategies above a win ratio of 0.1 seem to require at least 0.05 bits of unique mutual information. The second scatter plots contains data points for all strategies in regard to the partial unique relevant information for the three one-bit variables that encode the player's own ability to retaliate. In comparison to the first graph those random variables contain very little unique relevant information for higher performing strategies.

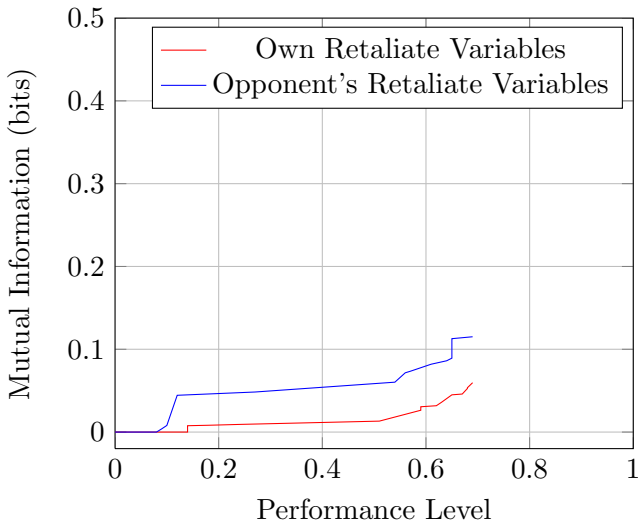



Figure 3.9: A plot showing the approximated partial relevant information function for the opponents retaliation variables (blue) and the player's own retaliation variables (red). This indicates that higher performance levels require more information processing of the opponent's retaliation variable then of one's own retaliation variable.

# Chapter 4

# Digested Information

## 4.1 Chapter Overview

The main question in this chapter is: "Is there something special about how one agent processes information that makes interaction beneficial for other agents?" Starting from our initial non-semantic, agent-centric model, the sensor input of the agent is represented as a set of random variables. No meaning is associated to them, and there is no *a priori* distinction between the random variables associated with sensor inputs from other agents, and those associated with sensor inputs from the rest of the environment. This chapter presents an argument and supporting simulations regarding the special properties of information coming from other agents, which would provide agents with a motivation to focus their attention on the information provided by another agent's actions.

I will first present an argument as to why special properties should arise for the information present in an agent's actions, which will also introduce the concept of *Digested Information*. I will then demonstrate for two models, the Fishworld and the Treasure Hunter model, how those properties can be measured, and that they actually rise to a non-trivial level.

In the larger context of this thesis, the aim of this chapter is to demonstrate that even a single agent, just motivated by increasing its own performance, will encode relevant information in its actions.

## 4.2 Digested Information Argument

Before we look at the results of actual simulations, I will outline the general argument regarding the class of models we defined in chapter 2. We established that, for all interesting

models, there is a certain non-zero amount of relevant information. The agent not only has to obtain this information, but also has to act upon it. If the agent's behaviour were not influenced by the obtained information, then there would be no point in investing the computational power to obtain and process it in the first place. This manifests itself in a non-zero amount of mutual information between the agents action variable $A$, and the state of the environment $R$.

### 4.2.1 Presence of Relevant Information in Actions

So the agent, by virtue of trying to optimize its own strategy, will change the communication channel between its sensors $S$ and its actuators $A$ to ensure that there is a certain amount of mutual information in $I(A;R)$. Following from the symmetry property of mutual information this also causes the agent's actions $A$ to contain information about $R$.

Furthermore, the very definition of relevant information, as the minimal mutual information between $R$ and $A$ for an optimal acting agent, suggests that this is not just any kind of information about the environment, but in essence the information relevant to the agent in question. To paraphrase, without any direct intent to communicate information, the agent nonetheless will cause its actions to contain the relevant information for its own strategy.

So, for example, if an agent encounters a hazard like a fire, and then starts to move away from it its actions contain some relevant information. Namely, that it would be good to move in a certain direction (away from the fire). Its actions do not contain the information that there is a fire, since the agent could also be fleeing from a predator, or some other kind of hazard. But then this kind of information does not seem to be relevant right now, because fleeing seems the right action in regard to both situations, and therefore all the relevant information is present.

### 4.2.2 Relation between Performance Increase and Relevant Information

Furthermore, there is a similar argument as to why an agent that performs better is also likely to encode more relevant information in its actions. In the chapter on "Relevant Information" I outlined how relevant information can also be defined for any suboptimal performance level. For every performance level the relevant information is the minimal mutual information of $R$ and $A$, for all strategies that perform at least this good, or better. It follows from the definition, that any higher performance level has either the same amount of relevant information, or more relevant information than the performance level it is compared with. This means that any increase in performance is likely to cause

an increase in the relevant information as well, or at least, keep the relevant information on the same level. Again, an agent just motivated by increasing its own performance, is now also motivated to increase the amount of information about the environment.

To illustrate, assume a scenario were agents are interested in moving away from possible predators. The optimal strategy keeps you as far away from any predators as possible. But not all predators are equally likely to be spotted, and some agents are better at spotting predators than others. So, while all agent's movement contains some relevant information in regard to predator positions and possible movement routes, the agents that have better sensors are likely to perform better. They might react to some kind of environmental information (the location of hidden predators) that other agents might not have. Therefore, their actions then contain more relevant information than the actions of other, less observant, agents. This would indicate that it might not only be good to observe other agents, but also that it might be better to observe better agents.

### 4.2.3 Relevant Information Density in Environment and Action Variables

Taking a closer look at the two random variables, $R$ and $A$, for which the mutual information is calculated we realize that it is reasonable to assume that the number of states $R$ can assume is, in general, much larger than the states of $A$. $A$ only encodes the different actions one agent can take, while $R$ encodes the entire state of the world, apart from the agent itself. There might be rare cases where the agent is more complex, i.e., has a larger state space, than the entire environment it is in, and consequently $A$ might be larger or similar in size to $R$. But then we need to remember that at least for the multi agent case, were multiple, similar agents populate the environment, that those similarly complex agents are also part of the environment from the perspective of the first agent. Their mere presence again increases the space of $R$ well beyond that of $A$.

Lets for now assume that $R$ is indeed much larger than $A$. We also assume that there is a certain amount of relevant information, expressed in a non-zero amount of mutual information $I(A;R)$. The overall amount of information a variable $X$ can encode is limited by the entropy $H(X)$,

$$H(X) = -\sum_{x} P(X = x) \log(P(X = x)). \tag{4.1}$$

The entropy itself is limited by the size of the alphabet $\mathcal{X}$ of $X$, the maximum amount of entropy is $\log |\mathcal{X}|$. Since the state space of $R$ is larger than $A$, the amount of information

$R$ can encode is also larger. But both variables have to encode the same amount of *relevant* information. So it follows that $A$ will have to encode the same amount of relevant information, but will have to do so with less bandwidth, as its channel capacity is limited by its self information $I(A;A) \leq \log(|A|)$. Therefore $A$ should contain relevant information in higher density, meaning that the ratio between relevant information and entropy of $A$ is higher. Similarly, if we formalize the variables $R$ and $A$ as collections of random variables, consisting of several random variables with state spaces with similar cardinality, then $R$ would contain more variables than $A$. In that case $A$ would have to put all relevant information into a small number of variables, while $R$ could encode the same information, very inefficiently, spread out over many variables.

To illustrate, assume that we wanted to know the outcome of a presidential election with two candidates. We want to donate to the winner before the election ends to gain his favour. The information regarding who will win is fully present in the environment, since could go and ask everyone who they will vote for. This will eventually lead to us knowing who will win the election (assuming for simplicity, that no voter will change their vote). But the information is badly formatted: every time we ask one voter, we take up one bit of information, but we gain very little information in regard to the one bit of relevant information we are interested in, namely who will win the election. If there exists another agent like us, with similar motivation, who may have done this already, we could instead try to learn who they gave their money to. This again would be one bit of information, but it would tell us who he thinks will win the election. Assuming that the other agent has done its own research properly, the one bit associated with its action will contain all the information we want to know, in one single bit.

All in all, it seems reasonable that the limitation in the state space of $A$ will increase the "density" of relevant information, and if another agent only had limited sensor capacity, then it might be reasonable to focus on this other agent, rather than the environment.

### 4.2.4 Transport of Relevant Information through Memory

For this case we assume that the agents in question are equipped with some internal states that serve as memory, as defined in chapter 2. So, instead of being purely reactive to the current sensor input, the agent can also use information encoded in its internal states to take a decision. Since the only information that matters is the relevant information, it is reasonable to assume that an agent optimized towards performance, encodes relevant information in its memory.

When an agent now acts upon its memory, it basically encodes relevant information, from the past, possibly gathered in another location, in its actions. In the here an now of

the agent, this information might not be present in any other form. So, it is possible, that relevant information is present in one agent's action that is not available in this location and at this point in time.

Again, the example of the agent fleeing from a fire comes to mind. The agent moving past another agent might not tell the other agent that there is a fire nearby, but it fleeing behaviour might indicate to the other agent that there is some kind of danger in another location it does not know about. If the first agent was not present, and therefore did not flee, then the second agent might never have learned this piece of information, because it was not available at that point in time and space.

### 4.2.5 Digested Information

Summing up our previous argument, we see that one agent's actions have several properties that distinguish them from other environmental variables. If we consider one agent's actions to be part of the environment of another agent, then, via the argument from the last subsection, these actions contain information not only relevant to the first agent, but also to the observing agent. This information, which I will call *digested information*, is beneficial because:

1. Agents encode relevant information in their actions.
2. The better they do, the more likely they are to encode more information.
3. The actions of an agent might exhibit a higher density of relevant information than other parts of the environment.
4. These actions might, in addition, provide information not available at that point in space and time otherwise.

In essence, when we talk about digested information, we describe the relevant information in the environment that is visible in another agent's actions. Importantly, note that this phenomenon does not rely on another agent's willingness to communicate information, since the agent-internal, motivating factor for producing digested information is the agents own performance. This argument does not rely on some kind of group fitness motivation, or an interest in reciprocal, cooperative information sharing.

All we claim here is an agent, only motivated by its own performance, is already digesting the relevant information in the environment and "ejecting" it back into said environment with its actions.

To further support our theoretical argument I will now present two simulation models that demonstrate the digested information principle. The simulations will support the four properties I argued for with quantitative measurements. This will also demonstrate that the overall principle can be expressed in quantitative terms, and said measurements can be used by another agent to focus its attention on other agents, or interact with them.

## 4.3 Non-Social Agent Simulations

In this section I will present two different models to illustrate and support the digested information concept. The first simulation, the "Fishworld" model, is a grid based search task. In this chapter I will mainly focus on this model, which will also be the central simulation model for the rest of the thesis. In this chapter it will be used to demonstrate the first three properties of digested information outlined in 4.2.5.

The second simulation, the "Treasure hunter" model, features an even simpler decision model, and it will be used later to investigate some social learning and information replication phenomena, that could be seen, but not easily demonstrated in the Fishworld model. The main purpose of the second simulation in this chapter is to demonstrate the digested information properties for a second, different model, and also to investigate the difference between agent decisions, and the actual visible results of an agent's decision.

### 4.3.1 Fishworld Model

First we created a model where an agent has a simple information acquisition task. This model will serve as a baseline for our question regarding how the performance of that agent could be enhanced by observing other agents.

The single agent model considers an agent situated in a grid world of size $n \times m$ with periodical boundaries (torus shaped) in which there is one single food location. The agent's location, and the location of the food are randomly generated at the start of the simulation, and the goal of the agent is to determine the location of the food source, in the shortest time possible. At each time step the agent can execute a move action which moves it one cell up, down, left or right. The agent then gets new sensor inputs; it is able to see the state of the world in all cells not more than 2 cells away from it, and perceives whether those cells are empty or contain a food source. After the observation, the agent then decides where to move next. This behaviour is repeated until the agent finds (but not necessarily reaches) the food.

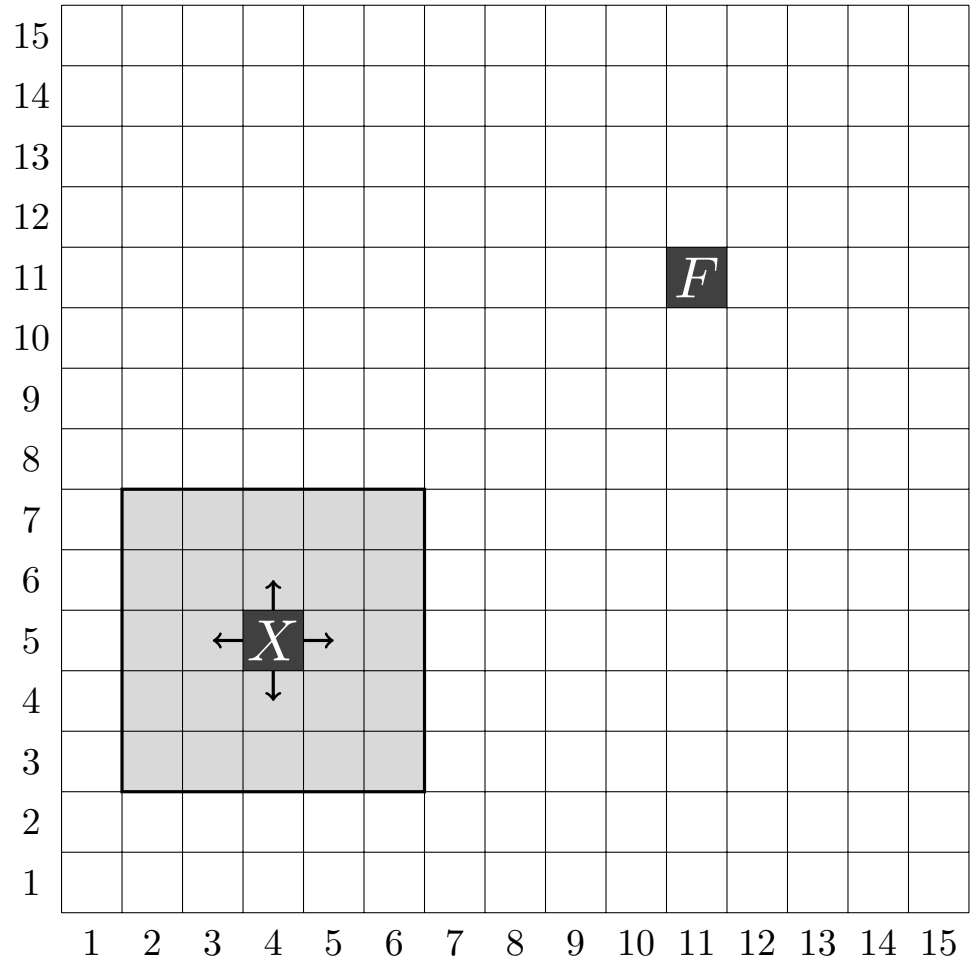



Figure 4.1: A sample grid world of the size 15 x 15. $X$ indicates the position of the agent, $F$ the position of the food source. The grey area is the cells visible to the agent in position $X$, and the arrows indicate the cells the agent can move to in the next time step.

### 4.3.2 Infotaxis Search

The basic algorithm to generate the single agent's behaviour is a modified version of the "Infotaxis" behaviour (Vergassola et al. 2007). The basic idea behind the infotaxis approach is for an agent to always act in a way that maximises the expected information gain. I modified this idea for a discrete grid world scenario. At each time step the agent chooses the action that has the highest expected reduction in entropy, with regard to the relevant information the agent is after. In case of the fishworld model this is the location

of the food source.

**Technical Details of Infotaxis**

My application of infotaxis to the fishworld scenario is a greedy information maximisation algorithm that selects a specific action from a list of possible action for each time step. The agent determines its actions by using an internal memory which stores information about the world. In fact, this internal memory acts as a Bayesian model for the location of the food source. More precisely, the internal memory is an array that has the same number of cells as the world. Each memory cell is associated with a cell in the world. Those cells store the probability for a cell to contain food, given the past experience of the agent.

Initially, all cells have the same probability of $1/(nm)$, in an $m \times n$ world. However, as the agent moves around, it discovers that some cells are empty or contain food. The distribution of probabilities is adjusted by setting either the probability of a cell to zero in the case that there is no food in it, or to one in case where the cell contains food. In all cases the probabilities of the remaining cells are normalized, to ensure that the sum of probabilities is always one. The remaining uncertainty about the location of the food source position is reflected by the probability distribution, and can be measured in terms of entropy $H(F)$, where $F$ is a random variable encoding the expected position of the food. As indicated before, the entropy computes to

$$H(F) = -\sum_{f} P(F = f) \log(P(F = f)). \tag{4.2}$$

To determine which way to go, the agent considers all its possible moves and decides which move has the highest expected reduction in remaining entropy, according to $\hat{F}$, its internal (Bayesian) model of $F$, the random variable encoding the food source. At each time step, the calculation of the expected entropy reduction of $\hat{F}$ is done by using the respectively current distribution of $\hat{F}$. Thus, the expected reduction of entropy is based on the agent's current "knowledge" about $F$.

To formalize this, we first have to define the set

$$W = \{w = (i, j) | 0 < i < (n + 1), 0 < j < (m + 1)\} \tag{4.3}$$

that contains the positions $w$ of all the cells of the grid world. The values $i$ and $j$ are the coordinates of the position on the grid world. Note that the random variable $F$ that encodes the food source position from the perspective of the agent uses $W$ as alphabet,

with $|W| = n \cdot m$ for an $n \times m$ world. Also, since we are considering a world with periodical boundaries both sides of the equation $(i, j) = (i + n, j + m)$ denote the same position. Depending on the position of the agent $w_a$, there is a set that includes all the positions that are visible to the sensor of the agent. If the agent now takes an action $a$ from a set of possible actions $A$ starting from the current position, one obtains a set $S_a$ as the new set of sensor inputs after the move.

To calculate the expected entropy reduction $\Delta H(a)$, depending on the action $a$, two main cases have to be considered. In the first case the actual location of the food source $f \in W$ would be in $S_a$, the sensor range after the action $a$ was taken by the agent. The agent assumes that this occurs with the probability of

$$P(f \in \mathcal{S}_a) = \sum_{f \in \mathcal{S}_a} P(\hat{F} = f) \tag{4.4}$$

in reference to the agent internal model $F$. In this case the agent's uncertainty after carrying out action $a$, $H(F_a)$, would be reduced to zero, and the reduction of entropy would be the difference $H(\hat{F}) - H(F_a) = H(\hat{F})$. In the other case, the location $f$ of the food source is not in $\mathcal{S}_a$. This occurs with a probability of $1 - P(f \in \mathcal{S}_a)$. In that case, we have to calculate an updated probability distribution for $\hat{F}$, called $F_a$. According to Bayes' rule, $P(F_a = f) = 0$ for all $f \in \mathcal{S}_a$, the resulting probability for all observed, empty locations to contain the food source is zero. The remaining locations are normalized accordingly to

$$P(F_a = f) := \frac{P(\hat{F} = f)}{\sum_{w \notin \mathcal{S}_a} P(\hat{F} = w)}, \; for\ all\ f \notin \mathcal{S}_a. \tag{4.5}$$

This divides the remaining non-zero probabilities, by the sum of their probabilities, normalizing the overall sum of all probabilities to 1. This updated version of $F_a$ can then be used to calculate the reduction of entropy in the second case, which is given by the difference $H(\hat{F}) - H(F_a)$. If we put all this together, the expected reduction of entropy for taking action $a$ is

$$\Delta H(a) = P(\hat{F} \in \mathcal{S}_a) \cdot H(\hat{F}) + P(\hat{F} \notin \mathcal{S}_a) \cdot \big(H(\hat{F}) - H(F_a)\big). \tag{4.6}$$

To summarize, each step the agent selects the action $a$ that maximises $\Delta H(a)$. If several actions lead to the same expected entropy reduction, the agent selects one of them at random. The sensors are then updated as described above, and this behaviour is repeated until the food source is located. Essentially, this behaviour implements a version of Vergassola et al.'s *infotaxis* search and I will refer to it as such in the subsequent text.

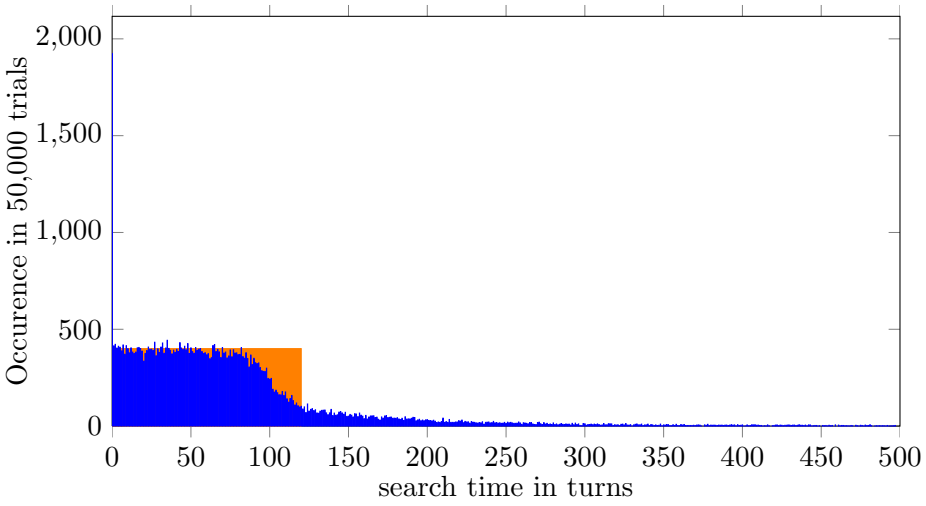



Figure 4.2: The distribution of time it takes the agent to locate the food source with an infotaxis search. The actual numbers correspond to occurrences in 50000 trial runs. The distribution approximates the theoretical optimum distribution (the orange rectangle in the graph), with a rough 4 % chance to find the food in round 1, and an even distribution of search time between the first 120 rounds.

#### Random Search

As a baseline for comparison I also implemented an alternative method of behaviour generation, i.e. random search. The random search agent basically checks its sensor every turn to determine if it can sense the food source. If it can sense the food it has finished its task of finding the food. If it cannot, then the agent will move into a random direction, with a chance of 1/4 for each direction. This behaviour will be continued until the agent finds the food.

### 4.3.3 Performance of Infotaxis

As a measure of performance I record the time it took the agents to locate the food source. On average, the agents with the infotaxis behavior outperform agents that chose their direction at random by a significant factor. For a $25 \times 25$ world, the average search time for the location of food, measured over 50000 trials, is ca. 76 turns for infotaxis agents, and around 450 turns for random walk agents.

### Optimal Searchtime

This compares well against the theoretically optimal searchtime for a non-social strategy which can be calculated as follows. A $25 \times 25$ grid world has 625 positions. The agent perceives 25 positions in round 0, therefore it has a chance of 25/625=1/25 to find the food source in round 0. No matter how the agent moves, the maximum amount of positions that could enter its sensor range that were not previously seen is 5. So, it would take at least 120 turns to sense all positions and thus have a probability of 1.0 to find the food. None of those positions are more or less likely to contain food, so the order in which they are searched can be considered arbitrary. By this reasoning we can deduce that the probability to discover the food source prior to or in round $t$ grows linearly with $t$. The average search time for the second case is half of the maximal search time of 120 turns, i.e. 60. Hence, the two different cases have the expected search time of 0 and 60 respectively, therefore the optimal search time calculates to

$$0 \cdot \frac{1}{25} + 60 \cdot \frac{24}{25} = 57.6. \tag{4.7}$$

In general, the optimal search time for an $n \times m$ grid world, with a sensor range of $r$ is

$$0 \cdot \frac{(2r+1)^2}{(n \cdot m)} + \frac{n \cdot m - (2r+1)^2}{2r+1} \cdot \left(1 - \frac{(2r+1)^2}{n \cdot m}\right). \tag{4.8}$$

### Search Time Distribution

Taking a look at Fig. 4.2 we see that the distribution of search times approximates an agent with an optimal strategy. There is a high probability to find the food in the first round, and it looks roughly equally probable to find the food in any of the next 100 rounds. This is not too surprising because in this simple scenario infotaxis behaves very similar to exhaustive search, which would be optimal.

The main difference I observed were a few instances in which the agents take a substantially longer time to find the food source. Closer inspection of those simulations shows that the agents sometimes get trapped in a local optimum of the greedy infotaxis search. Since the agent only considers the information gain for its immediate next step, it is possible that it ends up in a situation where all the cells it could reach with one step are already explored. In this case, the next direction is chosen at random. The agents do not necessarily move towards the closest patch of unexplored territory. Visual inspection of the agent's behaviour indicates that in those cases it is entirely possible for the agent to perform a random walk for considerable time before finding an explorable area again. A

possible way to circumvent this for future research would be to give the agent the ability to consider several future steps in deciding on an action in order to give it a more directed walk towards areas where its internal model has non-vanishing probabilities.

### 4.3.4 Relevant Information Encoding

I now want to address one of the core theses expounded earlier: Using the information-theoretic framework, I aim to verify the assumptions about the relevant information in the agent's action. The question I want to answer is how much information does the agent's action contain about the position of the food source?

To do so quantitatively, I ran 100,000 single agent trials and recorded the states of the random variables:

- $A$: the action of the agent, specifically the direction of its last move.
- $F$: the location of the food source relative to the agent.

In each simulation the agent started from a random position and repeated the infotaxis search until it found the food source. Once the food source was discovered the simulation ended immediately, and the next trial run started, with a new initial position. The agent's actions and the relative food source position were logged after each time step.

Based on this data I am able to calculate the joint distribution of $P(A, F)$. This makes it possible to calculate the conditional entropy of $H(F|A)$ by simply summing over the conditional entropies for each of the four actions

$$H(F|A) = -\sum_a P(A = a) \sum_f P(F = f|A = a) \log(P(F = f|A = a)). \tag{4.9}$$

Figure 4.3 shows the probability distributions of the relative food source locations for each move action. Two things become clear from the picture. The resulting distribution for the different actions are similar, if rotated regarding their corresponding action. Nothing in this simulation favours any specific direction, the action called "north" is just labelled thus by arbitrary convention. Also, the conditional distributions for the food source location given the agent's last action are not uniform. There is an area of zero probability in that part of the world that was observed by the agent before it decided to move, and there is an area of high probability in the area the agent moved towards. This non-uniform distribution of $H(F|A)$ indicates that there is information in $A$ about $F$ as the mutual information should be larger than zero:

$$I(F; A) = H(F) - H(F|A) \geq 0. \tag{4.10}$$

Based on the data visualized in Fig 4.3 this value can be computed. In this specific case we consider a 20 world, which contains 400 possible locations for the food source. *A priori* nothing is know about the location, so I assume a uniform distribution, resulting in an entropy for $F$ of

$$H(F) = -\sum_{f} p(f)\log(p(f)) = 400 \cdot \frac{1}{400}\log\left(\frac{1}{400}\right) \approx 8.643856 \tag{4.11}$$

Following from Eq.(4.9) I can calculate the conditional entropy $H(F|A)$ based on the data visualized in Fig. 4.3, by calculating the conditional entropy for each action $a$ as $H(F|A = a)$, and then calculate the weighted sum over all actions $a \in A$. The result is ca. 8.514056. Subtracting one value from the other, as in Eq. (4.10), I get 0.1298 bits of mutual information between $A$ and $F$. This value indicates how many bits of information the action of an agent contains, on average, about the location of the food source. Note though, that this calculation is based on the uniform prior for $F$, which was chosen for this general, objective look at the mutual information, as it corresponds to the actual distribution of the food source location in the world. If an agent were to evaluate the mutual information with a different prior for $F$, then the results may change.

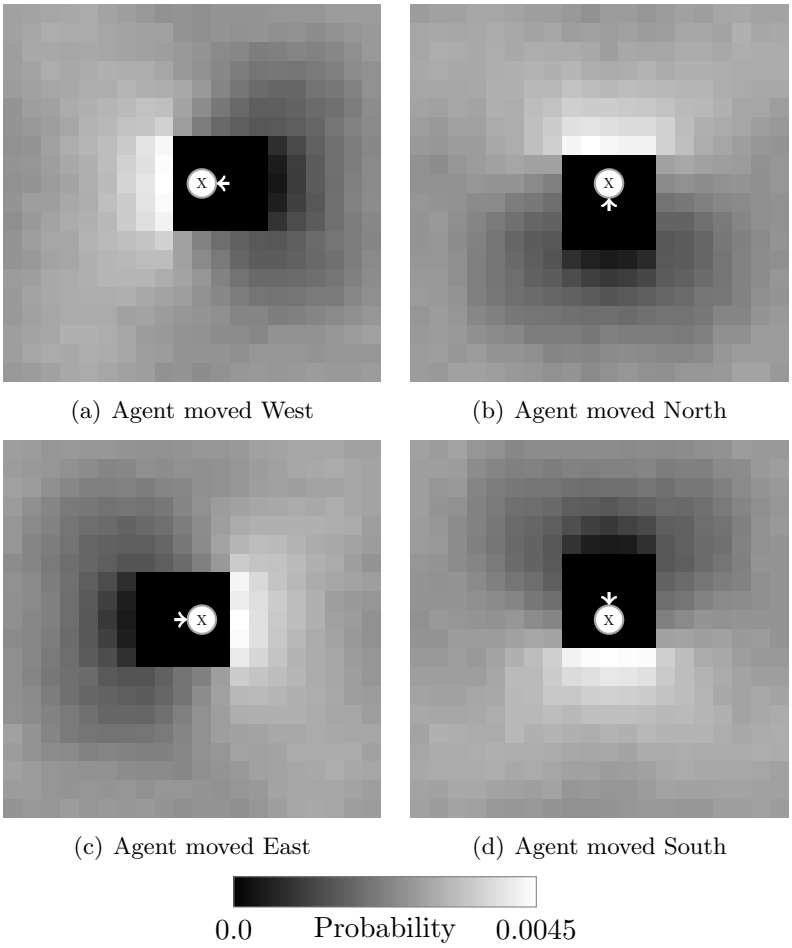
**Stigmergy vs. Digested Information**

An alternative way to determine the mutual information would be to use the marginal distribution of $F$ as a prior. This distribution can be obtained from our sampling by summing up the probabilities for specific outcomes in $F$, over all outcomes in $P(A, F)$. Fig. 4.4 shows the probability distribution of the relative food source location, regardless of actions $A$. To avoid confusion, I will call this distribution $F_p$.

It is noticeable that $F_p$ is not uniformly distributed. Calculating the entropy of $H(F_p)$ results in 8.599144 bits, which is lower than the 8.643856 bits of the uniform distribution. If I were to calculate the mutual information between the agents action A and the relative food source location with this marginal distribution $F_p$ as a prior, rather than with an assumed uniform distribution $F$, the resulting information would be lower.

$$I(F_p; A) = H(F_p) - H(F|A) \approx 0.084088 \tag{4.12}$$

The conditional probability here is still the same $H(F|A)$ obtained from statistics, as $F$ and $F_p$ are the same random variable, just with different prior distributions. Their conditional distributions after observing $A$ are identical.




(a) Agent moved West (b) Agent moved North

(c) Agent moved East (d) Agent moved South

Figure 4.3: The four figures show the probability distribution of the food source location in a 20 × 20 world relative to the agent and dependent on the agent's last move. The agent's current position is denoted with an X, and the arrow indicates the agent's last movement. The black areas have zero probability, as they are all within sensor range in the agent's last position.

Since this is a different value than the one calculated for $I(A;F)$ the question that now arises is: Which of those two calculations actually tells us how much information an agent's actions contains about the location of the food source? Or, what does the value in Eq. (4.12) actually mean?

To answer this it will be useful to decompose the information gained from an agent further. Looking at Fig. 4.4 we can see that, even if we could only observe an agent's position but *not* its last movement, we would still get some information about the food

source location. The distribution of $F_p$ is not uniform, and if we were to compare it to the *a priori* uniform distribution of the food source location $F$ we could measure an average information gain for observing an agent's position as

$$H(F) - H(F_p) \approx 8.643846 - 8.599144 = 0.0044712. \tag{4.13}$$

While the agent's position is initially random in relation to the food source, the agent's repeated actions change the world in a systematic, non-random way. In general, when an agent acquires information from the environment and acts upon it then this can change the environment in a way that reflects this information in the environment itself. In theory, this could be used to store information outside an agent, or communicate said information to other agents. This effect has been described as a form of stigmergy in (Klyubin, Polani and Nehaniv 2004), where the effect was also quantified for a similar grid world scenario. The data suggests that this is a similar phenomenon. The agent explores the area, and its position alone contains information about the location of the food source. In our specific case this information essentially conveys that the source is less likely to be close to the agent, because the agent would then have been more likely to have found it, and the simulation would have already been over.

Therefore, I call the information described by the difference between the uniform distribution, and the food source distribution knowing an agent's position, formalized as $H(F) - H(F_p)$, *Stigmergy information*.

Realizing that the agent's position in itself already contains information, we can then rephrase our initial question. How much *more* relevant information does knowing an agent's action provide, if we already know the agent's position? This requires a decomposition akin to the unique relevant information analysis discussed in chapter 3.

To compute the unique information in the agent's action, I compare the remaining average entropy for just knowing an agents position $H(F_p)$, with the entropy of $F$ when I also know the agent's last move, which is $H(F|A)$. This essentially is a mutual information computed in Eq. (4.12), $H(F_p) - H(F|A)$, which I will call *unique action information*(UAI). The resulting value for the UAI in bits for a $20 \times 20$ world is 0.084088.

Table 4.1 gives an overview of the different values and offers a comparison to the random strategy's values. The digested information $I(A; F)$, which is the overall reduction in entropy from an *a priori* position of maximum ignorance to the *a posteriori* distribution after observing an agent's action decomposes into the sum of a.) the stigmergy information gained from observing an agent's position and b.) the information the agent's actions provide on top of that (unique action information).

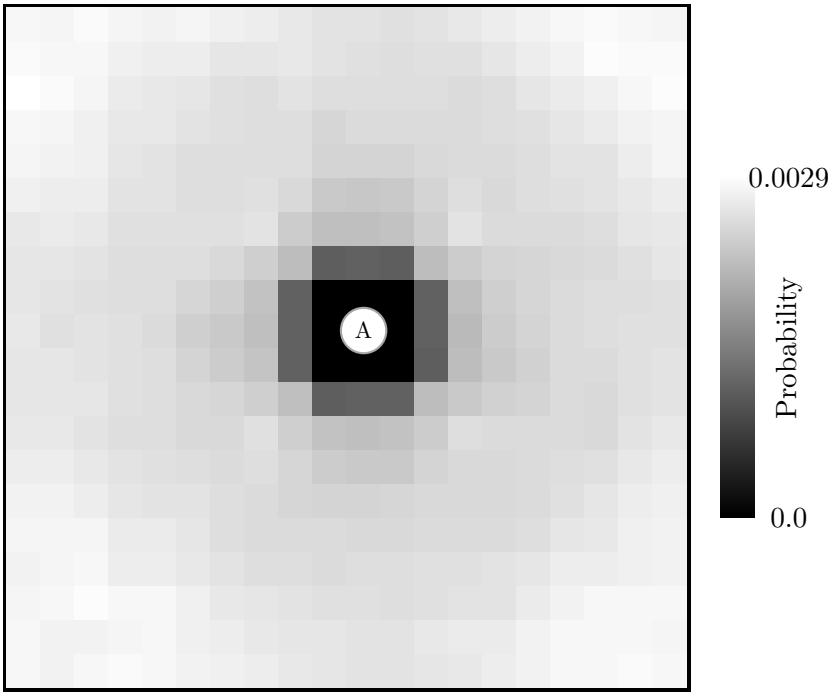



Figure 4.4: The probability distribution of $F$ given the position of an agent (with infotaxis behaviour) who just moved, averaged over the different possible actions. This illustrates how the agent's position, even without observing the agent's action, contains information about the food source location.

| | Infotaxis Agent | Random Agent |
|---|---|---|
| Uniform Dist.: $H(F)$ | 8.643856 | 8.643856 |
| Position Dependent Dist.: $H(F_p)$ | 8.599144 | 8.531202 |
| Stigmergy: $H(F) - H(F_p)$ | 0.044712 | 0.112654 |
| Action Dependent Dist.: $H(F\|A)$ | 8.514056 | 8.519220 |
| Unique Action Information: $H(F_p) - H(F\|A)$ | 0.084088 | 0.011983 |
| Digested Information: $I(A;F)$ | 0.129800 | 0.124636 |

Table 4.1: Overview for the different amounts of information to be gained by observing the infotaxis and random agent. All measurements in bits.

### Comparison to Random Strategy

Table 4.1 also contains the information values gained from observing an agent using the random strategy. Unless the food is found, the agent will move in a randomly chosen

direction. Observing this agent's actions provides less information than observing the infotaxis agent, but still contains a considerable amount of information. This might first be counter-intuitive, as the agent seems to not react to any of its sensor inputs (choosing its actions at random), and thereby it is unclear how information about the food source can be observed in the agent's actions. But in fact, the agent does react to its sensor input, specifically when it decides whether it is going to move. At that point the agent checks if the food source location is in the sensor input, and then reacts accordingly, possibly ending the simulation, and thereby also the recording of data.

This is reflected in the composition of the gained information. As we see in Table 4.1 most of the information gained comes from the actual position of the agent (via stigmergy), and very little is encoded in the actions itself. Note though, that the random agent has more information about the source location stored in its position than the infotaxis agent. Comparing the action independent distributions of Fig. 4.4 and Fig. 4.5, it looks like the random agent created a more informative gradient of probabilities around its position. This might be an effect of a longer time spent in the environment per simulation. As the random agents roughly need six times longer to locate the food source, this might give them more time to inject information into the environment via their actions, and thereby their position might contain, on average, more information.

Nonetheless, the overall information gained from the random strategy is worse, and if we were to just focus on the information gained from the actions, then this difference becomes even larger.

#### Comparison to Non-Agent Environment

Another aspect of the digested information concept suggests that there might be more information to be gained from observing other agent's than from the rest of the environment. This leads to a comparison between how much information can be gained from observing an agent vs. observing the environment, minus the agent. This is also helpful to get a scale to measure information gain against. Currently all we know is that there is some information in the agent's actions, but it is unclear if 0.1 bit of information is a substantial amount.

Again, let's look at a $20 \times 20$ grid world, which has 400 cells. 399 of them are empty, one contains the food source. If the agent were to observe one cell at random, two cases are possible. If the food source is observed, the entropy will be reduced to zero immediately. In the other 399 cases the entropy will be reduced to a uniform distribution with 399

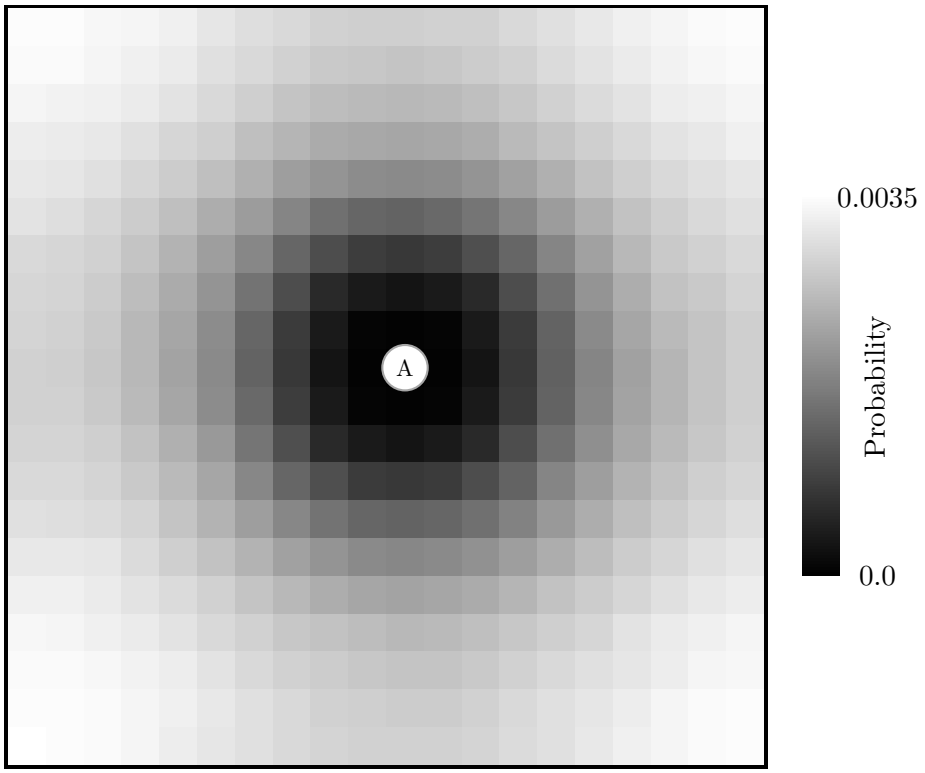



Figure 4.5: The probability distribution of $F$ dependent on the observation of a random agent who just moved, without taking the last move into account. Even the random agent encodes information about the food source in its location, because every turn the agent checks if it can sense the food, and only moves if it cannot.

instead of 400 states. The average reduction in entropy can therefore be computed as

$$\frac{399}{400} \cdot (\log 400 - \log 399) + \frac{1}{400} \cdot \log(400) \approx 0.02523. \tag{4.14}$$

If we compare the information gained from the action of an agent with sensor range of 2 ($\approx 0.13$ bits), to the information gained from observing a single cell ($\approx 0.03$ bits), then the information from the agent is significantly higher. This is true, even if we take into account that the cell only has two states, while the agent's actions have four states, and hence twice the bandwidth of the cell.

Note that both the information gained from another agent, and the information gained from a cell in the environment are determined here based on a uniform prior, so the indicate the information gained if nothing was yet know about $F$. If an agent had prior knowledge

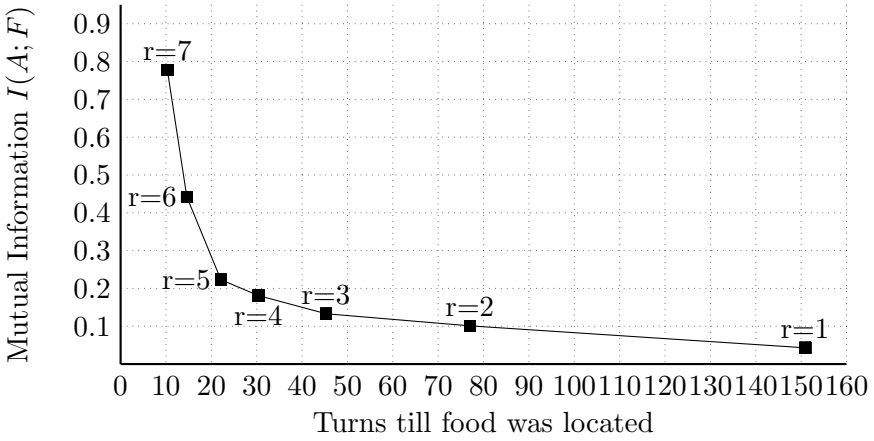



Figure 4.6: Graph depicting the relation between encoded mutual information and search time for different sensor ranges. Data taken over 10.000 simulated trials. Labels on each note indicate the associated sensor range. An increase in sensor range lead to an increase in both performance and mutual information.

about $F$, then those values would be different. Here I only consider the objective outside view, which is based on a prior of maximal uncertainty.

### 4.3.5 Performance Dependency

I also predicted that an increase in performance would lead to an increase in digested information. To support this claim I implemented two modifications of the original fishworld model that should increase the performance of the individual agent. This should allow us to observe how the encoded relevant information is affected by a change in performance.

#### Sensor Range Increase

The first modification is an increase in sensor range. Instead of being able to sense only those world cells that are not more than 2 squares away, the agent can now see squares up to $r$ square away. This should allow an agent to take in more information per turn, increase its performance, and also increase its encoded relevant information. Note that this change also increases the agent's capacity for information intake, which in turn changes the optimal search time of the agent, since this new agent can take in more unexplored cells per turn than the more limited agent.

The graph in Fig. 4.6 shows how the change in sensor range $r$ affects both the performance and the mutual information $I(A;F)$. With increasing range the performance of the

agent increases, as predicted. The agents with longer sensor range take less time to find the food source. More importantly, the increase in range also leads to the agent's action containing more information about the food source location. So, in this specific case the performance increase is accompanied by an increase in the mutual information between the agent's actions and the food source location, which is in essence an increase in relevant information.

Note though, that the increase in sensor range also increased the agent's intake of information. While this increases both the agent's performance and the encoded relevant information, it is unclear if this is, as argued for earlier, a result of moving to a different point on the relevant information trade-off curve, or simply a result of more information being available to the agent, which is then processed through to the other end. Either case would support the original argument, but I thought it prudent nonetheless to design a scenario were the increase in agents performance is caused by a change of the processing alone.

#### Horizon of Information Maximization

My second method to increase the agent's performance is based on changing how far into the future the agent maximises his information gain. In the original, greedy implementation, the agent would only consider what the result of its next time step would be. As described earlier, this would sometimes result in the agent "being stuck" in an explored area, where all adjacent cells have been explored. The agent would then act randomly, where a better strategy would be to move directly for an unexplored area further away. Therefore, I modified the original infotaxis adaptation for the gridworld scenario with a changeable horizon, so it would now optimize its behaviour for expected information gain over several turns.

I established earlier that in the state $t$ the information gain for the next turn can be computed as

$$\Delta H(t,a) = P(F \in \mathcal{S}_a) \cdot H(F) + P(F \notin \mathcal{S}_a) \cdot \big(H(F) - H(F_a)\big). \tag{4.15}$$

This equation consists of two terms, one that corresponds to the information gain if the food source is found in the next time step: $P(F \in \mathcal{S}_a) \cdot H(F)$, and another that corresponds to the information gain if the food is not found: $P(F \notin \mathcal{S}_a) \cdot \big(H(F) - H(F_a)\big)$. If we want to expand the potential information gain following from an action $a$ further, we only need to look at the case where the food was not found; in the other case the simulation would end anyway, and there would be no information to gain. For the case in which the food

was not found it is also clear what situation we are in. The agent is in a new position, and every cell in the sensor area does not contain the food. Lets call this state $t + a$.

Thus, it is possible for $t + a$ to recursively construct a virtual future state of the agent's internal model, and apply the infotaxis algorithm to that state. The same formula as for $\Delta H(a)$ can be used to calculate which new action $a_1$ will yield the highest expected reduction in entropy. The resulting information gain can be expressed as

$$\max_{a_1 \in A} \Delta H(t + a, a_1). \tag{4.16}$$

This amount of information can then be included into the agents consideration in the first step, as a potential information gain available in the situation resulting from its action $a$. This can be expressed as

$$\Delta H(t, a) = P(F \in \mathcal{S}_a) \cdot H(F) + P(F \notin \mathcal{S}_a) \cdot \big(H(F) - H(F_a) + \max_{a_1 \in A} \Delta H(t + a, a_1)\big). \tag{4.17}$$

This recursive function allows us to compute the potential information gain of an action $a$ for several steps into the future. Two things have to be considered though. Obviously, as a recursive formula the potential information term itself contains more potential information terms. So, to compute the value, one has to determine a cut-off horizon, a point in the future after which the potential information gain is not considered any more. Furthermore, this computation requires the creation of a lot of "virtual" memory states on which infotaxis is computed. Each step multiplies the number of virtual states with the number of available actions. This makes computing this value for long horizons infeasible.

Nevertheless, for short horizons it is possible to compute, and should improve the performance of the agent. Especially regarding the previously mentioned performances where an agent would get "lost" in a previously explored area, and had to resort to random search. Now the agent would be able to detect unexplored areas up to $n$ squares away, were $n$ is the horizon of its search.

In Fig. 4.7 we see the results from another 10 000 simulations, for different look ahead horizons. We can observe that an increase in the temporal horizon increases the performance of an agent, and at the same time also increases the agent's mutual information between its actions and the food source location. Both increases here are much smaller than in the sensor range example, which is likely due to the fact that the sensor input capacity of the agent remained unchanged. The agent could only improve based on better information processing, and not because it had access to more information, as in the previous example. This also means that in this case the agent could not surpass the lower

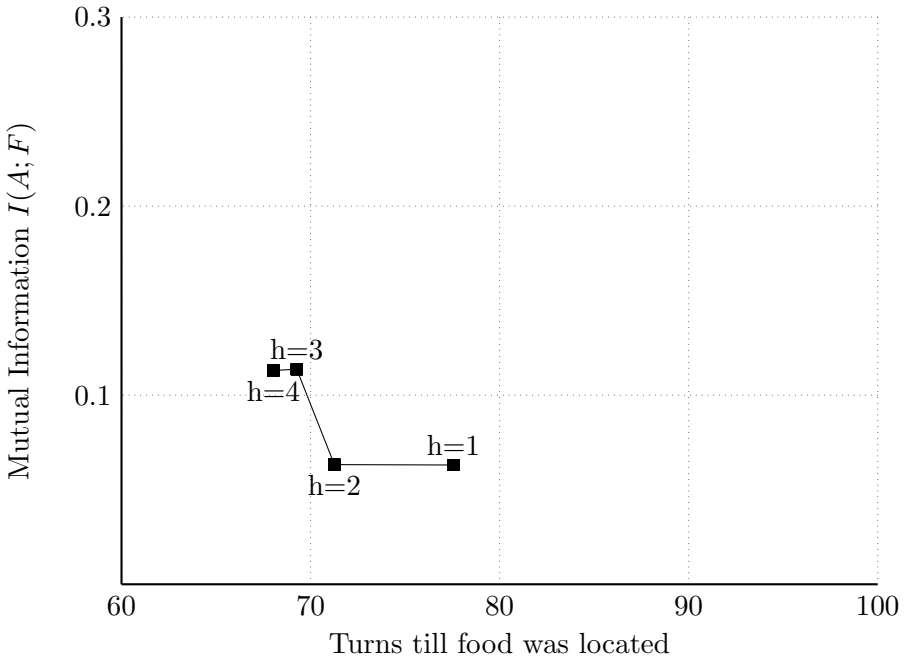



Figure 4.7: A graph depicting the relationship between the agent's performance and mutual information between actions and food source location for different look ahead horizons $r$. The value of r indicates for how many steps into the future the agents tries to maximise its expected reduction in entropy, with $r = 1$ being greedy infotaxis.

bound on optimal search time calculated earlier. Concluding, the simulations seem to support, for both cases, that an increase in performance will lead to an increase in digested information.

### 4.3.6 Discussion of Fishworld Model

Summarizing the result, the Fishworld model seems to support the properties predicted in the digested information argument in section 4.2. For these model it is true that the single agent's actions contain information about the relevant information (the food source location), and this is achieved without any motivation on the agent's part to communicate said information. Even the random behaviour, which did very little in terms of information processing, still injected a certain amount of information about the food source location into the agent's actions.

It also appears that the per bit density of information about the food source location is higher in an agent's action than it is in the cells in the environment. This, of course, could be averted by choosing a different representation of the states of the environment,

or by designing the simulation differently. Nonetheless, this should at least indicate the possibility of such a higher density in a model that is not particularly contrived.

Furthermore, both the external modification in sensor range and the behaviour modification of longer look ahead for the information maximisation, demonstrated a relationship between an increase in performance and an increase in relevant information.

Only the last point, the transportation of information through memory, remains somewhat unsupported. Looking back at Fig. 4.4 and Fig. 4.2 we can see that both an agent's current location, and action contain information about the probability of the food source location in cells far out of the agent's current sensor reach. This could be used to argue that somehow this information must have gotten into the agents current actions through its memory. But then the distribution in Fig. 4.5 shows a similar distribution for the random agent, where it is clear that this agent did not act on any kind of memory. The implementation of a random agent could be made as a purely reactive, memoryless agent. Nonetheless, the random agent's location contains information about the food source location. The likely explanation here is a process called stigmergy, as discussed in (Klyubin et al. 2004). Since the agent's actions change the environment, it is possible to systematically change the environment to contain some information. Technically, most search algorithms in this scenario would do this, since they are likely to move the agent closer to the goal. As a result the agent's position should generally contain some information about the goal location. So, in a sense, the agent uses the environment as an external memory, in this case specifically its current position. This or course further complicates an analysis of how the information contained currently in both the agent's actions and the agent's position has gotten there.

The decomposition into stigmergetic and unique actions information is helpful here, as it shows a clear difference between the random and the infotaxis strategy. The comparatively high amount of unique action information for the infotaxis strategy indicates that the actual action of the agent contains the majority of the digested information, while the random agent provides mainly stigmergetic information. Also, if we compare how the probability distribution in cell far away from the sensor reach changes with and without observing the last action, we also see that knowing the actions of the infotaxis agent has a much higher impact here. This further indicates that in the case of the infotaxis agent information about these locations is actually transported in the internal memory.

In any case, it seems to be clear that both agents somehow transported information from a different location or time to the present location and time. Furthermore, the question if an agent's actions contain information that is not available otherwise, should become much clearer in the next chapter, when I will demonstrate how the information

displayed in one agent's actions can be used by another. Concluding, this specific model acts as predicted by the digested information argument, and its results support the initial concept.

### 4.3.7 Treasure Hunter Model

I will also introduce a second scenario where it is possible to only observe the decision the agent takes, without taking in additional information, such as the agents position, or the outcome of that decision. This simulation shares several properties with the simulations discussed in the related work section on Social Bayesian Learning, but we will deal with the social aspect of this scenario only in the next chapter. For now it will just serve to demonstrate the main properties of digested information for a second model. Therefore, the agents in this simulation are not able to observe anything about other agents. I will describe what could potentially be observed, to determine if there is digested information in those observable actions, but in this simulation, the agent itself are not capable of observing anything related to another agent.

The agents in this scenario are treasure hunters, they are looking for a specific treasure located in one of $n$ locations. Each turn an agent can choose one location to go to, and look there for the treasure. It will then leave the harbour where all agents are located and will be able to observe the state of the chosen location (containing the treasure, or not). Once the agent found the treasure it stops playing. It is the replaced by a new, ignorant agent, so the agent population remains constant. The treasure is placed in a random location at the beginning of the simulation.

The agents' actions can be observed when they are leaving for a specific location, and it is then clear what location they are going to. The agents cannot be observed coming back, and it cannot be observed if the agents found the treasure.

Since we are asking how much relevant information is present in a single action I have to introduce some additional constraints. If agents were identifiable, and we could use the context of the simulation to inform our decision, it would be easy to just look for an agent that does not return. But what we want to model is the information one can get if one were to turn up at a random time during the simulation, and just observe the next agent leaving. To capture this I added the following constraints: Once agents stopped playing, they are replaced with new agents which have no information about the world. Also all agents are indistinguishable, and the order of their moves can be considered as random. Also, agents cannot be observed returning from any location, regardless if they found treasure or not. Formally, all that can be observed is one random variable $A$, which has as many states as there are locations $|A| = n$, and which state indicates which location

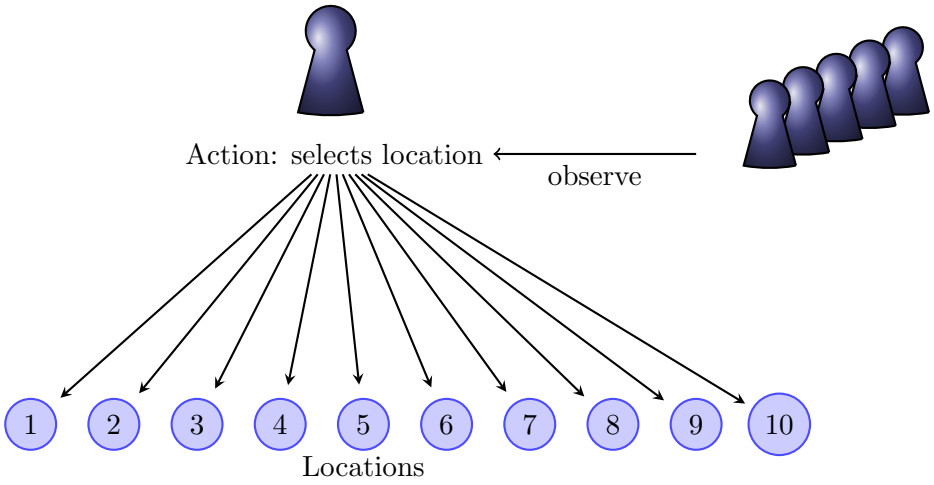



Figure 4.8: Diagram of the Treasure Hunter Scenario. The active agent chooses on each round which location to explore. The observers can only observe that choice.

an agent is locking for the treasure.

### Agent Behaviour Generation

Similar to the grid world simulation the agents have an internal memory variable that is basically a Bayesian Model of the treasure location. The treasures location will be denoted by $T$, and the internal model as $\hat{T}$. When the agent has to decide which location to visit it will act in a way consistent with the infotaxis approach, going to that location which will lead to the greatest expected information gain. For this simple model this is identical to going to the location with the highest probability of the treasure being there. In case of a tie, the agent would chose one of the optimal locations at random. Given that the agents initialize their model $\hat{T}$ with a uniform distribution; this means the non-social single agent basically choses a random location it has not previously visited. Since nothing else can be done to gain more information, as agents in this simulation are not capable of observing each other, this implements an optimal strategy for the agent. For a world with ten locations it takes $\approx 5.5$ tries to find the location with the treasure.

### Encoded Information

Recording the observable actions taken by the agents we can observe a distribution for $A$ as depicted in Fig. 4.9. Keep in mind that those agents that have found the treasure

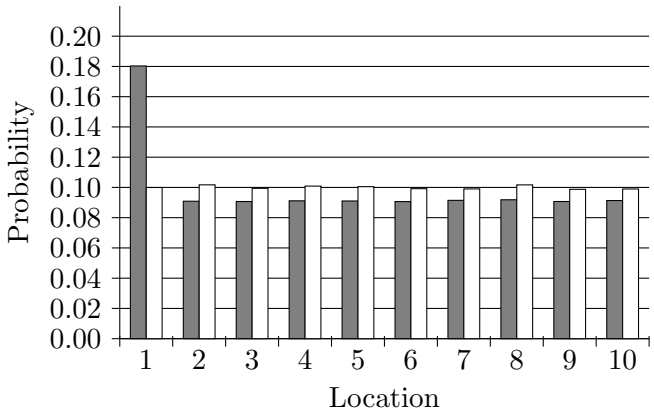



Figure 4.9: A plot showing the probability of observing an agent going to a specific location, if the treasure is located in position 1 and there are 10 locations. The grey bars are for observing infotaxis agents, while the white bars show the observations of random agents. The data was gathered by 100,000 observations on simulated agents.

are replaced with ignorant new agent. $A$ is just the recording of all actions taken by the agents, as they are indistinguishable.

Fig. 4.9 shows that the actual location of the treasure has significantly more visits than the other locations, even though none of the agents going out know where it is. This again, is an effect based on agents making decisions on previously processed information. The agents know where the treasure is not, and this influences their decisions. In turn, this information is detectable in their actions.

For this specific case this can be explained by looking at all possible outcomes of an agents search. Any agent is just randomly and exhaustively searching through every location, never visiting one twice. It cannot gain any insight into which location would be more likely (in the single agent case), so the time it takes to find the treasure is uniformly distributed between 1 and the number of possible location. So all search time duration from 1 to $n$ are equally likely. For a world with 10 locations this means the average search history of an agent has a length of 5.5, but each of those search histories contains the location of the treasure at the end. While we cannot know if an agent is currently going to the location with the treasure, we know that 1 out of 5.5 actions are going to the right location.

Based on the observed joint distribution of $A$ and $T$ it is possible to compute the mutual information between the actions taken and the location of the treasure. In this case the marginal distribution of $T$ for all observed events is uniform, which indicates

that there is no stimergic information in this model. This is consistent, as all that can be observed are the actions itself, and the actions do not change anything in the state of the world as the move action in the fish world scenario did.

So, all the mutual information in $I(A;T)$ is entirely contained within the actions. The mutual information in this case computes to $\approx 0.042799$. We can compare this to a random strategy, where the agents just choose a location at random and do not have any memory. Taking a look at the sampled distributions of actions in Fig. 4.9 we can see that the distribution is nearly uniform, hence the (empirical) mutual information for this case is $\approx 0.000026$.

We can also compare those values to the information gained from observing a random location, which is the information an agent gains when it makes the decision to visit a specific location. The information for observing one random location is $\approx 0.468996$. This is actually higher than the information provided by the agent's actions, but not necessarily violating the earlier argument. Higher density was assumed for cases where the state space of an agent's action is significantly smaller than the state space of the environment. While this seems to be a reasonable assumption in real world scenarios, it is not the case here, where the agent's actions have as many states as the environment itself. Due to the simplicity of the model it is not really feasible to implement a better strategy for a single agent, so there is no comparison for different performances here, apart from comparing the actual strategy with a random strategy.

One thing that is highlighted in this model is that the storage of information in the memory of one agent is crucial for it to display this information when it takes the action to embark to another location. The decision taken at the harbour is influenced by information only available elsewhere, and is then displayed at the harbour, via the agents actions. In summary, this simulation also seems to display the properties outlined by the digested information concept.

## 4.4 Chapter Conclusion

The initially outlined properties of digested information regarding the presence of relevant information in an agent's actions seem to hold for the discussed simulations. This now allows me to address the question if the information in the environment that is produced by another agent is somehow special? As demonstrated in the chapter, the answer is "yes". Especially the information related to another agent's actions not only contains relevant information (this follows already from chapter 3), but an agent is also motivated to maximise this relation. Assuming we are in a scenario where there exists a non-zero

trade-off between information processing and performance (a type 3 world, as discussed in section 3.5.10), then an agent that tries to increase its performance is at the same time increasing the relevant information encoded in it actions. Whether the agent can actually perform this adaptation during its lifetime (via learning) or this adaptation only happens through an evolutionary process in the population is secondary, as both lead to a similar conclusion: As a side-effect of a beneficial adaptation for the agent itself, the agent also creates a high-capacity channel from the relevant information to its own actions. Furthermore, since the actions of the agent are likely to have a much smaller state space than the environment, this will also lead to a higher per-bit density of relevant information, as demonstrated in the simulations. In essence, agents are motivated to be efficient preprocessors for just the type of information needed by other agents of their own kind.

Connecting this to the overall question driving this thesis, it should be clear that the digested information can offer a powerful incentive for sensor evolution and adaptation. As I demonstrated in chapter 3, the relevant information in parts of the sensor input can be quantified with agent-internal measurements, such as the unique relevant information measure. Let us consider the perspective of an agent that treats all its inputs as simple, indistinguishable data, and has so far only adapted to use the information provided by the environment to locate the food. Such an agent could then recognize (again, either by some active learning method, or through evolutionary adaptation) that the information related to other agents actually contains a lot of information relevant for its own actions, and adapt both its sensors, and it strategy accordingly. This should then lead to the minimal definition of social interaction made at the beginning of this thesis, where the actions of one agent become directly influenced by the actions of another agent. The following chapter will deal with what actually happens when an agent makes this adaptation, and demonstrates how this results in phenomena also present in real, biological agents.

In summary, the main message is that an agent that does better than random in a world were information matters *has* to encode this information in its actions. The agent can try to obfuscate this information, or only act when it is not observed, but the bottom line is this: If the agent wants to act according to its information, then this information is contained in its actions. Furthermore, this chapter should also demonstrate that information theory is able to quantify this effect, and can be used to demonstrate, as for example in the random agent's case, that there might still be information present in observed systems, even if we do not see it at first.

# Chapter 5

# Social Bayesian Update

## 5.1 Chapter Overview

The last chapter demonstrated that an agent's actions at a specific performance level have to contain a specific amount of information about aspects of the environment that are needed to achieve this performance level. As discussed in the chapter of relevant information, I assume that the amount of relevant information for higher performance levels is higher than zero for those worlds we are interested in.

I did then argue that under an evolutionary perspective this should give a selective advantage to an agent adapted as to incorporate this information into its own decision making process. In this chapter I will make the assumption that the agent can differentiate between the environment at large and other agents, and investigate how the digested information stemming from other agents can be incorporated from an agent-centric perspective, and what kind of problems are likely to arise from this.

Keeping with the information theoretic framework, I will use Bayes' Theorem to incorporate the additional information gained form other agents into the agent's internal Bayesian Model. I will also outline the shortcomings of the naive Bayesian Approach used by the agents, as it is at the root of some of the problems social agents encounter.

I will then demonstrate how Bayes' Theorem can be adapted to the Fishworld and Treasure Hunter Scenarios, and take a look at the resulting behaviour. I will demonstrate that the inclusion of social information is beneficial if a single agent develops it, but can become quite harmful if this adaptation spreads through the agent population. I will then link the result here to phenomena observed in Social Learning literature, especially those on Bayesian Social Learning, and Social Learning in Random Networks.

## 5.2 Bayes' Theorem

To integrate the action information into the agent's internal model, I will use Bayes' Theorem, which is usually stated as:

$$P(X|Y) = \frac{P(Y|X)}{P(Y)} \cdot P(X) \tag{5.1}$$

The random variables $X$ and $Y$ can be interpreted as propositions (facts about the world), or hypothesis that are either true or false. Bayes Theorem then addresses the question how the probability of $X$ changes if one was to observe the event $Y = y$, or consider the evidence that $Y$ is either true of false.

To illustrate, image there are two urns, one contains two black and one white marble, and the other contains two white and one black marble. They are indistinguishable otherwise. I now chose one urn at random and blindly draw a marble, which is white. What would be the probability that this was the urn containing two white marbles?

$X = white$ in this case would be the hypothesis that the urn drawn from was the white majority urn. So the *a priori* probability that I picked that urn would be $P(X = white) = 0.5$. $Y = white$ is the proposition that the marble drawn is white. $P(Y = white)$ is the marginal probability that I would draw a white marble, assuming only my *a priori* knowledge the system. That would be $P(Y) = 0.5$; considering that I could draw randomly from the black or white majority urn, it is equally likely to draw a black or a white marble. The conditional probability of $P(Y = white|X = white)$ then quantifies how likely it is to draw a white marble, if the urn drawing it from is actually the white majority urn. In our case, this would be $P(Y = white|X = white) = 2/3$. Putting those values into the formula I get:

$$P(X = white|Y = white) = (2/3)/(1/2) \cdot 1/2 = 2/3 \tag{5.2}$$

So after drawing one white marble the probability that I am standing in front of the white majority urn is 2/3. If we turn the question around and assume $X = black$ to mean I am in front of the black majority urn, then the formula remains largely unchanged. The a priori probability of $P(X)$ is the same, and also $P(Y)$ remains unchanged. $P(Y|X)$ equals 1/3 , and as a result, $P(X|Y)$ is also 1/3. This is the expected result, since both results should add up to 1, because in this case one of them should be exclusively true. Also note $P(Y)$ remains unchanged, independent of what hypothesis $X$ we are choosing. It can be considered as a normalization factor, making sure that the probabilities add up correctly to one.

### 5.2.1 Naive Bayes'

For the models in this dissertation it is necessary to integrate more than one observed event, since the agents observe several other agents and parts of the environment repeatedly. The general and optimal solution would be to treat all observed events as one large compound random variable and perform a single Bayesian Update with them.

Coming back to the urn example, this is like repeatedly drawing marbles from one urn (putting them back after each draw). For each of the $n$ draws, the marble's colour is formalized in the random variables $Y_1, Y_2, ..., Y_n$. They all can be expressed as one compound variable $Y_a = (Y_1, Y_2, ..., Y_n)$. Based on our knowledge about the system, it is possible to determine the probabilities for each state of $Y_a$, and thereby it is possible to calculate the marginal distribution of $Y_a$, and the correct *a posteriori* probability of $X$, which encodes if the urn is a black- or a white-majority urn.

This approach becomes problematic if the probabilities of $Y$ cannot be determined via model assumptions, but have to be obtained from statistical sampling. In such a case one would have to obtain enough samples to determine the probabilities of every state of $Y$ to a sufficient degree of accuracy. For example, imagine we were to look a medical data and wanted to know the probability of a having a specific medical condition, encoded in $X$, based on a list of 100 binary symptoms $Y_1, Y_2, ..., Y_{100}$. To obtain good statistics it would then be necessary to find a large enough group of patients for each of the $2^{100}$ symptom combinations to then determine the probability of a patient in that group having illness $X$. This is obviously not feasible.

A solution to this problem is called the *Naive Bayesian* Approach. To apply it, one makes the assumption that all observations $Y$ are independent conditioned on $X$, or that the systems in question approximates this property close enough so the error resulting from this assumption is negligible. This can be formalized for a range for observations $Y_1, Y_2, ..., Y_n$ as

$$I(Y_i, Y_j|X) = 0 : 0 < i \leq n, j : 0 < j \leq n, i \neq j \tag{5.3}$$

In this case the chain rule can be applied to decompose the multi-variate update into several consecutive single Bayesian updates so that

$$P(X|Y) = \frac{P(Y_1|X)}{P(Y_1)} \cdot \frac{P(Y_2|X)}{P(Y_2)} \cdot \dots \cdot \frac{P(Y_n|X)}{P(Y_n)} \cdot P(X) \tag{5.4}$$

The naive approach has several advantages:

- It is only necessary to gather enough statistical data to determine the influence of each single observation on $X$ separately. This greatly reduced the amount of needed

data.

- Updates can be applied in arbitrary order, so it is not necessary to sort or prioritize the observations.
- Later additional observations can easily be integrated by a multiplication of the current probability assumption at a later point.

These points make the approach well suited for an agent who can potentially observe other agents, and then wants to integrate the information gained at a specific point in time with its current prior (which might already contain information gained from the environment). The alternative, a complete and "correct" Bayesian Model on the other hand is infeasible for several reason. The necessary statistics to model any possible sequence of interdependent observations would be extremely large, which makes both obtaining and storing them difficult. Furthermore, it would also require the agent to store all its observations into a separate memory, so after each new observation it could then look up the appropriate update for the overall sequence of observations and then apply this to an initial prior. With the naive model the only need for persistent memory is the storage of the current probability assumption for $X$.

The same advantages also lead to the widespread application of the Naive Bayesian Theorem in areas such as machine learning, network monitoring, and others. Related literature reports good classification results for different examples of real world data (Hand and Yu 2001), even when the independence assumption was violated. Furthermore, (Domingos and Pazzani 1997) shows that it depends on the nature of the dependencies between the observations how far the naive models differs from the optimal actual Bayesian classification. Consistent dependencies (those that support a certain classification) are worse than inconsistent dependencies (those that cancel each other). If the dependencies fully cancel each other out, e.g. if they are symmetrically distributed, then the naive Bayesian classification even achieves optimality.

### 5.2.2 Adaptation to the Fishworld model

To further investigate the use of digested information for a agent in a multi-agent world I now want to modify my original fishworld model so agents are able to include information from other agents via Bayesian Update. As outlined in the last section, I will use the simplified Naive Bayesian Model for reasons of feasibility (feasible both for implementation in a computational model, but also more feasible in terms of ease of adaptation for the agent)

In our specific model we want the agent to use Bayes Theorem to update its hypothesis about the food source location when it obtains evidence in the form of other agents movement. The *a priori* hypothesis is the internal probability distribution $\hat{F}$, which assigns each cell in the world a probability for it to contain the food. Since $\hat{F}$ is not a single proposition, but a random variable with $n \times m$ states, we treat this as $n \times m$ separate *a priori* hypotheses. Bayes Theorem can be applied to each of them separately, just as we demonstrated with the white and black hypothesis.

$P(\hat{F} = f)$ is the probability that the hypothesis that the food is in location $f$ is true, where $f$ is an element of $W$, the set of all world cells. We immediately see that all $P(\hat{F} = f)$ are mutually exclusive, and that one of them must be true. As a result, we know that

$$\sum_{f \in W} P(\hat{F} = f) = 1. \tag{5.5}$$

A similar argument can be laid at the posterior probabilities of $P(\hat{F} = f|A = a)$ that quantify how likely a certain food source position is if an agent was observed to perform the move action $a$. They also have to add up to one:

$$\sum_{f \in W} P(\hat{F} = f|A = a) = 1. \tag{5.6}$$

The marginal probability $P(A = a)$ can also be determined quite easily. Either by argument, where it follows from the rotational symmetry that any move action $a$ is equally likely, and therefore $P(A = a) = \frac{1}{4}$. Alternatively, this can also be determined with the statistical measurement of the infotaxis agents, which supports our assumption of $P(A = a) = 1/4$, disregarding noise.

The last value we need to determine for each location $f$, is the conditional probability of $P(A = a|F = f)$. The probability of a certain action $a$, if the food source is in $f$. This value can be calculated, for every $f$ and $a$, from the statistics of the infotaxis agent. For example, if the action $a$ is north, and the position is 3 cells directly north of the agent, we can then look at the statistics and count how many times in 10 000 trials the agent has been 3 squares south of the food and moved north. This value is then divided by the overall amount of times the agent has been 3 squares south of the food. So, in context, the question $P(A = north|F = 3north)$ answers how likely is the agent to move north, if the food is three squares north of it. Note that the position $f$ in this case is calculated in relation to the position of the observed agent, and is relative only to the observed agent.

Putting all those values together we can calculate for every $f$:

$$P(\hat{F} = f|A = a) = \frac{P(A = a|F = f)}{P(A = a)} \cdot P(\hat{F} = f) \tag{5.7}$$

- $P(\hat{F} = f)$, the *a priori* probability, is the internal model of the agent for mapping the probability distribution of $F$, as gained by their own experience so far;
- $P(A = a)$ is the probability of an agent taking the move action $a$. Rotational symmetry suggests a probability of $1/4$ for each action $a \in \{north, west, south, east\}$. Measurements in our single agent simulation confirm this. This is a normalisation factor, so the overall sum of probabilities is still 1.
- $P(A = a|F = f)$ is the probability of another agent performing action $a$ if the food is in position $f$. Note that the position $f$ in this case will always be calculated in relation to the position of the observed agent.

Note that the agents I used to gather the statistics were non-social and thus blind to the actions of other agents. They behaved as described in the Infotaxis part of this paper. So even though agents in the simulation have the ability to sense other agents and update their internal world models they still calculate their Bayesian update under the assumption that all others are non-social agents. For reasons of brevity I will use the term *social* to denote agents that use the Bayesian Update with information gained from other agents.

Also keep in mind that we used those agents to create the statistics to calculate the probabilities for $P(A = a|F = f)$, so the $F$ in this formula refers to is the actual position in the word, rather than the assumed probability distribution internal to the agent.

## 5.3 Social Fishworld Simulation

In the next experiment I will now look at simulations that contain multiple agents, where some of the agents have the ability to perform the previously described Naive Social Bayesian Update. To focus on the effects produced by the Bayesian Updates I limited other channels for agent interaction. There is no collision detection, so agents can freely move into a similar space. There is also no competition for scarce resources, so the food source will not be used up by other agents finding it.

The general goal of the agents is similar to the single agent scenario, the agent wants to detect the food source location in the shortest time possible. The agents still employ the infotaxis strategy. The *social* agents add an additional step to their decision making

process. First they will check, as usual after a move, if any cell around them contain the food source. If not, they will then check within their sensor range if they can observe any agents. If this is the case they will then, for each agent separately, perform a Naive Bayesian Update, based on the last move of the other agent. The order of application here is irrelevant, because the Naive Bayesian Update creates the same result, regardless of order. After the agent updates $\hat{F}$, it resumes the previously described infotaxis behaviour to generate its next move action.

Note that agents which have successfully located the food stopped moving and are no longer perceivable by other agents. This was done to increase the challenge, since it would have been trivial for another agent to infer from seeing another non-moving agent that the food must be within the sensor range of that agent. As a result, the agents cannot see any agents which know where the food actually is.

This model, which includes the possibilities for the agents to use the Bayesian update not only on the environmental variables, but also on other agents they accidentally encounter will be called the *Social Bayesian* model. Apart from the update of the internal model before another infotaxis action is chosen it is identical to the infotaxis model.

### 5.3.1 Results of Social Bayesian Fishworld

First I equipped a single agent with the ability to perform the social Bayesian Update, where all other agents in the simulation would use the non-social Infotaxis approach. I then varied the number of agents in the simulation, and ran 10,000 simulations each in a $25 \times 25$ grid world to measured the average time it took the one social agent to locate the food source. At the beginning of each of the simulations all agents were put in a random location, and their internal world model was initialize as uniform distribution.

The resulting performance of the social agent can be seen in Fig. 5.1. The search time of the one social agent is influenced by the number of other, non-social agent in the simulation. The social agent search time decreases with an increasing agent population until there are about 50 agents in the simulation, and then increases for larger numbers of agents. The performance of the non-social agents is not depicted, as it remains constant throughout the different simulations.

The most obvious conclusion here supports the claim that Digested Information provides information otherwise not available to the agent. The dotted line in Fig. 5.1 shows the lower bound for search time achievable to any non-social agent. As we can see, the agent using information gained from the actions of others can perform better, e.g. has a lower search time, than this lower bound. The only information the agent obtains, in addition to the information about the local cells picked up from its environmental sensors

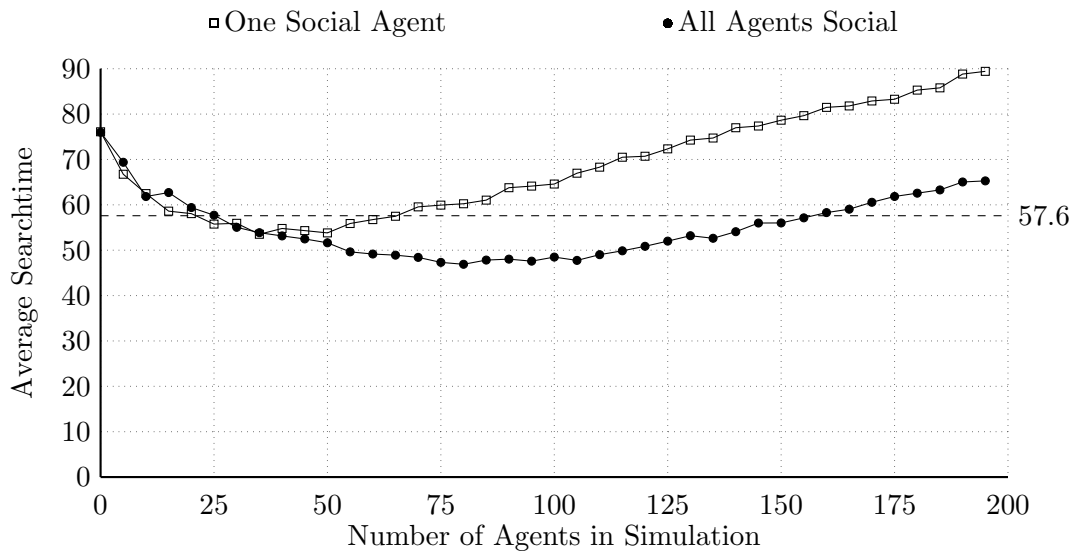



Figure 5.1: A graph showing the dependencies between the number of agents in the simulation and the average search time of one agent. Each data point is the average of 10,000 simulation for each number of agents. The values are calculated for one specific agent, who is either the only social agent in a world with non-social agents (white box), or is a social agents in a world where all others are also social agents (black circle).

is the information contained in other agent's action. Since the agent could not perform this well with only the locally available environmental information, this requires the other agent's actions to provide information that is not locally available right now. This shows that the other agent has to transport relevant information either through space or time to then display it in its actions here and now.

Since the agent seems to profit from this new ability (at least if there are less than 100 agents in the simulation) I also modified the simulation further, so that every agent is now able to perform the social Bayesian Update. All other aspects of the simulation remain unchanged. The resulting average performance for different numbers of agents can also be seen in Fig. 5.1. Again, the performance first increases with a growing number of agents, up to about 80 agents. If the amount of agents in the simulation increases further, then the performance drops again, indicated by an increase in the average search time.

### 5.3.2 Interpretation

The obvious question here is "Why does an increase in other agents beyond a certain point worsen the performance of the Social Bayesian Update agent?". This is particularly

puzzling, since we established that the other agent's actions contain useful information, and that an agent can actually profit from observing said information (as seen for the simulations with less than 100 agents in the environment).

Some common explanations, which might be very reasonable for this phenomenon in natural system, can be excluded due to the design choices of the simulation. This was done intentionally to focus on the effect the information has. Since there is no collision in the simulation, meaning several agents can be in the same cell, there is no overcrowding effect, or agents blocking others with their presence. Similarly, there is no competition for scarce resources, so the food source does not get depleted by other agents. This also means more agents do not limit the access to the food source for other agents. Basically, the other agents can only see and be seen, but not be interacted with further. Therefore, the explanation to the worsening performance with too many agent's should related to either the information transfer from other agent's actions to the social agent, or to the information transfer from the social agent's actions to the sensors of other agents.

Another possible problem for the agent could be the lack of good statistical data for the other agent's behaviour. As all the data is gathered for non-social agents, using the resulting conditional probabilities for the social Bayesian update might introduce a certain amount of error, if the observed agent's behaviour differs from the non-social agent. Obviously, other social agents act somewhat different to non-social agents (this can be easily seen from the difference in performance), so this could explain why updating with their information might have a negative effect. But this also seems unlikely to be the main cause for the increase in search time. We see in Fig. 5.1 that the worsening effect of too many other agents in the environment is larger when the other agents are non-social, in which case the used statistics would be correct.

Nevertheless, the problem has to be connected somehow to the social Bayesian Update process, as it is the only way how one agent can affect another. Therefore, a closer look at the information in the observed actions is warranted.

## 5.4 Single Symbol Information

After examining the overall effect of the Social Bayesian Update on agent performance, I will now analyse the effects of a single sensor input. I will introduce two measures, *internal certainty* and *external correctness*. The first is the resulting actual reduction in entropy of the internal model, the very value which infotaxis aims to maximise. The second, external correctness, measures how well aligned the agent's internal model becomes with the actual state of the world.

Both measures are very similar to the ones presented in (DeWeese and Meister 1999); the main difference being that the measures are taken in regard to a specific agent's perspective. So, I do not ask how a single symbol would affect certainty on average, which would basically be mutual information, but how it does affect a specific agent at a specific time. The difference manifest itself mainly in the selection of the priors. This is particulary interesting for correctness, where a symbol can be misleading in general (worsen the average correctness), but still "correct" for a particular agent, or vice versa.

Note also, that this analysis is done regarding an agent's perspective, not *from* an agent's perspective. Meaning that the measurement reflects how much an agent would gain from a single symbol, but this measurement cannot be taken by the agent itself. If it was possible for the agent to evaluate the correctness of a Bayesian Update, then it would be pointless to do so, as the agent would already know where the food source is. Therefore, the correctness can only be evaluated from an omniscient observer perspective, or after the agent learns where the food source is.

I will use this analytical tool to investigate two questions. Once again I take a look at how the information gathered from agents differs from the information gathered from the environment. I am also interested to see what happens to the information gained from other agents if there are too many agents in the simulation.

### Internal Uncertainty

Information, in regard to information theory, can be classified as the average reduction of uncertainty (or entropy) caused by observing a specific variable. This aligns well with the use of information as mutual information in this thesis, since mutual information $I(X;Y)$ between $X$ and $Y$ measures how much the entropy of $X$ would be reduced if $Y$ was known, and vice versa.

In our specific scenario the two variable in question are $\hat{F}$ (the internal probability distribution of the food source location) and $F$ (the actual food source position). Infotaxis aim is in fact to reduce the entropy $H(\hat{F})$ of the internal distribution, which I will call *internal uncertainty*, as quickly as possible.

Thus, one way to evaluate the quality of a given sensor input would be to measure the actual reduction in entropy to the internal variable $\hat{F}$. For this measurement, let $s$ be a specific sensor input $S = s$, were $S$ is the random variable that encodes all sensor input states. Then $\hat{F}_b$ is defined as the internal variable's state $F$ before $s$ was observed, and $\hat{F}_s$ its state after $s$ was observed. Then the actual reduction of uncertainty for $s$ can be defined as

$$\Delta H_s = H(\hat{F}_b) - H(\hat{F}_s). \tag{5.8}$$

Note that this measurement only relies on the agent's internal variables, so this value could be measured by the agent internally. It measures the increase or decrease of certainty the agent has about the world state. A piece of information that increase this certainty the most would have the highest $\Delta H_s$. Also, since the measurement is defined as the difference of entropy between two points in time, it is obvious that all reductions in entropy over a single simulation have to sum up to the overall reduction from maximal entropy to zero entropy once the food is found. In other words, to find the food, the agent basically has to reduce its internal entropy to zero, meaning the agent has to process enough sensor input to provide it with enough entropy reduction.

**External Correctness**

The second way to evaluate a single piece of information would be to check if the agents internal assumption are more or less correct after processing that piece of information. In terms of probability distribution this can be done be evaluating how well the agent's internal probability model $\hat{F}$ approximates the actual food source position encoded in $F$. This measure will be called *external correctness*.

The formalism used here to evaluate this is called the Kullback-Leibler divergence. It can be computed over two probability distribution that are defined on the same alphabet. In our specific case both $F$ and $\hat{F}$ are both defined over $\mathcal{W}$, the set off all world cells. The Kullback-Leibler divergence is then defined as

$$D_{KL}(F||\hat{F}) = \sum_{x \in \mathcal{W}} P(F = x) \log \frac{P(F = x)}{P(\hat{F} = x)}. \tag{5.9}$$

The KL divergence is, per definition, non-negative, and will attain its minimal value 0 when the internal distribution $\hat{F}$ is identical to the actual distribution $F$. If $f \in \mathcal{W}$ is the actual location of the food source, then

$$P(F = x) = \delta_{xf}. \tag{5.10}$$

The KL divergence will be zero when $P(\hat{F} = f) = 1$, meaning that the agent has located the actual food source, and is correct about it. By convention the KL divergence is infinite if we have to divide a non-zero probability by zero. In this specific example, this is the case when we have a state in $F$ with a non-zero probability, where the corresponding state of $\hat{F}$ has a probability of zero. The only non-zero probability in $F$ is the one for $P(F = f) = 1$, where $f$ is the actual location of the food source. So, for the KL divergence to be infinite, the probability for $f$ in the internal distribution has to be zero, $P(\hat{F} = f) = 0$. This

also means that no Bayesian update could create a state where $P(\hat{F} = f) = 1$, making it impossible for the agent to ever arrive at a fully correct model. The agents model about the world in this case is not just very wrong, but basically broken. An infinite KL divergence captures this well. For all other cases the KL divergence is finite.

Following from the definition of the KL divergence in Eq. 5.9 and the property of $F$ to vanish for all cases were $F$ is not the food source $f$ (see Eq. (5.10)), the KL divergence can be calculated as

$$D_{KL}(F||\hat{F}) = P(F = f) \log \frac{P(F = f)}{P(\hat{F} = f)} = \log \frac{1}{P(\hat{F} = f)} \tag{5.11}$$

If we accept this as a measure of how correct our agent's internal modelling of the world state is, we can then also check how its correctness was affected by a single symbol $s$. Again, $\hat{F}_b$ is defined as the internal variable's state $\hat{F}$ before $s$ was observed, and $\hat{F}_s$ its state after $s$ was observed. The change in correctness is then measured as

$$\Delta KL_s = D_{KL}(F||\hat{F}_b) - D_{KL}(F||\hat{F}_s). \tag{5.12}$$

Note though, that this value can only be evaluated if one either has an outside view on the overall system, or if an agent would actually store all its internal probability distributions over time, and evaluate their correctness after finding the actual food source location. This measurement cannot be used by the agent at the time when the agent actually does the update. Also, just as for the other measurement, this value has to eventually add up to the overall reduction in KL divergence of the initial uniform model to zero KL divergence, when the food source is finally located.

### 5.4.1 Single Agent Experiment

First I will take a look at the single, non-social agent case. For this I reran the original grid world search task outlined in chapter 4. The simulation contains a single agent, in a $25 \times 25$ world, its behaviour generated by the infotaxis algorithm. What I want to determine now is how each sensor input changes the internal probability distribution of the food source location.

The sensor input event $s$ contains the states of all the world cells in sensor range of the agent for one time step. This corresponds to all the information the agent can process before it has to make the next decision. For a sensor range of 2 this means there were 25 cells to evaluate, each with two possible states. The processing of those potential 25 bits of information was considered as one event. For each of those sensor inputs $s$ the two

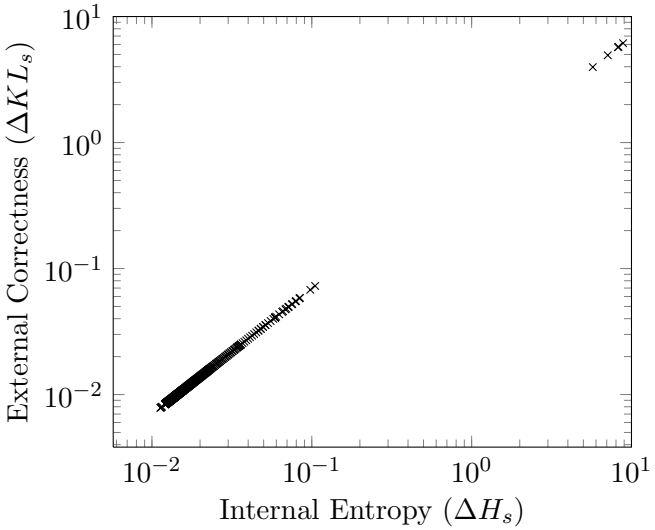



Figure 5.2: A scatter plot showing the change in both internal entropy $\Delta H_s$ and external correctness $\Delta KL_s$ for different sensor input events. Each data point depicts the values for the observation $s$ of all world cells in sensor range in one time step. The graph shows the accumulated data for 5 full search tasks, recorded in a $25 \times 25$ world, with sensor range 2. The outlying points are those observations in which the agent actually finds the food source location.

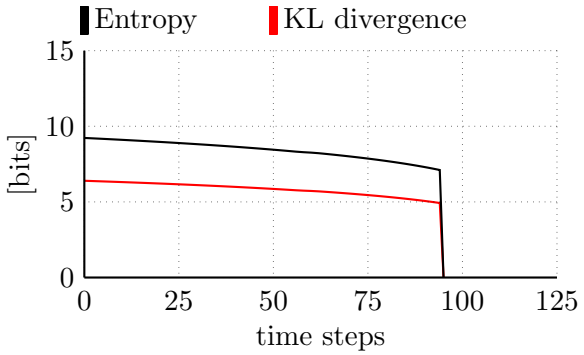



Figure 5.3: A graph showing the development over time of both internal uncertainty (Entropy) and external correctness (KL divergence) for a single agent. Measurements were taken for a single simulation in a $25 \times 25$ world, with a sensor range of 2. The steep drop at the end corresponds to the time when the agent actually discovered the food source location.

values, for $\Delta KL_s$ and $\Delta H_s$, were recorded. The measurements in Fig. 5.2 were taken for

5 agent search tasks[1], each data point is the combination of a single $\Delta H_s$ and $\Delta KL_s$ for a specific $s$.

Looking at the data in Fig. 5.2 we see that the values for uncertainty reduction $\Delta H_s$ and increase in correctness $\Delta KL_s$ correlate perfectly for the single agent case. The five large outliers are the values associated with the last steps in each simulation, the one where the agent actually locates the food source, and thereby reduces its remaining entropy. Apart from the final observations the recorded values are always positive, or at least zero, but relatively small. They were plotted on a logarithmic scale to show that most of them actually have non-zero values.

Fig. 5.3 is a plot of the development of both the entropy $H(\hat{F})$, and the KL divergence after each time step for one specific simulation. The development seen here is typical for a single agent simulation. Each time step reduces the uncertainty, and increases the correctness of the agent. The steep drop at the end happens when the agent actually finds the food. The only major difference difference between simulations is the time step at which this actually happens. Also, for some rare cases there are short plateaus where the uncertainty and correctness remain constant. Those correspond to the phenomenon mentioned earlier, where the agent explored all locations in its immediate reach, and therefore will move around randomly, without any immediate information gain.

Based on the observed data in both figures, it is clear for the specific single agent fishworld case that every reduction in uncertainty is accompanied by a reduction of the KL divergence with a proportional amount, and vice versa. This single agent simulation therefore seems to be extremely well suited for the agent's current mode of information processing and decision making because infotaxis tries to maximise the reduction in entropy and thereby maximises its gain in correctness.

While both $\Delta$ values could, in theory, be negative, this is not the case here. Furthermore, the correlation between both values, meaning those inputs that decrease the uncertainty also increase the correctness, suggests that it might be possible for the agent to actually determine the correctness of a single input by relying on its internal measurement of entropy reduction. But, as we will see in the next section, the general case, or a case involving other agents might not be so accommodating to the agent.

### 5.4.2 Results for Social Bayesian Update

This sections applies the single-symbol information analysis on a multi-agent simulation where all or some agents use Social Bayesian Update. It contains a series of experiments,

[1]The low number of trial runs here is not statistically meaningful, but was chosen as to not clutter the graphical representation. The interpretation, therefore, should only be used qualitatively.

all of them are situated in a $25 \times 25$ gridworld, with sensor range 2. There are 4 different parameter combinations:

- 40 Agents, one agent using Social Update
- 40 Agents, all using Social Update
- 200 Agents, one agent using Social Update
- 200 Agents, all using Social Update

In the *single social* case only one agent will be able to use the Social Bayesian Update. The data recorded in this case will be for this one social agent. In the *all social* case every agent in the simulation has ability to perform the Social Bayesian Update, and the data will be recorded for one, arbitrarily chosen agent.

The simulations with 40 agents were chosen because both the single social, and all social case behave similar in terms of performance for this set of parameters, as seen in Fig. 5.1. The simulations with 200 agents were chosen to take a closer look at why the agents suddenly perform worse with more agents being present in the environment.

This time two different kinds of data points were collected. As in the last simulation, one type of data point contains the differences for KL divergence and entropy before and after processing all the *environmental* sensor information (the state of the world cells). The other type is the difference before and after processing all information gained from other agents in sensor range, which will be called the *Social Update*. So, the second type of data collates all the information gained from other agents in one time step. Furthermore, I also record the development of both KL divergence and internal entropy over time.

### Development over Time

First, lets take a look at the development of the measurements over time. In Fig. 5.4 we can see the development of both KL divergence and internal entropy for 10 different search tasks for a social agent in a world with 40 other social agents. Again, this is only a randomly chosen sample of 10 trials, but we can still gain some qualitative insights here. First, note that only 8 of the 10 search task actually show development over time, as 2 ended immediately, with the food source being sensed in the initial round.

For the other 8 search tasks, there seems to be a general trend for both the KL divergence and entropy to decrease over time, with a sharp drop at the end, when the food source is actually located. Different from the simulations where the agent used only the environment to update its internal state, we can now see that both external correctness

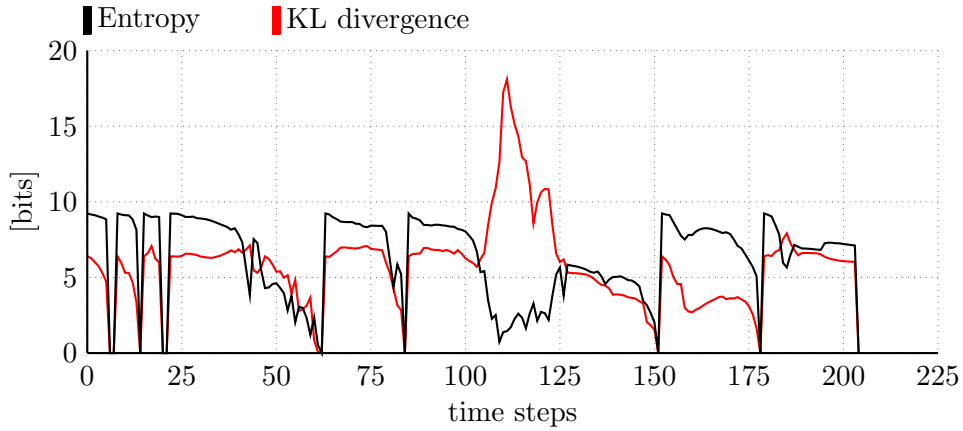



Figure 5.4: The graph shows the development of both internal uncertainty (Entropy) and external correctness (KL divergence) over the course of 10, randomly chosen, consecutive search tasks. The measurments were taken for one specific agent using social update, in a $25 \times 25$ world, with a sensor range of 2. The world contained 40 agents, all also using social update. Two of the search tasks ended immediately, as the agent could sense the food source location in the first turn.

(KL divergence) and internal certainty (entropy) can also become worse as time progresses. For example, the 6th simulation shows a very high peak, where the KL divergence becomes nearly trice as large as its starting value, indicating that the agent's internal model assigned a very low probability to the location where the food source actually was. Note that this also coincides with the 6th search task beeing the one that took longest (ca. 70 time steps). Likewise, the internal certainty also can become worse over time, as can be seen in several search tasks in Fig. 5.4.

The second thing we can already see from the rough analysis of Fig. 5.4 is that the clear correlation between entropy and KL divergence, which existed for the case with only the environmental update, is not present here. In several cases one of the two values increases, while the other values decreases at the same time, or vice versa.

The other parameter sets (different amount of agents, either all social or only one agents social) produce qualitatively similar graphs where it also becomes clear that both KL divergence and entropy can increase, and that they are no longer perfectly correlated. The graphs itself are not shown here, as the differences between graphs of the same parameter set are larger than the differences between parameter sets, so it seems that no further insight can be gained from comparing the temporal development of KL divergence and entropy for different parameters on a qualitative level.

### Decomposition in Environmental and Social Update

As a next step I looked at the data points for the environmental and the social update separately. In the scatter plots in Fig. 5.5 and Fig. 5.6 each data point indicates the difference in KL divergence and entropy for a single update, either the sensing off all cells in sensor range (environmental update), or the processing of all visible agent's actions (social update). So one data point each is collected for every time step in the simulation. The graphs contain the accumulated data points for 50 simulated search tasks, in the usual $25 \times 25$ gridworld, with a sensor range of 2.

Looking at Fig. 5.5 and Fig. 5.6 we can see that the collected data points now can be in any of the four quadrants, indicating that the information gain from a single sensor input can now affect the agent in the following ways:

**upper right** more correct, more certain

**upper left** more correct, less certain

**lower left** less correct, less certain

**lower right** less correct, more certain

In general, the upper quadrants should be preferable for the agent, as they indicate a gain in actual correctness, rather than just an increase in certainty about the world that might or might not be wrong, but it should be noted that in order to complete the search task the agent has to both reduce its uncertainty to zero, and be completely correct about it.

Regarding the data there are some general observations that can be made. Each of the 4 different experiments has sensor observations in each of the 4 quadrants. There is however a tendency for the social update information to be on the right side, where the agent becomes more certain. In the next section, we will look at the average values for the single symbol information, where table 5.1 will support this visual impression with quantitative data, taken for larger samples sizes.

The information for single environmental updates on the other hand is never "misleading", in a sense that is does never lower the external correctness for any of the four experiments. The outlying points, those that have both a high gain in certainty and correctness are those associated with actually finding the food, i.e. the complete reduction of the remaining uncertainty and KL divergence. Apart from those events, most of the environmental updates seem to decrease the agents certainty, while increasing the correctness.

Specifically, looking at the plots for the simulation where all agents have the social ability in Fig. 5.5, there is a comparatively large deviation from the center for both the

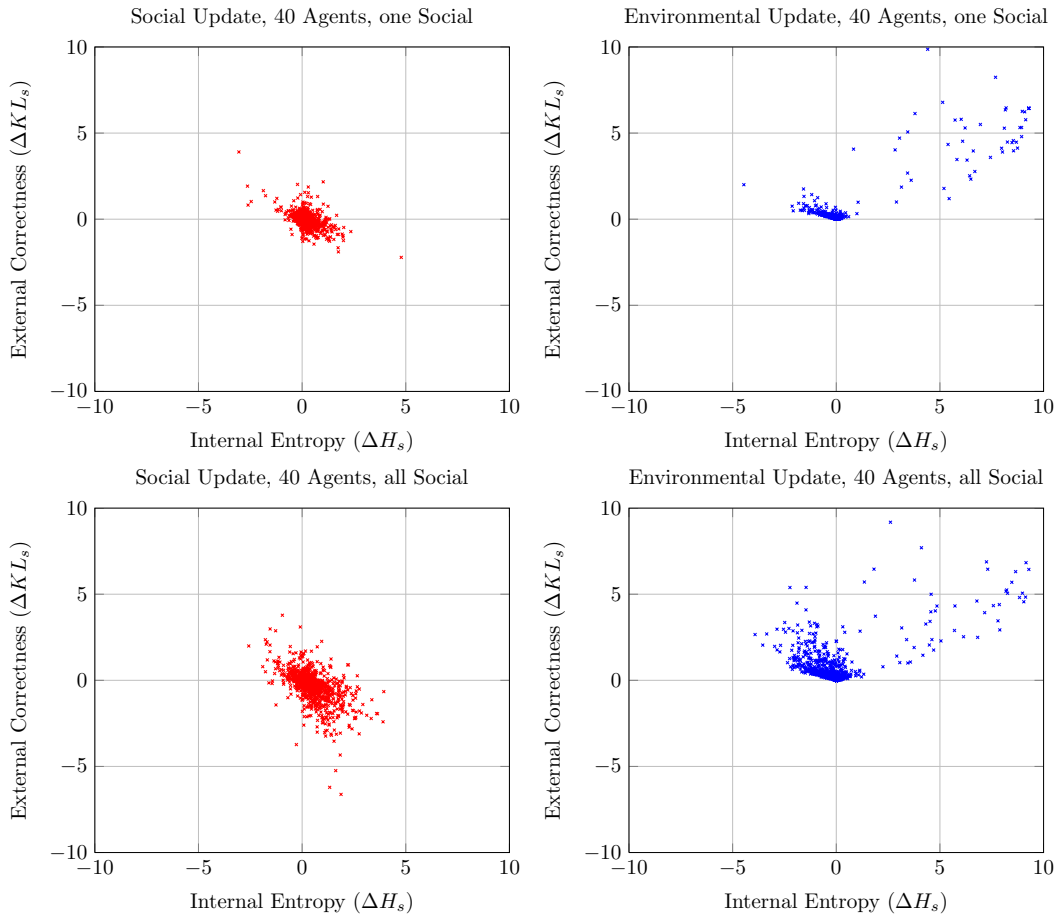



Figure 5.5: Four plots showing the gain in external correctness and reduction in internal entropy for single sensor events. The two left plots show data for sensor inputs coming from other agent, while the right graphs show data for environmental sensor input. The two upper graphs are data from 10 search tasks of one social agent located, in a world with 40 non-social agents. The two lower graphs are data from a social agent located in a world where the other 40 agents are also social.

social update and the environmental update. The social update events here make the agent more certain, but less correct, while the environmental updates make the agent more correct, but remove certainty.

Little difference can be seen for the simulations with 200 agents. Both, the single social, and and the all social case, have quite similar plots. The social update information is clustered around the center, and the environmental update seem to mainly reduce

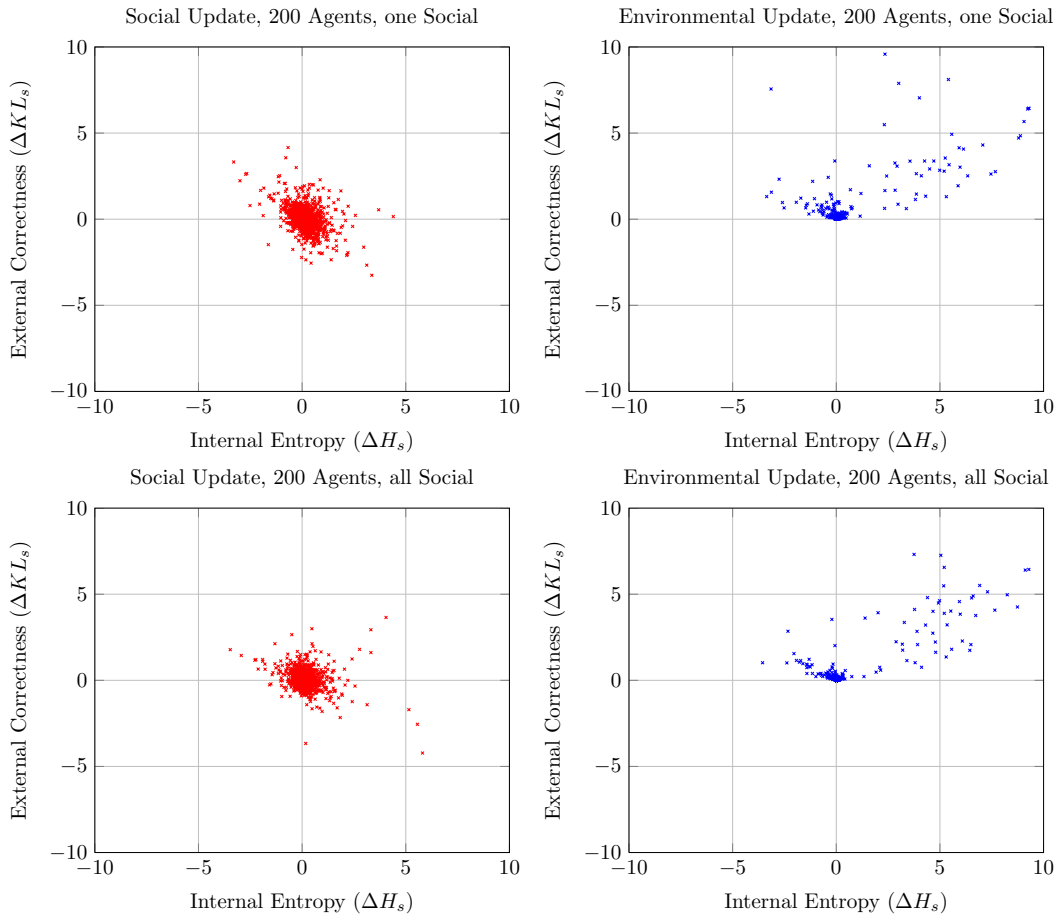



Figure 5.6: Four plots showing the gain in external correctness and reduction in internal entropy for single sensor events. The two left plots show data for sensor inputs coming from other agent, while the right graphs show data for environmental sensor input. The two upper graphs are data from 10 search tasks of one social agent located, in a world with 200 non-social agents. The two lower graphs are data from a social agent located in a world where the other 200 agents are also social.

certainty, while increasing correctness, apart from those events where the food source is actually located.

### Quantitative Comparison of Social and Environmental Update

To evaluate those differences quantitatively the average value over 10,000 search task simulations for both $\Delta H_s$ and $\Delta KL_s$ was computed. The resulting measurements can be

| | KL Div. Social | KL Div. Envi. | H Social | H Envi. |
|---|---|---|---|---|
| 40 Agents, one social | -0.0113 | 0.1348 | 0.0889 | 0.0893 |
| 40 Agents, all Social | -0.1542 | 0.2676 | 0.2031 | -0.0396 |
| 200 Agents, one Social | 0.0139 | 0.0599 | 0.0567 | 0.0498 |
| 200 Agents, all Social | 0.0313 | 0.0653 | 0.0669 | 0.0725 |

Table 5.1: This table gives an overview of the average reduction in internal entropy and the gain in external correctness for different scenarios. The values are averaged over 10.000 simulations, and are separated by social update information and environmental information.

found in table 5.1.

Note, that the sum of the social and environmental reduction in KL divergence and the sum of the social and environmental reduction in entropy are directly related to the average search time. This follows from the previous definition of $\Delta H_s$ and $\Delta KL_s$ as differences between a state before and after observing a symbol $s$. The full entropy and KL divergence have to be reduced to zero to complete the search task, so the average reduction in entropy and in KL divergence is the overall reduction divided by the time steps it takes to find the food. The measurements taken over 10,000 trials reflect this. If one was to add the average $\Delta H_s$ for the environmental update, and the average $\Delta H_s$ for the social update, then the sum would be average $\Delta H_s$ for one turn. This value corresponds to the overall reduction divided by the average search time. The same is true for adding the values for $\Delta KL_s$. The values for the different scenarios all fulfil this property.

Since both simulation with 40 agents, have roughly the same average search time, the average gain per time step (the sum of social and environmental $\Delta H_s$ and $\Delta KL_s$) are also roughly similar. But the decomposition for those two simulations looks very different. In the case where only one agent uses the social update 3 of the 4 values are positive, while the remaining one, the $\Delta KL_s$, for the social update is negative, but close to zero. So the Social Update Information is slightly wrong on average.

The simulation where all agents are social has very different distribution of values. Here the $\Delta KL_s$ is clearly negative, but this is compensated by a much larger $\Delta KL_s$ for the environmental update. But the environmental update has a negative value in $\Delta H_s$, the average reduction in entropy. This again in compensated by a large reduction of entropy for the social update.

The measurements for the simulations with 200 are all positive, and show no particular differences in regard to decomposition between the one social and the all social case. All average gains for the one social case, the one that performs worse, are lower, as would be expected given that the agent has a longer average search time. Taking a closer look

at the underlying data shows that these lower averages, especially in the environmental update values, results from a large amount of updates with zero information gain.

### 5.4.3 Interpretation

After looking at the resulting data gathered from the different simulation utilizing the social update there are several questions that still remain to be answered.

- Why does the environmental update never have a negative $\Delta KL_s$, i.e. why is no single environmental update misleading?
- What causes the average gain in certainty or correctness to become negative?
- Why does an increase in the agent population beyond a certain number worsen the performance of social agents?
- Why is this worsening effect stronger for the case where only one agent is social?

#### Deterministic Observations

One major difference between the environmental and the social updates is that the information in the environmental updates is never misleading to the agent, i.e. never increases the KL divergence. This can be explained by the fact that the environmental sensor inputs, the states of the world cells, are fully determined by the location of the food source. Assume that $C$ is a random variable encoding the state of a world cell, with $C = 0$ meaning it is empty, and $C = 1$ meaning it contains the food source. $F$ encodes the food source location; the information the agent wants to learn about. We see that for a given state of $F$ we can determine the state for every $C$. If we know where the food is, we can tell if a specific cell contains it or not. So, $H(C|F) = 0$, or in other words, $C$ is determined by $F$. If the agent now observes a cell $C$, we can calculate how this would affect the agent's Bayesian model of $P(\hat{F} = f)$, the probability for the actual location, $f$. If the cell is empty we know that $P(C = 0, F = f) = 1$, and therefore Bayes' theorem calculates as:

$$P(\hat{F} = f|C = 0) = \frac{P(C = 0, F = f)}{P(C = 0)} \cdot P(\hat{F} = f) = \frac{1}{P(C = 0)} \cdot P(\hat{F} = f). \tag{5.13}$$

Since $P(C = 0)$ has to be smaller of equal to one, it follows that

$$P(\hat{F} = f|C = 0) \geq P(\hat{F} = f). \tag{5.14}$$

The analogous argument can be made for any other state of $C$, which in this specific case would be containing the food source. It follows that for any observed state of $C$ the probability of the actual location $f$ to contain the food source is increased, which leads to a decrease in KL divergence. This simply follows from $C$ being fully determined by $F$. Contrary to this, the state of an agent's actions are not determined by $F$, so in general $H(A|F) \geq 0$. As a results, updating with another agent's action contain the possibility of actually increasing the KL divergence, i.e. becoming less correct.

This offers another nice distinction between social and environmental information for this simulation, but we should be careful to generalize this result. I would argue that is it pretty safe to assume that an agent's actions are not fully determined by the information it is looking for, simply because the agent is lacking this information in the first place. Furthermore, in a more complex scenario the agent might have several concerns it needs to address, resulting in a form of hybrid action selection that depends on different aspect of the environment.

Nevertheless, I doubt that it is generally safe for other models to assume that environmental observations are fully determined by the information an agent seeks. Any form of noise, be it in the sensor input, or in the environment itself, would already violate this constraint. Therefore, it would be more reasonable to assume, that it is quite possible for environmental information to mislead the agent as well.

### Negative Average Information Gain

In the simulation with 40 Agents, where each of them has the social update ability, the average reduction in entropy becomes negative for the environmental update. This is counter-intuitive at first. It is well understood that a single update can result negative entropy reduction, but the average reduction in entropy should be positive. Especially since the mutual information can be defined as the average reduction in entropy, and we established in chapter 4 that there is a non-zero amount of mutual information between the cells in the environment, and the position of the food source location. But taking a closer look at one formalization of mutual information, as in

$$I(X;Y) = H(X) - H(X|Y), \tag{5.15}$$

it becomes clear that this average reduction is quantified in regard to a prior of $H(X)$. In our specific example, this prior is $\hat{F}$, the internal model of $F$, which can assume a state of high certainty and low external correctness. A subsequent update from this state with environmental information, which is always correct in our case, will then result in lowered

certainty (an increase in entropy), and a increase in correctness. This wrong, but certain prior in the internal model is created by the social update. Specifically for the case with 40 social agents we see that the average $\Delta KL_s$ for the social update is negative, meaning that, on average, the social update is wrong. What is happening here is that the agents encounters another agent and then uses that agent's action to update it own internal state. This creates the assumption that the food source location is likely just outside its reach, in the direction the other agent was just going. This assumption is likely wrong, as the measurement indicates. The agent then explores the location of high probability, likely finds it empty and subsequently updates it internal model. The model is now much more correct, but less certain, since the location which was likely to contain the food does not actually contain it. This also explains how the inclusion of the social update changes the nature of the information gained from the environmental update as seen in the scatter plot in Fig. 5.2 to the one seen in Fig. 5.5. The information gain changes because of the systematic change in the priors, not due to any change in the information itself.

Furthermore, even though the social update is systematically wrong (as in average decrease in correctness), the overall performance of the agent is still improved by it. Just disregarding the social update would return the agent to its non-social performance, which was worse than the performance of the agent that incorporates the incorrect social information. So taking up this "misleading" information is beneficial. In this specific case it improves the correctness gained from the environmental information substantially, compared to the correctness gain from environmental information in the non-social agent. Furthermore, the social update still contributes a huge amount of reduction in uncertainty, which in this case also helps to rule out a lot of locations where the food is not.

#### Systematic Dependency for Naive Social Update

Understanding how specific information leads to an average gain in uncertainty, at least for selected parts of the sensor input, still leaves the question why the social update is on average incorrect. Similar to the argument regarding the average reduction in entropy the average reduction in KL divergence should not be negative. While single symbols can be misleading, they should not be misleading on average. If they are wrong on average than this means that the model (namely, the conditional probabilities) used to perform the update are wrong, i.e., not reflecting the actual probabilities as they are. This means, we should take a closer look at the models used for the update.

Our model of the agent's behaviour was created by observing non-social agents, so it is possible that it fails when this model is used to incorporate the behaviour of other *social* agents. One the other hand, we know that that the statistics used in the update are

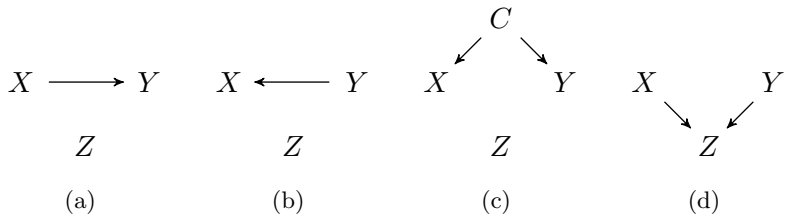



Figure 5.7: Illustrations of the four principle Causal Bayesian Networks that explain non-zero conditional mutual inforamtion, $I(X; y|Z) > 0$. (a) causal path from $X$ to $Y$, (b) causal path from $Y$ to $X$, (c) common cause $C$ or (d) being conditioned on a common descendant of $X$ and $Y$.

correct for the case where only one agent is using the social update, as the other agents are in fact non-social agents, just like the ones used for gathering the statistics. Their behaviour is identical to those in the multi-agent simulation, as the non-social agents in the multi agent simulation are not aware of other agents, or interfere with them in any way. The negative average gain in correctness is still present in this case, so we need to look for a different explanation. In this specific case, a likely candidate is the violation of the independence assumption for the Naive Bayesian Update, formalized as

$$I(A_i, A_j|F) = 0 : 0 < i \leq n, j : 0 < j \leq n, i \neq j, \tag{5.16}$$

where $A$ are the observed actions of other agents, and $F$ is the actual location of the food source.

In general, there are four different causal structures, illustrated in Fig. 5.7, which can create mutual information between two variables $X$ and $Y$, conditioned on another variable $Z$ (Pearl 2000). For two variables $X$ and $Y$ to have positive, non-zero mutual information there has to be either:

- a causal path from $X$ to $Y$, not containing $Z$
- a causal path from $Y$ to $X$, not containing $Z$
- a common cause $C$, leading to both $X$ and $Y$, neither path containing $Z$.
- a causal path leading from both $X$ and $Y$ to $Z$, the variable the mutual information is conditioned on.

In our specific case, $F$, the location of the food source is determined at random, so

it cannot be in any causal path from one action $A$ to another. Similarly, since $F$ has no parent nodes, it can only be the common cause variable itself, instead of a variable lying on the path from the common cause to the variables in question. For our specific example this means that to show that the independence assumption is violated, one has to demonstrate either a direct path from one variable to another, or show a common ancestor in the causal graph that is not $F$ itself.

There is one further possibility, the fourth case, where dependency is induced by conditioning on a variable which is the descendant of both variables the conditional mutual information is calculated for. To illustrate, it is possible to select a subset of the events of the two variables in question, so that they have mutual information. Since the mutual information for the remaining subset cannot be negative, the overall mutual information will then also be positive. A classic, fictional example is the dependency relation between smart and athletic people in elite universities. The events, or samples in this example are people. Each of them can be smart, with a probability of 10%, which is formalized with the variable $S$, and each of them can be athletic, with an independent probability of 10%, which is formalized in the variable $A$. For the overall population, there is not dependence between being smart, or being athletic, so $I(A;S) = 0$. But if we now select a subset of the population, this changes. Assume that an elite college would accept anyone who is either smart or athletic, or both. Acceptance will be formalized as $C = t$, the random variable for college acceptance assumes the state true. A simple calculation would then show that $I(A;S|C = t) > 0$, since it is more likely for someone who is smart and in college to not be athletic. This also means that $I(A;S|C) > 0$ is larger than zero, as the general conditional mutual information is just the weighted sum for all states of $C$.

This leaves in total three different ways on how the different agents' action variable can become correlated:

1. direct causal path from one action to another action,
2. common cause for different actions that is not the food source location,
3. conditioning of the overall system on a variable that is a descendant of different actions.

In the following part I will demonstrate how the fishworld simulation in particular could realize these violations of the independence assumption. I will also speculate how those fishworld specific violations generalize to generic social Bayesian Learning Scenarios.

### Common Cause

As the agent's actions are partially determined by its own internal state, it is conceivable that the agent's actions in several different time steps are a result of the same internal state. This would then introduce a dependence of the agent's actions in different time steps. And indeed, taking a close look at multi-step action sequences, one can observe that the agent nearly never goes back the way it just came. Since the agent knows that all cells in that direction are empty, it would only move there if its actions were determined at random, because all cells in reach were already explored. In general, the existence of memory makes this dependence possible, since the very point of having memory is to use a particular piece of information later, and possibly repeatedly to inform ones actions. On the other hand, this effect should not worsen depending on the number of agents present.

### Direct Cause, from Action to Action

Taking into account that the average gain of KL divergence is significantly worse in the case where all agents are social, compared to the case where only one agent is social, it stands to reason that this difference should somehow account for the lack of correctness. Allowing all agents to observe each other makes the system susceptible to information cascades. One agent might go into a certain direction, another might follow, base on the assumed private information of the first agent, etc. And indeed, if we take a closer look at the agent behaviour, we can see that the agents in the simulation seem to synchronize their behaviour by aligning the directions of their movement. This will be studied in greater detail in the next section, but here it should help to illuminate the differences between the "all social" and "single social" model. The model where all other agents are social allows those agents to each observe each other, and thereby to synchronize their resulting actions. This then leads to statistical dependency, and furthermore, to errors when using the Naive Bayesian Update.

Furthermore, the behaviour of the social agents is slightly different than the non-social behaviour, which was used as a basis to construct the statistics for the Social Bayesian Update. In addition, this could also introduce a source of error that would worsen the update.

### Dependency through Selection

Even with the information cascade and the faulty statistics as explanation we still need to explain why the KL divergence for the single social case still indicates that the information gained from other agents is wrong, on average. One explanation for this was the

dependency through memory, but there is another possibility. When the agent observes two or more agents, then those are not randomly chosen from the population of all agents, but they are chosen by virtue of being in sensor range of the observing agent. This means the observed agents are likely to have been in close proximity to each other, and therefore observed similar parts of the environment in the past. This means their actions basically give information about similar parts of the environment, and updating with said information becomes redundant, and thereby wrong. The underlying assumption for the update is that the observed agents are randomly selected from the population of all agents, but this is not the case. Not in this simulation, and not in general, as observation is often limited by location and time, as agent mostly can only observe other agents that are close to them, both in terms of time and location.

Concluding this excursion into possible systematic dependencies it seems there are several possible explanations for why the social update can be on average incorrect. Furthermore, the likely explanation for the difference between the all-social and the single-social case seems to be some kind of information cascade, indicated by the alignment of agent's movement, and by the significant change in average KL divergence in the case where agents can observe all other agents.

In general, this area does warrant further studies. The information theoretic tools I utilized here gave some insight into what is happening, and indicated certain tendencies, but it would be nice to further disentangle the different effects. But even this simple simulation already generates a lot of complexity, which makes it hard to further subdivide the different measures. To partially address this I will look at a second simulation, where similar effects can be seen and differentiated with more clarity.

### Lack of information gradient

Moving on to the simulation with 200 agents, it still remains unclear why the agents performance becomes worse when the number of agent increases? Looking at the average decrease in entropy and KL divergence, which are all positive, it also looks like the systematic incorrectness of the social update offers no explanation here. If anything, the information gained from the social update seems to be more correct than for the case with 40 agents. But a closer look at $\Delta H_s$ reveals that the updates now contain a larger number of update events where $\Delta H_s$ is either zero, for the environmental update, or very close to zero for the social update. Similarly, the change in $\Delta KL_s$ also is very small in these events.

An increase in sensor events with little or no information- and correctness gain would explain the lowered average for both the entropy reduction and the reduction of KL di-

vergence. Furthermore, there is a very clear explanation what happens when the $\Delta H_s$ of an environmental update is zero. Since the agent chooses its actions to maximise the gain in entropy reduction a zero indicates that any possible action would have resulted in zero information gain. Therefore, all cells within on time step must have already had an assumed probability of zero, meaning all cells directly around the agent have already been ruled out as food source locations by the agent. This is bad for the agent's performance, as there is no gradient that infotaxis can use to guide the exploration. The agent has to resort to random action selection until it finds an area that still has non-zero probabilities. This has happened even to single agents before, but for larger agent population this effect seems to be more common.

The specific problem here causes by the large number of agents is the increase in likelihood of the following scenario. Imagine the agent is surrounded by other agents, who are all just at the edge of its own sensor range. All of them have just moved, and now the agent performs a social update. One thing that is clear from the statistics is that the agent that just moved had not seen the food in the last round. So, there is an area around the observed agents previous position that becomes completely explored. This area reaches beyond the sensor range of the observing agent. Now, if the agent is surrounded in all directions, then those explored areas might overlap in a way so that the fully remove all information from the world in immediate moving distance. The area becomes informationally dead, and the agent, in our current model, has to resort to random search. This surrounding scenario becomes more likely when there are more agents in the simulation.

#### Alignment vs. Lack of Gradient

What remains now is the difference between the "single social" and the "all social" simulation for 200 agents. Or more specifically, why do the agents in the simulation, where all agents are social, perform better. Looking at the data, it seems that they are less often subject to the lack of an information gradient. This can be seen by locking at the actual data plotted in Fig. 5.7, but is not clearly visible in the actual plot. The agent in the all-social simulation has less environmental updates that are completely zero than the agent in the simulation where the other agent are non-social. This only occurs when all four directions offer no new environmental information, and the agent then chooses one at random. This indicates, that the agent in the non-social simulation is more often in a situation where all local informational gradients are gone.

One difference between the all-social and the single-social case is the possibility for the agent who all use social update to align their movement directions (I will study this in

a quantitative fashion in the next chapter). The lack of a good gradient, as explained in the last section, is caused when agents move into the observing agent's sensor range from all directions. If we now assume that the agents align, or partially align their movement directions, then this becomes less likely. So, the presence of an information cascade could protect the agents here from landing in a fully explored zone without a gradient.

## 5.5 Conclusion for Fishworld Model

The most direct conclusion drawn from the analysis of the fish world simulation, extended with the social Bayesian update, is that the information gained from other agents is indeed helpful in some cases. Using the social update ability has allowed agents, under specific circumstances, to perform far better than any non-social agent could. Therefore, it seems reasonable to conclude that such an adaptation would be reasonable for an agent to have.

But a closer look has also revealed that there are several problems that can arise in which the social Bayesian update ability can be harmful. One such limitation is very specific to the modified infotaxis search, which is in essence a gradient ascent along an informational gradient. If this gradient vanishes around the agent, then the search becomes undirected and inefficient. As demonstrated in the model, this can occur when there are to many agent around to gather information from.

Furthermore, there were also examples of how the Bayesian Update, specifically the Naive Bayesian Update, can fail in a more general way. The simulation where all 40 agent were using the naive Bayesian Update demonstrated clearly, that the gathered information is not just misleading in a specific event, but can be misleading on average. There are several different mechanism that can lead to a systematic dependency, which then can cause the Naive Bayesian Update to be wrong. Several of these dependencies, such as a dependency through memory, or the dependency introduced by observing only those agent in close proximity, are likely present in a wide range of scenarios, and the resulting problems should therefore also be expected across a range of social learning scenarios.

Additionally, there seems to be some form of alignment between the agents, just based on gathering information from the environment and other agents. This phenomenon, which look similar to some form of coordination, will receive some further attention in the next chapter. First however, I will take a closer look at how the treasure hunter model is affected by social Bayesian update.

## 5.6 Multiple Agents Treasure Hunter Scenario

In this section I will integrate the Bayesian Social Update into the Treasure Hunter Simulation and demonstrate how using the information gained from others can change an agent's digested information.

To recap the model: Consider a world with $n$ locations, one of them containing treasure. The location of the treasure is encoded in the random variable $T$, with $|T| = n$. The agents try to locate in which of the $n$ locations in the world the treasure is located. Once per turn an agent action consists of visiting one of those $n$ locations and observing whether it contains the treasure. Should an agent find the treasure, it is immediately replaced by a new, ignorant agent.

The last chapter demonstrated that there is some information about the treasure location $T = t$ in the distribution of the agent's actions $A$, as observed when they embark to a location. So, by observing where other agents are looking for treasure one can gain information about the treasure location.

In the multi-agent simulation in this chapter it is now possible for the agents to observe each other. Observing an agent will tell one where the observed agent is looking for the treasure. It will not reveal whether the treasure was found, or whether the location contains treasure. This information is encoded in the variable $A$.

If an agent observes another agent's action, it will integrate the obtained information into its own internal model. The observing agent will perform a Naive Bayesian Update, based on the statistics gathered from the non-social treasure hunter simulations (Fig. 4.9), to update its own internal probability distribution $\hat{T}$ which encodes the agents assumed probability for the state of $T$. This uses the same formalism introduced in more detail in section on Social Bayesian Update for the fishworld model.

Since the order of actions is important for the results, here is the exact order of what an agent does in its lifetime:

1. initialize internal distribution $\hat{T}$ to the uniform distribution
2. if observing other agents, update $\hat{T}$ with Bayes' Theorem
3. decide to search one of the locations
    4. if treasure not found $\rightarrow$ update $\hat{T}$
    5. if treasure found $\rightarrow$ reset $\hat{T}$ to uniform distribution
6. repeat from 2. onwards

All data discussed here is the average value for 1,000 simulations, each running for 1,000 turns.

### 5.6.1 Single Social Agent

In the first experiment we are looking at 10 agents in a world with 10 locations. Only *one* of the agents has the ability to observe the others. The location of the treasure is fixed, and determined at random at the beginning of the simulation. Every time any of the agents finds the treasure, its internal memory is reset.

Unsurprisingly, the remaining non-social agents perform exactly as in the single agents simulation. Their distribution of actions matches the one recorded in Fig. 4.9, and their performance ratio is 0.180. Performance is measured as the ratio of discovered treasure vs. turns. So, if an agent finds treasure on average once every five turns, it then has a performance ratio of 0.2. This measurement is also identical to the fraction of agent actions that are looking at the right location.

The one social agent in the simulation is performing better; it reaches a performance of 0.30. This agent benefits from the information the other agents gather. As discussed in the "Digested Information" argument, the other agents act as information preprocessors for the social agent. Also, note that the distribution of actions of the social agents is even more concentrated on the actual treasure location, hence the mutual information between its actions and the treasure location, $I(A, T) = 0.220$, is higher than the same mutual information for the non-social agents, which was 0.042. The social agent performs better and its actions encode more relevant information

As an additional side-note I should point out that this is another simulation where the improvement in an agent's performance coincidences with an improvement in the relevant information in that agent's actions. If the agents could be distinguished, then a Bayesian Update with the social agent's distribution could yield even more relevant information than observing a non-social agent.

### 5.6.2 All Social Agents

Based on the success of this strategy, I assume that in the second demonstration all agents have adopted the social update approach. This turns out to be extremely beneficial. The performance of the overall population, which is also the performance of every separate agent is $\approx 0.99$. Once the food is located by one agent, everyone always finds the food. The mutual information between actions and treasure location is nearly maximal, $I(A; T) \approx \log(10)$.

Basically, the relevant information that the treasure is in location $t$ propagates through the agents. It is displayed in agent's actions, then used to update another agent's internal model, and then that agent uses the information to determine which action to take, which is going to be $A = t$. The agent will then find the treasure and reset its internal model. But it will perceive others before it has to act again, biasing its internal model again towards taking action $A = t$. This will continue unless environmental information conflicts with this information, meaning the agent will not find the treasure in the location it was looking. In that case, the observed location's probability to contain treasure it set to zero, and the agent will look at other locations. This will initially get the agents to explore all locations until they find the treasure, and then they will all copy each other, finding the treasure every turn from that point onwards. Note, that the treasure does not move when it is found, just the agent who found the treasure is reset.

As we see, the important information is preserved by continuously flowing through the agents population. Even when agents die and are replaced, the information is not lost. This looks like a very desirable feature for an agent population, and therefore the Social Bayesian Update seems like a reasonable adaptation for the whole population. But the next simulation will show that the very same adaptation can have negative consequences for the agent's performance, if the simulation is just slightly altered.

### 5.6.3 Changing World State

In the next simulation the locations of the treasure will change during the simulation to a different random location. This will happen ever turn with probability of 0.01. On average this should change the location every 100 turns. The behaviour of the agents is left unchanged.

First, let's again take a look at the simulation for a single agent. The performance of the agent drops from 0.182 for the static world state simulation, to 0.148 for the simulation where the world state changes. A closer analysis shows that the agent's original behaviour has problems dealing with the changed scenario. Consider that the agent visits a location $x$, and finds it empty. Then the probability for $T = x$ will be set to zero in $\hat{T}$. If the location now changes to $T = x$ *after* the agent visited $x$, then the agent will first explore all other locations, finding all of them empty. This, in itself, is not problematic. But once the agent looked at every locations once, all probabilities are assumed to be zero, given that the agent still assumes there is one, non-moving treasure location. This is inconsistent with the basic properties of probabilities, which is a result of the incorrect assumption about the immovability of the treasure location. In this specific implementation, this means that the agent now resorts to random search. This behaviour has, as we have seen,

a lower performance rate, and therefore lowers the agent's overall performance.

### Modelling Uncertainty

To address the problem in the changing world scenario, I will introduce uncertainty into the Bayesian model. This will also make the model more correct in general, as it produces a more exact model of the actual probabilities from the agent's perspective.

Assume the treasure changes its location with a probability of $P(change) = 0.01$ and assumes another of the random $n$ locations. This can be modelled by assuming that the world is in one of two cases. Either, with $P(change) = 0.01$, it is in a case where the location has just changed, so $T$ should be uniformly distributed with every $t \in T$ having the probability $P(T = t) = 1/n$. The other case, with a probability of $1 - P(change)$, is the one where the locations remains unchanged, so the agent should continue to assume the distribution represented by its internal model $\hat{T}$. These two cases can be combined in a weighted sum to determine a new internal distribution $\hat{T}'$. The probability for every state $t$ in this new distribution can be computed as

$$P(\hat{T}' = t) = P(change)\frac{1}{n} + (1 - P(change)) \cdot P(\hat{T} = t). \tag{5.17}$$

To model the uncertainty, the formula should be applied to the agent's internal model each turn. Note that this leaves the ordering of probabilities from the most likely to the least likely event intact, unless the probability of change would be 1. Therefore, the single agent behaviour with modelled uncertainty performs still just as well as the agent without, assuming the location is not changing. But, applying the above uncertainty model to a single agent in a world where the treasure location does change increases its performance from 0.148 (for the agent without uncertainty) to 0.180.

The performance increases here because the agent modelling uncertainty retains some information about the order in which it explored the previous locations in its internal model. The location that has been visited first and found empty had uncertainty applied to it for nine times, once the agent cleared the last, tenth location. It therefore has the largest probability to contain the treasure, and will be the first location to be visited again. This actually reflects the fact that this location is most likely to contain the treasure, since it is unclear when the treasure changed location. If it changed location after round 1, then it would have to go to the first location. If it changed in round 2, it could either go to the first, or the second location, etc. After the 10th round, when every location has been visited once it is clear that the location has changed at one of the nine times in between searches. The resulting probability $p(1st)$,that the treasure is in the first visited location

can then be computes as:

$$p(1st) = \frac{1}{9}\left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{9}\right), \tag{5.18}$$

whereas the second locations has a probability of

$$p(2nd) = \frac{1}{9}\left(\frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{9}\right). \tag{5.19}$$

The later the location was visited the lower the probability becomes. The first location is the one location the treasure has most likely changed to.

This also shows why modelling the uncertainty works better than simply resetting the probabilities after all locations were visited and found empty. This would reset the internal model and prevent the agent from having to use random search, but it would not preserve the ordering of the previous search, which could be used to the agent's advantage.

### 5.6.4 Uncertainty and Social Bayesian Update

**One Social Agent:**

The next simulations now has multiple agents with the ability to model uncertainty and a changing treasure location. First, lets take a look at a simulation were only one agent observes the other agents. The one social agent will observe every single action taken by the other agents and update its model accordingly. It will also apply uncertainty to its own model after taking its own action.

The non-social agents again perform just as in the single agent simulation (with uncertainty), as their behaviour remains unchanged. Their performance is 0.18. The agent that does use the social update is doing worse than that, having a performance of 0.174.

This is the result of internal uncertainty combined with repeated social Bayesian updates. A closer look at the agent's behaviour reveals that there are certain situations in which the agent revisits a previously explored location rather than exploring those it had not yet visited. Lets say the agent has explored location 3 already, and has then later applied a degree of uncertainty to it assumed probability distribution. It then assumes, internally, that there is a small chance that the treasure is in location 3. If several other agents now all take action 3 in the next round, then observing each of those agents will repeatedly update this small probability to a larger probability. This might cause the agent to explore location 3 again, even though it had just been explored. This could not happen in the previous simulation, since without uncertainty the prior for the update would be a

probability of 0, which would remain a probability of 0 after the update. But the small amount of internal uncertainty made the agent susceptible to the influence of other agents.

**All Agents Social:**

Extending the Social Bayesian Update ability to all agents has even worse results. If all agents update their internal probability distribution with the other agents' actions, and also apply uncertainty, then the performance of all agents falls to 0.1. A closer look at the distribution of overall actions reveals that all agents are always exploring the same location. This behaviour is somewhat similar to what happened in the case where all agents had the social update ability, but without uncertainty or changing treasure location. The difference here though is that the agents will all go to the same specific location regardless of where the treasure actually is.

What happens is this: Initially, one agent chooses a location $x$ at random, and all others update their internal distributions accordingly, making this location more likely. The first agent then updates its own model, assigning a probability of 0 to that location. Then it will apply a degree of uncertainty to account for the possibility that the treasure changed location. The other agents then all visit the same location $x$ as the first agent, as it is more likely to contain the treasure than any other location. This is according to their own internal model, based only on the actions of those agents that acted before them. The first agent now observes all other agent going to location $x$, updating it own internal model. When it is the first agent's turn to act again, all the repeated updates from the other agents will have "convinced" the first agent that $x$ is the most likely location for the treasure to be in, and the whole process will be repeated from the beginning. Basically all agents reinforce each others behaviour, getting stuck in a feedback loop that is not dependent on the actual input from the environment. This is a classic example of the previously mentioned information cascade.

The roughly 10% of found treasure simply result from the fact that the treasure changes location and coincidentally actually appears where the agents are looking anyways in 10% of the cases. If the treasure location would remain unchanged and the agents would initially pick the wrong location, then the performance rate would go down to 0.

So, while the Social Bayesian Update is very beneficial for the agents in some cases, it turns out that it can even be harmful, specifically when combined with a more accurate model of uncertainty. The next simulation takes a closer look at the problem that a repeated update from other agents' actions seems to dominate the information from other, non-agent sensor inputs. I will show that this can be alleviated by neglecting some of the agent's input.

### 5.6.5 Partial Observability

For the next Treasure Hunter simulation I assume that all agents apply uncertainty to their model ($P(change) = 0.01$) and also use the Social Bayesian Update with a distribution based on the non-social agent's behaviour whenever they observe the actions of another agent. The treasure location does change, also with a probability of 0.01. Different to the other models, only a fraction of the other agent's actions can be observed. Every time an agent takes an action ever other agent has a probability of $p_o$ to observe this action and update its internal model. Whether an agent can observe a specific action is determined for each observing agent separately.

This basically creates several simulations interpolating between two previously studied simulations. If $p_o = 0$, then the model would be identical to the non-social agent simulation, and if $p_o = 0$, then it would be identical to the one where all agents could observe each other, which led to a feedback loop and very bad performance ratios.

#### Changing Observation Probability for all Agents

Varying the parameter $p_o$ for all agents results in performance ratios as depicted in Fig. 5.8. As expected the extremal points have similar performance to the non-social and all-social models. In the case where no agents observe each other the agents find the treasure on average 0.18 times per round. The performance ratio increases as the chance to observe other agents increases, up to ca. 30 % observation probability, where all agents have a performance ratio of ca. 0.32. Increasing the observation chance further lowers the performance again down to about 0.1 at an observation chance of more than 50 %. The performance stays this low for larger observation chances for the population.

The second line in Fig. 5.8 is the mutual information between the agent's actions $A$, and the treasure location $T$. We see that $I(A;T)$ has the same value as for a non-social agent when the observation probability is zero, then it rises to a peak of ca. 0.45 bits for an observation probability of 30 %. The mutual information then decreases for larger observation chances, down to zero mutual information for values above 60 %.

### 5.6.6 Interpretation as Information Cascade

Following from earlier arguments it is unsurprising that increasing the observation probability from zero upwards leads to an increase in performance. Agents do encode relevant information about the environment, and when other agents occasionally observe others, and use this information, their performance increases, since they have more relevant information about the environment. The interesting effect here is the decrease of performance,

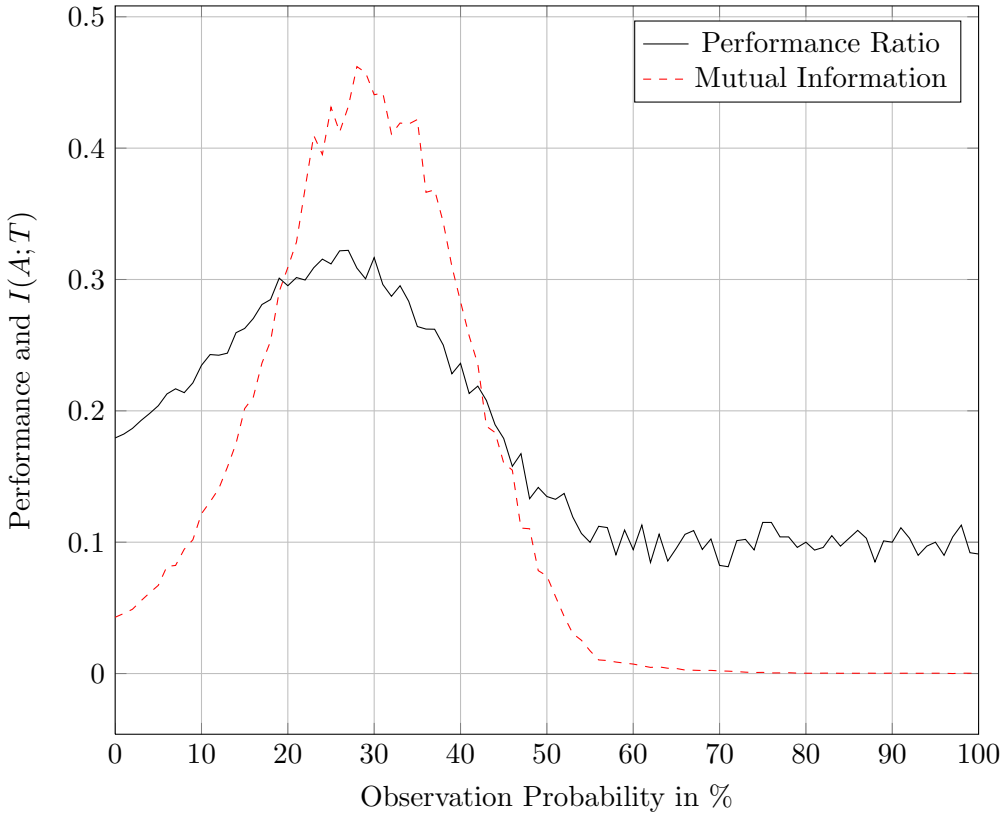



Figure 5.8: A graph depicting both the average performance of an agent population, and the mutual information between its actions and the treasure location, depending on the probability to observe the actions of other agents. The values are calculated for 100,000 recorded actions of the agents for each percentage of observation probability.

once all agents have a observation probability higher than 30 %. Why is a further increase in obtained social information suddenly detrimental to the agent's performance? The problem here can be understood as an information cascade.

Both simulations, the one where agents model uncertainty and the one where agents do not, exhibit clear signs of an information cascade in the case where all agents observe each other. This is not unusual, as both simulations fulfil all the previously identified criteria:

- The simulations contain agents that perform actions sequentially.
- There is private information for the separate agents

- The agents employ social Bayesian learning
- Everyone can observe the action of everyone else

In the context of information cascades, the fact that one simulation models uncertainty can basically be understood as creating a different set of priors for the updates.

A detailed account of the process that leads to the information cascade is as follows. Initially, the first agent makes a random choice where to look for the treasure. The next agent updates its internal model, and the position the first agent looked at becomes more likely to contain the treasure, so the subsequently acting agent also looks there. All other agents will likely follow. If there is *no* uncertainty, then this "cascade" will end after all agents looked at the location once. If there is uncertainty, and if it is applied directly after the agent visited a location, then we are dealing with a very similar situation to the one discussed earlier, where an agent was convinced by others that a previously visited location could, with high probability, actually contain the treasure. But in the current case, where all agents are using the social update, this is not just a random co-occurrence of the other agents action, but a population-wide synchronization, resulting from the Bayesian Update. The data from the simulations indicate that basically all agents always move to the same location. So, after applying uncertainty, every other observed agent would indicate that the food is in a specific location. This then causes all observing agent to also go to that location.

In the first simulation, the one that does not model uncertainty, this leads to an information cascade that makes all agents move to the right location. Every other cascade is aborted after one round, as all agents realize that the treasure is not actually there, and their zero probability prior makes them insusceptible to social information from other agents. This leads to a whole population of agents finding the treasure nearly every turn, which results in nearly perfect agent performance.

In the simulation where the agent's internal model has added uncertainty it is possible for the agents to synchronize on a specific location in a similar fashion, but cascades for incorrect locations will not be aborted. Therefore, the location the population synchronizes on might be the wrong one. Since none of the agents are sure that this is not the right location (because the location could have switched), this wrong synchronization becomes persistent. Finding the treasure then comes down to random chance, the chance being that the treasure location switches to the position the population already believes the treasure to be in. Alternatively, the agents could get lucky, and the agent who moves first might choose the actual location of the treasure at random.

So, while it is clear what happens when all agents can observe all other agents, the

more interesting cases here are those with a limited chance to observe the other agents. With very low observation probability the agents act very similarly to non-social agents, and have a similar performance. As the observation chance increases, so does the agent performance. The information from the other agents is used to improve performance and agents are able to find the treasure ca. 32 % of the time. This also is accompanied by a significant increase in the mutual information between the treasure location and the agent's actions. The agent population as a whole has good information about the location of the treasure and retains this information to a degree. On the other hand, once the treasure location changes the population is able to switch their internal models, and then prefers going to the new location. Those ca. 68 % of the actions that do not locate the treasure can be understood as an investment in exploration.

Once the observation chance gets higher than 30 %, the performance starts to drop. The agents still synchronize, but this synchronization is not subject to environmental information. This can be clearly seen in the development of the mutual information, which drops to zero. Above 70 % observation chance the actions of the agent population have no correlation with the actual location at all. The increase in observation probability makes the population more susceptible to information cascades.

The observed phenomenon can also be understood in regard to the underlying network topology. As Gale and Kariv (2003) prove, social Bayesian learning in a network leads to uniformity if the network has a certain connectivity. As the chance of random observation increases the network describing which agents observe each other transforms from one of separated clusters to a fully connected network. Complementing this work, Acemoglu, Dahleh, Lobel and Ozdaglar (2011) prove which network topologies will lead to asymptotic learning, meaning that eventually all agents will converge on the right solution or behaviour. Both their work applies to learning with persistent agents in a network with random but persistent structure, while our model here removes and add agents, and has probabilistic observation probabilities. Therefore, their work does not directly apply to my model, but still suggests that the network structure and specifically the connectedness would play an important role in learning process.

Particularly interesting here is that there seem to be two different transitions when the network of observation becomes more complete. First there is a increase in uniformity that also leads to a high degree of correct behaviour. Then, the uniformity rises even more, but tends to converge on a random location, with no correlation to the actual information in the environment. So there seems to be a trade-off here between getting a lot of information from other agents, but at the same time still being able to incorporate the information from the environment.

### Changing Observation Probability for one Agent

If the observation probability is understood as the result of an agent's effort invested in observing others, then it could be treated as a behavioural parameter that the agent, or at the least the process that adapts agents, could control. This could be realized by deliberately degrading the agent's sensors to save resources in case of an adaptation process on the agent's population, or by simply discarding some of the sensor input at random if this is realized as an agent strategy. In this context it would make sense to ask if an individual agent could perform better than the rest of the population, by unilaterally changing the probability to observe others.

Given that the actions of the remaining population provide a high degree of mutual information it might be useful to obtain more of this information than others do. On the other hand, there were also indications that taking in too much information from others might override the information from the environment, and thereby degrade the agent's performance. So deliberately lowering the social information intake might also improve the agent's performance compared to the rest of the population.

In the next simulation we will look at one agent that can change its observation probability independent from the rest of the population. The observation probability for an agent determines how well it can see others, not how well it can be seen. That means that whenever this agent would observe another agent's action, its own observation probability would be used to determine whether this agent could actually sense what action the other agent took.

All other agents in the simulation have a fixed observation probability of 30 %, since this was the value that lead to the best performance for the overall population, and also encoded the most relevant information.

In Fig. 5.9 we see the resulting performance ratio and mutual information $I(A;T)$ for varying $p_o$ for the one agent that can change its observation probability. Overall, the graph looks very similar to the previous graph where all agents could change their observation probability. The performance for that one agent is still optimal at $\approx 30\%$. Scaling down the observation probability to zero, obviously has the same performance as the non-social agent. Increasing observation probability still also still lowers the performance to ca. 0.1.

This is particularly interesting, because for this specific simulation it creates something akin to a game theoretic equilibrium at the 30 % point. Even if all agents could change their own observation probability at will, none of them could change it away from 30 % without also decreasing its performance, all other factors being equal. Additionally, the mutual information $I(A;T)$ for the specific agent is also largest at 30%, which is at the same time the relevant information the agent's actions provide to other agents. While not

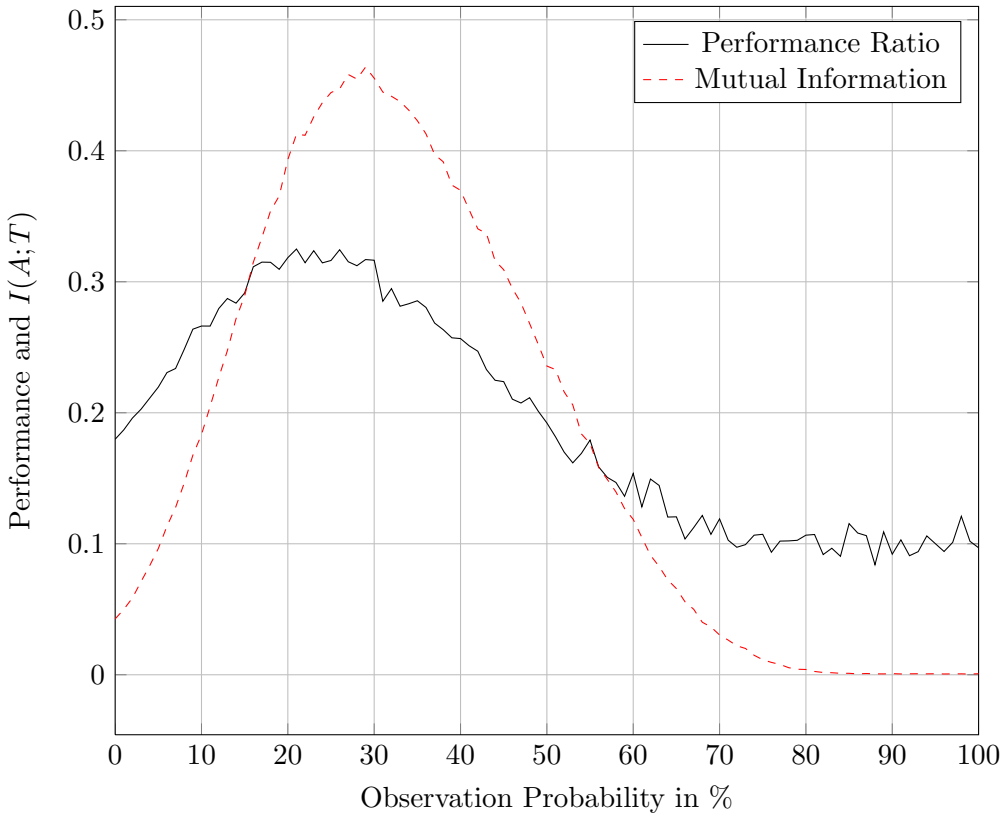



Figure 5.9: A graph depicting the performance of a single agent, and the mutual information between this agent's actions and the treasure location, depending on the probability to observe the other agents in the population. All other agents observe each other with a probability of 30%. The values are calculated for 100,000 recorded actions of the agents for each percentage of observation probability.

doing so deliberately, agents still provide valuable information to each other. In this case, they provide the most at the same point where they have the best performance, as seen in Fig. 5.9. Thereby, an agent that is interested in improving its own performance, is also motivated to process and provide as much relevant information as possible.

### 5.6.7 Comparison to Relevant Information Function

The idea that more performance leads to more encoded relevant information relies on the assumptions that the relevant information function $RI(u)$, which tells us how much information is needed for a given performance level, is monotonically increasing, and on

the assumption that the agents strategies actually lie on, and not above the trade-off function. Since it is possible to compute the actual RI function for the treasure hunter model, we can compare the achieved values to the function, and thereby how efficient the agents use their information.

### RI(u) for the Treasure Hunter Model

The relevant information for the treasure hunter model is determined by the distribution of the treasure, encoded in $T$, and a specific agent's action distribution, encoded in $A$. Both random variables are defined over the same alphabet, which corresponds to all possible locations in the world.

As relevant information is a property of the environment, and not of a specific agent, it considers all possible strategies $P(A|T)$, regardless of how any specific agent would acquire the information needed to actually implement this strategy. To determine the value for $RI(u)$ we have to answer the question, which joint distribution of $A$ and $T$ that has at least a performance level of $u$ has the lowest mutual information.

Since the treasure relocates randomly we know that the marginal probabilities for any specific state $t$ of $T$ are $p(t) = 1/|T|$. Now, for any specific state $t$, to achieve an average performance of $u$, with $0 \leq u \leq 1.0$, the agent has to employ a strategy that chooses the right action with the probability of $u$, hence $P(A = t|T = t) = u$. It follows that all other states of $A$ together share the remaining probability. Since the distribution of the other states does not matter in terms of performance, the remaining probabilities should be distributed uniformly in $A$ to minimize the mutual information:

$$P(A \neq t|T = t) = \frac{1-u}{|A|-1}. \tag{5.20}$$

This allows us to compute the conditional probability P(A—T=t) for a strategy that both achieves performance level $u$, and has minimal mutual information. With this we can compute the conditional entropy as

$$\begin{aligned} H(A|T) &= -\sum_t P(T=t) \sum_a P(A=a|T=t) \log(P(A=a|T=t) && (5.21)\\ &= -|T|\frac{1}{|T|}\left(u\log(u) + (|A|-1)\frac{1-u}{|A|-1}\log\left(\frac{1-u}{|A|-1}\right)\right) && (5.22)\\ &= -\left(u\log(u) + (1-u)\log\left(\frac{1-u}{|A|-1}\right)\right) && (5.23) \end{aligned}$$

The relevant information then is the mutual information,

$$I(A;T) = H(A) - H(A|T). \tag{5.24}$$

For our specific example of a world with ten locations we can therefore compute the relevant information function as

$$RI(u) = \log(10) + \left( u \log(u) + (1-u) \log\left( \frac{1-u}{9} \right) \right). \tag{5.25}$$

Note that this is the function that computes the minimal mutual information for being on a specific performance level $u$, not for having a strategy that at least has the performance level $u$. But looking at the actual function, which can be seen in Fig. 5.10, it becomes clear that the function is, for values of $u$ over 0.1, strictly increasing. Therefore, the minimal mutual information for a specific performance level above 0.1 is also the actual relevant information needed to perform at least that well. There is no strategy that performs better with less mutual information, and as a result, the graph computed with Eq.(5.25) is the actual relevant information function for all values above 0.1.

The previous distinction is necessary, though, because in this case it is necessary to process information to have a performance level lower than 0.1. A performance of 0.1 can be achieved with a random strategy, and therefore has no relevant information. Eq.(5.25) does reflect this, as it is zero for $u = 0.1$. For values of $u$ lower than 0.1 the function in Eq.(5.25) computes values higher than zero, which would be the information necessary to actually perform *at* this level. One would have to actively avoid the treasure. But by previous definition relevant information should return the information needed to at least attain a specific level, and since random performs better, and has no relevant information all performance levels below $u = 0.1$ have zero relevant information. This is reflected in the graph in Fig. 5.10, which therefore differs from Eq.(5.25) in values below 0.1.

The data points plotted in Fig. 5.10 are taken from the two previous simulations, those where all agents changed their observation probability, and those where only one agent changed its observation probability and all other agents had an observation probability of 30 %. Each point is the combination of the mutual information $I(A;T)$ and the achieved performance ratio for a specific percentage of observation probability. Different observation probabilities result in different strategies, i.e. different conditional probabilities $P(A|T)$.

The data points gathered here are, as expected, all above or on the RI trade-off curve. The values developed very similarly for both simulations. For an observation probability of 0.0 the data point is located at a performance of 0.18, and actually on the trade-off

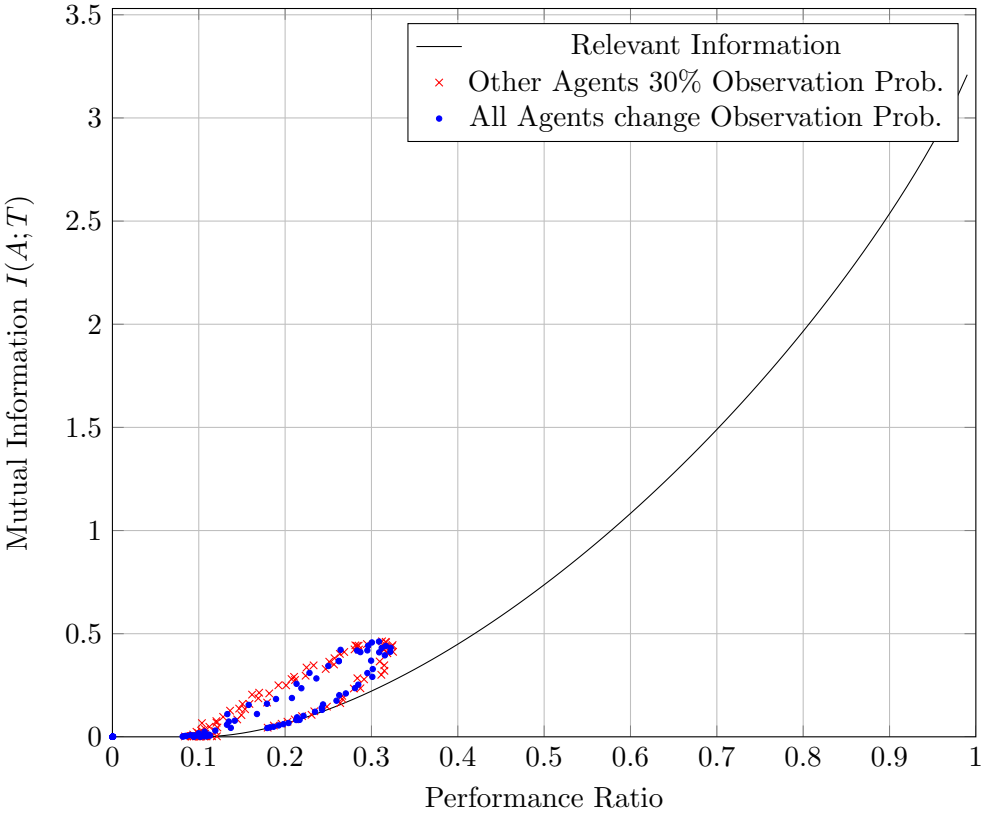



Figure 5.10

curve. As the observation probability increases so does the performance. The strategies still stay on the trade-off curve for the lower percentages of observation probability, and since the trade-off curve is strictly increasing, so does the encoded relevant information.

As the observation probability gets larger we see that the resulting data points leave the trade-off curve, which means the resulting strategies encode more mutual information about the environment than necessary. The strategies for further increases in observation probability are located in the upper loop where they gravitate towards a point of no mutual information and a performance of 0.1. This indicates that they also encode more information about the environment than necessary.

This comparison of the mutual information in the actual strategies to the actual relevant information illustrates how observing more and more agents leads to processed information, which might not necessarily be relevant. The strategies with low observation probability are located on the actual relevant information trade-off curve, meaning

they are efficient in the regard that they do not process non-relevant information. The strategies which are subject to the information cascade on the other hand do display a lot of information about the environment in their actions which is non-relevant. At the same time, as seen here, their performance diminishes as well. Fortunately for the agent population, the point where agents display the most relevant information about the environment is also roughly the point where the agent performs best, so it would be possible for an agent population, which could adjust their observation probability, to stabilize in the point which benefits all agents the most.

### 5.6.8 Conclusion for Treasure Hunter

The treasure hunter model offered a chance to perform a clearer analysis of the information provided by the agent's actions, and how it does affect other agents in turn. Also, it allowed us to better study how information changes when it is processed through several agents.

Similarly to the fishworld model the naive conclusion here is that the information in other agent's action can increase an agent's performance. Especially the lone social agent benefited from observing others. Furthermore, the simulation provided additional evidence that an increase in performance leads to an increase in digested information, i.e. an increase in the mutual information between agent's actions and the treasure location. As we saw, the mutual information between the food source locations and the agent's actions was maximal at roughly the same observation level.

Also, similar to the fishworld simulation it became clear that the usefulness of the acquired information depends on several factors, such as the application of uncertainty to the internal model. Without uncertainty the system proved to be relatively stable; more information was always more helpful. All agents observing each other led to an information cascade, but one that preserved the location of the treasure in the agents population, even though none of the agents in the population had actually seen the treasure. This was particularly helpful for the agent population when new agents entered the population.

The interesting case was the simulation with uncertainty in the internal model of the agent, where more processed information would lead to problems, as several sources for information now had to be balanced against each other. An increase in the chance to observe each other first leads to an increase in overall performance, but if too much social information was coming in it eventually overrode the actual environmental information, and the agents would synchronize on an arbitrary choice.

While this might look bad for the digested information hypothesis, it is in fact addressing the first research question of this thesis. In the area of artificial life the main interest is to reproduce the behaviour of living systems. Systematic errors of living systems are

particularly interesting, as they offer insight into how a systems operates. Making an evolutionary argument for a system that always operates perfectly is simple, the benefit of such a system are clear. But producing a reasonable systems that produces systematic errors similar to those observed in natures is far more interesting, as it indicates that nature might operate with the same, or functional similar mechanisms.

For this specific case there are rough similarities to phenomena such as mass hysteria, cargo cult believes and run-away fashion fads. As discussed in the introduction, these are often subsumed under the general concept of information cascades. While they can be very harmful, they also seem to be an existing phenomena in biological systems. The fact that our model produces similar phenomena is therefore interesting, rather than problematic.

Relating the last model to information cascades in general also leads to the question what additional insight the fisherman model can provide here. Of particular interest here would be the existence of a Nash equilibrium of social information intake, where no agent by itself could change the chance of how often it would observe others without a decrease in its own performance. As an agent would (roughly) provide the most relevant information by operating at the observation level where it would also perform best, the agent is motivated to remain at that observation level, for its own benefit. Thereby it would also provide the largest amount of relevant information. No single agent could thus switch its strategy unilaterally without losing performance.

At the same time this also showed that the process of providing information to others and using such information is dynamically linked. Using information from other agents changes the information an agent provides itself. In general this leads to a game theoretic scenario, where the question of how much information from others an agent should process is not just a static optimization process. By collectively processing more information from others a situation might arise in which the very information agents provide to others vanishes. So, if everyone tries to process as much social information as possible, this might lead to no social information for anyone. In this specific scenario there existed a specific equilibrium point where an individual agent was both performing optimally and providing the most information to others with the same strategy. But this begs the question if such equilibria for information processing always exist?

Similarly, there are also a lot of open questions regarding the network structure and its influence on social Bayesian learning. Recent work, as discussed earlier (Gale and Kariv 2003, Acemoglu et al. 2011), shows that both the network structure and the internal priors influence if the population will converge, and if it will converge on the optimal solution. If an agent can influence how much "resources" are spent on specific sensors, or at least determine where to steer its attention, then this agent might actually be able to

change the very network structure it is located in. From the perspective of information maximisation it might in fact be reasonable to actually discard certain inputs, in order to increase the overall "quality" of the information that is provided via the network of possible observations.

# Chapter 6

# Flocking Behaviour

## 6.1 Chapter Overview

This chapter demonstrates how the previously introduced infotaxis behaviour, combined with the social Bayesian update, can lead to flocking behaviour. I aim to demonstrate that the maximisation of relevant information alone is sufficient to generate behaviour similar to flocking.

I will first give a short introduction to flocking behaviour in nature, in general, and to Reynolds boids rules (Reynolds 1987), in particular. It will serve as a baseline to compare our results against. I will then present a slightly modified version of the earlier gridworld search task, incorporating both infotaxis and Social Bayesian Update. I will also introduce some measurements for alignment, local density and collision, to investigate the prime properties boids-like flocking should display. Finally, I will present the results, and discuss in a less technical frame how information maximisation leads to those properties.

## 6.2 Introduction

### 6.2.1 Motivation

Observation of the agent's movement in the social infotaxis simulations in the last chapter indicated that the agents might move around in groups, forming something akin to swarms or flocks. The purpose of this chapter is to verify the existence of this behaviour in a more quantitative fashion.

The main hypothesis is that agents controlled by infotaxis and Social Bayesian Update, as described in Chapter 4 and 5, will form flocks of agents, similar to the behaviour

generated by the boids rules (Reynolds 1987).

This is particularly interesting in regard to research question 1, which asked if optimization of information processing leads to agent-agent interaction. Flocking or swarming is clearly an interactive behaviour that requires some form of coordination. Embracing the bottom-up, artificial life perspective, it would be good to demonstrate that optimization of information processing could lead to such behaviour.

For the following model we will therefore assume that agents indeed optimize their intake of relevant information. We will also assume that these agents somehow adapted to display behaviour functional equivalent to infotaxis and the Social Bayesian Update described in the last chapters. We can then ask if it is possible that these behaviours generate behaviour similar to flocking?

## 6.3 Related Work

### 6.3.1 Animal Aggregation in Nature

Flocking behaviour is a natural phenomenon found in a diverse selection of life forms. Spatial aggregation of animals have been observed in bird flocks, fish schools, mammalian herds, and bee swarms (Allee 1931, Lissaman and Shollenberger 1970), just to name a few examples. Dyer, Ioannou, Morrell, Croft, Couzin, Waters and Krause (2008) even demonstrate that humans, under specific circumstances, exhibit similar flocking behaviour in large crowds.

In general, the flocking phenomenon is a widespread and well documented example of local, agent-centric self-organization. There is no central entity that controls or creates the flocking, but it emerges nonetheless, as a result of the individual agents' behaviour, which in turn is based on the local information available to those agents.

The possible explanations for flocking behaviour, or animal aggregation, are numerous. Depending on what kind of animal we are talking about, flocking offers several benefits. It protects the individuals from predators, offers an increased choice of mates, and adds the possibility that other flock members might be aware of food sources, predators or migratory routes that the individual is not (Camazine 2003). Several of those reasons can be conceptualized as forms of information transfer. This might be the information about mates, food sources, predators, or other factors in the environment that are important for the agent. In essence, these cases are examples of relevant information being shared between the flocking agents.

Incidentally, if we look at several of the earlier biological examples for Danchin's "Inad-

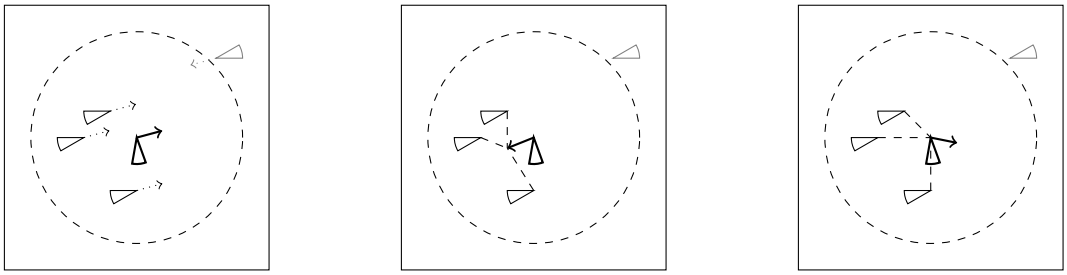



Figure 6.1: Three figures illustrating the three basic boids rules: alignment, cohesion and separation. The agent in the middle of each figure determines its movement direction by observing the locally visible agents (those in the dashed circle). The resulting direction for each rule is the thick arrow. The actual resulting movement is a weighted linear combination of the three indicated direction.

vertent Social Information" (Danchin et al. 2004) we see that the animals who exhibited the ability to use the digested information of other agents, such as bees (Baude et al. 2008) and birds (Parejo et al. 2008) are also animals that exhibit flocking or swarming behaviour.

Further supporting evidence for the relation between swarm behaviour and information transfer are several recent studies into the informational properties of artificial swarms. Couzin, Krause, Franks and Levin (2005) demonstrate that information known to only a subset of the swarming agents is still sufficient to guide the overall movement of the swarm. Also, Wang, Miller, Lizier, Prokopenko and Rossi (2011) demonstrate that agent aggregations exhibit certain information theoretic properties if their behaviour is created with the boids flocking rules. Specifically, information storage (Lizier, Prokopenko and Zomaya 2007) and transfer entropy (Schreiber 2000) between agents becomes larger when the swarm organizes from a more to a less fragmented configuration.

So it seems plausible that flocking behaviour enhances the ability to use the information of other agents and in theory it could even be caused by maximising one's own information intake.

### 6.3.2 Boids

One of the first models to recreate this behaviour in a computer simulation is the *boids* steering model, introduced by Reynolds (1987). Originally developed to animate the movement of fish and birds for graphical presentation, the boids model has developed into a *de facto* standard for flocking algorithms.

The three basic rules, alignment, separation and cohesion, are agent based and local, so they allow every agent to determine its own actions by itself, using only local data:

**Alignment:** Steer towards the average heading of local flock mates. The agent adds all the movement vectors of the locally visible agents. The resulting vector is the agent's desired movement direction, based on the alignment rule.

**Cohesion:** Steer towards the average position of local flock mates. The agent averages the position of all visible agents. The vector pointing from its own location to that center of mass is the cohesion component of the agent's movement.

**Separation:** Steer to avoid crowding local flock mates. The agent determines the difference vectors between itself and each other locally visible agent. Based on those vectors the agent creates and repulsion vector for each visible agent. This vector points in the exact opposite direction, and is longer the closer the other agent is. The sum of those repulsion vectors is the separation component of the agent's movement.

The agent's actual direction of movement is a weighted linear combination of the three vectors for alignment, cohesion and separation. Each of them weighted with a coefficient that determines how strong that specific component, or rule, influences the agent's overall behaviour.

This model, or variations thereof, are not only the basis for many current flocking and swarm simulations, but are also a powerful example for how simple, local rules can lead to the emergence of complex, life-like properties.

Furthermore, artificial flocks based on boids rules have also been used to perform geographic location tasks (Macgill and Openshaw 1998), demonstrating how flocking makes agents better at processing data related to locations. This is related to my model, as the agents in the fishworld model also try to find a specific location. In contrast, Macgill and Openshaw (1998) introduced flocking explicitly to increase the performance, while in our model flocking is a by-product of the optimization of information processing.

## 6.4 Information based Flocking

What I want to investigate in this chapter is if the phenomenon of self-organised flocking can be produced by the optimization of information processing. But instead of motivating the individual atomic rules for separation, cohesion and alignment, I will investigate if infotaxis and Social Bayesian Update will generate a group behaviour similar to flocking.

In my model the individual agent's actions, and the resulting global flocking behaviour, is created and motivated by obtaining as much relevant information about the environment as possible. This is an additional result of the previous efforts to extend information theoretic-behaviour generation in general, and in particular the biologically inspired *infotaxis* model by Vergassola et al. (2007), to a multi agent system.

In the original infotaxis model the sensor inputs from the environment are used, via a Bayesian Update, to update an internal probabilistic model about a specific location. Actions are chosen based on how much expected information gain they provide for the internal model. In the multi-agent model, the actions of other, observable agents are treated with the same Bayesian update.

The focus of this chapter is to evaluate this claim by looking at some quantitative data regarding the agents' flocking behaviour. I will use a slightly modified version of the earlier infotaxis driven grid world search, and introduce some measurement to verify the existence of flocking behaviour.

### 6.4.1 Experimental Model

As before, we are looking at a grid world model with periodic boundaries. There is one single location of interest, which I will call the location of the *food source*, but one can interpret it as any other relevant location information, such as position of shelter or mates. The goal of the agents is to determine (not reach) this location in the shortest possible time.

The agents' initial location, and the location of the food are randomly initialized at the start of the simulation. The agents all use the infotaxis behaviour to locate the food source, and all of the agents are using the Social Bayesian Update when they encounter another agent incidentally. Both behaviours have been described in detail in the chapters on "Digested Information" and "Social Bayesian Update", respectively.

Different from the other models, this simulation includes collision detection. If an agent tries to move into a cell already occupied by another agent it will remain in its originating cell.

Once an agent finds the food, the agent still disappears. An agent that has disappeared does not block other agents, cannot be observed, and its behaviour is not taken into account for the statistical measurements. Note also that the food source itself is unaffected from agents finding it.

The above scenario determines the basic properties of our setting. Now, as I am interested in flocking behaviour, for an effective evaluation, the simulation will be run continuously, so the agents have time to form a swarm. Thus, instead of reinitializing the

simulation every time one or all agents find the food source, at each time step there is a 3 % chance that the food will be randomly relocated. In this case, the internal models of all agents are reset, so they start a new search. Those agents which have disappeared because they found the food will also be put back into the world in the location they previously disappeared from. The purpose of this is to allow swarms that have already formed to continue their coordinated movement.

If several agents are on the same cell when they re-enter the world they will be put into the same cell. They can still not move into a cell were there is another agent, but they can leave from a cell that contains several agents.

## 6.5 Measurements

While flocking behaviour is visible at this point in our model, defining an objective overall measure which quantitatively captures the emergent flocking behaviour seems difficult. A direct action-to-action comparison between boids rules and infotaxis is problematic. First, because flocking in its original form is not well defined for a discrete grid world. Second, the question here is not if the underlying micro-behaviour is identical, but if infotaxis can lead to similar macro-behaviour of the overall swarm.

Instead, I aimed to measure the immediate effects that behaving according to the boids rules should have. For that, I defined the following measurements.

### 6.5.1 Alignment

To quantify the *alignment* of the different agents, I added up all the agents' movements and took the length of the resulting vector and normalised it. I.e., every agent $x \in \mathcal{X}$ has an associated vector

$$\vec{v}_x \in \{(1,0),(0,1)(-1,0)(0,-1)\} \tag{6.1}$$

corresponding to the last direction it moved in. The *global alignment* is then calculated as the length of the sum of all agents' vectors, divided by the number of agents:

$$alignment = \frac{|\sum_{x\in\mathcal{X}} \vec{v}_x|}{|\mathcal{X}|} \tag{6.2}$$

This results in a value between 1.0 and 0.0. The maximum value is reached when all agents move in the same direction, and the lowest value of 0.0 is attained when the movement of all agents is distributed evenly among those moving north and south, and those moving west and east, respectively. Note again, that agents which have found the food are not

taken into consideration for this measurement, since it would be irrelevant to measure how well aligned they are, once they are not moving anywhere.

This measurement is taken for every simulation step, and an average over all simulation steps is then calculated for the whole simulation.

### 6.5.2 Cohesion

To measure *cohesion*, I simply count, for every agent, how many other agents are within the agent's sensor range for any given time step. This value is then averaged over all agents, and over all time steps, and the result is the *local agent density*, or simply density. This value, different from the global alignment, is only taken locally, and reflects how well agents keep other agents within their own sensor range.

### 6.5.3 Separation

The hardest value to measure is *separation*, since it basically quantifies an objective of what should not happen. To approximate this, we measure how often one agent tries to enter the cell of another agent, and thus is colliding with it. In this case, the agent trying to move will simply fail doing so. The resulting number of overall collisions is then divided by the number of time steps, providing an average amount of collisions per round, or simply *collisions*. This number is of course also dependent on the number of agents in the simulation, but this dependence is not linear. Therefore I did not normalise with respect to agent number. Thus, one needs to take care to only compare values where similar amounts of agents have been involved. Again, agents that have found the food are not considered for collision detection.

### 6.5.4 Results

All measurements were taken in a open ended simulation where the food had a 3 % chance of being moved every time step. When this happens, all agents' internal models are reset, and those agents who have already found the food earlier are put back into the simulation. The simulations were run for 100,000 time steps, with 20 agents, in a $20 \times 20$ torus-shaped grid world, with a sensor range of two. As a baseline for comparison, we also measured those values for a group of agents that chose their actions at random, only stopping if they chanced upon the food source. The other two behaviour modes considered here are non-social infotaxis, and infotaxis where all agents have the ability to use a Social Bayesian Update. The last is called *Social Bayesian* for comparison.

| | Alignment | Density | Collisions |
|---|---|---|---|
| Random | 0.23 | 1.03 | 0.72 |
| Non-Social Infotaxis | 0.29 | 1.33 | 1.31 |
| Social B. Update | 0.39 | 1.68 | 0.49 |

Table 6.1: Flocking indication measurements taken for three behaviour models. (Random, Infotaxis, Social Bayesian

Comparing the random behaviour to the non-social infotaxis search, we notice that both the local agent density and the number of collisions are larger for the infotaxis model. The agents are not reacting to each other in the non-social infotaxis model, so this is a result of the improved search algorithm alone. If we measure how long it takes, on average, for a *random* agent to find the food (ca. 450 time steps), and compare it to the time it takes an *infotaxis* agent to find the food (ca. 70 time steps), we see that the infotaxis search has a much better performance, resulting in agents actually finding the food before it changes position. This causes a local concentration of agents, as more agents get to the area around the food location faster. This, in turn, is likely to result in increased density and collisions.

Also note regarding the alignment indicator, that even for a group of agents which move at random the average alignment is not 0.0, but 0.23. This is a statistical effect and not surprising, since it would actually take coordination to ensure that all agents' movements are always balanced between the different directions.

The interesting comparison is now between the two simpler models and the Social Bayesian Update. In the latter, we see a further increase in alignment, indicating that a high number of agents now move in similar directions during most of the simulation. Keep in mind that to achieve an average of 1.0, all agents would have to move in that same direction, in every turn. We also get a further increase in local agent density, while at the same time the number of collisions is reduced. So while there are even more agents within the sensor range of each other, the agents manage to collide much less.

Furthermore, if we take a look at a graphical representation of the agent's behaviour (two sample images can be seen in Fig. 6.2) we can see that small groups of agents are forming when agent's happen to encounter each other, and those groups then start to move together. The "tails" in Fig. 6.2) indicate the last few movements of an agent, and we can see even in the still image, that those agents that are closely group together also have well aligned movement vectors for the last few moves.

While relying solely on visual results is problematic when identifying swarm behaviour (as discussed in (Sayama 2011)), together with the quantitative measurements this gives

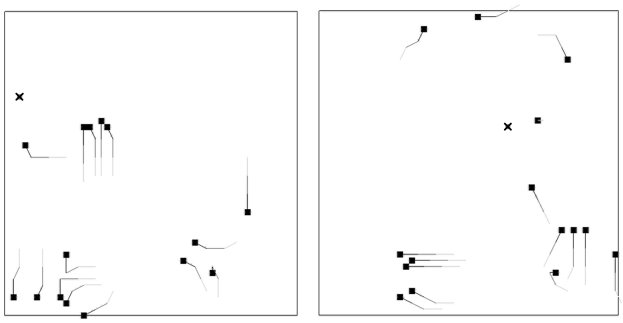


Figure 6.2: Two visualizations from a social infotaxis simulation with 15 agents, sensor range 5 in a 50 x 50 world. The crossed out box is the food source, the black boxes are agents. The "tails" attached to the agents visualized the movement of the agent for the last 9 time steps. Each tail consists of 3 line segment, each representing the vector of past agent movement for 3 time steps.

further evidence that the agents behaviour now exhibits some form of coordination resembling swarm behaviour.

## 6.6 Interpretation

I presented a model in which the agents' behaviour is motivated by one single principle or goal, namely to gain as much information about a relevant variable in the environment. To achieve this, the agents take any kind of sensor variable, be it an environmental variable, such as the state of a grid world cell, or the action variables of another agent, and perform a naive Bayesian update on its internal probabilistic model regarding said relevancy variable. The agent's own actions are chosen in regard to which of them provides the greatest expected reduction of entropy, based on the agents' own internal model.

In this section, I would now like to discuss possible explanations on how this information maximisation model may lead to the three different rules which create the boids-like flocking behaviour.

### 6.6.1 Alignment

The alignment behaviour seems to result mainly from the agent's estimation of where the food source is. Looking for an actual location, be it food or some other relevant place, would be necessary to generate this part of the behaviour. At the very minimum, the agents would have to believe that there is a relevant location out there and look for it.

What happens in more detail is this: When an agent is controlled by non-social infotaxis behaviour moves, then its action contains information about the relative position of the food source. If we take a look at an agent moving north (due to rotational symmetry, the actual direction is exchangeable), then the food is more likely to be in a position north of the agent, and less likely to be in a position south of it. This effect, even though the agent does not know where the food is, results from the fact that the agent knows where the food is *not*. As seen in Fig. 4.2, the probability distribution has its highest peak directly north of the agent, and the minimum of the distribution is in the area south of the agent. Both peaks flatten out the further the cells are away from the agent.

Another agent who observed the first agent move north would perform a Bayesian update on its own assumed probability distribution of the food source. Everything else being equal, this would lead him to "believe" that the food is more likely to be north. The resulting move action would also be to rather move north than in any other direction. A flock of agents, each observing each other, could thereby create a "travelling wave" of high probability immediately outside of their sensor range, driving them all in a similar direction.

The generalised principle here is that an agent 1 observing actions by an agent 2 assumed to have similar goals would lead the original agent 1 to conclude that agent 2 has information that would make such an action reasonable, and in turn, this would make the same action more reasonable for agent 1.

### 6.6.2 Separation

Whenever agent 1 observes an agent 2 moving in the grid world model, it performs a Bayesian update for the position of the food source. The largest impact of this update is on the probabilities of the area immediately around agent 2. The cells of the world agent 2 observed in its previous turn are definitely empty; the Social Bayesian Update would therefore assign a probability of zero to every cell that the agent could have seen in its last turn. As the agent has only moved the distance of one cell its current location and all cells around its current location that are one less than its sensor range away still have a probability of zero. Once a probability is zero, there is no event that would cause the Bayesian Update to assign a non-zero probability to that cell. Therefore, observing any cell with zero probability will not yield any change in the internal probability distribution, and will therefore result in zero information gain.

Since agent 2 is in an area surrounded by cells which agent 1 assumes to have a zero probability, it would be bad for agent 1's information gain to observe the cells around agent 2. The area around agent 2 has become informationally "dead" because of observing

agent 2.

While observing another agent is an efficient way to gain information, the immediate environment around that agent becomes informationally unrewarding afterwards. Everything else being equal, an information-driven search would therefore try to steer away from the immediate area around an observed agent.

In general, if an agent 2 in a specific position reveals information it gets from being in that position to agent 1, then the more information agent 1 gets from that agent, the less informationally interesting does being in the same position as agent 2 become.

### 6.6.3 Cohesion

In the current model, most of the cohesion seen in our agent groups seems to be a direct result of the high amount of agent alignment. If agents that meet each other move into a similar direction, with similar speed, then they also happen to stay together.

While it was possible to generate a higher level of cohesion than random in the studied model, it is unclear if such an effect would also hold in a more general model. If more movement direction where to be included, or if the agents could use variable speed, this part the flocking behaviour might not be generated.

One way to counter this, would be to further modify the infotaxis formalism. In general, it would be reasonable to include a further term into the infotaxis formalism which would account for the amount of information gained from other agents. Following from the "digested information" principle, it is informationally advantageous to keep other agents in sensor range, to be able to use them for a Social Bayesian Update. Seeing another agent, and being able to use the information in its actions increases each agent's expected entropy reduction. This information reduction could either be estimated from past experience, or combined with an expected action formulation even explicitly computed.

The agent would then need to maintain an additional probability distribution, which would model the expected number of agents in each cell of the environment. This model would be updated when another agent is actually encountered. Upon leaving the sensor range the other agent would then be modelled with some diffusion kernel, assuming that it would move at random. Similarly, it would be possible to use a more advanced behavioural model. Based on this the agent could calculate how many agents it could expect to see if it moved in a certain direction, and adjust its action selection accordingly.

I would speculate that this additional behavioural term would lead the agent to be attracted to each other. Furthermore, I would also speculate that this effect would be more robust in regard to different models of the world.

Based on the previous argument, I would argue that, from a perspective of maximising the relevant information intake, the best position to be in relation to another agent would be as far apart, but just in sensor range. This way each agent could gain the digested information from the other agent, but at the same time observe the most area that is not informationally "dead". Maintaining a specific distance, just within sensor range, would then be the resulting macro-behaviour.

## 6.7 Future Work

Since all agents observe each other I would suspect there is the distinct possibility that a positive feedback loop can emerge, which detaches itself completely from the environmental information. As an example, an agent might take, for lack of better information, a random action; for example to move up north. Another agent might observe the first, and if it did not know anything apart from the fact that another agent moved north, he also would move north. The first agent in turn might now see the second, observe that the other agent moved north, and take this as good reason to also move north. This vicious feedback circle then continues, reaffirming both agents internal beliefs that "they are doing the reasonable thing". This phenomenon warrants further study, since it could illuminate how in social settings seemingly reasonable assumptions lead to strong "convictions" that are utterly wrong and detached from reality.

Furthermore, it might also be interesting to move the present model from a grid world scenario into a continuous world. This would not only create more realistic animations, but would also be necessary to establish that the observed effects are not just artefacts of the grid world model. The challenge here would be the extension of previously described information theoretic tools to the continuous domain.

## 6.8 Chapter Conclusion

This chapter offered a quantitative analysis of the flocking behaviour resulting from the maximisation of relevant information intake, as described in chapter 4 and 5. The measured qualities indicate that a group of infotaxis agent with Social Bayesian Update has increased alignment and cohesion and collision avoidance, when compared to random or non-social agents. Since these factors by itself are able to generate flocking behaviour (as demonstrated by the boids rules), it seems reasonable to call this behaviour flocking.

Furthermore, I also outline in argument, how maximization of relevant information can conceptually be used to generate all three basic boids behaviours. Further work would have

to be done to evaluate if this can be realized across a variety of models, or in the continuous domain. But in theory, there is a basis for flocking based on information maximisation, and this chapter can serve as a proof of concept, that flocking can be generated in this way.

As outlined in the introduction, this result is particularly interesting in regard to the first research question, which puts this into an artificial life context. For the specific example of this model I demonstrated that flocking-like behaviour can indeed arise from information theoretic principles. If we were to fully accept the assumption that organism are indeed actively trying to maximise their information intake, and also have adapted in a way to realized the previously outlined abilities, then this indicates that it would be conceivable that these organisms also develop some kind of flocking behaviour under certain circumstances, or at least be inclined towards flocking.

# Chapter 7

# Conclusion

This chapter gives an overview of the larger argument running through the chapters, and reiterates the important conclusion related to it. It also uses the insights gained from the previous chapters to answer the main questions motivating this thesis.

## 7.1 Thesis Summary

Chapter 2 outlined the information theoretic agent-world model used in this thesis, and discussed its main properties. Information was introduced as a non-semantic quantity, independent of a specific agent's perspective. Then causal Bayesian Networks were used to define an agent model that does neither require semantic grounding for its symbols, nor presupposes basic social abilities, such as the identification of other agents. The model reflects the idea of situated and embodied cognition, where the agent has to figure out the world from the agent's perspective through interaction with the world defined by the agent's embodiment.

Chapter 3 revisited Polani's concept of relevant information. Information is relevant when it is necessary for an agent to perform better. This makes it possible to quantify how much relevant information an agent needs to process in order to perform on a specific performance level. In this context I then returned to the central assumptions of relevant information maximisation and information parsimony. In a world where relevant information increases for higher performance levels the maximisation of relevant information can be easily motivated. An agent that wants to perform better needs to acquire more relevant information. Keep in mind, though, that due to the definition of relevant information as the minimal mutual information over all strategies with a certain performance level, there is a clear upper limit of how much relevant information an agent can obtain. An agent

can, of course, always obtain more information, but it is not necessary to do so. At this point we return to the idea of information parsimony, which was itself not studied in this thesis. The general argument here is that information processing has a cost, so processing more information than necessary is a waste, which should be avoided. As a result, we would assume that adapting agents would try minimize the amount of mutual information between their inputs and outputs, ideally down to the level of the relevant information. In summary, this motivated how the maximisation of relevant information and information parsimony would likely cause the agents strategy to end up on the trade-off curve between performance and necessary mutual information.

Furthermore, I argued that if the agent is actively processing information, it is likely that the agent is situated in a world where the trade of between performance and mutual information is "non-trivial", i.e. the agent actually has to process *more* information to perform better. Those two insights already suggested that an agent would be intrinsically motivated to increase the mutual information between its own actions and the environment.

I also introduced the concept of unique sensor information in chapter 3, to demonstrate how an agent could, within the same adaptation process used previously, determine how much relevant information is located in a specific part of the sensor input. This would allow the agent to determine that some sensor inputs are more valuable than others.

Chapter 4 introduced the *Digested Information* argument, where I argue that an agent is likely to encode not just any piece of information about the environment, but specifically those bits of information that are relevant to its own behaviour, the same information that is also relevant to the behaviour of other agents with similar agendas. I discussed two simulation models to demonstrate the existence of relevant information in an agent's action, and also discussed some related properties of digested information. Namely, that an agent is likely to

1. encode more relevant information when its performance increases,
2. encode more information than the environment,
3. and transport relevant information from other locations and points in time to the here and now.

Following from the arguments and simulations in chapters 3 and 4 I then concluded, that

**from Chap. 4** An agent is encoding information relevant for its own behaviour in its actions. In the specific simulations I looked at this information was also relevant for other agent's with similar goals.

**from Chapt. 3** Assuming that the agent manages to realize a strategy on the relevant information trade-off curve (either through adaptation of learning) it can determine which part of its sensor input contains how much relevant information.

Combining these two insights it seem plausible for an agent, especially one motivated by maximising its relevant information, to realize that there is a certain amount of relevant information located in the part of the environment that is the other agent's action. This should motivate the agent to specifically pay attention to the actions of other agents similar to itself. Furthermore, the simulations in chapter 4 also indicate that relevant information is likely to be present in higher density in the actions of others. Combining this with the principle of information parsimony would make the information in other agent's actions even more attractive, as less overall information would have to be processed to gain a given amount of relevant information.

Chapter 5 then demonstrates several possible effects when an agent actually incorporates the information of others via Bayesian Update. The three main conclusion from that chapter are:

1. Using other agents' digested information can actually increase an agent's performance; this can exceed the performance level theoretically attainable for a single agent.

2. Not all information gained from other agents is necessarily useful for the observer; processing other agent's information can be detrimental to agent performance because:

   (a) In populations where only one agent is social too much information can destroy the information gradient used in infotaxis search.

   (b) Location based selection of observations can lead to a conditional dependency between other agent's actions, which violates the central assumption of the Naive Bayesian Update.

   (c) In an all social population information cascades can propagate misleading information through the population, which then overrides the correct information gained from the environment.

3. Processing another agent's action information becomes another factor that changes the digested information that an agent provides to others.

The concrete examples in this thesis, specifically the treasure hunter model, demonstrated that using the information of other agents changes the information an agent provides. An interesting phenomenon here is the possible existence of game theoretic equilibria. An agent has to balance its own performance against the information it provides. The agent itself is just interested in optimizing its own performance. But if this means that the agent would not provide any information to others, then the same reasoning could be applied to other agents, and as a result there would be no information available to anyone, and no one would profit from a Social Bayesian Update. But in the specific case of the treasure hunter model with partial observability there was a specific behaviour (the one with ca. 30% observability) that was both optimal in terms of performance and providing information. Moving away from it unilaterally would not be possible for any agent without incurring a loss of performance. Assuming that the chance of observation is a parameter controlled by an adaptive process this would allow the overall population to stabilize in this equilibrium state, where all agents would use a specific level of observation to perform Social Bayesian Updates.

Chapter 6 demonstrated, using the same basic formalism of previous simulations, that information maximisation can lead to flocking behaviour. This indicates the possibility that the classical boids rules for flocking need not necessarily be assumed primitive, and that more fundamental information theoretic principles could be used generate similar flocking behaviour. This further demonstrates how a information theoretic approach can lead to some form of agent-agent interaction; all that is needed is some localized relevant information, and the ability to integrate the digested information of other agents.

## 7.2 Research Questions Revisited

With an overview of the whole thesis in mind we can now return to my initial research questions. The first one was:

> Does the optimization of information processing lead to agent-agent interaction?

This question has been refined throughout the thesis, both in scope and meaning. First, I needed to clarify what exactly I meant by *optimization of information processing*? I decided to use the principles of relevant information maximisation and information parsimony as assumptions on how an organism might optimize its information processing to

see where this would lead. Other options, such as the maximisation of channel capacity between an agent's actions and sensors (empowerment) or adaptation towards better prediction of an agent's future state where left unstudied.

I also limited the scope of interaction that I looked at, focussing on the most basic agent-agent interactions, cutting out more complex social interactions. This lead me to another question, namely, is there something special, in information theoretic terms, about the information in another agent's actions? I believe that this is a necessary first step towards an information-theory guided sensor evolution that can account for attention towards other agents. The early chapters then were focussed on demonstrating, with a formal mathematical basis, that the information in another agent's actions does indeed have some special properties.

Assume that agents with similar goals need to acquire the same information. The relevant information formalism together with the digested information argument show that this necessary information inadvertently needs to be displayed in their actions. While there are a lot of model and strategy-dependent differences the bottom line is that if an agent wants to react appropriately to the information in the environment, then it has to display at least the relevant information in its actions. Whether another agent is able to use this relevant information form its own perspective is another matter, but both the argument, and the supporting simulations show that the information is there. This results from the agent's drive to improve its own performance, and does not necessitate a desire of the agent to communicate, or a joint pay-off matrix that rewards cooperation or coordination.

For a specific agent that tries to make sense of an unstructured environment this means that other agents are processes in the environment that are intrinsically motivated a.) to extract the information they need and b.) to provide this information in their actions. Arguably, they are the only part of the agent's sensor input or environment with that motivation. This already provides a reason for an agent to adapt in a way that pays special attention to other agent's actions, as they are likely to provide the relevant information an agent with similar goals would need.

With this as a basis, we can then return to the larger question, and ask how this could lead to actual interaction between the agents. Relying on information theoretic measures, the special properties of digested information can be quantified from the agent's perspective, using methods such as the unique relevant information formalism. Therefore, an agent that is using information maximisation as a guiding principle for sensor adaptation would indeed favour a sensor adaptation that pays special attention to other agents.

This, together with the actual processing of information via Bayesian Update, demonstrates that information theoretic principles alone can already lead to a rudimentary form of social interaction, meaning that now ones agent's actions would causally depend on another agent's actions. Keep in mind that this did not require any joint pay-off matrices, or the ability for agents to directly influence each other.

In the later chapters the simulations exhibited several collective behaviours which are also present in real, biological systems, such as flocking and information cascades. So, in conclusion, it seems that within the scope and assumptions chosen in this thesis the answer to the first question is positive.

Tying this back into the original, larger motivation for the question from the perspective of artificial life connects this back to our understanding of nature. Initially, I presented the hypothesis that adaptation in nature can be understood in terms of optimizing certain information theoretic principles. Especially in the area of sensor adaptation and basic cognition this has lead to interesting and life-like behaviour (Klyubin et al. 2005b, Klyubin et al. 2007, Der et al. 1999, Ay et al. 2008, Sporns and Lungarella 2006, Prokopenko et al. 2006) . This work is part of an effort to further extend this information theory based behaviour generation to also include the interaction with other agents. By reproducing phenomena observed in nature, such as flocking or information cascades, I am aiming to bridge the gap from basic cognition to higher social abilities. Importantly, the fact that those different phenomena are generated with similar informational principles leads further support to the original hypothesis, namely that natural agent are guided by informational principles in their adaptation process. Specifically in this dissertation, one of the main insights was the observation that agents are inclined to provide relevant information to other agents with similar goals. This leads to the possibility to differentiate agents with information theoretic measurements from the environment, and further creates a gradient for the development of attention, and the ability to integrate the information provided by others. In regard to our understanding of nature, these insight might also create a change of perspective. Understanding life in terms of information processing is not only about organisms that process information to improve themselves, but the environment is also filled with other organisms providing information they already processed to others.

The second question looks at this thesis from a different perspective:

> What insights can the analytical framework of information theory provide into agent-agent interaction?

As a scientist studying nature it might be interesting to use the tools discussed in this thesis to study the phenomena I tried to generate. As I did not actually apply any of

the tools discussed within here to empirical data, the answer to this question remains, for now, also theoretical.

One advantage of the information theoretic approach is that measures, such as entropy, mutual information, etc., are extremely versatile as they can, in principle, be applied to virtually anything that can be expressed as random variables. In this thesis I focussed mainly on how to apply these measure to the simulated interactions of agent-agent interaction.

The measure of unique relevant information, specifically within an agent's sensor input, can give insights into where relevant information is located in the environment. While this is not directly linked to agent-agent interaction it allows us to quantify how information is contained in the actions of a specific agent. This might, as demonstrated, help to differentiate this information from other, less interesting, environmental variables. Furthermore, it might also allow us to differentiate different agent's by how much information they provide. Both insight might lead to an agent paying attention to other agents in general, and well performing agents in particular.

Furthermore, information theory allowed us to decompose the digested information in an agent's actions into stigmergy and action information. This allows us to quantify how much information is in the actual action selection of an agent, and how much information is "around" the agent, because the agent's actions have injected it back into the environment. This allows us to better understand where the information is located, and what is needed to facilitate good information transfer from one agent to another. By understanding what the current medium for information transfer is, we can better understand how sensor have to adapt to capture this information.

One illustrative example here is the information contained in the actions and positions of the random agent. Counter to my intuition the random agent still displayed some information in its actions. Even though its decision where to move was random, and therefore independent of its sensor input, the decision to move at all was not. The agent would stop if it found the food. This led to a considerable amount of information encoded in the agent's position, which was measurable with the methods discussed in this thesis.

The decomposition of the partial information into the different forms of sensor input (social and environmental) in the fishworld model also helped to explain why more agents in the environment where bad for the social Bayesian update. The different possible causes, such as systematic dependencies of lack of informational gradient, could be differentiated by their different profiles for the partial information properties. While not done in this thesis, I also believe that this analysis could be further extended, and be performed in more detail by relying on specific measures for information flow, which where not used in

this thesis.

In the simpler treasure hunter model, the comparison of the different strategies to the actual relevant information graph illustrated easily which strategies were informationally efficient and which were not. This is helpful to understand how exactly other agents provide information, especially once the information they provide changes due to their own information intake.

In summary, there where several examples detailed throughout the thesis on how specific insight about a system with several agents can be reached with the help of information theoretic tools. So if the question was just aimed at the analysis in models, then the answer is also positive, and well demonstrated through this thesis. In a more general context, namely the real of nature, it remains to be seen if the tools developed in this thesis will be of use.

## 7.3 Discussion and Future Work

In this section I like to discuss some general issues arising from this thesis, and outline questions that could be addressed with further work. I will also speculate in regard to what might be likely phenomena to arise from continuing research in this direction.

### 7.3.1 Deceit

The question that was raised most often in relation to this work is about deceit. I make the claim that agents *have* to encode a certain amount of information into their actions. But what if there is, different from the presented model, a shortage in resources? What if the other agent's actions do matter for an agent, and suddenly an agent is motivated to hide its information? I have argued that, in our model, the minimal mutual information (the relevant information) has to be displayed in the agent's actions. Using any less mutual information between the environment and the agent's action would results in a lower performance level. So, in our model, reducing the information is only possible if the agent is willing to reduce its own performance.

Before we look at more general models I like to point out a possible confusion regarding what we are talking about. The information I mean is the perspective invariant mutual information which is calculated from an omniscient perspective; the information that is there regardless of any specific observer. This should not be confused with the information that another agent can obtain from one agent's actions. The common ideas of deceit rely on using the difference between what is actually happening, and what another agent can

either perceive or infer from its observation.

A classic example is the kind of deceit where an agent would first determine if it is observed by another agent it is competing with. If not, it would then perform whatever strategy is best, regardless of the information displayed. If it is observed, the agent would act in a misleading way or not at all.

Furthermore, an agent could also utilize the other agent's inability to model its perception-action mapping correctly, in order to systematically mislead it. So, the agent could act as if a specific thing was the case (while it is not), and thereby lead the other agent to perform suboptimal.

In both cases the agent's actions would contain more information than the other agent would perceive. As there seems to be a different between the actual information in an agent's action and the obtainable information it would be nice to clarify this further, possibly finding a way to measure how much relevant information can be obtained from another agent's actions given a specific model of the other agent perception-action mapping.

I believe that two things would be most helpful here to address this question in further detail. First, I would be good to better understand how an agent would adapt to obtain and utilize this information. Secondly, making agents compete for resources, or more general, the inclusion of joint pay-off matrices leads to a dynamic environment, in which other agent's action could now directly affect an agent's performance. To deal with this analytically we would need to extend the basic notions of game theory by incorporating information theory.

### 7.3.2 Adaptation of the Bayesian Update

The thesis relies heavily on the idea that an agent would adapt to obtain and utilize the information displayed by other agents. I modelled this by equipping agents with the ability to perform Bayesian Updates for variables in their environment. Even if I was just to talk about an ability that is functional equivalent with a Bayesian Update, it is still questionable how and if such an ability would develop. Showing such an adaptation in a simulation model would be a good step to further support the idea that something functionally equivalent to a Bayesian Update could arise. Ideal would be to demonstrate how this could lead to a generic Bayesian Update ability; so an agent is not just able to perform something "like" a Bayesian Update for a specific context, but could demonstrate and apply this ability to new contexts.

Approaching this problem from a different direction would be to study actual biological agents to determine what methods they are using to incorporate information from other agents. Bayesian Update was chosen for my model because it is optimal in the sense that

it gives the best estimate of the world given the available information. But it would be possible to consider other models of information integration, social learning and decision making and apply similar information-theoretic analysis to them.

Even when sticking with the Bayesian Update, another aspect should be addressed in the future. A proper Bayesian Update does not only require the ability to utilize the Bayesian Theorem, but also requires the agent to somehow obtain a conditional probability distribution to perform the update with. This could be part of the evolutionary adaptation (for a functional equivalent of a Bayesian Update), but greater flexibility would be gained form being able to "learn" this conditional probability distribution during an agent's lifetime. In our model we considered this distribution fixed, but if an agent could update its distribution then information gain should also incorporate possible changes to the model. This area has not been touched upon in this thesis, but it would be relevant in order to understand the development towards social information integration.

### 7.3.3 Game Theory and Information Theory

In this thesis I deliberately assumed a model where an agent's action has no direct influence on the performance of other agents. This was done so I would not have to deal with the recursive complexity that arises from adapting your behaviour in regard to a likewise adapting environment.

The Nash equilibrium I did describe in chapter 5 demonstrated that even in this case, there is still a possibility to influence other agents in a way that changes one's own environment. By passing on "bad" information, the same information could be passed back to an agent and influence it towards "bad" behaviour. In our specific case there was an evolutionary stable strategy for processing a specific amount only, so that no single agent could unilaterally change its processing without losing performance. An interesting question would be to ask under what circumstances such equilibria do arise? One speculation I would offer here is that the model we observed was basically cooperative in nature. Agents did not gain anything by other agent's performing better or worse, but they could gain more information from agents that would perform better. So it was in the general interest of all agents that all agents performed well.

A more general approach would be take a look at what happens if agents act in a competitive or zero-sum scenario. We could cut out dedicated communication and assume that the only way agents pass on information is through their actions. So every time an agent acts it has to consider both the effect of the action on the world, and what information it transmits with this action. Arguably, one could even say that these two aspects are now the same, as transmitting information into the world is just another way

of affecting it.

In classical game theory, where all information is know to everyone, an agent has to determine how to act optimally by taking into account that other agents will also act optimally while again including how other agents will act into their decision making. In specific cases, for example the complete-information, sequential, zero-sum game, this results in the convergence on a specific set of strategies which are optimal in the sense that it is not possible to act better, given that the other agents are competent.

In this extended model we could now assume that agents have a.) limited information about the world b.) a non-perfect model of how other agents map sensor inputs to actions. This then requires a decision making agent to not only calculate how its actions would affect the world and the other agent's decisions, but an agent would then also have to take into account how its action would change another agent's world information and the other agent's model about itself. A complete solution in the game theoretic sense then would require the agent $A$ to not only have a (probabilistic) model of agent $B$, but also be able to model how agent $B$ would model agent $A$, etc. This would have to be a recursive probabilistic model of models up to the point where the interaction ends.

A nice example for a scenario of this kind would be the game of poker, or even better, online poker. Betting is the only action a player can take, and in the online version also the only way how to communicate with other players. Each player only knows its own cards, and relies on an imperfect model of the other players behaviour to determine what cards the other player has based on the other player's actions. Both the player's assumption about the world and how others act change over time.

This proves to be quite complicated, and as far as I am aware there is no general solution for how to act in this model. So, while it would be interesting to extend the model in this thesis towards a more competitive model, it is unclear how an agent would determine how to actively deceive others if information about the world is limited. On the other hand, one could approach this scenario from a brute force perspective, and just create a scenario where deception could be useful, and enable the agents to adapt their strategy. It would then be interesting to see what kind of deceptions arise, and how they would be reflected in the information theoretic properties.

### 7.3.4 Detachment of Social Information Update

Another phenomenon that would warrant further study is the detachment of "believe" regarding some state of the environment from actual environmental evidence. An information cascade in the treasure hunter scenario demonstrated that repeated social updates can transfer a "common" shared believe that the treasure is in a specific location, even

though it is not. This, by itself, is a known phenomenon, and has been conceptually linked to the spread of religion, fashion fads and mass hysteria.

I would speculate that a similar phenomenon arose in the fish world scenario, once noise was introduced into the agents internal models. Then agents could perpetually move into one direction, reinforcing each others believes that whatever they are looking for would be just out of their sensor range. This would be interesting, as it not a convergence on a specific false assumption, such as the food is at coordinate x and y, but a convergence on an ever changing assumption, i.e. the food is 5 spaces to the east.

This also raises the general question if the information transfer realised by Bayesian Modelling of the world and Social Bayesian Updates could give rise to a systems that allows for replication, modification and adaptation of information patters, common to what is sometimes described as "memes". The general idea of memes is that ideas or cultural units are subject to a similar evolutionary process as biological organisms.

The idea of memes is mostly associated with Richard Dawkins (Dawkins 1990), who introduced an early concept of them and possibly coined the term "meme". He presented them as a non biological analogy to the biological replicators, the genes. Both replicators are, given the right environment, able to create copies of themselves, despite there being no "intention" present on their part. He also introduced the idea that the fitness of those replicators is mainly determined by three properties:

- copying fidelity: how similar, or errorless the new copies are
- fecundity: how often the replications create new copies
- longevity: for how long a particular replicator is able to make copies

These properties work well, for both memes and genes, but while the replicators of the genetic evolution are well identified, it remains unclear what the replicators in question for the memetic evolution are, how their self-replication process is realized and how the properties of a specific meme are determined by the underlying dynamics.

Another major contributor is Aaron Lynch (Lynch 1999) who introduced similar concepts under the name of "Thought contagion". He modelled the spread of memes with a model borrowed from epidemiology and defined the meme's contagiousness as:

$$F(m) = A(m) \cdot R(m) \cdot E(m) \cdot T(m)$$

- $A(m)$: proportion of individuals assimilated on encounter
- $R(m)$: proportion of individuals that retain m in their memory

- $E(m)$: number of expressions of m by a host for a given interval
- $T(m)$: number of potential new hosts

This model still has problems identifying the replicators, but is able to use actual numerical values once it is possible to determine whether a host is infected with a meme or not. His work also describes a wide array of social phenomena, from the spread of religion, to sexual moral, to political views, in terms of thought contagion and therefore offers a good repository of phenomena worth explaining.

While the meme analogy might be attractive regarding our intuition, there is the question what the model of memes adds as a scientific theory (Edmonds 2002). Specifically, Edmonds outlined three challenges that memetics needs to address. I speculate that information optimization might be a possible route to address the third challenge, to produce "a simulation model showing the true emergence of a memetic process", but there are still a lot of problems that would need addressing. First, to make the model credible, the previously discussed assumption that organisms use an ability similar to a Bayesian Update would have to be connected to nature. If this was possible, then it might be possible to use the physical expression of behaviour or action selection as a testable medium to track the transfer of memes. Here information theory could be used to construct a metric that does not rely on a semantic interpretation of the actions to ascertain the closeness of different action or information patterns. In a model it would also be possible to compare the internal models and track similar similarities, but this would be hard to verify in connection to biological phenomena later.

The advantage of an information optimization model that incorporates some form of Bayesian update would be that it does not include an explicit replication mechanism. The purpose of the original Bayesian Update could be just to understand states of the environment and act accordingly. Information would in this case only flow from the environment to an agent once. Only the addition of other similar agent would then create the "information flow" from agent to agent, creating a new environment for information, in which it would become detached from the original source and subject to slow changes due to noise. Agents which perform badly might be less likely able to pass on the pattern, which would then introduce a mechanism for selection.

# Bibliography


Acemoglu, D., Dahleh, M., Lobel, I. and Ozdaglar, A.: 2011, Bayesian learning in social networks, *The Review of Economic Studies* **78**(4), 1201–1236.

Adami, C.: 1998, *Introduction to Artificial Life*, Telos, Springer Verlag.

Allee, W.: 1931, *Animal Aggregations: A Study in General Sociology*, The University of Chicago Press.

Almeida e Costa, F. and Rocha, L.: 2005, Introduction to the special issue: Embodied and situated cognition, *Artificial Life* **11**(1-2), 5–12.

Ashby, W.: 1956, *An Introduction to Cybernetics*, Taylor & Francis.

Ashlock, D., Kim, E. and Leahy, N.: 2006, Understanding representational sensitivity in the iterated prisoner's dilemma with fingerprints, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **36**(4), 464–475.

Attneave, F.: 1954, Some informational aspects of visual perception, *Psychological Review* **61**(3), 183.

Axelrod, R.: 1997, *The Complexity of Cooperation: Agent-based Models of Competition and Collaboration*, Princeton Univ Press.

Axelrod, R. and Hamilton, W.: 1981, The evolution of cooperation, *Science* **211**(4489), 1390–1396.

Ay, N. and Polani, D.: 2008, Information flows in causal networks, *Advances in Complex Systems* **11**(1), 17–41.

Ay, N., Bertschinger, N., Der, R., Güttler, F. and Olbrich, E.: 2008, Predictive information and explorative behavior of autonomous robots, *European Journal of Physics B* **63**(3), 329–339.

Banerjee, A.: 1992, A simple model of herd behavior, *The Quarterly Journal of Economics* **107**(3), 797–817.

Barlow, H.: 1959, Possible principles underlying the transformations of sensory messages, *Sensory Communication: Contributions to the Symposium on Principles of Sensor Communication* pp. 217–234.

Barlow, H.: 2001, Redundancy reduction revisited, *Network: Computation in Neural Systems* **12**(3), 241–253.

Baude, M., Dajoz, I. and Danchin, É.: 2008, Inadvertent social information in foraging bumblebees: Effects of flower distribution and implications for pollination, *Animal Behaviour* **76**(6), 1863–1873.

Bialek, W., Nemenman, I. and Tishby, N.: 2001, Predictability, complexity, and learning, *Neural Computation* **13**(11), 2409–2463.

Bikhchandani, S., Hirshleifer, D. and Welch, I.: 1992, A theory of fads, fashion, custom, and cultural change as informational cascades, *Journal of Political Economy* pp. 992–1026.

Bongard, J., Zykov, V. and Lipson, H.: 2006, Resilient machines through continuous self-modeling, *Science* **314**(5802), 1118–1121.

Call, J. and Carpenter, M.: 2002, Three sources of information in social learning, *Imitation in Animals and Artifacts* pp. 211–228.

Camazine, S.: 2003, *Self-Organization in Biological Systems*, Princeton studies in complexity, Princeton University Press.

Capdepuy, P.: 2010, *Informational Principles of Perception-Action Loops and Collective Behaviours*, PhD thesis, University of Hertfordshire.

Capdepuy, P., Polani, D. and Nehaniv, C.: 2007a, Constructing the basic Umwelt of artificial agents: An information-theoretic approach, *Advances in Artificial Life* pp. 375–383.

Capdepuy, P., Polani, D. and Nehaniv, C.: 2011, Perception–action loops of multiple agents: Informational aspects and the impact of coordination, *Theory in Biosciences* pp. 1–11.

Capdepuy, P., Polani, D. and Nehaniv, C. L.: 2007b, Maximization of potential information flow as a universal utility for collective behaviour, *Proceedings of the First IEEE Symposium on Artificial Life*, pp. 207–213.

Couzin, I. D., Krause, J., Franks, N. R. and Levin, S. A.: 2005, Effective leadership and decision-making in animal groups on the move, *Nature* **433**(7025), 513–516.

Cover, T. M. and Thomas, J. A.: 1991, *Elements of Information Theory*, 99th edn, Wiley-Interscience.

Danchin, E., Giraldeau, L., Valone, T. and Wagner, R.: 2004, Public information: From nosy neighbors to cultural evolution, *Science* **305**(5683), 487–491.

Darwen, P. and Yao, X.: 2002, Co-evolution in iterated prisoner's dilemma with intermediate levels of cooperation: Application to missile defense, *International Journal of Computational Intelligence and Applications* **2**, 83–108.

Darwin, C.: 1859, *On the Origin of Species by Means of Natural Selection*, John Murray.

Dawkins, R.: 1990, *The Selfish Gene*, Oxford University Press.

Der, R., Steinmetz, U. and Pasemann, F.: 1999, Homeokinesis — A new principle to back up evolution with learning, *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA'99), Vienna, 17-19 February 1999.*

DeWeese, M. and Meister, M.: 1999, How to measure the information gained from one symbol, *Network: Computation in Neural Systems* **10**(4), 325–340.

Domingos, P. and Pazzani, M.: 1997, On the optimality of the simple Bayesian classifier under zero-one loss, *Machine Learning* **29**(2), 103–130.

Dretske, F.: 1981, *Knowledge and the Flow of Information*, MIT Press.

Dyer, J., Ioannou, C., Morrell, L., Croft, D., Couzin, I., Waters, D. and Krause, J.: 2008, Consensus decision making in human crowds, *Animal Behaviour* **75**(2), 461–470.

Easley, D. and Kleinberg, J.: 2010, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press.

Edmonds, B.: 2002, Three challenges for the survival of memetics, *Journal of Memetics-Evolutionary Models of Information Transmission* **6**(2), 45–50.

Engelbrecht, A., Peer, E. and Pampara, G.: 2010, Computational intelligenc library.

Floridi, L.: 2011, *The Philosophy of Information*, Oxford University Press.

Gale, D. and Kariv, S.: 2003, Bayesian learning in social networks, *Games and Economic Behavior* **45**(2), 329–346.

Gibson, J.: 1986, *The Ecological Approach to Visual Perception*, Lawrence Erlbaum.

Hand, D. J. and Yu, K.: 2001, Idiot's Bayes—not so stupid after all?, *International Statistical Review* **69**(3), 385–398.

Harder, M., Salge, C. and Polani, D.: 2013, Bivariate measure of redundant information, *Physical Review E* **87**, 012130.

Holland, J.: 1992, Genetic algorithms, *Scientific American* **267**(1), 66–72.

Jeffery, W.: 2005, Adaptive evolution of eye degeneration in the mexican blind cavefish, *Journal of Heredity* **96**(3), 185–196.

Jung, T., Polani, D. and Stone, P.: 2011, Empowerment for continuous agent-environment systems, *Adaptive Behavior* **19**(1), 16–39.

Juul, J.: 2003, The game, the player, the world: Looking for a heart of gameness, *Level Up Digital Games Research Conference Proceedings*, Vol. 3, Utrecht University.

Kennedy, J. and Eberhart, R.: 1995, Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks, 1995*, Vol. 4, pp. 1942 –1948 vol.4.

Klyubin, A., Polani, D. and Nehaniv, C.: 2004, Tracking information flow through the environment: Simple cases of stigmergy, *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems, edited by Pollack, J.*, MIT Press.

Klyubin, A., Polani, D. and Nehaniv, C.: 2005a, All else being equal be empowered, *Advances in Artificial Life* pp. 744–753.

Klyubin, A. S., Polani, D. and Nehaniv, C. L.: 2005b, Empowerment: A universal agent-centric measure of control, *Congress on Evolutionary Computation*, pp. 128–135.

Klyubin, A. S., Polani, D. and Nehaniv, C. L.: 2007, Representations of space and time in the maximization of information flow in the perception-action loop, *Neural Computation* **19**, 2387–2432.

Koster, R.: 2005, *A Theory Of Fun In Game Design*, Paraglyph press.

Laughlin, S. B.: 2001, Energy as a constraint on the coding and processing of sensory information, *Current Opinion in Neurobiology* **11**(4), 475 – 480.

Linsker, R.: 1988, Self-organization in a perceptual network, *Computer* **21**(3), 105–117.

Lissaman, P. B. S. and Shollenberger, C. A.: 1970, Formation flight of birds, *Science* **168**(3934), 1003–1005.

Lizier, J., Prokopenko, M. and Zomaya, A.: 2007, Detecting non-trivial computation in complex dynamics, *Proceedings of the 9th European Conference on Advances in Artificial Life*, Springer-Verlag, pp. 895–904.

Lizier, J., Prokopenko, M. and Zomaya, A.: 2008a, A framework for the local information dynamics of distributed computation in complex systems, *arXiv preprint arXiv:0811.2690*.

Lizier, J., Prokopenko, M., Tanev, I. and Zomaya, A.: 2008b, Emergence of glider-like structures in a modular robotic system, *Proceedings of ALife XI* pp. 366–373.

Lynch, A.: 1999, *Thought Contagion: How Belief Spreads Through Society*, Basic Books.

Macgill, J. and Openshaw, S.: 1998, The use of flocks to drive a geographic analysis machine, *International Conference on GeoComputation*.

McCowan, B., Hanser, S., Doyle, L. et al.: 2004, Quantitative tools for comparing animal communication systems: information theory applied to bottlenose dolphin whistle repertoires, *Animal behaviour* **57**(2), 409–419.

Millikan, R.: 1989, Biosemantics, *The Journal of Philosophy* pp. 281–297.

Nash, J. et al.: 1950, Equilibrium points in n-person games, *Proceedings of the National Academy of Sciences* **36**(1), 48–49.

Olsson, L., Nehaniv, C. and Polani, D.: 2004, Sensory channel grouping and structure from uninterpreted sensor data, *Proceedings of the NASA/DoD Conference on Evolvable Hardware, 2004*, IEEE, pp. 153–160.

Olsson, L., Nehaniv, C. and Polani, D.: 2006, From unknown sensors and actuators to actions grounded in sensorimotor perceptions, *Connection Science* **18**(2), 121–144.

Parejo, D., Danchin, É., Silva, N., White, J., Dreiss, A. and Avilés, J.: 2008, Do great tits rely on inadvertent social information from blue tits? A habitat selection experiment, *Behavioral Ecology and Sociobiology* **62**(10), 1569–1579.

Pearl, J.: 2000, *Causality: Models, Reasoning and Inference*, Cambridge University Press.

Philipona, D., O'Regan, J. and Nadal, J.: 2003, Is there something out there? Inferring space from sensorimotor dependencies, *Neural Computation* **15**(9), 2029–2049.

Polani, D.: 2009, Information: Currency of life?, *HFSP journal* **3**(5), 307–316.

Polani, D., Martinetz, T. and Kim, J. T.: 2001, An information-theoretic approach for the quantification of relevance, *ECAL '01: Proceedings of the 6th European Conference on Advances in Artificial Life*, Springer-Verlag, London, UK, pp. 704–713.

Polani, D., Nehaniv, C. L., Martinetz, T. and Kim, J. T.: 2006, Relevant information in optimized persistence vs. progeny strategies, *Artificial Life X : Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, The MIT Press (Bradford Books), pp. 337–343.

Prokopenko, M.: 2009, Guided self-organization, *HFSP Journal* **3**(5), 287.

Prokopenko, M., Gerasimov, V. and Tanev, I.: 2006, Evolving spatiotemporal coordination in a modular robotic system, *From Animals to Animats 9,* pp. 558–569.

Prokopenko, M., Polani, D. and Chadwick, M.: 2009, Stigmergic gene transfer and emergence of universal coding, *HFSP journal* **3**(5), 317–327.

Rapoport, A. and Chammah, A.: 1965, *Prisoner's Dilemma: A Study in Conflict and Cooperation*, Vol. 165, University of Michigan Press.

Reynolds, C. W.: 1987, Flocks, herds and schools: A distributed behavioral model, *SIGGRAPH Comput. Graph.* **21**(4), 25–34.

Ross, D.: 2011, Game theory, *in* E. N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy.*

Salge, C. and Mahlmann, T.: 2010, Information as a formalised approach to evaluate game mechanics, *2010 IEEE Symposium on Computational Intelligence and Games (CIG)*, IEEE, pp. 281–288.

Salge, C. and Polani, D.: 2009, Information-driven organization of visual receptive fields, *Advances in Complex Systems* **12**(3), 311–326.

Salge, C., Glackin, C. and Polani, D.: 2012, Approximation of empowerment in the continuous domain, *Advances in Complex Systems* **16**(1 & 2), 1250079.

Sayama, H.: 2011, Seeking open-ended evolution in swarm chemistry, *IEEE Symposium on Artificial Life (ALIFE), 2011*, IEEE, pp. 186–193.

Schreiber, T.: 2000, Measuring information transfer, *Physical Review Letters* **85**(2), 461–464.

Shannon, C.: 1951, The redundancy of English, *Cybernetics; Transactions of the 7th Conference, New York: Josiah Macy, Jr. Foundation.*

Shannon, C. E.: 1948, A mathematical theory of communication, *Bell Systems Technical Journal* **27**, 379–423.

Shi, Y. and Eberhart, R.: 1998, A modified particle swarm optimizer, *The 1998 IEEE International Conference on Evolutionary Computation, ICEC'98*, pp. 69 –73.

Spohn, W.: 2000, Bayesian nets are all there is to causal dependence, *Stochastic Dependency and Causality* pp. 157–172.

Sporns, O. and Lungarella, M.: 2006, Evolving coordinated behavior by maximizing information structure, *Artificial Life X : Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, International Society for Artificial Life, The MIT Press (Bradford Books), pp. 323–329.

Stephens, D.: 1993, Learning and behavioral ecology: Incomplete information and environmental predictability, *Insect Learning: Ecological and Evolutionary Perspectives* pp. 195–218.

Surowiecki, J.: 2005, *The Wisdom of Crowds*, Anchor.

Tishby, N., Pereira, F. C. and Bialek, W.: 1999, The information bottleneck method, *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pp. 368–377.

Touchette, H. and Lloyd, S.: 2000, Information-theoretic limits of control, *Physical Review Letters* **84**(chao-dyn/9905039. 6), 1156.

Touchette, H. and Lloyd, S.: 2004, Information-theoretic approach to the study of control systems, *Physica A: Statistical Mechanics and its Applications* **331**(1-2), 140 – 172.

van Dijk, S. G., Polani, D. and Nehaniv, C. L.: 2010, What do you want to do today? Relevant-information bookkeeping in goal-oriented behaviour, *in* H. Fellermann, M. Dörr, M. Hanczyc, L. L. Ladegaard, S. Maurer, D. Merkle, P.-A. Monnard, K. Stø y and S. Rasmussen (eds), *Artificial Life XII: The 12th International Conference on the Synthesis and Simulation of Living Systems*, MIT Press, Odense, Denmark, pp. 176–183.

Varela, F., Thompson, E. and Rosch, E.: 1992, *The Embodied Mind: Cognitive Science and Human Experience*, The MIT Press.

Vergassola, M., Villermaux, E. and Shraiman, B. I.: 2007, 'Infotaxis' as a strategy for searching without gradients, *Nature* **445**(7126), 406–409.

von Neumann, J.: 1928, Zur Theorie der Gesellschaftsspiele, *Mathematische Annalen* **100**(1), 295–320.

Von Neumann, J. and Morgenstern, O.: 1944, *Theory of Games and Economic Behavior*, Princeton University Press.

von Uexküll, J.: 1909, *Umwelt und Innenwelt der Tiere*, Springer.

Wang, X., Miller, J., Lizier, J., Prokopenko, M. and Rossi, L.: 2011, Measuring information storage and transfer in swarms, *In Proceedings of the European Conference on Artificial Life 2011*, pp. 838–845.

Ward, P. and Zahavi, A.: 1973, The importance of certain assemblages of birds as "information-centres" for food-finding, *Ibis* **115**(4), 517–534.

Williams, P. and Beer, R.: 2010, Nonnegative decomposition of multivariate information, *arXiv preprint arXiv:1004.2515*.