Ahsum Nimity: exploring the possibilities of crowdsourcing Bayesian network structure learning through a video game

Steven T. Rekké
Radboud University Nijmegen
Correspondence: steven@rekke.net

June 23, 2012

A thesis submitted in partial fulfillment of the requirements for a degree of Master of Science in Artificial Intelligence.

Academic supervisor:
dr. Iris van Rooij
Department of Artificial Intelligence
Donders Institute for Brain, Cognition, and Behaviour
Radboud University Nijmegen

External supervisor:
Willem Vervuurt, CEO
Rodo - Intelligent Computing

Academic supervisor:
dr. Marina Velikova
Department of Model-Based System Development
Institute for Computing and Information Sciences
Radboud University Nijmegen

Internship project

This thesis is part of the end result of an internship project at Rodo - Intelligent Computing, a creative software studio that aims to develop fun and useful software for large audiences by combining technical skills with an academic background in social sciences and artificial intelligence. Profit maximization is not Rodo's main goal: the company firmly believes that optimal results are achieved when the interests of companies are combined with those of end users and academic institutions. Ahsum Nimity as a product, and the internship that led to the end result are concretizations of this vision.

Acknowledgements

I would like to express my sincere gratitude towards all of my supervisors, dr. Iris van Rooij, dr. Marina Velikova and Willem Vervuurt, for their enthusiasm, guidance and support. Special thanks to Willem Vervuurt for his friendship and the investments that allowed me to perform this research at Rodo with great joy and achieve the personal growth that came with it. I would also like to thank my parents and my sister; without their support and care I would never have come to this point in the first place. Finally, I would like to thank Femke Hesselink for pulling me through the process of writing this thesis. On a more general note, my gratitude goes out to all people that were involved in my education at the Radboud University.

Abstract

Games With A Purpose (GWAPs) are new and promising research tools that apply human-based computation through computer games. Human-based computation is a technique in which part of a computational problem is delegated to humans. Several GWAPs, such as the Foldit game, have shown that in some cases human players can produce good solutions to hard problems. The present research explores the possibility of developing a GWAP for applying human-based computation to such a problem: Bayesian network structure learning. Bayesian networks (BNs) are versatile graphical probabilistic models that are employed in a wide range of fields, both for practical applications and research. They encode knowledge about variables and their (in)dependencies, allowing probabilistic inference and reasoning under uncertainty. Unfortunately, learning the structure of BNs from data is NP-complete. In the present research a first attempt is made at crowdsourcing Bayesian network structure learning through a computer game.

KEYWORDS: Game With A Purpose (GWAP), human-based computation, crowdsourcing, Bayesian network (BN), structure learning

Contents

1 Introduction

As a species, we humans spend an enormous amount of time playing games. At the time of writing, it is said that more than three billion hours per week are spent playing video games (*TED Conversations*, n.d.). While playing these games, the players use their problem solving skills to make progress in the game. As such, all the hours of all the players combined represent a huge problem-solving effort. Apart from providing joy to the players this enormous effort goes unused. Several games such as the "ESP game" (von Ahn, 2007) and "Foldit" (Cooper et al., 2010), however, have shown that it is possible to harness that problem-solving effort. These games are commonly referred to as "games with a purpose" (von Ahn & Dabbish, 2004). Games with a purpose apply a technique called human-based computation, in which part of a computational problem is delegated to one or more humans. Recently, this technique has developed similarities to the technique of crowdsourcing, in which a task is delegated to distributed groups of people. Although they share some similarities, these techniques are not the same and they may both be present in a single game with a purpose. Games have been used successfully to crowdsource simple image- and text-recognition tasks (Law & von Ahn, 2009), and the success of Foldit has shown that complex scientific problems can benefit from crowdsourcing through games (Cooper et al., 2010).

The research presented here, is an effort at developing such a game with a purpose for a hard problem called Bayesian network structure learning, and an exploration of the challenges of game-design in the context of scientific research. Although games with a purpose have shown to be a promising new research tool, they are far from abundant. To our current knowledge, there exist only a handful of projects that are of a similar scientific nature. In the present work, we hypothesize that players of a casual puzzle game can contribute to the construction of Bayesian networks in any domain by inferring conditional dependence relations from joint observations presented in a visually abstract manner. No such approach previously existed. Like Foldit, this research is part of a pioneering movement that explores the possibilities of applying games with a purpose to hard problems.

Bayesian networks are a type of probabilistic graphical models used for reasoning under uncertainty. In other words, they are used to reason about the influence that events in the world have on the probabilities of others. These events can be facts or observations and they are represented by variables. In the context of Bayesian networks, the influence of variables on each other is referred to as dependence. If variables influence each other they are called dependent; otherwise they are called independent. We chose to develop our game with a purpose for application to the field of Bayesian networks, because they are very hard to learn from data and they have a very broad application domain.

The outline of the thesis is as follows. In the next section we give a more detailed explanation of the similarities and differences between crowdsourcing and human-based computation, we further explain Bayesian networks and what makes them hard to learn and we present some related work. The methodology and theoretical contributions of the research are presented in Section 3, with a description of the research questions and experimental setup. In Section 4 we describe the results of the research. Finally, in Section 5, we discuss the main findings, some lessons we learned and open questions.

2 Background

2.1 Crowdsourcing

Crowdsourcing is a distributed problem-solving and production model that involves outsourcing tasks to a distributed group of people (a crowd). Although it is common for this process to occur online, technically it can also occur offline. One of the main differences with ordinary outsourcing is that a task is not outsourced to a specific (affiliated) body, such as paid employees. The definition of crowdsourcing in the literature varies greatly and after studying more than 40 definitions of crowdsourcing, Estellés and González propose a new integrating definition (Estellés-Arolas & Guevara, 2012):

> Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on the type of activity undertaken.

From this new definition, we can see that crowdsourcing is thought to be mutually beneficial. The crowdsourcer utilizes what the crowd has provided, while the crowd receives some type of reward. In the case of a game, the reward can be the fun experienced while playing the game, but possibly also other forms of reward.

2.2 Human-based computation

Human-based computation is a technique from computer science in which a computational process performs its function by delegating some of the steps to (one or more) humans. This approach achieves a form of symbiotic human-computer interaction by considering the abilities and costs associated with the human and the computer and splitting the workload accordingly. The origins of human-based computation are often considered to be in the early work on interactive evolutionary computation. The idea behind interactive evolutionary computation is due to Richard Dawkins (Dawkins, 1986). Software accompanying his book "The Blind Watchmaker" asks a human to be the fitness function of an evolutionary algorithm. In other words, the user is tasked with judging which solutions are "good" and thus guiding the evolutionary algorithm. Victor Johnston and Karl Sims extended this concept by harnessing power of many people for fitness evaluation (Sims, 1991; Caldwell & Johnston, 1991). The growth of the internet has led to a shift of research on human-based computation from using single users to using large crowds of users, i.e. crowdsourcing.

2.3 Games with a purpose

Human-based computation forms the basis for games with a purpose (GWAPs), which is why they are also commonly referred to as human-based computation games. This type of human-based computation is made popular by Luis von Ahn with his work on games such as the "ESP game" (von Ahn & Dabbish, 2004; von Ahn, 2007), a game in which players are challenged to correctly label images. A more recent successful application of the GWAP paradigm is "Foldit" (Cooper et al., 2010; Cooper, 2012), in which humans apply their spatial problem-solving abilities to solve protein folding problems (Khatib, Cooper, et al., 2011; Khatib, DiMaio, et al., 2011). The potential of this new scientific method is illustrated by the fact that the initiator Seth Cooper has recently won the ACM Doctoral Dissertation Award 2011 (ACM, n.d.-a).

Although the terms "games with a purpose" and "serious games" are often used synonymously (Dugan et al., 2007; Stone, 2009), in our opinion these are not the same. Serious games are defined as games which have a primary goal other than entertainment. Although this can also be true for games with a purpose, serious games lack the human-based computation component and were actually introduced well before electronic games were common in entertainment (Abt, 1970). A serious game generally attempts to realize some form of progress in an individual player, such as therapeutic games or educational games, while the progress realized in games with a purpose does not lie with individual players but with the task that is being performed. Thus, to summarize, games with a purpose are human-based computation games in which the purpose of the game lies with the task being solved and not with the individual players, whereas serious games aim primarily to educate and train (Michael & Chen, 2005; Siorpaes & Hepp, 2008).

A game with a purpose can have advantages over traditional research techniques. A well-designed game produces incentive for the users to participate in the experiment. By incorporating a competitive element into the game, we can stimulate the users' motivation to try their best in producing good solutions (von Ahn, 2007). Furthermore, if the game should become popular the possible income could fund further research.

2.4 Related work

We make a distinction between games with a purpose that have a research-oriented nature and those that have a more practical nature. Luis von Ahn has developed several GWAPs that have a relatively practical nature. They intend to delegate a task to humans that cannot be solved by computers alone, but they generally do not intend to investigate *how* humans solve the task. Examples include the ESP game (von Ahn & Dabbish, 2004; *Website of several GWAPs*, n.d.), reCAPTCHA (re*CAPTCHA Website*, n.d.) and Tag a Tune (*Website of several GWAPs*, n.d.). Other projects with similar goals include:

- Phrase Detectives - University of Essex - Phrase Detectives allows players to indicate relationships between words and phrases to create a database of linguistic information. (*Phrase Detectives Website*, n.d.; Chamberlain, Poesio, & Kruschwitz, 2008)

- OnToGalaxy - University of Bremen - In OnToGalaxy players help to acquire common sense knowledge about words. (*OnToGalaxy Website*, n.d.; Krause, Takhtamysheva, Wittstock, & Malaka, 2010)

- EyeWire - MIT and Max Planck Institute for Medical Research - EyeWire attempts to find the connectome of the retina. (*EyeWire Website*, n.d.)

The more research-oriented projects generally attempt to improve automated problem solving techniques by observing how humans solve the problems they are given. Examples of such projects include:

- Foldit - University of Washington - This game lets players fold proteins in the form of 3-dimensional puzzles. The researchers attempt to improve their folding algorithms by investigating how humans perform the task. (*Foldit Website*, n.d.; Cooper et al., 2010; Cooper, 2012; Khatib, Cooper, et al., 2011)

- EteRNA - Carnegie Mellon University and Stanford University - EteRNA is a game in which players are tasked with designing RNA sequences that fold into a given configuration. The solutions provided by players are evaluated to improve the predictions of RNA folding by computer models. (*EteRNA Website*, n.d.)

- Phylo - McGill Centre for Bioinformatics - In the Phylo game, players align colored squares. While doing this, they contribute to solving the problem of multiple sequence alignment. Ultimately, the goal is to understand how and where functions of an organism are encoded in their DNA. (*Phylo Website*, n.d.; Kawrykow et al., 2012)

Our project is different from the projects above in that we intend to explore the possibilities of building a GWAP for an entire modeling framework instead of specific problem instances. This means that, unlike the GWAPs above, our game could have an impact in any problem domain in which that modeling framework can be used. As we will explain in the next section, Bayesian networks are highly versatile and have many application domains so our GWAP could have impact in a broad range of domains. To our current knowledge, our GWAP is the only one in existence that targets the Bayesian network structure learning problem. The ultimate goal of our GWAP is to see if we can extract and automate the techniques used by human players in order to improve Bayesian network structure learning algorithms, but this is only after our GWAP has proven to be applicable in the more practical sense discussed above.

2.5 Bayesian networks

Bayesian networks are probabilistic models that provide a framework for reasoning under uncertainty. As we have already indicated, BNs have applications in a vast range of domains: they are used for modeling knowledge in areas such as computational biology (Friedman, Linial, Nachman, & Pe'er, 2000), bioinformatics (Zou & Conzen, 2005), medicine (Long, 1989), information retrieval (Fung & Del Favero, 1995), image processing (Luttrell, 1994), decision support systems (Horvitz & Barry, 1995), engineering (Pernkopf, 2004), gaming (Becker, Nakasone, Prendinger, Ishizuka, & Wachsmuth, 2005) and law (Thagard, 2004).

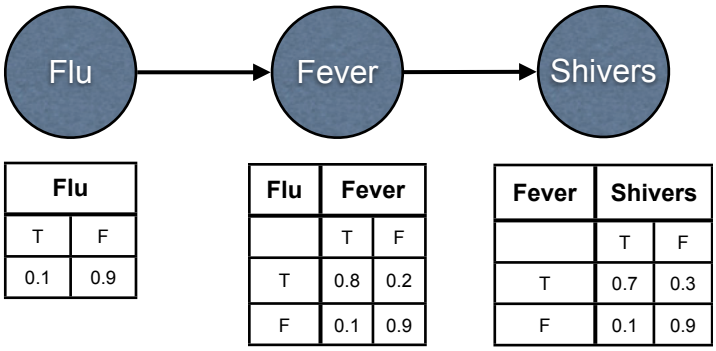


Figure 1: A very simple example of a Bayesian network. The figure shows the Directed Acyclic Graph as well as the probabilities of each node given the values of its parents. In this case, each node has zero or one parents and their states are binary: True or False.

See e.g. Charniak (1991); Haddawy (1999); Heckerman, Mamdani, and Wellman (1995) for overviews. Judea Pearl, one of the pioneers of the probabilistic approach to Artificial Intelligence (Pearl, 1982) has been credited with the invention of Bayesian networks for the algorithm he proposed for belief propagation in graphical models (Pearl, 1982, 1988) and has recently received the ACM Turing Award 2011 (ACM, n.d.-b) for his achievements in this area of research.

Formal definition A Bayesian network is defined as a pair $BN = (G, P)$, where G is a directed acyclic graph (DAG) $G = (V, E)$ and P is a joint probability distribution of the random variables X. There exists a 1-1 correspondence between the nodes in V and the random variables in X; the (directed) edges, or arcs, $E \subseteq (V \times V)$ correspond to direct causal relationships between the variables. A Bayesian network BN offers a compact representation of the joint probability distribution P in terms of local conditional probability tables (CPTs), by taking into account the conditional independences represented by the DAG (Pearl, 1988).

Conditional (in-)dependence Let us have a look at a simple example from the medical domain in Figure 1. The example shows three variables that tell us something about a person. If the person has the flu, there is a high probability that the person has a fever. If the person has a fever, that increases the probability of him shivering. So Shivers is dependent of Fever and Fever is dependent of Flu. That means that Shivers is also dependent of Flu. But now let us say that at some point in time we know for a fact that the person has a fever (e.g. by measuring his temperature). Then knowledge about whether or not the person has the flu will not have an effect on the probability that he is shivering. This is because knowledge about Flu has an indirect effect on Shivers through the variable Fever. In this case, we say that Flu is conditionally independent of Shivers given knowledge about Fever. As we will discuss further on, there are also situations in which knowledge about a variable makes two other variables dependent. Together, these conditional (in-)dependencies form what we call the

conditional dependence relations.

d-Separation Now that we have introduced conditional (in)dependence and the fact that knowledge about a variable can alter the dependency between other variables, we will proceed to introducing *d-separation*. d-Separation is a criterion for deciding, given a DAG, whether a set of variables U is independent of another set V, given a third set Z. It was introduced by Pearl (1988) and has since become a common notion in Bayesian network theory (Korb & Nicholson, 2004). The general idea is to associate dependence with the existence of a connecting path and independence with the absence of such a path (i.e. "separation"). The set Z represents the set of variables for which there is knowledge of their states. In other words, with d-separation we can tell whether given knowledge about the states of variables in Z, the variables in U and V are dependent or not. For two variables u and v d-separation is defined as follows: Let P be a trail (that is, a collection of edges which is like a path, but each of whose edges may have any direction) from node u to v. Then P is said to be d-separated by a set of nodes Z if and only if (at least) one of the following holds:

- P contains a chain, $x \to m \to y$, such that the middle node m is in Z
- P contains a chain, $x \leftarrow m \leftarrow y$, such that the middle node m is in Z
- P contains a fork, $x \leftarrow m \to y$, such that the middle node m is in Z
- P contains an inverted fork (or collider), $x \to m \leftarrow y$, such that the middle node m is not in Z and no descendant of m is in Z

So u and v are said to be d-separated by Z if all trails between them are d-separated. If u and v are not d-separated, they are called d-connected.

Causal networks Although Bayesian networks are often used to represent causal relationships, this is not necessarily the case. A directed edge from vertex a to vertex c does not require that the variable represented by c is causally dependent on the variable represented by a. This can be illustrated with an example: consider the Bayesian networks represented by the graphs $a \to b \to c$ and $a \leftarrow b \leftarrow c$. According to the definition of BNs they are equivalent, because they encode the same conditional independence relations (Pearl, 1988).

A BN which is explicitly intended to encode causal relationships is referred to as a Causal Bayesian network or simply as a causal network. Causal networks have additional semantics in place that specify that if a node X is actively caused to be in a given state x, then the probability density function changes to the one of the network obtained by cutting the links from X's parents to X, and setting X to the caused value x. This operation was dubbed $do(X = x)$ by Pearl (Pearl, 2000). The *do* operator allows us to perform 'graphical surgery' on Bayesian networks, disconnecting a variable from its normal causes. Using these semantics, one can predict the impact of external interventions from data gathered prior to the intervention. Intervention in causal BNs can give insight in how probabilities of variables behave in the circumstances that have our interest. This feature of Bayesian networks is particularly powerful as it allows us to use BNs as predictors and decision models.

Inference From the formal definition specified above, we can see that vertices in a BN represent random variables in a Bayesian sense, they may be observable quantities, latent variables, unknown parameters or hypotheses. The edges in the BN represent conditional dependence relations. Assigned to each vertex is a probability distribution that describes the probabilities of the values of that vertex given the values of its parent vertices. Figure 1 shows an example of a Bayesian network and its probability distribution. Because a BN encodes the variables and relations between them, it can be queried to gain knowledge on the state of a set of variables given that another set of variables has been observed. The process of computing the posterior distribution of variables given some evidence is called *probabilistic inference* (Pearl, 1988). An example use of this technique is calculating probabilities for the presence of diseases given observed symptoms, making medical diagnostics a popular application domain (Nikovski, 2000; Pang, Zhang, Li, & Wang, 2004; Xiang, Pant, Eisen, Beddoes, & Poole, 1993; Jr, Roberts, Shaffer, & Haddawy, 1997; Lisboa, Wong, Harris, & Swindell, 2003; Milho, Fred, Albano, Baptista, & Sena, 2000; Long, 1989).

Learning Bayesian networks Before we can perform inference on a Bayesian network it needs to be constructed first. Constructing a BN consists of two main sub-tasks: structure learning and parameter learning. The first is involved with the (causal) structure of the graph, while the latter concerns itself with the probability distributions on the vertices. Specifying the parameters of a Bayesian network involves specifying for each node X the probability distribution for X conditional on X's parents. As the parents of X are generally unknown and can become known after structure learning, parameter learning is often performed only after structure learning.

A traditional BN construction method involves a Bayesian modeler and a domain expert who manually construct a Bayesian network. In relatively simple cases this is a viable method but as the number of variables grow, the more time-consuming, error-prone and tedious it becomes. More recently, several automated BN learning techniques have appeared which are used to learn BN structures from sets of joint observations. A set of joint observations is a series of simultaneous observations on all variables under consideration. Table 1 shows an example of such joint observations for the simple network in Figure 1. These structure learning algorithms generally belong to the classes of constraint-based or score-/metric-based search algorithms although hybrid algorithms exist (see e.g. (Korb & Nicholson, 2004) for an overview). The constraint-based approach attempts to find a minimal structure that satisfies the conditional independence relations in the data set. The score based approach attempts to find a structure that maximizes the fit of the model to the data. Examples of software packages implementing these algorithms are the Python Environment for Bayesian Learning (PEBL) (Shah & Woolf, 2009), bnlearn for R (*bnlearn for R Website*, n.d.) and BNT for Matlab (*BNT for Matlab Website*, n.d.).

$$f(N) = \sum_{i=1} N(-1)^{i+1} C_i^N 2^{i(N-i)} f(N-1) \tag{1}$$

Although these algorithms are generally considered an improvement over the domain-expert approach, they all share a common problem: the sheer number of possible structures. Equation 1 shows a recursive expression for the num-

	Variable		
	Flu	Fever	Shivers
Observations	T	F	T
	T	T	T
	T	F	T
	F	F	F
	F	T	T
	T	F	T

Table 1: Example of joint observations for the Bayesian network in Figure 1.

ber of possible DAGs given N variables (Robinson, 1977). It follows from the expression that with 3 variables we have 25 DAGs, with five there are 25,000 DAGs and with ten variables we have $4.2 * 10^{18}$ possible DAGs. This number grows super-exponentially in the number of variables. In fact, learning Bayesian networks has been shown to be NP-complete (Chickering, 1996), so to search this space of possible DAGs for the optimal structure is intractable. Note, however, that some NP-hard problems can be computed by algorithms that are polynomial in the overall input size n and non-polynomial only in some small aspect of the input called the input parameter. These problems are said to be fixed-parameter tractable for that input parameter (Downey & Fellows, 1999). The algorithms discussed above merely attempt to find "good enough" solutions in a reasonable amount of time. However, intractable Bayesian computations are not generally tractably approximable (Kwisthout, Wareham, & van Rooij, 2011), which suggests that perhaps these algorithms do not approximate, or BN structure learning may in fact be fixed-parameter tractable. In any case, in practice the algorithms still require a very large set of joint observations to be able to come up with a good structure. This is problematic, because generally these observations are not readily available.

Humans as Bayesians There is ongoing debate in Cognitive Science about whether humans are 'Bayesian' or not (Chater, Tenenbaum, & Yuille, 2006). This debate is concerned with the question whether cognitive judgments should be viewed as following optimal statistical inferences (in which case humans would be 'Bayesians'), or as following error prone heuristics that are insensitive to priors. Interestingly, there is evidence supporting both views. For instance, Kahneman & Tversky (Kahneman & Tversky, 1972) concluded from their experiments that humans are no Bayesians at all, while Griffiths & Tenenbaum (Griffiths & Tenenbaum, 2006) suggested that everyday cognitive judgments follow the same optimal statistical principles as perception and memory. They argued that there is a close correspondence between peoples' implicit probabilistic models and the statistics of the world. It has been suggested that when reasoning under uncertainty in everyday life humans do seem to follow optimal statistical inferences, while when explicitly asked to reason about probabilities they do not (Griffiths & Tenenbaum, 2006). Although the evidence on whether humans are 'Bayesian reasoners' is inconclusive, evidence does exist that suggests humans follow some form of Bayesian inference rules in everyday cognition. Several models postulating that a part of human cognition performs some type

of Bayesian inference have been proposed in various cognitive domains, including vision (Yuille & Kersten, 2006; Kersten, Mamassian, & Yuille, 2004), language (Chater & Manning, 2006), decision making (Sloman & Hagmayer, 2006), motor planning (Wolpert & Ghahramani, 2005), eye movement control (Engbert & Krügel, 2010), and theory of mind (Baker, Saxe, & Tenenbaum, 2009; Cuijpers, Schie, Koppen, Erlhagen, & Bekkering, 2006). The ability of some of these models to successfully predict human behavior, albeit in relatively small tasks, has led us to believe we might be able to harvest these Bayesian reasoning abilities from humans to help guide a Bayesian network learning algorithm. Furthermore, as there is belief that causality is central in how humans understand the world (Steyvers, Tenenbaum, Wagenmakers, & Blum, 2003; Sloman, 2005) we are interested to see whether humans can infer causal structure from observations generated by a BN.

2.6 Conclusion

In the previous two sections, we have described the relatively novel "games with a purpose" technique and we have introduced the Bayesian network structure learning problem we wish to apply the GWAP methodology to. Now that we have provided the necessary background information, we will proceed to the Methodology section where we will restate the main hypothesis and explain how we will investigate this hypothesis.

3 Methodology

This section will introduce the methodology of the present research. We will start by stating our main hypothesis and explaining how we mapped the problem domain of Bayesian networks and observations to the game domain on a conceptual level. Then we will proceed to explain how we will investigate our main hypothesis and what the experimental setup will be. Finally we will provide implementational details about the game.

3.1 Main hypothesis

We hypothesized in the introduction that players of a casual puzzle game can contribute to the construction of Bayesian networks in any domain by inferring conditional dependence relations from joint observations presented in a visually abstract manner. When we say that we want to present the joint observations in a visually abstract manner we mean that the way of presenting the observations should be independent of the domain in which the observations were obtained. In other words, with the present research project, we wanted to show that it is possible to build a casual game that allows non-experts to contribute to the construction of Bayesian networks.

As we have explained in Section 2, learning Bayesian networks from data consists of two subtasks: learning the structure and learning the parameters. We have also explained that learning the parameters usually comes second to learning the structure. Here we take the same approach: we will focus our research on finding the structure of a Bayesian network. In order to investigate our hypothesis, we first needed to develop a mapping from Bayesian network structures and observations to a game. The following subsection will describe how we came to that mapping.

3.2 GWAP: Conceptual design

Here, we will report some of the steps we have taken and problems we have encountered in mapping causal structures and observations (the research domain) to the user-friendly game world of *Ahsum Nimity* (the game domain).

Goal For our game to be successful as a game with a purpose we needed the game to attract players. For the present research in particular, we needed the game to attract enough players to be able to investigate our hypothesis and obtain significant results. For those reasons, we needed our game to be fun. We also wanted the game to support the players in achieving the task we wanted them to achieve: inspecting joint observations and providing dependency information about the underlying Bayesian network. We refer to this underlying network as the ground truth. To achieve our research goals, we set the following goals for our game:

- Provide a usable interface that allows inspection of joint observations of discrete variables from any domain.
- Provide a usable interface that allows users to describe the conditional dependence relations they infer from these observations.

- Motivate and train the user by giving feedback in the form of rewards (points, stars) and info about the user's progression throughout the game.
- Provide a storyline that captivates the user and wraps the abstract notions of joint observations and conditional dependence relations into metaphors the user can easily work with (the game domain).

Inspection of joint observations In order to allow users to inspect the joint observations, we needed a way to present the joint observations to the users in a domain-agnostic manner. As we have explained in the previous section, one joint observation consists of a series of states: one state for each variable under observation. We first considered how to present one single joint observation. While exploring the possibilities, we quickly came up with using different colors for the different states of the variables. Then we needed to find a representation for the variables and after a while we settled on using floating cities. (How we came to this representation will be elaborated in Section 3.5.) The combination of a city representing a variable and a color representing its state allowed us to present joint observations from any domain as a simultaneous coloring of cities.

The next challenge was to allow the player to inspect multiple joint observations. We decided it would be confusing to the players to show the same variable multiple times in different states, so we came up with the alternative to have variables change states. In our newly found representation this means that the floating cities change colors. Multiple joint observations could now be shown by consecutively recoloring the cities. Because we did not know at what rate humans would prefer these consecutive presentations of joint observations we have decided to allow the users to specify the presentation speed themselves. More information on the manner in which players are allowed to do so will follow in Section 3.5. To prevent any unmonitored effects from the order of the joint observations on the performance of the players, the joint observations are presented in a random order.

Input from users In the research domain, we want players to judge whether two variables are (conditionally) dependent or not because such information can be used directly to guide structure learning algorithms. With enough of these "dependency statements" we might even already be able to construct (undirected) BN structures without the use of separate structure learning algorithms. In the game domain one of these statements consists of choosing a pair of variables to judge, and a decision: connected or disconnected. Say a user picks variables A and B (cities A and B) and judges them to be connected, this means that (s)he thinks A and B are dependent. We chose not to include direction in these dependency statements, because of the equivalence between directions as discussed in Section 2.5 and because we wanted to prevent the added complexity for the users. We only allow dependency statements between two variables because allowing more variables would, to our opinion, overly complicate the task. Any (in)dependency between groups of variables can be expressed in terms of individual (in)dependency statements between two variables, so no loss of generality occurs.

To give the subjects an investigatory tool, as well as more expressive power, we allow them to use an operator we refer to as clamping. Using this operator the user can fix the value of one or more variables to a randomly chosen state.

Input	D-separated	
	Yes	No
Connected	INCORRECT	CORRECT
Disconnected	CORRECT	INCORRECT

Table 2: This table shows the relation between d-separation, the player's decision (connected/disconnected) and the correctness of their decision. The table shows for example that if the player says two variables are connected while they are d-separated, their decision is incorrect.

The operator effectively prunes the list of observations to only those in which the clamped variables have the fixed value. Judging two variables to be connected while a set of variables is clamped is considered as the equivalent of stating that those two variables are dependent given that set of clamped variables. We train users to this equivalency by giving them feedback about their performance. Apart from the purpose of clamping to increase expressiveness of the dependence relations (they add the possibility of encoding conditional dependence relations), clamping has another purpose in our research. Several researchers argue that humans learn how the world works from intervening in the world instead of merely observing (Steyvers et al., 2003; Sloman, 2005; Sloman & Hagmayer, 2006). This suggests that humans may be more capable of forming theories about the underlying Bayesian network structure if they are allowed to intervene. In the game, clamping allows for (a manner of) such intervention. Sloman and Hagmayer (2006) place the notion of making choices in the world within the framework of Bayesian networks and relate this type of intervention to the *do* operator mentioned earlier. Note that our clamping intervention is not exactly the same as the intervention achieved by the *do* operator, as clamping only filters the set of joint observations instead of operating on the ground truth network.

Giving feedback Providing feedback of the player's performance during the game allows us to train the player to use the tools we give them effectively. It also allows us to teach the player what is correct and what is incorrect. Our intention is to have players provide conditional dependence or independence statements about variables, so our game should correctly encode and evaluate those decisions in terms of conditional dependence. We check the correctness of the subject's decisions in the Bayesian network domain by applying the Bayes-Ball algorithm (Shachter, 1998) to the structure of the ground truth network. Bayes-Ball is proven to be a correct implementation of the principles of d-separation (Shachter, 1998) and under the faithfulness assumption, which we will explain further on, d-separation is a correct equivalent of conditional independence (Pearl, 1988). Using the Bayes-Ball algorithm we compute the relevant and irrelevant nodes for either one of the two connected nodes. If the other node is in the irrelevant nodes given the clamped nodes then the nodes are d-separated and thus, under the faithfulness assumption, conditionally independent. Otherwise, they are dependent.

The faithfulness assumption (Pearl, 1988) is relevant here, because there

exists a scenario in which the dependence of two variables cannot be assessed by looking at the structure alone. When two nodes in the network are *not* d-separated (and thus normally considered dependent), their parameters may still make them independent. This is the case when the nodes have uniform probability distribution given their parent nodes. In that scenario, one would need to assess the parameters of the BN to see whether the nodes are conditionally independent or not. The faithfulness assumption, however, states that this scenario can be ignored because conditional independence only occurs as a result of causal (structural) independence.

Given our approach encoding players' decisions in terms of d-separation, the correctness of a subject's statements is evaluated as presented in Table 2. Note that in real-world problems it would be impossible to give this type of feedback, because the so-called ground truth is unknown; the ground truth is actually what we are trying to find when using our GWAP. We therefore only intend to use this direct feedback as a training tool. As part of this research we will investigate whether users also perform well without direct feedback.

Providing a storyline In the implementation section (Section 3.5), we will describe the storyline that we came up with to create an understandable "world" in which it would make sense that there are floating cities that change color. The storyline is also intended to create player engagement, such that he/she is compelled to play the game.

Other relevant decisions Here we will report some of the various other relevant decisions we have made about the conceptual design of the game.

- Number of clamps allowed.
  We allow players to clamp multiple variables at the same time. We did this for multiple reasons. We wanted to allow people to be able to indicate particular types of situations in the ground truth (e.g. that A and D are independent of each other given both B *and* C), which would be impossible with only one clamp. Also, we did not yet know how exactly people were going to use the clamping operator, so to specify a limit to the number of clamps ourselves without prior investigation seemed unwise. We thought it better to allow players to find the optimal strategy. We considered to give an extra score bonus for clamping more variables, but we chose not to do so because that would give motivation to clamp as many nodes as possible, regardless of the relevance of the clamped variables for the discussed pair. We decided that this conflicted with our goals of gathering maximally relevant data from the user.

- Pairing under clamps.
  In principle, we have made it possible to pair any two variables, except in the situation where both variables are clamped. We decided not to allow this due to the fact that we can be fairly certain that when both variables are fixed to a value, there is no information for the user to decide whether they are dependent or not. We do allow pairing between one clamped and one unclamped variable. This may seem to be just as meaningless, but we hypothesized that people would go about clamping searching for effects to 'occur' (become visible). We reasoned as such: if clamping variable A

suddenly fixed variable B to one value, that would be a strong indication of a positive connection. We did not want the user to first have to unclamp A before saying that A and B are connected because that would not be good from usability perspective and we could not explain to the user why that would be necessary. We therefore felt it was better to allow pairing with 1 clamped variable, and interpreting that decision as though that variable was not clamped.

- Correcting decisions.
  We chose not to let players correct their decisions for several reasons, the two most important being the following. First, we were afraid the players' performance would start to depend even more on the direct feedback. Second, if the players could always correct their mistakes, we thought there would not be enough incentive to really learn to use the observations and be correct in one try.

- Pairs under different clamp sets.
  Technically, it would be possible to make dependency statements about two variables under different clamp sets. We decided not to allow this for multiple reasons; primarily because it was hard to explain the concept to the players and secondly because we expected we would have enough subjects so we would not need multiple statements about one pair from a single user. Also, we were afraid that people would use this feature to easily score points. For instance, if a user should find out that A and B have a direct connection, (s)he would be able to score points for indicating that connection under each possible set of clamps, and thus skip through levels without providing a broad inspection of the level. In hindsight, allowing this to happen might have actually been a good approach, because it could provide insight to which variables have a direct connection in the ground truth. In future research we suggest investigating this approach.

- Replaying levels.
  We decided to allow players to replay a level they had already played before. This is a feature that is generally expected in a game and allows the players to train and improve as much as they want. The benefits are that players can choose for themselves which levels they need to play to improve their skills, and if a lot of players play several levels multiple times, this would allow us to investigate learning effects. Furthermore, if we did not allow it, the game might lose some players due to them expecting the feature. The feature could, however, potentially be a problem for within-subject factors as the player then has some control over the order in which levels are presented. But because we have the ability to only consider each subject's first attempt at a level, we decided not to remove the replay feature.

- Target scores.
  We considered to have the player finish a level whenever (s)he wanted to. However, this was not experienced as fun and game-like enough. So we came up with target scores, which give a clear goal and reduce the number of decisions a user has to make (not having to think about when to end the level). We chose to set the same target score for all experimental

levels, because we wanted to prevent any influence of the target score on subjects' performance.

- Multiple tutorial levels.
  In our beta tests, we noticed that having no tutorial was simply not an option. After that we tried to build a one-level tutorial, but this level quickly became far too complex and immediately demotivated players. Finally, we introduced multiple tutorial levels in order to introduce users to the game's interface and complexities gradually, hopefully drawing them into the complex puzzles without scaring them off.

3.3 Research questions

When we had created the essential mapping from Bayesian networks to the game, we needed to split our main hypothesis into directly investigable components. To see if non-expert users can indeed contribute to the construction of Bayesian network structures, we posed several research questions that the present research aims to provide an answer to. In this subsection, we will first list the abstract questions and how we split these up into more concrete questions. Then, we will explain what the goal of each particular question is and how we are going to answer it.

Textbox 1 Research questions

RQ1 How well do players perform?

- (a) Do players perform better than chance?
- (b) Do players perform better than chance with decision schemes?
- (c) Is the players' performance similar on all ground truth BNs?
- (d) Do performances drop when subjects have fewer observations available?
- (e) Do performances drop when players no longer get direct feedback?
- (f) Is there a cross-network learning effect?

RQ2 How well do the players use the tools provided in the game?

- (a) Do players show a preference for dependence vs independence?
- (b) Do players use clamping effectively?

RQ3 Can we already build BN structures with players' input?

The research questions presented in Textbox 1 should provide insight into whether our main hypothesis is correct. They serve this purpose by telling us whether the users are capable of performing the task (RQ1), how they use the tools we provided them (RQ2) and whether the information they provide is actually useful (RQ3). We further split some of these questions to more concrete ones as is also shown in Textbox 1. In the next few pages, we will explain these research questions and the experimental factors we introduced to answer them.

3.3.1 RQ 1a: Do players perform better than chance?

Goal To investigate whether the human players had *any* information available when making their decisions about the dependence relations. If they were not performing better than can be expected purely on the basis of chance, this indicates that players were "just guessing" and were thus unable to use any information. If, however, they perform better than chance, this indicates the presence of such guiding information. The presence of such information would indicate that the human players were somehow able to extract information from the visual presentation of observations and might be able to contribute that information to constructing Bayesian networks. We expect the human players to perform significantly better than the random players.

Method To see whether this is the case, we have a computer player randomly make the same types of decisions as the human player and compare their performance. For this purpose, performance is defined as the number of correct decisions (given the same amount of decisions in total).

Remember the three components that make up a human decision: clamps, a pair of variables and connected vs disconnected. The random player will pick a random set of clamped variables, then pick two random variables to make a dependency statement about and finally randomly decides whether they are dependent or not. It will do this for every decision made by a human. As the computer player is completely random, this is a very weak baseline. So if the humans do not outperform the random player in any condition we can conclude that our game has failed to effectively encode the problem, or indeed that it isn't possible at all.

3.3.2 RQ 1b: Do players perform better than chance with decision schemes?

Goal Because the purely random player is a rather weak baseline to compare the human players to, we want to have a stronger baseline as well. For this research question, we compare the human players to several different types on random players. These random players each have what we call a connectivity bias in their decisions. We expect the human players to perform better than all the random players with connectivity bias.

Method We have a random player that says "connected" in 0% of the cases, in 10% of the cases, 20% of the cases, etc. up to a random player that says the variables are connected in 100% of the cases. We refer to these as the random decision schemes. The way in which the ground truth BNs differ from each other (their connectivity) makes it either more rewarding to always say two variables are connected or never say that they are connected, or something in between. If the human players perform better than all these different random players this indicates that they are not merely randomly choosing based on some predetermined distribution. It would be more evidence that they are indeed using some information present in the observations to judge dependency on a case-by-case basis. If they perform as good as the random player, it might suggest that they only pick a random distribution intelligently. If they perform worse, it might suggest that the information is actually working against their

judgment. Note that in real-world scenarios the connectivity of the ground truth is unknown, so even if players just pick a decision distribution intelligently they would actually have to make use of some information present in the observations.

3.3.3 RQ 1c: Is the players' performance similar on all ground truth BNs?

Goal To give some intuition for whether our findings will generalize to all Bayesian network ground truths. Although the space of all possible BNs is very large, a property as simple as the number of variables in the network may already give a difference in performance. We have no real expectations concerning the results of this research question. It might be the case that subjects perform better when levels are small, because there is less information to process, but it might also be the case that they perform better on larger networks because they can pick the most obvious connections.

Method For this research question we have introduced a within subject factor: the ground truth BN. As we explained earlier, the observations of each level are generated using a Bayesian network. For the last three levels in the game, we will randomly vary for each subject which BN is first, second and third. As described in Section 3.4, we have chosen three Bayesian networks that are commonly referred to in BN literature and which vary in size from relatively small to moderate size. As this is a very small selection of networks the results of this analysis are likely not to be very conclusive, but if we do find a large difference in performance it may spark more ideas for future investigation. In the experimental setup section we will explain what these BNs look like and how exactly they are placed within the game.

3.3.4 RQ 1d: Do performances drop when subjects have fewer observations available?

Goal To investigate whether people perform worse when there are fewer observations available. This is important, because in real-world problems there may not be a lot of observations available. Structure learning algorithms in particular start to fail when there is no abundance of data. We do expect some difference in performance for the number of observations, although we expect this difference to be largest between the smallest number of observations and the medium number, because we do not think the medium and large numbers will make a large difference. We expect this difference to be largest in the largest ground truth network. This is because in that network it is possible to clamp a large number of variables, which causes a lot of pruning on the observations. When there are few observations available, these large clampsets will quickly lose value.

Method For this research question we have introduced another experimental factor, namely the number of observations available to subjects. This is a between-subject factor with three levels: 300, 3000 and 30000 observations. Each subject is randomly assigned to one of these categories. The observations themselves are randomly drawn from the set of all observations (which is a set of 30000 observations). This also causes the order of observations to be random.

3.3.5 RQ 1e: Do performances drop when players no longer get direct feedback?

Goal This research question is important because the fact that we give direct feedback to train people could potentially be the reason for good performance. It might be the case that without it, people are no longer able to perform well. This would be a real problem to our GWAP because in real-world applications we would not be able to provide the same kind of feedback. Other forms of feedback are possible, but the feedback mechanism we chose uses the ground truth from which the observations are obtained. We would not have access to this ground truth in real-world scenarios, because the ground truth is exactly what we are trying to find. By the time the players reach the final level, we expect them to have learned a strategy that is independent of the direct feedback. (Especially because we do not allow people to correct their mistakes after the tutorial levels.) As such, we do not expect to see a significant drop in performance when players no longer receive direct feedback.

Method To see whether players perform differently without direct feedback we have made the final level in the game a "blind" level. This means that there is no direct feedback about the subjects' decisions during the game, so the player cannot see how well (s)he is doing during that level. Only after the level is completed, the player will be able to see their performance. Because of the within-subject ground truth factor, this level can be any of the three ground truths. We compare subjects' performances on the blind level to their performances on non-blind levels.

3.3.6 RQ 1f: Is there a cross-network learning effect?

Goal The answer to this research question should show us whether (despite every level being a different Bayesian network) players get better over time. More specifically, we want to see whether there is an increase in performance between consecutive levels. We would expect people to gradually become better at the game despite there being different BN ground truths.

Method To provide an answer to this question, we will measure performance in the last four levels (except the blind level) and compare them. We chose to exclude the blind level due to the interaction that might occur with the blind experimental condition. Furthermore, including the blind level would lead to a decrease in the number of subjects we could use for this analysis.

3.3.7 RQ 2a: Do players show a preference for (in-)/dependence?

Goal To see whether the players have a preference for specifying connected versus disconnected (dependent vs independent relations) and if there is a difference in their performance on these two types. The idea is that if we know which users prefer and which they are better at, we can improve future versions of the game and maximize the usefulness of their input. We have no real expectations about the results of this study although intuitively we tend to think that strong correlations are likely to "pop out" and that would probably lead to more dependence relations being specified.

Method For this study, we will be comparing the number of input relations that were said to be connected against the number of relations that were said to be disconnected and how many of them were correct and incorrect.

3.3.8 RQ 2b: Is clamping used effectively?

Goal To see if the clamping tool we have provided is being used effectively by the players. If this tool is not used effectively, then it has no real purpose in our game and it should be removed for sake of simplicity. However, there is a possibility that clamps are used to create "order" in the observations, but that they do not really contribute to the correctness of their decisions directly. We currently have no way of seeing whether this is the case, so in fact this research question can only be answered partially. (One way would be to also include the ability to clamp as an experimental factor. But because we did not do this in the present research we can only recommend such a methodology for future research.) We would expect people to make effective use of the clamping tool, but whether this means they use them directly in their decision or whether they only use them to create order is not clear.

Method To answer this research question we will look at the effectiveness of the clamps. Using the Bayes-Ball algorithm we will compute for each human input whether the absence of the set of clamps chosen by the player would have changed the correctness of the dependency relation. In other words, we will compute whether the set of clamps has contributed to the correctness or incorrectness of the dependency relation. This will result in a number of decisions that was correct due to clamps, that was incorrect due to clamps and a set in which the clamps had no effect on the correctness. By comparing these numbers, we can have some insight into how clamping is used. To have a clean comparison, for this study we have limited the dataset to only the input where a single variable was clamped. We will also provide some descriptives on the input where the number of clamps was greater than 1.

3.3.9 RQ 3: Can we already build BN structures with their input?

Goal Up to now, we have only looked at performance of players as an individual and we have only looked at performance as the proportion of correct decisions. But similar to the ideas put forth in books about the "wisdom of crowds" (Surowiecki, 2005), we want to know whether the whole is greater than the sum of its parts. The goal of this study is to see if we can already use the information the whole group of users has provided to build BN structures and to see if the group as a whole is performing better than the individual. We expect the accuracy of decisions made by the group to be higher than that of an individual. Given this assumption, we expect to be able to form undirected graphs that are pretty similar to the ground truth.

Method In order to provide an answer to this research question, we have developed a voting system that allows us to mark dependencies as either present or absent. Starting out with a fully connected graph, it allows us to prune the graph according to the decisions of the user group. We will then compare the resulting graph to the ground truth to see if the result is anything like the ground

truth. Furthermore, we will investigate what the performance is of the collective by computing the number of correct and incorrect decisions that came out of the voting system. We have decided not to include clamping in this preliminary investigation, because a voting scheme that includes clamping is not trivial and needs to be developed first.

3.4 Experimental Setup

In this section we will summarize the experimental factors we have introduced in the previous section and explain how we have incorporated them into the game.

Bayesian networks The Bayesian networks we used are called Asia, Stud Farm and NHL. We chose these Bayesian networks because of their variation in network size, shape and presence in literature from the field. Asia (also known as Chest Clinic) is a small Bayesian network that calculates the probability of a patient having various lung diseases based on several factors, such as whether or not the patient has been to Asia recently (*Hugin Samples Website*, n.d.). The stud farm network is used to calculate the probabilities of horses in a stud farm being carriers of a recessive gene causing a life threatening disease (*Hugin Samples Website*, n.d.). The NHL Bayesian network is used to choose the appropriate treatment for (gastric) Non-Hodgkin Lymphoma and incorporates variables that are widely used in choosing the appropriate therapy for patients (Lucas, Boot, Taal, et al., 1998). A description of the structure of these Bayesian networks is given in the Appendix (section 7.1).

In our research, these networks were used as the ground truth from which the observations were generated. This way we had as many observations available as we needed to create our experimental conditions. The observations are generated using the "Generate Simulated Cases" function of SamIam (*SamIam Website*, n.d.) which generates joint observations according to the structure and parameters of the Bayesian network. The same networks were used with the Bayes-Ball algorithm for providing feedback to the user about their performance. This method allowed us to provide feedback in the first place, but because the observations were obtained from the network it also allowed us to be sure that our feedback corresponded correctly to the observations.

Experimental factors Some of the experimental factors are applied after the results from the game are obtained, such as the player type (random vs human), while others had to be incorporated into the game. The latter is true for the following factors:

- Number of observations. (300, 3000, 30000)
- Ground truth network. (Asia, StudFarm, NHL)
- Direct feedback (feedback, blind)

The number of observations was introduced as a between subject factor, while both the ground truth network and the presence of direct feedback are within-subject factors. Figure 2 shows the level structure of the game and how the factors play a role in that structure. Players are randomly assigned to one of

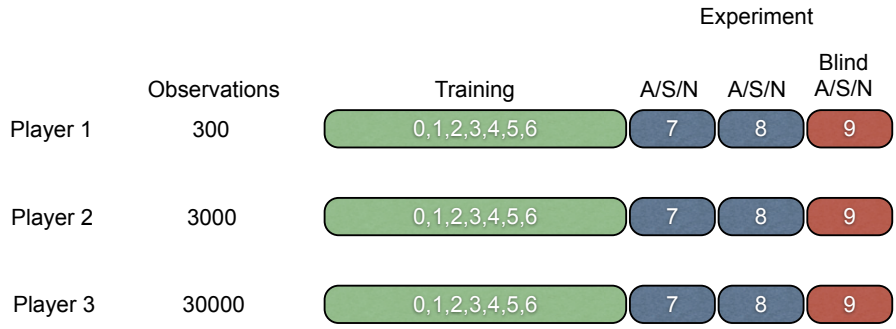


Figure 2: The experimental set-up of the game. For levels 7 through 9 it is decided at random whether they use the Asia, StudFarm or NHL (A/S/N) sets of observations.

the groups with different numbers of observations available. These observations are randomly selected from the largest set of observations. Their order is also random due to the random selection. The ground truth networks that form a game level are chosen in a random order for game levels 7, 8 and 9. The last level is always blind, so without direct feedback. Due to the random order of the ground truth networks, effectively the blind levels are distributed randomly across ground truth networks.

3.5 GWAP: Implementation

We developed the game for the Apple iOS operating system. The internship company has expertise and a keen interest in iOS development. Additionally, the Apple Appstore allows for relatively easy deployment to, and accumulation of, a potentially very large userbase. Having a large userbase would allow us to design an experiment which assigns users to several different conditions while still enabling us to find statical significant results. We began by developing a prototype to test basic gameplay elements. For developing the final game, we worked together closely with a professional illustrator to develop the game's storyline and all visual artwork. We developed a beta version, release version, and three iterations with post-release improvements.

Prototype The prototype is a simple Java application that visualizes observations generated from a manually constructed Bayesian network. Every variable in the network is visualized as a colored circle on a black background; the color of the circle depicts the state of the variable. No game metaphor or rewards are present in this version, but the prototype does provide users with an interface to inspect joint observations and input decisions about the (supposed) structure of the underlying Bayesian network. Using the prototype we performed a limited pilot study to see if, at face value, users are able to extract information about the underlying Bayesian network based on the joint observations. This study, albeit very limited, gave us strong confidence that we can design a fun game that allows humans to do that.

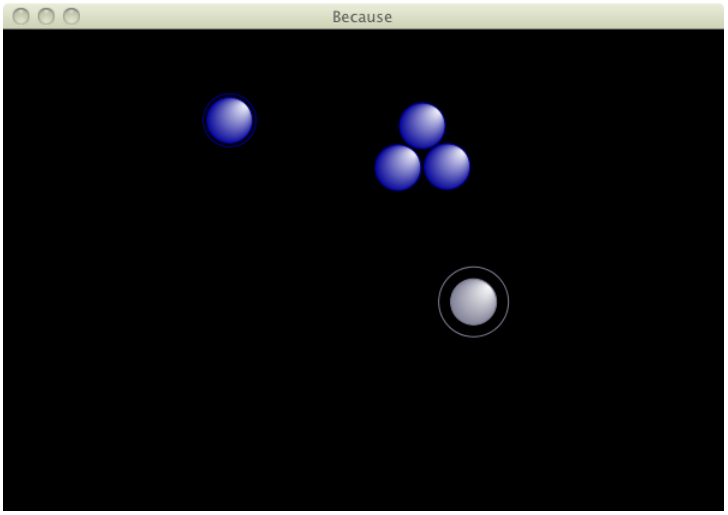


Figure 3: The user interface of our proof of concept application. The 'bubbles' represent variables (vertices) in a Bayesian network. The color of a bubble represents its value. The bubbles with a (pulsating) circle around them are clamped.

The game metaphor As the basic gameplay became known, we started designing the game metaphor. We developed a storyline that turns the complex notions of variables and their interdependencies into understandable concepts, allowing the user to relate to the game and understand (part of) its inner workings without having any knowledge about the underlying concepts, models or science. Based on that storyline all visual artwork was created. Initially, we focused on finding a metaphor for the variables: what should a variable 'be' in the game world? It was not until we realized that the relationships between the variables were the abstract notions that were difficult to communicate, that we could find a proper metaphor for the game.

For the game metaphor we decided to represent variables as cities. A city's color reflects the state of the underlying variable. Virtual tunnels between the cities represent the dependence relationships between the variables. As every possible pair of variables in the bayesian network is either d-connected or d-separated, the tunnels between the corresponding cities are either intact (d-connected) or broken (d-separated).

Furthermore, players are allowed to fix one ore more variables at certain values. As we have explained earlier, we refer to this as clamping. A clamped variable is visualized as a city that is flagged (see Figure 11). It was difficult to find a proper way of explaining this rather complicated concept in terms of the game metaphor. Initially, we wanted to explain flagged cities to the player as 'via'-cities, indicating a possible route between the paired variables via the clamped variables. This, however, turned out to be too complex. Therefore, we later simplified this to a more abstract notion of flagging, dropping the 'via'-metaphor completely.

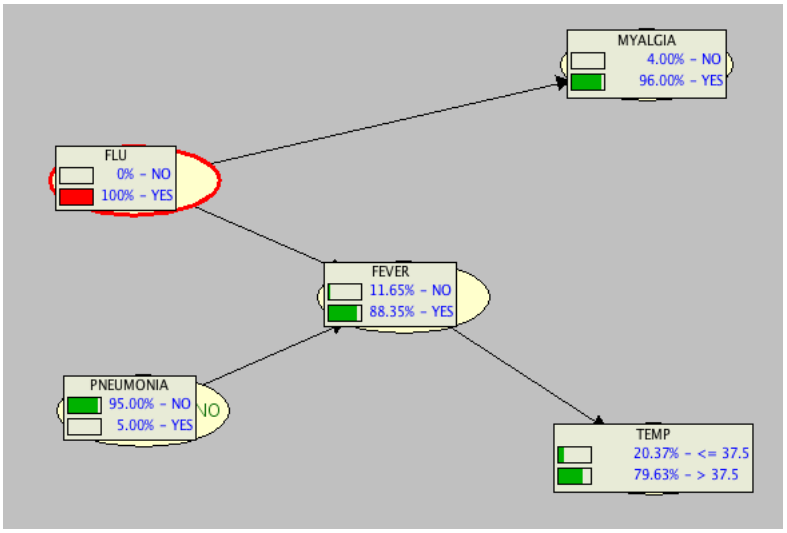


Figure 4: The Bayesian network that produced the cases for our pilot study. The value of FLU is clamped to YES similar to how the value of variables can be clamped in our proof of concept interface.

Presenting observations For each joint observation, the cities are colored according to the state of their underlying variable. At first we randomly chose a color for each state of a variable, but we soon realized that some colors were so similar that they became indistinguishable. To improve the distinguishability of the colors we designed an algorithm to randomly pick colors while maximizing the contrast between the colors. To allow users to inspect the joint observations at their preferred speed, we introduced the ScrollScroll: a paper scroll that sits horizontally at the bottom of the screen and allows users to control the speed at which the joint observations are refreshed. In other words: the ScrollScroll allows players to change the speed with which the variables change their state, and thus how fast the cities change colors (see Figure 11).

Storyline Finally, we wrapped the entire game into a storyline and created a movie telling the storyline with custom visual artwork and music. This movie is freely available on www.ahsumnimity.com. The storyline presented in the movie is as follows:

> Our adventure takes place on the mysterious planet of Dunya, whose inhabitants live in peace and luxury. But this wasn't always the case... Every generation still tells the story of the Nyx: a space-traveling horde of horrible creatures that raided the planet in vast numbers. In utter despair, the people of Dunya called upon a wise sorcerer to help them survive the vicious attacks of the Nyx. The mighty wizard, descendant of the powerful family of Nimity, created a network of magical portals through which the armies of Dunya could travel at near light speeds. Many brave men died, but eventually, the Nyx were defeated... Today, thousands of years later, the

> resources of the planet are depleted and the magical portals have worn out... Even worse: because the planet has weakened, the Nyx are returning! And they're coming in numbers even greater than before... To save their lives, the people of Dunya need the ancient magical portals. But after so much time, nobody knows if they can still be used safely... Most are broken, some are intact: no mere mortal can tell... Yet again, an appeal is made to a descendant of the wizard family: Ahsum Nimity. With all his power, he rips the cities from the ground and into the air, to inspect the magical portals. Can you help him discover which are intact, and which are broken...? Most are broken, some are intact: no mere mortal can tell...

Beta Version The beta version is a full implementation of the game interface, artwork and gameplay. We let several users unknown to the project try out the game to get feedback with regard to possible optimizations. We especially learnt that people loved the storyline and production quality, but didn't understand what they were supposed to do in the game. This made sense, as the task we expected users to perform is unlike any game tasks they were familiar with.

Final Version To tackle the difficulty problem, we introduced seven tutorial levels that gradually explained the interface and concepts of the game. The tutorial levels introduce connecting cities, moving them about on the screen, and clamping them, in levels that only gradually increase in difficulty. Users were not allowed to begin the real game levels without finishing every tutorial level successfully, forcing them to become familiar with the game's concepts and rules before entering the real experiment.

Improvements After releasing the game, we learnt that the game did not succeed in motivating users to finish all tutorial and normal levels, resulting in too little data coming in for the experiment. We therefore introduced the following improvements over three successive (minor) updates:

- Re-balanced target scores: we lowered the target scores for problems so it would be easier to finish the entire game. This lowers the number of decisions gathered per player, but did increase the number of players because more players finished the levels.
- Removed 1 problem for shorter gameplay: we removed one problem from our experimental problems (Flu) in order to decrease the number of experimental conditions, thereby lowering the required number of subjects for a statistically sound analysis.
- More feedback on game progression: we updated the level screen to show all levels, including those that are still locked because earlier levels have to be finished first. This gave players a better overview of the level sequence of the game and their progression in it, hopefully stimulating them to finish the entire game.
- Added Nyx mini-game: to improve the fun factor of the game, we introduced a mini-game where the Nyx (horrible flying creatures from space)

enter the screen from the top and the user has to prevent them from reaching the bottom by tapping the screen to create explosions. By playing the main game, users gather magic by making correct decisions about the connectedness of cities. In the mini-game this magic is used up to create the explosions that defeat the Nyx, thereby increasing the fun factor of the game while hopefully increasing a user's motivation to perform well on the main task.

- Added help screen: although users were required to finish all tutorial levels for training, we noticed they sometimes wanted to look back at the lessons taught by the tutorial levels. Instead of forcing them to re-play the tutorial levels we introduced a help screen that is available from all levels and explains the basic game interface. This allows users to peek at the game's instructions while solving a problem, hopefully increasing a user's ability to solve it. (See Figure 10.)

Player profiles We implemented a profile system, so multiple users can play on the same device without being seen as a single subject. However, we did need to make it very easy for the player to switch profile and/or create a new one while not making it too easy for several users to just always log in to the same profile. To establish this, we present a profile picker to the user every time (s)he starts a new game (see Figure 7). If no profile is constructed yet, the user is immediately presented with a small form (see Figure 8) so a new profile can be created.

Figure 5: The main menu of the game. When a player taps "Play!" (s)he is asked to select a profile (Figure 7) or to fill in some information if no profile has been created yet (Figure 8).

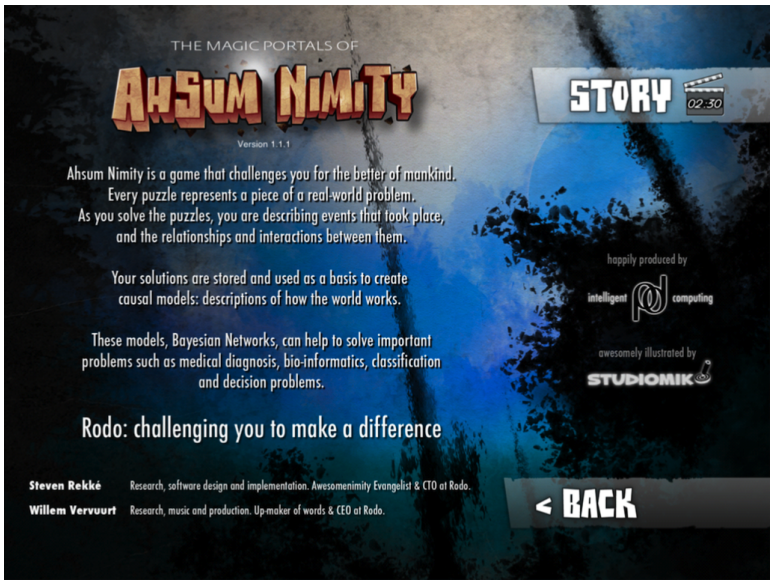


Figure 6: The game's about screen. Although we did not want it to be obvious in the game that it was a research tool and not a game as such, we do give some information about our motives on this screen.

Figure 7: The profile picker. The player is always presented with this picker when (s)he presses the play button, to increase awareness of the profile the player is using.

Figure 8: This is the form players are asked to fill in for a new profile. Only the username is required.

Figure 9: The game's level picker with flags representing the levels and stars showing the performance on each level. The blind level is presented as the "boss level".

Figure 10: The very simple help screen players can access from the game after playing the tutorial levels. Although this gives a simplified view of the task, our hope was that it would nonetheless help players to remember their training.

Figure 11: A level in the game. The player is moving his/her finger across the screen from one city to another, leaving a glowing trail. This is called pairing and means the player will make a dependency statement about the variables represented by these two cities. The flag on one city shows that it is clamped.

Figure 12: The two options presented to the player after pairing two cities. One bottle represents "dependent" while the other represents "independent".

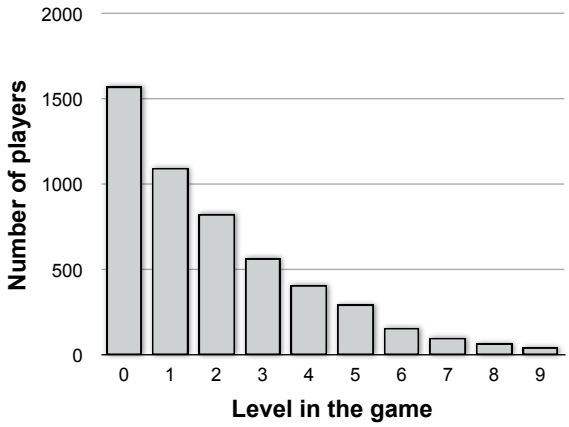


Figure 13: A graph illustrating the player drop-off rate by showing the number of subjects per level.

4 Results

4.1 Game

Over a period of 4 months, the game was played by a total of 1498 users on 1381 unique devices (iPads). These users analyzed the d-connectedness of 29,543 pairs of variables in 5,377 levels, totaling almost 350 hours of recorded gameplay. At the time we started analyzing the results, we had over 1500 subjects that had sent in their data. Figure 13 shows the number of users for each level in the game at the time of analysis. As is apparent from the graph, unfortunately the game suffers from a large drop-off rate. We started out with more than 1500 subjects in the first level, but were left with only about 40 in the final level. The age distribution of the users who chose to enter their age is as shown in Table 3. The gender distribution of all users is as shown in Table 4.

Although we have reached a large audience and collected a lot of data, the results in Figure 13 show that the game was not successful in drawing the users into the game and keeping them there. As these numbers clearly show, we lost 97 out of every 100 players somewhere between the first and the last level. This shows that the task presented to the users is too difficult or that we have not succeeded in making it a fun enough challenge to solve our puzzles, or both.

4.2 Performance (RQ1)

In the following section, we will investigate the performance of the players. This corresponds with research questions RQ1a through RQ1f from Section 3.3. For each research question we will show the results, discuss them and provide a concluding answer to the research questions.

Better than chance The results of this analysis show whether humans, on average, perform better than the random player baseline (RQ1a). If we look at

Age	Percentage
none provided	29.7%
1-10	7.4%
11-20	15.3%
21-30	18.4%
31-40	13.4%
41-50	7.3%
51-60	4.3%
61-70	1.9%
71+	2.3%

Table 3: Self-identified age of players.

Gender	Percentage
none provided	42.7%
man	36.9%
woman	20.4%

Table 4: Self-identified gender of players.

the graph in Figure 14, we can see that the average performance of the human players is higher than the average performance of the random players in all experimental conditions. To see if this difference is indeed reliable (significant), we ran a univariate generalized linear model (GLM) analysis with the proportion of correct decisions as the dependent variable. The independent variables and their corresponding number of subjects are presented in Table 5. The results of this analysis are presented in Table 6. If we look at the results from the analysis, we can see that there is no significant effect of either the number of observations available to the players (Observations) nor the ground truth. There were also no interactions between these factors. As anticipated however, there is a significant effect of playerType on the average proportion of correct decisions. This means that the difference that we observed in Figure 14 is most likely present in the population.

The next analysis concerns research question RQ1b: to see whether the human players systematically outperform random players with different decision schemes. In this analysis we compare human performance to a series of random players, each with their own decision distribution. The graphs in Figure 15 shows the results of this comparison. The bars in the graph indicate the average performance for each player type. As the graph shows, the human players have a higher average performance than all but one decision scheme. Although this involves a significant amount of speculation, the figure does seem to suggest that the human players did not "merely" use a connectivity bias to decide on the dependence relations and that they were able to use some other information present in the observations to judge their dependence. However, even if players are only being efficient at selecting such a decision bias this could provide us valuable information about the ground truth, and thus also about the structure to be learned.

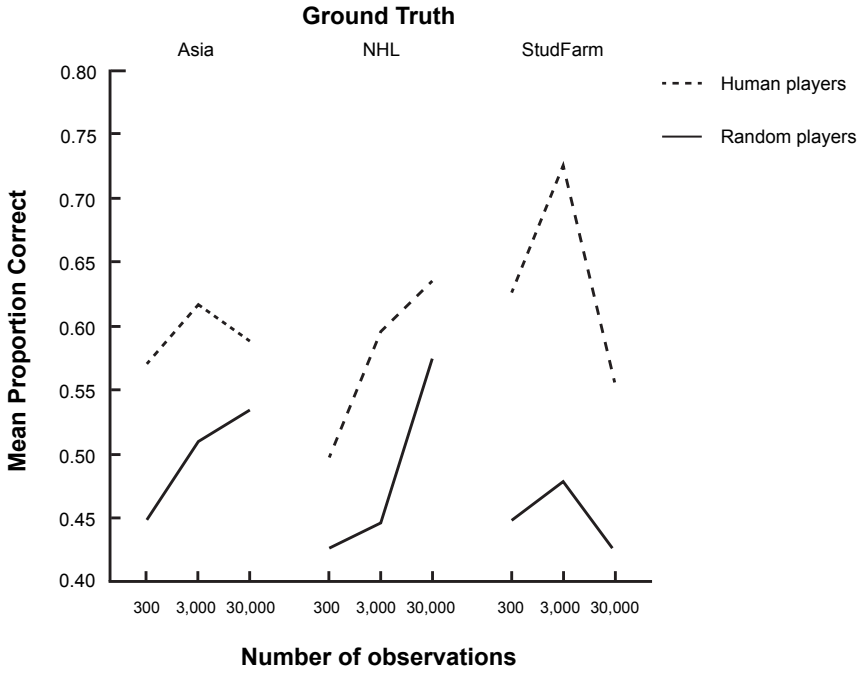


Figure 14: Mean proportion of correct decisions for the human players versus the random players. (RQ1a: Do players perform better than chance?)

Factor	Level	N
Observations	300	52
	3000	30
	30000	42
Ground Truth	Asia	50
	NHL	40
	StudFarm	34
Player Type	Human	62
	Random	62

Table 5: Sample size (N) for the experimental factors. (RQ1a: Do players perform better than chance?).

Factor	p	Effect
Observations	0.13	none ($\eta^2 = 0.04$)
Ground Truth	0.87	none ($\eta^2 = 0.00$)
Player Type	0.00	moderate ($\eta^2 = 0.17$)

Table 6: Results of GLM Univariate, showing a moderate effect of the player type on the mean proportion correct decisions. (RQ1a: Do players perform better than chance?)

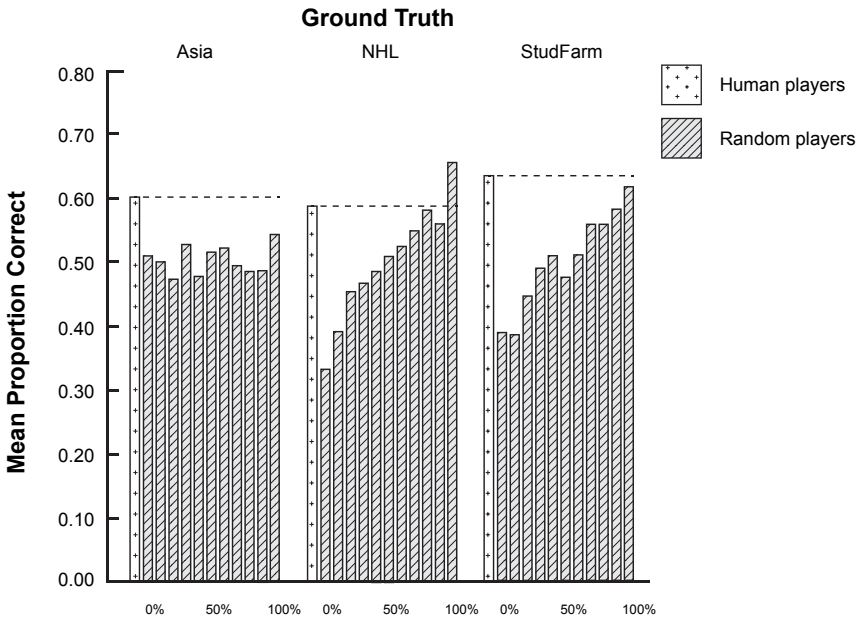


Figure 15: Human players' average performance on each ground truth network compared to several random decision schemes. The three light bars represent average human performance. Each of the other bars corresponds to a random decision scheme. The percentage indicates the decision scheme's bias towards choosing dependence over independence. (RQ1b: Do players perform better than chance with decision schemes?)

Factor	Level	N
Observations	300	26
	3000	15
	30000	21
Ground Truth	Asia	25
	NHL	20
	StudFarm	17

Table 7: Sample size (N) for the experimental factors. (RQ1c: Is the players' performance similar on all ground truth BNs? and RQ1d: Do performances drop when subjects have fewer observations available?)

Factor	p	Effect
Observations	0.26	none ($\eta^2 = 0.05$)
Ground Truth	0.48	none ($\eta^2 = 0.03$)

Table 8: Results of GLM Univariate, showing no significant effects. (RQ1c: Is the players' performance similar on all ground truth BNs? and RQ1d: Do performances drop when subjects have fewer observations available?)

Number of observations and ground truth Now that we have seen that the human players have a higher average performance than the random baseline players, we are interested in the effect of the experimental factors on the average performance of the human players (RQ1c and d). The dotted lines in the graph in Figure 14 shows the average human performance. The results of statistical analysis in Table 8 shows that there is no significant effect of the Bayesian network (Ground Truth) nor of the number of observations (Observations), nor an interaction thereof. As can be seen in Table 7, however, we have a very limited number of subjects in each cell. This means that in further research with higher number of subjects an effect might still be found. If this is not the case, it might indicate that the performance of human players is actually rather stable across the number of observations available and the ground truth.

Direct feedback and learning effect Our next analysis is intended to show whether there is a cross-network learning effect present (RQ1e) and whether direct feedback is necessary for the performance of the players (RQ1f). We will start by looking at the effect of direct feedback. In Figure 16 we have plotted the estimated marginal means of the performance on the levels five through nine of the game. As you may recall, levels seven through nine are the experimental levels, while levels five and six are still tutorial levels. We have included levels five and six anyway, because the gameplay in levels five and six is already representative of the experimental levels and they allow us to have a broader view of the performance differences across levels. As you may also recall, the ground truth networks are presented in a random order, causing levels seven through nine to have several possible ground truth orders. Level nine is always the blind level. To see if not having direct feedback available has affected the average performance, we ran a Repeated Measures GLM in which level 9 was set

Factor	Level	N
Observations	300	11
	3000	8
	30000	13
(Game) Level	5,6,7,8,9	32

Table 9: Sample size (N) for the experimental factors. (RQ1e: Do performances drop when players no longer get direct feedback?)

as the reference category for the within-subject contrasts. Then we compared the within-subject contrasts to see if the average performance on any level was significantly different from those on the blind level.

Table 9 shows the sample size (N) for each of the between-subject factors. N is constant for the within-subject factors as we have only used subjects who completed all the levels. The results from the analysis in Table 10 show that there is no main effect of the level progression and no interaction between level and the number of observations. However, for this analysis we are more interested in the contrasts between level 9 and the other levels to see if there is an effect of the Blind level. As we can see, the only significant contrast was that between levels 6 and 9 and that contrast also showed an interaction with the number of observations. However, if we look at some of the other contrasts and keep in mind that we have a rather small N, it seems likely that more contrasts would have been significant with larger N. It is hard to draw conclusions from these results. But even if we find a significant effect, it might be the case this effect would disappear if subjects had the chance to practice the game without direct feedback. More research is required to provide a conclusive answer.

For the analysis in which we look at whether there is a learning effect (RQ1f), we were able to slightly increase the sample size (see Table 11) by leaving out the blind level (level 9). As we can see in Figure 17 this produces a more "stable" graph. Here we are not particularly interested in the individual within-subject contrasts by themselves, but in whether there is a main effect of Level progression. In other words, we are interested to see whether people gradually become better at playing the game as they progress in the game. Table 12 shows the results of this analysis. It tells us that no main effect of Level progression was found and also no interaction with the number of observations was found. Due to the higher N for the level progression we can be more sure that the effect is indeed absent.

4.3 Usage of tools (RQ2)

In this section we will look at how the players use the tools we have provided. The tools we will investigate are the decision tool (RQ2a), and the clamping tool (RQ2b). We are interested to see whether players show a preference for either dependence or independence and whether they use clamping effectively.

Use of decision tool For this analysis we have computed the proportion of decisions for each player that indicated a dependence and the proportion that indicated an independence. For both of these sets, we have computed the pro-

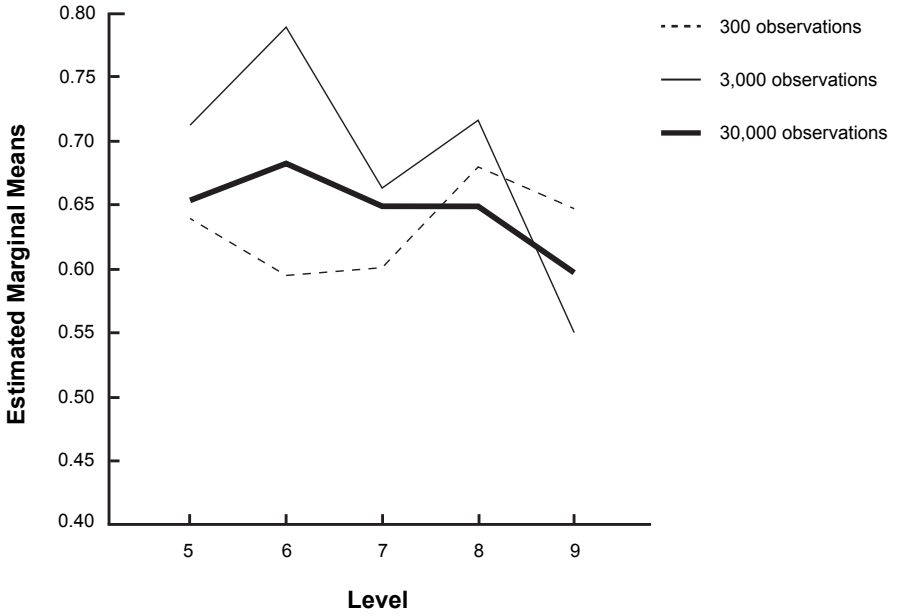


Figure 16: Shows the estimated marginal means of proportion correct for the levels in the game. Level 9 is the blind level. (RQ1e: Do performances drop when players no longer get direct feedback?)

Main Effect	p	Effect Size
Level	0.167	none ($\eta^2 = 0.213$)
Level * Observations	0.368	none ($\eta^2 = 0.147$)
Contrasts Level		
Level 5 and 9	0.083	none ($\eta^2 = 0.100$)
Level 6 and 9	0.030	moderate ($\eta^2 = 0.152$)
Level 7 and 9	0.301	none ($\eta^2 = 0.037$)
Level 8 and 9	0.054	none ($\eta^2 = 0.122$)
Contrasts Level * Observations		
Level 5 and 9	0.258	none ($\eta^2 = 0.089$)
Level 6 and 9	0.028	large ($\eta^2 = 0.218$)
Level 7 and 9	0.255	none ($\eta^2 = 0.090$)
Level 8 and 9	0.427	none ($\eta^2 = 0.057$)

Table 10: Results of GLM Repeated Measures, showing significant effects for one within-subject contrast. (RQ1e: Do performances drop when players no longer get direct feedback?)

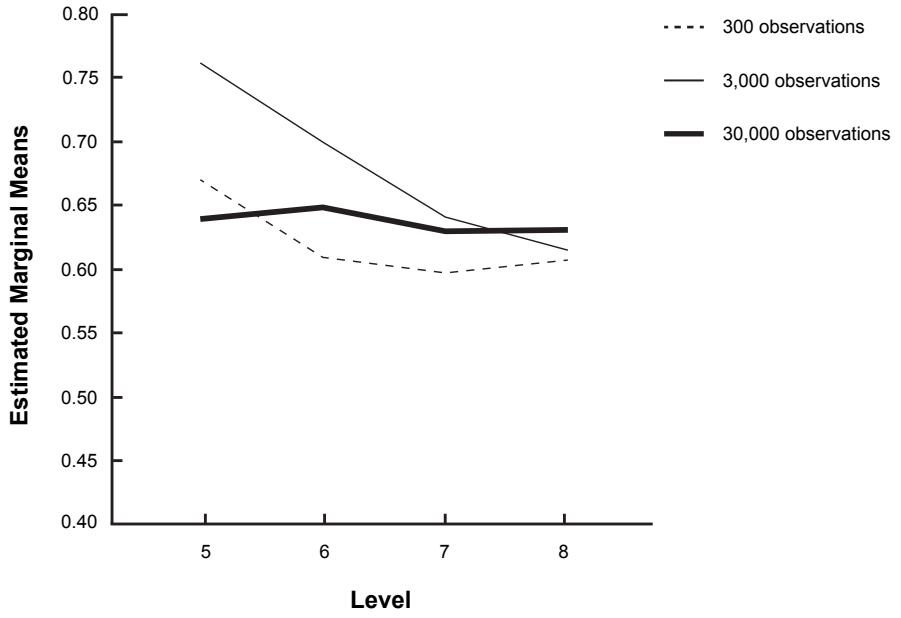


Figure 17: Shows the estimated marginal means of proportion correct for the levels in the game. The blind level is not included. (RQ1f: Is there a cross-network learning effect?)

Factor	Level	N
Observations	300	18
	3000	12
	30000	16
(Game) Level	5,6,7,8	46

Table 11: Sample size (N) for the experimental factors. (RQ1f: Is there a cross-network learning effect?)

Factor	p	Effect
Level	0.141	none ($\eta^2 = 0.123$)
Level * Observations	0.799	none ($\eta^2 = 0.036$)

Table 12: Results of Repeated Measures GLM showing no significant effects. (RQ1f: Is there a cross-network learning effect?)

Measure	N	Mean
Proportion dependence	144	0.61
Proportion independence	144	0.39
Proportion dependence correct	144	0.67
Proportion independence correct	144	0.40

Table 13: Means and sample size (N). (RQ2a: Do players show a preference for dependence vs independence?)

Pair	p
Proportion dependence - proportion independence	0.00
Proportion dependence correct - proportion independence correct	0.00

Table 14: Results of a pairwise t-test, showing that the proportions differ significantly. (RQ2a: Do players show a preference for dependence vs independence?)

portion of correct and incorrect answers. This results in four numbers: proportion dependence statements, proportion independence statements, proportion dependence statements correct and proportion independence statements correct. The means of these are presented in Table 13. By computing a pairwise t-test on these numbers for each subject we can see if these means differ in the population. Table 14 shows the results of this analysis. They tell us that there were significantly more dependency statements made (61% vs 39%) and that the proportion of correct decisions for dependence statements was significantly higher than the proportion of correct decisions for the independence statements (67% vs 40%) (note that the latter do not need to add up to 100%). It seems to us that the players show a preference for specifying dependence over independence as they have provided more of them and are on average better at them.

Use of clamping tool In the next analysis we will be investigating the use of the clamping tool (RQ2b). We want to see whether people use the tool and whether they use it effectively. First we will look at the proportion of decisions of all the players that were made while a variable was being clamped and whether the number of observations or the ground truth has an effect on that proportion. The graph in Figure 18 shows that clamping is indeed being used and that it seems there is an effect of the number of observations. To investigate this, we performed a univariate GLM with the proportion of decisions that were made under a clamp as the dependent variable (proportion clamped). The results of this analysis are shown in Table 16 and the sample size (N) is shown in Table 15. They show us that there is indeed a small effect of the number of observations, but not of the ground truth and there is also no interaction. It is interesting to see that having more observations available seems to lead to more clamping.

Effect of clamping Next, we are going to inspect whether players use the clamping tool effectively. As explained earlier, the effectiveness will be measured in terms of the effect of the clamps on the correctness of the decision. For

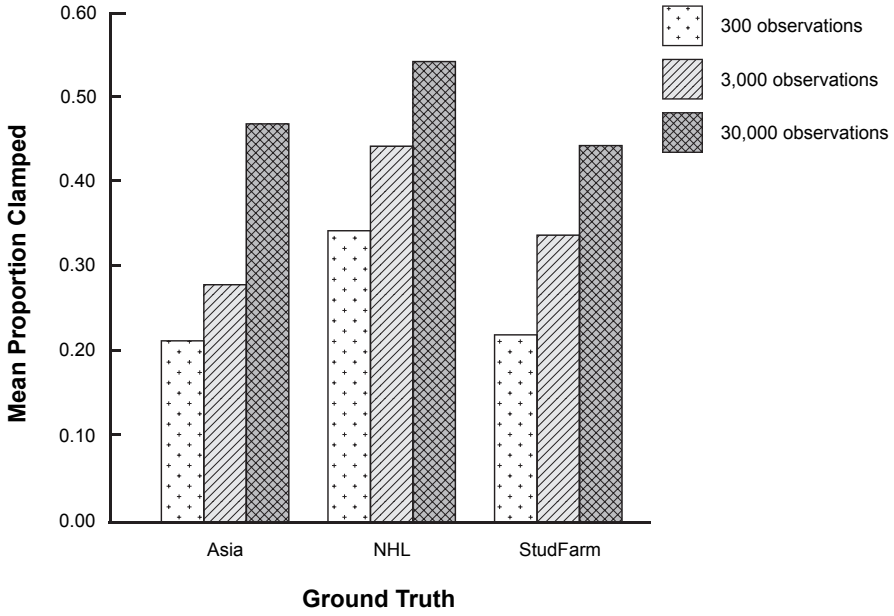


Figure 18: Shows the mean proportion of decisions that were made under clamps for the number of observations and ground truth network. (RQ2b: Do players use clamping effectively?)

Factor	Level	N
Observations	300	46
	3000	28
	30000	38
Ground Truth	Asia	44
	NHL	38
	StudFarm	30

Table 15: Sample size (N) for the experimental factors. (RQ2b: Do players use clamping effectively?)

Factor	p	Effect
Observations	0.044	small ($\eta^2 = 0.059$)
Ground Truth	0.363	none ($\eta^2 = 0.019$)
Ground Truth * Observations	0.996	none ($\eta^2 = 0.002$)

Table 16: Results of GLM Univariate, showing a significant effect of the number of observations on the mean proportion clamped decisions. (RQ2b: Do players use clamping effectively?)

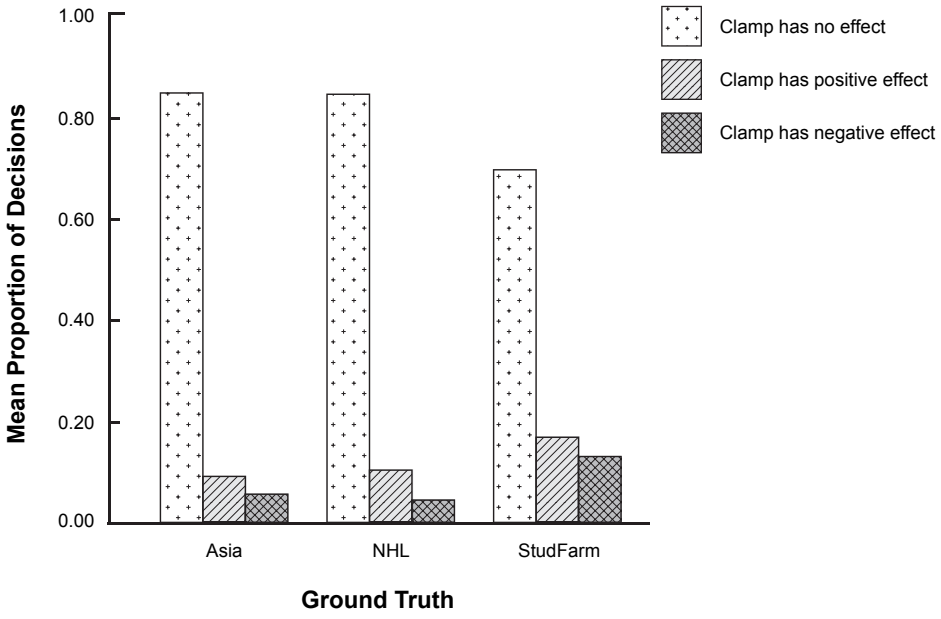


Figure 19: Shows the effect of the clamps on the correctness of decisions that were made under clamps. (RQ2b: Do players use clamping effectively?)

Clamp effect	N	Mean proportion of decisions
Positive clamp effect	871	0.12
Negative clamp effect	871	0.08
No clamp effect	871	0.80

Table 17: Means and sample size (N). (RQ2b: Do players use clamping effectively?)

this analysis we only look at decisions in which one or more variables were clamped. The graph in Figure 19 shows that the vast majority of decisions was neither correct nor incorrect *due* to the clamps. In other words: for these decisions, leaving out the clamps would have had no effect on their correctness. (Although the clamps may still have provided the user with information, as we will discuss in Section 5.1.) Furthermore, we see that the proportion of decisions that was correct due to the clamps does not differ greatly from the proportion of decisions that was incorrect due to the clamps, although the former is larger. A pairwise t-test shows us that all proportions differ significantly. The results of this analysis are shown in Table 18 and the sample size (N) and means are shown in Table 17.

Pairs of proportions of decisions	p
Positive clamp effect - negative clamp effect	0.003
No clamp effect - positive clamp effect	0.000
No clamp effect - negative clamp effect	0.000

Table 18: Results of a pairwise t-test, showing that the proportions differ significantly. (RQ2b: Do players use clamping effectively?)

4.4 Building BN structures (RQ3)

For this analysis, we have developed a voting scheme so we could combine all the decisions of the players. In order to create a Bayesian network structure using the voting scheme (RQ3), we started out with a fully connected undirected graph (see Figure 20a) using all variables that were present in the observations. Then, by using the information from the users we started pruning this structure. Because of the exploratory nature of this analysis we have decided not to consider the clamps and to only use decisions that were not made under clamps. An addition to our voting system will first need to be developed in order to also effectively use the extra information present in decisions under clamps. But for this analysis we ignore all decisions made under clamps because, due to the results for the effect of clamping, we do not think it makes a large difference for the quality of the decisions. Note that this is not to say that clamping was not used at all while making the decisions.

Our voting mechanism includes a parameter N_p that specifies how many people need to have specified the pair to allow a decision on the pair to be used. It also includes a confidence parameter C that specifies what fraction of people need to agree on the decision. Because the results have shown that players were more often correct about dependence relations and they specified more of them we decided to restrict the voting mechanism to dependence statements. This means that if enough decisions were made about a pair, and the ratio of people who agreed on that dependency is larger than the confidence parameter, a direct dependency is assumed to exist. If the ratio of people who agreed on the dependency is smaller than the confidence parameter, the direct dependency is assumed not to exist. If not enough decisions were made about the pair, no action is taken and thus the pair is assumed to be dependent.

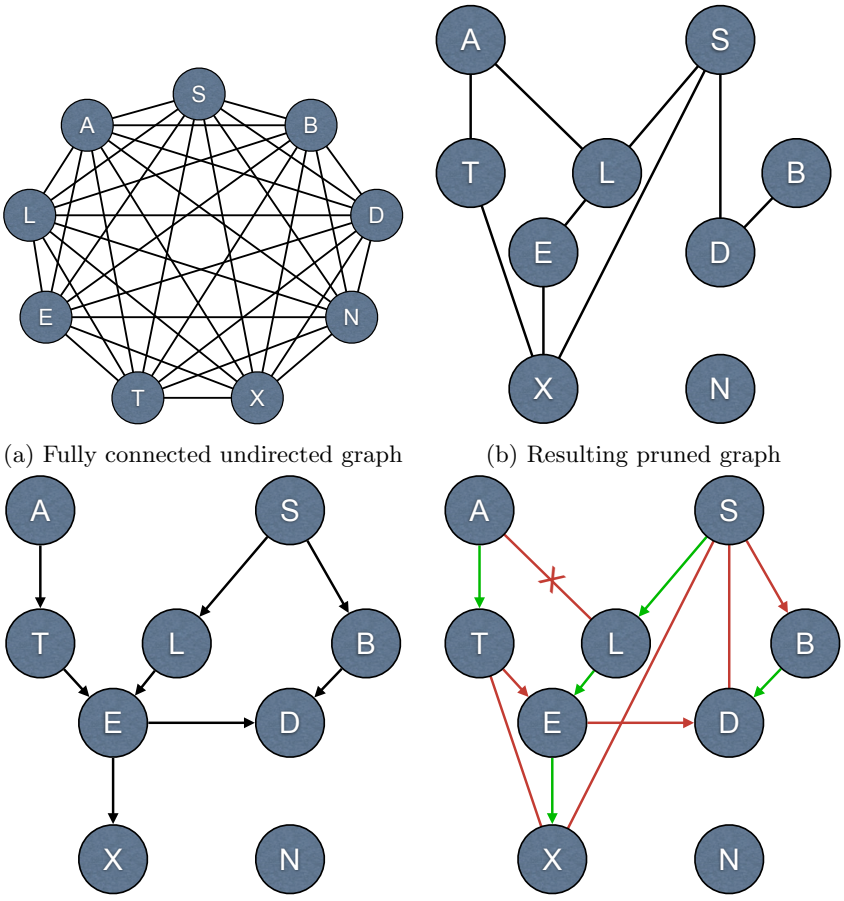
Table 19 shows for each ground truth for different settings of the parameters how many of the aggregated decisions were correctly interpretable as direct dependency or absence of direct dependency*. It also shows how many of the aggregated decisions that were incorrect, were correct in the ground truth if interpreted as presence or absence of indirect dependence**. Finally, it shows which proportion of the aggregated decisions is correct if we interpret them the same way as for a single player (using the Bayes-Ball algorithm to determine whether they are dependent or not)***.

Figure 20 shows an example of this technique applied to the aggregated decisions from the Asia network. For this example we have used parameters $N_p = 18$ and $C = 0.7$ because, as can be seen in Table 19, these resulted in the highest proportion correct decisions. Figure 20a shows the fully connected network we start out with. Figure 20b shows what we are left with after pruning

the connections as indicated by the aggregated decisions. If we compare that to Figure 20c we arrive at Figure 20d, where we can see that there is actually only one real mistake in the network, which is that between nodes A and L. That particular connection is the only one that cannot be explained by the ground truth. The only situation in which A and L would become dependent is if we have prior knowledge about variable E, but since we did not include clamped decisions in this analysis we need to consider this an error. The other undirected red lines in Figure 20d represent direct connections that were specified but not present in the ground truth while directed red lines are direct connections in the ground truth that were not specified by the voting system. The undirected lines are not correct if we interpret them as direct connections, but they are true in the ground truth in terms of dependence. The connection between S and D, for instance, is not a direct connection in the ground truth but they are in fact dependent through their connections with B.

GT	Parameters			Proportion correct		
	N_p	C	Pairs	Direct*	Not direct but correct in GT**	Indirect***
Asia	2	0.5	36	0.47	0.78	0.69
	2	0.7	36	0.78	0.86	0.56
	18	0.7	11	0.82	0.91	0.73
StudFarm	2	0.5	66	0.38	0.67	0.58
	2	0.7	66	0.58	0.74	0.53
	3	0.7	65	0.59	0.74	0.52
NHL	2	0.5	131	0.47	0.66	0.56
	2	0.7	131	0.73	0.82	0.62
	3	0.7	83	0.77	0.84	0.61

Table 19: Shows the parameters for the voting system and the corresponding proportions of correct decisions for each ground truth network. (RQ3: Can we already build BN structures with players' input?)



(a) Fully connected undirected graph

(b) Resulting pruned graph

(c) Real Asia DAG

(d) The resulting graph compared

Figure 20: Creation of an undirected graph for the Asia ground truth from aggregated decisions using the voting system. (RQ3: Can we already build BN structures with players' input?)

5 Discussion

In this section, we will first discuss our main findings, their relevance and their impact, thereby touching upon some open questions and suggestions for future exploration. Then, we will proceed to discuss some lessons for GWAP development that we think are useful to anyone engaged in a similar project. Finally, we will further elaborate the suggestions we made for the direction of future research.

5.1 Main findings, relevance and impact

The results of our research show that the human players are significantly better than can be expected on the basis of chance. This shows that there is indeed information present in the observations that players can utilize to judge the dependence or independence of variables. However, this is a rather weak baseline, so to make a stronger case we have also compared human players with several random decision schemes. The results show that humans outperformed all but one of those decision schemes. This may not immediately seem very interesting, because the performance of some of the decision schemes came close to the average human performance. But one must remember that in real-world scenarios it is impossible to know the right distribution because it is a property of the ground truth and one needs the ground truth to check which performs best. The fact that human players perform better than all but one decision schemes tells us that they are able to do more than just picking a random decision distribution, which is evidence that our approach has potential.

We did not observe a main effect of the number observations available to the subjects. This is interesting because that is one of the most important limits to current structure learning algorithms: they need a lot of observations. If humans perform equally well with smaller sets of observations this could be the "selling argument" to use our GWAP in combination with, or instead of regular structure learning techniques. We also did not observe a main effect of the ground truth. This is important because it suggests a consistent performance over different types of ground truth structures as well as the number of variables involved. However, it is important to note that most of our statistical analyses had a small number of subjects available, so it is recommended not to jump to conclusions. Also, we have only compared a relatively small set of ground truths and also a relatively small difference in number of variables. In future research we would suggest varying the number of variables in the network separately from the connectivity of the networks to be able to observe separate effects.

While one might expect there to be a learning effect across levels, it does not show in our results. However, we only investigated the learning effect across different networks. It might be the case that a clear learning effect would be visible if we had looked at the performance of several attempts at the same network. In that case the direct feedback could play a role in the increase of performance, so this learning effect could best be examined using the blind level. We also did not observe a main effect of not receiving direct feedback. We did see one significant contrast and the small number of subjects in those analyses suggests that more contrasts might have been significant given a larger number of subjects. However, given the small effect and the similar performances in other contrasts we are inclined to think that if we would let people practice without

direct feedback they would indeed be able to achieve similar performance.

According to our results, players specified more dependence than independence relations and they performed better on the former than on the latter. In our voting system, this property has allowed us to some extent to regard any direct connections as absent if players did not agree on them. Our investigation of clamping has shown that clamping is used, but the clamps made while making a decision do not seem essential for the correctness of that decision. We also found that people tend to clamp more when they have more observations available. This suggests that despite clamping not being essential for the correctness in a direct manner, it might have some value to the players nonetheless. It might for instance be the case that clamping is used to filter the observations such that it becomes easier to distinguish patterns. Although the distribution of states of the variables should be roughly the same for the small and large sets of observations, it might be the case that the larger set of observations allows for more and easier pruning by clamping. This might be due to the set of observations becoming too small when clamping in the small set of observations such that it becomes impossible to see any patterns, while when clamping in the large set of observations enough observations remain to be able to see the effect of the clamps. For future research we suggest to investigate whether being able to clamp at all actually affects performance.

To see if we could directly use the information provided by the players to build a Bayesian network structure, we have developed a voting mechanism that allows us to prune fully connected undirected graphs and compare the results to the ground truth. The results of this analysis have shown that even with the limited number of subjects we had available for the experimental levels, the performance of the group can surpass that of each individual. This phenomenon is in line with what is often referred to as the "Wisdom of the Crowd" (Surowiecki, 2005). Using the aggregated decisions of the crowd, we can create reasonably good undirected structures. Although these structures are undirected they can serve as prior knowledge for several structure learning algorithms (SLAs) in the form of structural priors (Shah & Woolf, 2009; Imoto et al., 2003; Langseth & Nielsen, 2003). Prior knowledge can boost an SLA's performance, in particular when there are only few observations available (Langseth & Nielsen, 2003). This leads us to believe that utilizing our GWAP as input for an SLA in situations where there are few observations available, might lead to a better result than possible when using the SLA alone.

Note that the number of players that need to have made a decision about a pair of variables is a parameter in our voting system (N_p). The N_p parameter has an impact on the quality of the generated network structure, which means that our voting system depends on the number of players. It may be suggested that our system is trading a dependency on the number of observations for a dependency on the number of players. Although this cannot be the case entirely (having only one observation available cannot be compensated by having more players), there might be some truth to it if having more observations available improves the result of the GWAP+SLA at a higher rate than the result of the SLA by itself. This could be a favorable development as in some cases it would be easier to obtain players for a game than observations. However, gathering enough players can be difficult and both gathering them and waiting for them to finish the game can be a time consuming process. So whether this development is favorable really depends on how successful the GWAP can become (more

players and shorter game completion times), how hard it is in the particular domain of application to gather more observations and the timespan in which a result must be obtained.

Unfortunately it falls outside the scope of this project, but we highly recommend investigating the potential of this approach in real world applications. In those situations, the ground truth is unknown and generally the number of observations available is small. It is our belief that the GWAP could add value in those situations. As we have discussed before, Bayesian networks have a very broad application domain, ranging from bio-informatics (Zou & Conzen, 2005) and medicine (Long, 1989) to gaming (Becker et al., 2005) and law (Thagard, 2004). They do not only serve practical purposes, like their use in diagnosing disease, but they also provide a framework for understanding human cognition. Models postulating that a part of human cognition performs some type of Bayesian inference have been proposed in various cognitive domains, including vision (Yuille & Kersten, 2006), language (Chater & Manning, 2006) and decision making (Sloman & Hagmayer, 2006). Improving Bayesian network structure learning techniques has a direct impact on all fields in which Bayesian networks have a possible application. (See Section 2.5 for more examples of practical purposes as well as cognitive models.)

5.2 Lessons for GWAP development

Looking back at the development process and feedback of players, we must conclude that although we have reached all of our goals at least partially, we did not succeed in motivating users to play the game as much as we wanted them to. The disappointingly high drop-off rates of players in the game lead to weaker results from our statistical analyses because we did not gather as many players as we expected. In terms of the lessons we learnt, however, the project has been very successful. Although some of the lessons we learnt are specifically related to the crowdsourcing of Bayesian network construction, the problems and challenges we faced taught us interesting lessons for anyone interested in doing scientific research or data collection through games with a purpose. Games with a purpose are a promising new research tool and an interesting multidisciplinary experience. It is our hope that more researchers will focus on this type of research such that the methods may be improved and more applications may be found. In this project we have identified some difficulties that might be useful for anyone engaging in a similar project:

- Between-subject factors (i.e. where groups of subjects are in separate experimental conditions, like the number of observations in our research) can be problematic because you want players to have more or less the same experience.
- Within-subject factors (i.e. where multiple experimental conditions are tested per subject, like the ground truth network in our research) can be problematic because of player drop-off and game progression. Usually you want a game to increase in difficulty so players are drawn in and challenged to complete more levels, which can be a problem for within-subject factors. Furthermore, with within-subject factors you will need a game that will make players come back, because subjects that have not been measured on all levels of the within-subject factors may become useless for the analysis.

- We have experienced it to be rather difficult to come up with fun gameplay that meets the requirements of the scientific purpose of the game. We suspect that many research domains suffer from the problem that no easy mapping is possible from the research domain to the game domain that will still result in fun gameplay.
- In general it is hard to maintain experimental control. Ability to replay levels is expected in most games and usually games can be stopped and started whenever the player wants. There is no guarantee that players play the game the way you intended (e.g. switch players, put the game down for several hours and then continue, etc.).

This leads to some very basic advice you may want to consider:

- Test your principal ideas in a more controlled setting. This way you can have the experimental control you need to prove your technique is viable. After this phase you can improve the quality of the game itself, to ensure enough players will be drawn to the game. This is not without risk though, if the gameplay does not receive early attention it may prove to be difficult to make the game fun.
- Prototype early and involve your users. This may seem rather obvious, but in most academic research the subjects are only involved during the actual empirical studies. It is advised to keep in mind that the success of the GWAP depends on the users. Including your target audience early also helps preventing you from assuming that what you think is fun is also fun for your target audience.
- Depending on your experimental design: focus heavily on game-play. The larger your experimental design, the more subjects you will need and the more fun the game needs to be. (Especially if you have within-subject factors.)

5.3 Open questions and future directions

In our discussion above we have indicated several options for future exploration and several findings that require verification using a larger group of subjects. Here we will summarize those options and describe how we think in the future our GWAP could be used as the basis for a Bayesian network structure learning system (BNSLS).

In our studies we looked at the effect of clamps on the correctness of players' decisions. We came to the conclusion that clamps were not *directly* essential for the correctness of their decisions, but we believe that players might still use clamping in some way. We suggest investigating this further by introducing a between-subject experimental factor for having the ability to clamp. This way the group of players that can clamp can be compared to the group that cannot clamp to see if the ability to clamp has an effect on average performance. We also looked at a number of different ground truth networks, but we did so in a relatively uncontrolled manner. We used only three different networks in which multiple aspects were varied at the same time, such as the connectivity, the number of variables and the number of states per variable. As discussed in Section 3.4, we chose these networks because of their variation on those aspects,

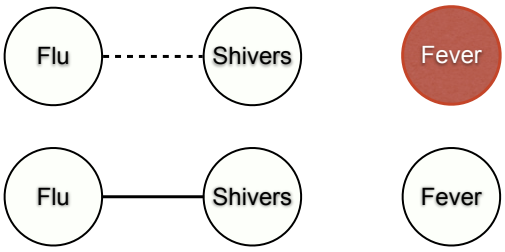
but for future investigation we recommend varying these aspects independently and using a larger set of networks. To get a better idea for the players' ability to perform without direct feedback, and to maximize their performance on levels without direct feedback, we suggest to include multiple "blind" levels in the training phase of the GWAP. Furthermore, we suggest that our findings concerning the absence of an effect for the ground truth and the number of observations be repeated with a higher number of subjects.

We would also like to point out that it might be possible to greatly improve the voting system by developing a way to incorporate decisions made under clamps. We illustrate this by means of the example in Figure 21, which shows how input from the users about the same variables under different clamps can be used to conclude which structures between these variables are plausible given the input. Furthermore, it might be possible to improve our system by further investigating the potential of the crowd. To harness the wisdom of the crowd in difficult decision problems, Zhang and Lee (2010) suggest that one should aggregate people's knowledge rather than aggregate their behavior directly (like we do in our voting system). In order to do this, Zhang and Lee suggest that one would need models of cognition that account for how latent knowledge manifests itself as observed behavior within the constraints of the task. Using these models, we could infer the players' knowledge from their behavior and use it for aggregation in our decision system, thereby potentially improving its performance. Future research could be aimed at investigating and developing models of cognition for that purpose.

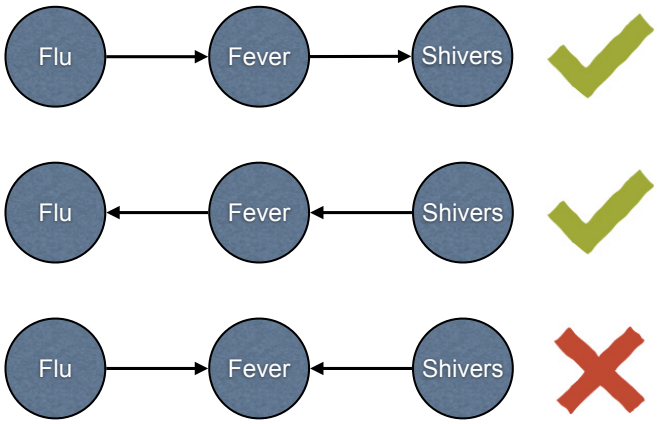
For the research suggested above to succeed it seems necessary to improve the GWAP so more players will be compelled to finish the game. An alternative is to conduct the research in a more laboratory-like setting so the players all finish the entire game. We envision that one day our GWAP or a derivative version could be at the core of a Bayesian network Structure Learning System (BNSLS) as we illustrate in Figure 22 and we encourage all research in that direction. The basic idea behind the BNSLS is that the output of a GWAP could serve as the input for a structure learning algorithm (SLA) so that the SLA can come up with good results even though only a relatively small number of observations is available. But for the game to be effective in such a system it is going to need more players and, as discussed before, we think it is necessary to make the game more fun in order to achieve that.

5.4 Conclusion

In the present research we have investigated the possibilities of developing a game with a purpose for crowdsourcing Bayesian network structure learning from joint observations. We have successfully developed a game that allows players to inspect joint observations generated from a ground truth Bayesian network and indicate which variables they think are dependent or independent. Even though the game suffered from a large drop-off rate, it was possible to collect sufficient data points for a first exploratory analysis and assessment of the promise of our approach. We investigated the players' performance by comparing their decisions to the ground truth networks. We have found that players perform significantly better than can be expected on the basis of chance. Interestingly, we did not observe any main effects of the number of observations the players had available, nor of the ground truth from which those observations



(a) Input from users who specified an independency between the variables Flu and Shivers when the variable Fever is clamped and a dependency between Flu and Shivers when Fever is not clamped.



(b) The conclusions drawn by the system about which structures are plausible given the input.

Figure 21: An illustration of how the voting system could use decisions under clamps to identify plausible structures.

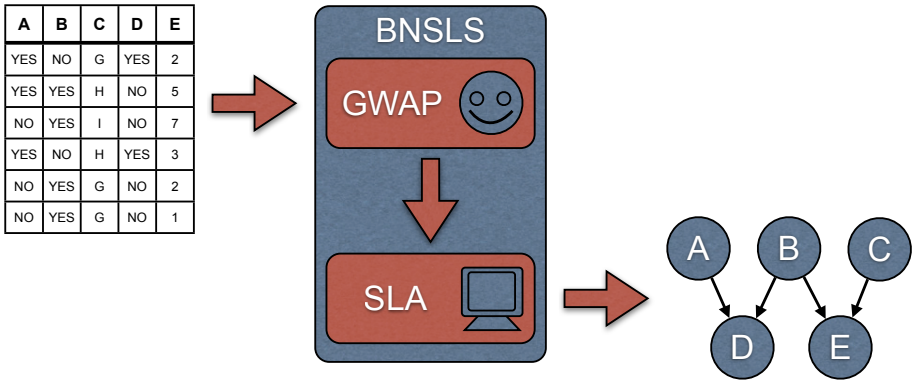


Figure 22: Schematic of a Bayesian network Structure Learning System (BNSLS) including a Game With A Purpose (GWAP) and a Structure Learning Algorithm (SLA).

were generated. Also no learning effect was observed across the levels of the game. According to our results, players specified more dependence than independence relations and they performed better on the former. The number of observations available to players had an effect on the number of decisions they made under clamps, but the clamps did not seem essential for the correctness of their decisions. After investigating the performances of individual players, we developed a voting system to aggregate all player's decisions and used that information to build undirected structures. The results showed that the aggregated decisions of the crowd reached a higher performance than the average performance of the individual players. The resulting structure contained relatively few errors when compared to the ground truth.

This project has made the first steps towards applying human-based computation through games on the structure learning problem of Bayesian networks. We have shown that such an approach does indeed have potential and we have indicated several possible directions for future research. Based on the results we have suggested that the output of our GWAP could be used as prior knowledge for existing structure learning algorithms, thereby reducing their dependence on very large sets of observations. We have also pointed out several challenges that arise in game development with a scientific purpose. We hope that these first steps towards a GWAP for Bayesian network structure learning inspire more research in this direction and that of GWAPs in general.

6 References

Abt, C. (1970). *Serious games.* The Viking Press, 625 Madison Avenue, New York, NY 10022.

ACM. (n.d.-a). *ACM Doctoral Dissertation Award 2011 Website.* `http://awards.acm.org/homepage.cfm?awd=146`.

ACM. (n.d.-b). *ACM Turing Award 2011 Website.* `http://amturing.acm.org/award_winners/pearl_2658896.cfm`.

Baker, C., Saxe, R., & Tenenbaum, J. (2009). Action understanding as inverse planning. *Cognition*, *113*(3), 329–349.

Becker, C., Nakasone, A., Prendinger, H., Ishizuka, M., & Wachsmuth, I. (2005). Physiologically interactive gaming with the 3d agent max. In *International workshop on conversational informatics, in conj. with jsai* (Vol. 5, pp. 37–42).

bnlearn for R Website. (n.d.). `http://www.bnlearn.com`.

BNT for Matlab Website. (n.d.). `http://code.google.com/p/bnt`.

Caldwell, C., & Johnston, V. (1991). Tracking a criminal suspect through "face-space" with a genetic algorithm. In *Proceedings of the fourth international conference on genetic algorithm* (p. 416-421). Morgan Kaufmann Publisher.

Chamberlain, J., Poesio, M., & Kruschwitz, U. (2008). Phrase detectives: A web-based collaborative annotation game. In *Proceedings of the international conference on semantic systems (i-semantics' 08), graz.* Graz.

Charniak, E. (1991). Bayesian Networks without Tears. *AI magazine*, *12*(4), 50.

Chater, N., & Manning, C. (2006). Probabilistic models of language processing and acquisition. *Trends in Cognitive Sciences*, *10*(7), 335–344.

Chater, N., Tenenbaum, J., & Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, *10*(7), 287–291.

Chickering, D. (1996). Learning Bayesian networks is NP-complete. *Learning from data: Artificial intelligence and statistics v*, *112*, 121–130.

Cooper, S. (2012). *A framework for scientific discovery through video games.* Unpublished doctoral dissertation, University of Washington.

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., et al. (2010, 08 05). Predicting protein structures with a multiplayer online game. *Nature*, *466*(7307), 756–760. Available from `http://dx.doi.org/10.1038/nature09304`

Cuijpers, R., Schie, H., Koppen, M., Erlhagen, W., & Bekkering, H. (2006). Goals and means in action observation: A computational approach. *Neural Networks*, *19*(3), 311–322.

Dawkins, R. (1986). *The blind watchmaker*. Longman.

Downey, R., & Fellows, M. (1999). *Parameterized complexity* (Vol. 3). springer New York.

Dugan, C., Muller, M., Millen, D., Geyer, W., Brownholtz, B., & Moore, M. (2007). The dogear game: a social bookmark recommender system. In *Proceedings of the 2007 international acm conference on supporting group work* (pp. 387–390).

Engbert, R., & Krügel, A. (2010). Readers use bayesian estimation for eye movement control. *Psychological Science*, *21*(3), 366–371.

Estellés-Arolas, E., & Guevara, F. González-Ladrón-de. (2012). Towards an integrated crowdsourcing definition.

EteRNA Website. (n.d.). `http://eterna.cmu.edu`.

EyeWire Website. (n.d.). `http://eyewire.org`.

Foldit Website. (n.d.). `http://fold.it`.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of computational biology*, *7*(3-4), 601–620.

Fung, R., & Del Favero, B. (1995). Applying bayesian networks to information retrieval. *Communications of the ACM*, *38*(3), 42–ff.

Griffiths, T., & Tenenbaum, J. (2006). Optimal predictions in everyday cognition. *Psychological Science*, *17*(9), 767.

Haddawy, P. (1999). An overview of some recent developments in Bayesian problem-solving techniques. *AI Magazine*, *20*(2), 11.

Heckerman, D., Mamdani, A., & Wellman, M. (1995). Real-world applications of Bayesian networks. *Communications of the ACM*, *38*(3), 24–26.

Horvitz, E., & Barry, M. (1995). Display of information for time-critical decision making. In *Proceedings of the eleventh conference on uncertainty in artificial intelligence* (pp. 296–305).

Hugin Samples Website. (n.d.). `http://www.hugin.com/developer/samples`.

Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., & Miyano, S. (2003). Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. In *Bioinformatics conference, 2003. csb 2003. proceedings of the 2003 ieee* (pp. 104–113).

Jr, C. E. K., Roberts, L. M., Shaffer, K. A., & Haddawy, P. (1997). Construction of a bayesian network for mammographic diagnosis of breast cancer. *Computers in Biology and Medicine*, *27*(1), 19 - 29.

Kahneman, D., & Tversky, A. (1972). Subjective probability: A judgment of representativeness. *Cognitive psychology*, *3*(3), 430–454.

Kawrykow, A., Roumanis, G., Kam, A., Kwak, D., Leung, C., Wu, C., et al. (2012, 03). Phylo: A citizen science approach for improving multiple sequence alignment. *PLoS ONE*, *7*(3), e31362. Available from `http://dx.doi.org/10.1371%2Fjournal.pone.0031362`

Kersten, D., Mamassian, P., & Yuille, A. (2004). Object perception as bayesian inference. *Annu. Rev. Psychol.*, *55*, 271–304.

Khatib, F., Cooper, S., Tyka, M., Xu, K., Makedon, I., Popović, Z., et al. (2011). Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences*, *108*(47), 18949–18953.

Khatib, F., DiMaio, F., Cooper, S., Kazmierczyk, M., Gilski, M., Krzywda, S., et al. (2011). Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature structural & molecular biology*, *18*(10), 1175–1177.

Korb, K., & Nicholson, A. (2004). *Bayesian artificial intelligence*. Chapman & Hall/CRC.

Krause, M., Takhtamysheva, A., Wittstock, M., & Malaka, R. (2010). Frontiers of a paradigm: exploring human computation with digital games. In *Proceedings of the acm sigkdd workshop on human computation* (pp. 22–25).

Kwisthout, J., Wareham, T., & van Rooij, I. (2011). Bayesian intractability is not an ailment that approximation can cure. *Cognitive Science*.

Langseth, H., & Nielsen, T. D. (2003, December). Fusion of domain knowledge with data for structural learning in object oriented domains. *J. Mach. Learn. Res.*, *4*, 339–368. Available from `http://dl.acm.org/citation.cfm?id=945365.945386`

Law, E., & von Ahn, L. (2009). Input-agreement: a new mechanism for collecting data using human computation games. In *Proceedings of the 27th international conference on human factors in computing systems* (pp. 1197–1206).

Lisboa, P., Wong, H., Harris, P., & Swindell, R. (2003). A bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer. *Artificial Intelligence in Medicine*, *28*(1), 1 - 25.

Long, W. (1989). Medical diagnosis using a probabilistic causal network. *Applied Artificial Intelligence*, *3*(2-3), 367-383. Available from `http://www.tandfonline.com/doi/abs/10.1080/08839518908949932`

Lucas, P., Boot, H., Taal, B., et al. (1998). Computer-based decision support in the management of primary gastric non-hodgkin lymphoma. *Methods of Information in Medicine-Methodik der Information in der Medizin*, *37*(3), 206–219.

Luttrell, S. (1994). Partitioned mixture distribution: an adaptive bayesian network for low-level image processing. In *Vision, image and signal processing, iee proceedings-* (Vol. 141, pp. 251–260).

Michael, D., & Chen, S. (2005). *Serious games: Games that educate, train, and inform.* Muska & Lipman/Premier-Trade.

Milho, I., Fred, A., Albano, J., Baptista, N., & Sena, P. (2000). A user-friendly development tool for medical diagnosis based on bayesian networks. In *Proceedings of the iceis* (pp. 176–180).

Nikovski, D. (2000). Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics. *Knowledge and Data Engineering, IEEE Transactions on*, *12*(4), 509–516.

OnToGalaxy Website. (n.d.). `http://dm.tzi.de/en/ontogalaxy`.

Pang, B., Zhang, D., Li, N., & Wang, K. (2004, oct.). Computerized tongue diagnosis based on bayesian networks. *Biomedical Engineering, IEEE Transactions on*, *51*(10), 1803 -1810.

Pearl, J. (1982). Reverend bayes on inference engines: A distributed hierarchical approach. In *Aaai-82.*

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann.

Pearl, J. (2000). *Causality: models, reasoning, and inference.* Cambridge University Press.

Pernkopf, F. (2004). Detection of surface defects on raw steel blocks using bayesian network classifiers. *Pattern Analysis & Applications*, *7*(3), 333–342.

Phrase Detectives Website. (n.d.). `http://www.phrasedetectives.org`.

Phylo Website. (n.d.). `http://phylo.cs.mcgill.ca`.

reCAPTCHA *Website.* (n.d.). `http://www.recaptcha.net`.

Robinson, R. (1977). Counting unlabeled acyclic digraphs. *Combinatorial mathematics V*, 28–43.

SamIam Website. (n.d.). `http://reasoning.cs.ucla.edu/samiam`.

Shachter, R. (1998). Bayes-ball: Rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In *Proceedings of the fourteenth conference on uncertainty in artificial intelligence* (pp. 480–487).

Shah, A., & Woolf, P. (2009). Python Environment for Bayesian Learning: Inferring the Structure of Bayesian Networks from Knowledge and Data. *The Journal of Machine Learning Research*, *10*, 159–162.

Sims, K. (1991). Artificial evolution for computer graphics. *Computer Graphics*, *25*(4), 319-328.

Siorpaes, K., & Hepp, M. (2008). Games with a Purpose for the Semantic Web. *IEEE Intelligent Systems*, *23*(3), 50–60.

Sloman, S. (2005). *Causal models: How people think about the world and its alternatives*. Oxford University Press, USA.

Sloman, S., & Hagmayer, Y. (2006). The causal psycho-logic of choice. *Trends in Cognitive Sciences*, *10*(9), 407–412.

Steyvers, M., Tenenbaum, J., Wagenmakers, E., & Blum, B. (2003). Inferring causal networks from observations and interventions. *Cognitive Science*, *27*(3), 453–489.

Stone, R. (2009). Serious games: virtual reality's second coming? *Virtual Reality*, *13*(1), 1–2.

Surowiecki, J. (2005). *The wisdom of crowds*. Anchor.

Ted conversations. (n.d.). `http://www.ted.com/conversations/44/we_spend_3_billion_hours_a_wee.html`.

Thagard, P. (2004). Causal inference in legal decision making: Explanatory coherence vs. bayesian networks. *Applied Artificial Intelligence*, *18*(3-4), 231–249.

von Ahn, L. (2007). Human computation. In *Proceedings of the 4th international conference on knowledge capture* (p. 6).

von Ahn, L., & Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 319–326). Association for Computing Machinery, New York.

Website of several GWAPs. (n.d.). `http://www.gwap.com`.

Wolpert, D., & Ghahramani, Z. (2005). Bayes rule in perception, action and cognition.

Xiang, Y., Pant, B., Eisen, A., Beddoes, M., & Poole, D. (1993). Multiply sectioned bayesian networks for neuromuscular diagnosis. *Artificial Intelligence in Medicine*, *5*(4), 293 - 314. (Probabilistic and Decision-Theoretic Systems in Medicine)

Yuille, A., & Kersten, D. (2006). Vision as bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, *10*(7), 301–308.

Zhang, S., & Lee, M. (2010). Cognitive models and the wisdom of crowds: A case study using the bandit problem. *Ratio*, *12*(13), 14.

Zou, M., & Conzen, S. (2005). A new dynamic bayesian network (dbn) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, *21*(1), 71–79.

7 Appendix

In this Appendix, we will give some additional information about the Bayesian networks and software used in the present project.

7.1 Bayesian networks

Here we show the structure of the Bayesian networks used in our GWAP, as well as provide URLs pointing to where the networks can be found online. The networks can be downloaded as Hugin Network (*.net) files and can be imported by SamIam, which will be described in the next section of this Appendix. Note that the parameters of the networks are not shown here, but they can be found online at the provided URLs because the Hugin Network files contain both the structure and the parameters.

- Flu (used for our prototype) - `http://cs.ru.nl/~marinav/Teaching/BDMinAI/networks/flu.net`
- Asia (Chest Clinic) - The smallest network of our ground truth factor. `http://cs.ru.nl/~marinav/Teaching/BDMinAI/networks/chest_clinic.net` or `http://download.hugin.com/webdocs/samples/asia.net`
- Stud Farm - `http://download.hugin.com/webdocs/samples/studfarm.net`
- NHL (Non-Hodgkin Lymphoma) - `http://cs.ru.nl/~marinav/Teaching/BDMinAI/networks/nhl.net`

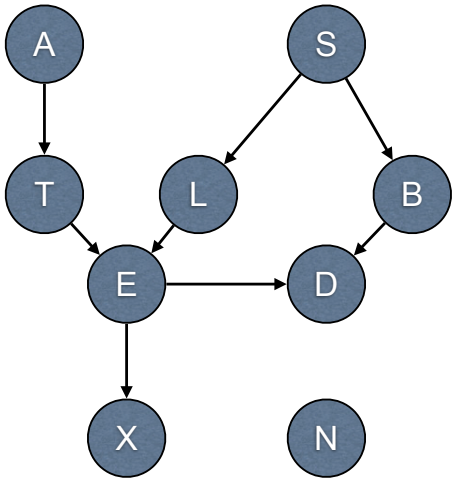


Figure 23: Schematic of the structure of the Asia Bayesian network.

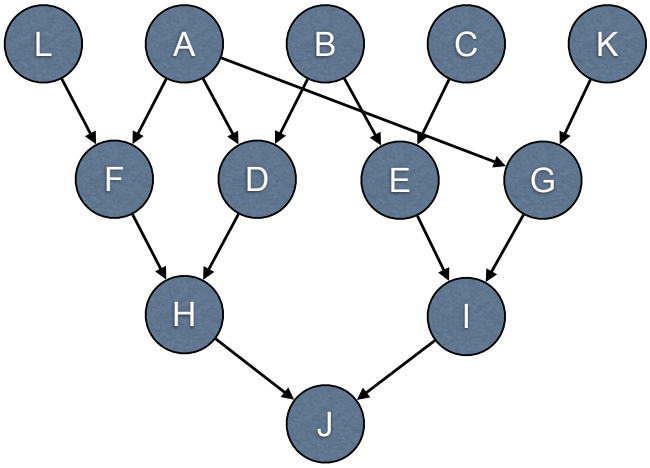


Figure 24: Schematic of the structure of the StudFarm Bayesian network.

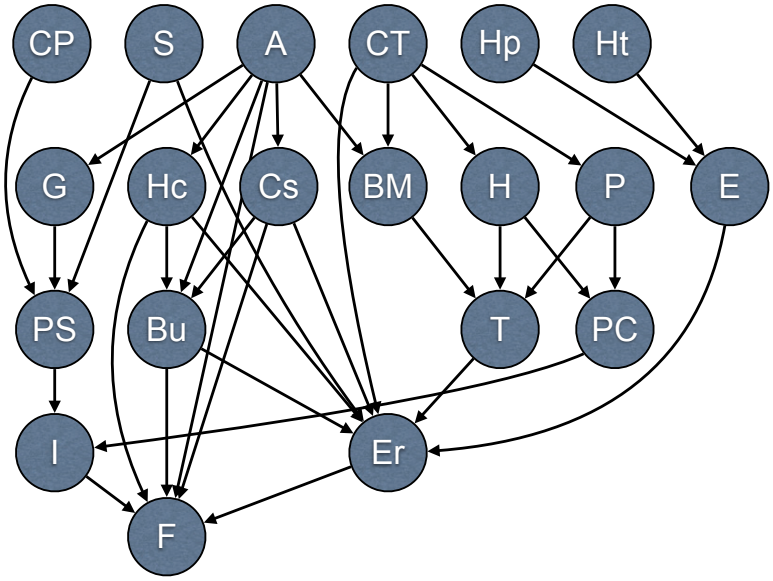


Figure 25: Schematic of the structure of the NHL Bayesian network.

7.2 Software used

Here we will briefly describe the software we used for this project and where more information can be found.

- SamIam - `http://reasoning.cs.ucla.edu/samiam`
  SamIam is a tool for modeling and reasoning with Bayesian networks, developed by the Automated Reasoning Group of Prof. Adnan Darwiche at the University of California Los Angeles (UCLA). SamIam is written in the Java programming language. We used SamIam to load the Bayesian networks described used for this research and to generate the case files which were used for the observations in our game.

- Apple iOS SDK - `http://developer.apple.com/devcenter/ios`
  We used the Apple iOS Software Development Kit (SDK) to develop our application for the Apple iPad. Development using the iOS SDK allowed us to distribute the game via the Apple AppStore.

- Cocos2d for iPhone - `http://www.cocos2d-iphone.org`
  Cocos2d for iPhone (and other iOS devices) is an open source framework for building 2D games and other graphical/interactive applications. Cocos2d for iPhone is written in the Objective C programming language.

7.3 Bayes-Ball implementation

Here we will present our Objective C implementation of the Bayes-Ball algorithm. Our implementation makes use of some utility classes we have developed for reading graphs from Hugin network files. Those classes are listed here as well. Please note that some inconveniently placed line and page breaks may be present in the code. We are happy to share some of our ideas and experiences if it could aid in your research and we are always interested to hear from you if you have made good use of our software. So if you would like to exchange knowledge and/or experiences, please feel free to send us an e-mail at: steven (at) rekke (dot) net -or- info (at) rodo (dot) nl.

Listing 1: GraphScoracle.h

```
//
//  GraphScoracle.h
//  Ahsum Nimity
//
//  Created by Steven Rekké on 18-04-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import <Foundation/Foundation.h>

@class P1Graph;
@class Coupling;

@interface GraphScoracle : NSObject {
@private
    P1Graph* groundTruth;
    NSMutableSet* schedule;
}
```

```
//Init with a path to a graph. (Hugin .net file.)
- (id)initWithGraphPath: (NSString*) path;

//Compute the score of a Coupling. Currently correct is 1.0, incorrect is 0.0.
- (float) computeScore:(Coupling *)coupling withIgnoreSelfClamps: (BOOL) ignore;

@property(readonly) P1Graph* groundTruth;

@end
```

Listing 2: GraphScoracle.m

```
//
//  GraphScoracle.m
//  Ahsum Nimity
//
//  Created by Steven Rekké on 18-04-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import "GraphScoracle.h"
#import "P1Graph.h"
#import "P1Vertex.h"
#import "Coupling.h"

typedef enum {
    fromChild,
    fromParent,
    fromBoth
} VisitType;

@implementation GraphScoracle

@synthesize groundTruth;

- (id)initWithGraphPath: (NSString*) path {
    self = [super init];
    if (self) {
        groundTruth = [[P1Graph alloc] initWithFilePath:path];
        schedule = [[NSMutableSet alloc] initWithCapacity:[groundTruth
    getNumberOfVertices]];
    }

    return self;
}

- (void)dealloc {
    [schedule release];
    [groundTruth release];
    [super dealloc];
}

- (NSSet*) getClampedVertices: (Coupling*) coupling withIgnoreSelfClamps:(BOOL)
    ignore {
    NSMutableSet* clampedVertices = [[[NSMutableSet alloc] initWithCapacity:[
    groundTruth getNumberOfVertices]] autorelease];

    for (NSString* name in coupling.clampedVariableIDs){
        if (ignore && ([name isEqualToString:coupling.variableID1] || [name
    isEqualToString:coupling.variableID2])){
            //Do not add this one
            NSLog(@"Ignoring self-clamp: %@", name);
```

```
        } else {
            [clampedVertices addObject:[groundTruth getVertexByName:name]];
        }
    }

    return [NSSet setWithSet:clampedVertices];
}

- (NSSet*) getConnectingVertices: (Coupling*) coupling {
    return [NSSet setWithObjects:[groundTruth getVertexByName:coupling.
     variableID1], [groundTruth getVertexByName:coupling.variableID2], nil];
}

- (void) scheduleVertices: (NSArray*) vertices withVisitType: (VisitType) type {
    for (P1Vertex* vertex in vertices){
        switch (type){
            case fromParent:
                vertex.visitFromParent = YES;
                break;
            case fromChild:
                vertex.visitFromChild = YES;
                break;
            case fromBoth:
                vertex.visitFromParent = YES;
                vertex.visitFromChild = YES;
                break;
            default:
                NSLog(@"ERROR: unknown VisitType in GraphScoracle");
                break;
        }
    }
}

//Returns whether the path is blocked
- (BOOL) isPathBlockedForCoupling: (Coupling*) coupling withIgnoreSelfClamps: (
     BOOL) ignore {
    NSSet* K = [self getClampedVertices:coupling withIgnoreSelfClamps:ignore];
    NSSet* J = [NSSet setWithObject:[groundTruth getVertexByName:coupling.
     variableID1]];
    NSSet* F = [NSSet set]; //Empty set. No deterministic nodes.

    //1.
    [groundTruth resetVertices];
    //2.
    [schedule removeAllObjects];
    [self scheduleVertices:[J allObjects] withVisitType:fromChild];
    [schedule addObjectsFromArray:[J allObjects]];
    //3.
    while ([schedule count] > 0) {
        P1Vertex* j = [schedule anyObject];                          //(a)
        [schedule removeObject:j];
        j.isVisited = YES;                                           //(b)
        if (![K containsObject:j] && j.visitFromChild){       //(c)
            if (!j.isMarkedTop){                                     //    i
                j.isMarkedTop = YES;
                [self scheduleVertices:[j.parents allObjects] withVisitType:
     fromChild];
                [schedule addObjectsFromArray:[j.parents allObjects]];
            }
            if (![F containsObject:j] && !j.isMarkedBottom) {          //    ii
                j.isMarkedBottom = YES;
```

```
                [self scheduleVertices:[j.children allObjects] withVisitType:
    fromParent];
                [schedule addObjectsFromArray:[j.children allObjects]];
            }
        }
        if (j.visitFromParent){                                 //(d)
            if ([K containsObject:j] && !j.isMarkedTop){              //    i
                j.isMarkedTop = YES;
                [self scheduleVertices:[j.parents allObjects] withVisitType:
    fromChild];
                [schedule addObjectsFromArray:[j.parents allObjects]];
            } else if (![K containsObject:j] && !j.isMarkedBottom){   //    ii
                j.isMarkedBottom = YES;
                [self scheduleVertices:[j.children allObjects] withVisitType:
    fromParent];
                [schedule addObjectsFromArray:[j.children allObjects]];
            }
        }
    }

    NSSet* irrelevantVertices = [groundTruth getVerticesNotMarkedOnBottom];
    if ([irrelevantVertices containsObject:[groundTruth getVertexByName:coupling.
     variableID2]]){
        return YES;
    } else {
        return NO;
    }
}

//Bayes-Ball:

//1. Initialize all nodes as neither visited, nor marked on the top, nor marked
     on the bottom.
//2. Create a schedule of nodes to be visited, initialized with each node in J to
      be visited as if from one of its children.
//3. While there are still nodes scheduled to be visited:
//  (a) Pick any node j scheduled to be visited and remove it from the schedule.
     Either j was scheduled for a visit from a parent, a visit from a child, or
     both.
//  (b) Mark j as visited.
//  (c) If j / K and the visit to j is from a child:
//      i. if the top of j is not marked, then mark its top and schedule each of
     its parents to be visited;
//      ii. if j / F and the bottom of j is not marked, then mark its bottom and
     schedule each of its children to be visited.
//  (d) If the visit to j is from a parent:
//      i. If j  K and the top of j is not marked, then mark its top and schedule
      each of its parents to be visited.
//      ii. if j / K and the bottom of j is not marked, then mark its bottom and
     schedule each of its children to be visited.
//4. The irrelevant nodes, Ni(J|K) are those nodes not marked on the bottom.
//5. The requisite probability nodes, Np(J|K), are those nodes marked on top.
//6. The requisite observation nodes, Ne(J|K), are those nodes in K marked as
     visited.

- (float) computeScore:(Coupling *)coupling withIgnoreSelfClamps: (BOOL) ignore {

    BOOL pathBlocked = [self isPathBlockedForCoupling:coupling
     withIgnoreSelfClamps:ignore];

    if(coupling.positive)
    {   // Positive coupling
```

```
        if(!pathBlocked)
        {   // Correct decision
            NSLog(@"POSITIVE coupling, CORRECT decision: path not blocked");
            return 1.0f; // [[NSNumber numberWithInteger:([coupling.
    clampedVariableIDs count] + 1)] floatValue];
        }
        else
        {   // Wrong decision
            NSLog(@"POSITIVE coupling, WRONG decision: path blocked");
            return 0.0f;
        }
    }
    else
    {   // Negative coupling
        if(pathBlocked)
        {   // Correct decision
            NSLog(@"NEGATIVE coupling, CORRECT decision: path blocked");
            return 1.0f; // [[NSNumber numberWithInteger:([coupling.
    clampedVariableIDs count] + 1)] floatValue];
        }
        else
        {   // Wrong decision
            NSLog(@"NEGATIVE coupling, WRONG decision: path not blocked");
            return 0.0f;
        }
    }

    NSAssert(false, @"Logic is flawed in computeScore.");
    return 0.0f;
}

@end
```

Listing 3: Coupling.h

```
//
//  Coupling.h
//  Ahsum Nimity
//
//  Created by Steven Rekké on 18-04-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface Coupling : NSObject {
@private
    NSString* variableID1;
    NSString* variableID2;
    BOOL positive;
    NSArray* clampedVariableIDs;
}

- (id)initWithVariableID1: (NSString*) var1 andVariableID2: (NSString*) var2
     andType: (BOOL) _positive andClampedVarIDs: (NSArray*) _clampedVars;

@property(readonly) NSString* variableID1;
@property(readonly) NSString* variableID2;
@property(readonly) NSArray* clampedVariableIDs;
@property(readonly) BOOL positive;

@end
```

Listing 4: Coupling.m

```
//
//  Coupling.m
//  Ahsum Nimity
//
//  Created by Steven Rekké on 18-04-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import "Coupling.h"


@implementation Coupling

@synthesize variableID1;
@synthesize variableID2;
@synthesize clampedVariableIDs;
@synthesize positive;

- (id)initWithVariableID1: (NSString*) var1 andVariableID2: (NSString*) var2
     andType: (BOOL) _positive andClampedVarIDs: (NSArray*) _clampedVars {

    self = [super init];
    if (self) {
        variableID1 = [var1 retain];
        variableID2 = [var2 retain];

        positive = _positive;
        clampedVariableIDs = [[NSArray alloc] initWithArray:_clampedVars];
    }

    return self;
}

- (void)dealloc {
    [variableID1 release];
    [variableID2 release];
    [clampedVariableIDs release];
    [super dealloc];
}

@end
```

Listing 5: P1Graph.h

```
//
//  P1Graph.h
//  Ahsum Nimity
//
//  Created by Steven Rekké on 20-03-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import <Foundation/Foundation.h>
@class P1Vertex;

@interface P1Graph : NSObject {
    NSMutableDictionary* vertices;
    NSMutableArray* edges; //An array of P1Edge objects
}
```

```
//Init with Hugin file
- (id) initWithFilePath: (NSString*) filepath;

- (P1Vertex*) getVertexByName: (NSString*) name;

- (void) resetVertices;

- (NSUInteger) getNumberOfVertices;

- (NSSet*) getVerticesNotMarkedOnBottom;

- (NSArray*) getAllVertexNames;

@end
```

Listing 6: P1Graph.m

```
//
//  P1Graph.m
//  Ahsum Nimity
//
//  Created by Steven Rekké on 20-03-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import "P1Graph.h"
#import "P1Vertex.h"
#import "P1Edge.h"

@implementation P1Graph

#define nodeIndicator @"node "
#define nodeSplitter @" "
#define nodeIndex 1

#define connectionIndicator @"potential"
#define connectionRegex @"\\((.*)\\|(.*)\\)"
#define connectionFromGroup 2
#define connectionToGroup 1

- (id) initWithFilePath: (NSString*) filepath {
    self = [super init];
    if (self != nil){

        //Init arrays:
        vertices = [[NSMutableDictionary alloc] init];
        edges = [[NSMutableArray alloc] init];

        //Reading the file:
        NSStringEncoding encoding;
        NSError* error;
        NSString* fileContents = [NSString stringWithContentsOfFile:filepath
    usedEncoding:&encoding error:&error];

        //Splitting in lines:
        NSArray *lines = [fileContents componentsSeparatedByCharactersInSet:[
    NSCharacterSet newlineCharacterSet]];

        //First the nodes:
        for (NSString* line in lines){
            NSRange range = [line rangeOfString:nodeIndicator options:
    NSCaseInsensitiveSearch];
```

```
            if( range.location == 0) {
                NSArray* nodeTokens = [line componentsSeparatedByString:
    nodeSplitter];
                NSString* nodeName = [nodeTokens objectAtIndex:nodeIndex];

                P1Vertex* newVertex = [[P1Vertex alloc] initWithName:nodeName];
                [vertices setObject:newVertex forKey:nodeName];
                [newVertex release];
            }
        }

        //Then the connections:
        for (NSString* line in lines){
            NSRange range = [line rangeOfString:connectionIndicator options:
    NSCaseInsensitiveSearch];
            if( range.location != NSNotFound ) {
                NSError *error = NULL;
                NSRegularExpression *regex = [NSRegularExpression
                                              regularExpressionWithPattern:
    connectionRegex
                                              options:
    NSRegularExpressionCaseInsensitive
                                              error:&error];
                [regex enumerateMatchesInString:line options:0 range:NSMakeRange
    (0, [line length]) usingBlock:^(NSTextCheckingResult *match, NSMatchingFlags
     flags, BOOL *stop){

                    NSRange toNodeRange = [match rangeAtIndex:connectionToGroup];
                    NSRange fromNodesRange = [match rangeAtIndex:
    connectionFromGroup];
                    NSString* foundFrom = [line substringWithRange:fromNodesRange
    ];
                    NSString* foundTo = [line substringWithRange:toNodeRange];

                    foundTo = [foundTo stringByTrimmingCharactersInSet:[
    NSCharacterSet whitespaceAndNewlineCharacterSet]];
                    foundFrom = [foundFrom stringByTrimmingCharactersInSet:[
    NSCharacterSet whitespaceAndNewlineCharacterSet]];

                    NSArray* split2 = [foundFrom componentsSeparatedByString:
    nodeSplitter];
                    for (NSString* token in split2){
                        if ([token length] != 0){
                            NSLog(@"TO: %@ FROM: %@", foundTo, token);
                            P1Edge* edge = [[P1Edge alloc] initWithFrom:[vertices
     objectForKey:token] andTo:[vertices objectForKey:foundTo]];
                            [edges addObject:edge];
                            [edge release];
                        }
                    }
                }];
            }
        }
    }
    return self;
}

- (void) dealloc {
    [vertices release];
    [edges release];
    [super dealloc];
}
```

```
- (NSUInteger) getNumberOfVertices {
    return [[vertices allValues] count];
}

- (NSSet*) getVerticesNotMarkedOnBottom {
    NSMutableSet* result = [[[NSMutableSet alloc] initWithCapacity:[self
     getNumberOfVertices]] autorelease];
    for (P1Vertex* vertex in [vertices allValues]){
        if (!vertex.isMarkedBottom){
            [result addObject:vertex];
            NSLog(@"IRRELEVANT NODE: %@", vertex.name);
        }
    }
    return result;
}

- (P1Vertex*) getVertexByName: (NSString*) name {
    return [vertices objectForKey:name];
}

- (NSArray*) getAllVertexNames {
    return [vertices allKeys];
}

- (void) resetVertices {
    for (P1Vertex* vertex in [vertices allValues]){
        [vertex reset];
    }
}

@end
```

Listing 7: P1Edge.h

```
//
//  P1Edge.h
//  Ahsum Nimity
//
//  Created by Steven Rekké on 20-03-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import <Foundation/Foundation.h>
@class P1Vertex;

@interface P1Edge : NSObject {
    P1Vertex* from;
    P1Vertex* to;
}

@property(readonly) P1Vertex* from;
@property(readonly) P1Vertex* to;

- (id) initWithFrom: (P1Vertex*) from andTo: (P1Vertex*) to;

@end
```

Listing 8: P1Edge.m

```
//
//  P1Edge.m
```

```
//  Ahsum Nimity
//
//  Created by Steven Rekké on 20-03-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import "P1Edge.h"
#import "P1Vertex.h"

@implementation P1Edge

@synthesize from;
@synthesize to;

- (id) initWithFrom: (P1Vertex*) _from andTo: (P1Vertex*) _to {
    self = [super init];
    if (self){
        from = [_from retain];
        to = [_to retain];

        [from addEdge: self];
        [to addEdge: self];
    }
    return self;
}

- (void) dealloc {
    [from release];
    [to release];
    [super dealloc];
}

@end
```

Listing 9: P1Vertex.h

```
//
//  P1Vertex.h
//  Ahsum Nimity
//
//  Created by Steven Rekké on 20-03-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import <Foundation/Foundation.h>
@class P1Edge;

@interface P1Vertex : NSObject {
    NSMutableArray* incomingEdges; //An array of P1Edge objects
    NSMutableArray* outgoingEdges;
    NSString* name;

    BOOL isMarkedTop;
    BOOL isMarkedBottom;
    BOOL isVisited;
    BOOL visitFromChild;
    BOOL visitFromParent;
}

@property(readonly) NSString* name;

@property(assign) BOOL visitFromChild;
```

```
@property(assign) BOOL visitFromParent;
@property(assign) BOOL isVisited;
@property(assign) BOOL isMarkedTop;
@property(assign) BOOL isMarkedBottom;

- (id) initWithName: (NSString*) name;

//Add an edge to this vertex
- (void) addEdge: (P1Edge*) edge;

//Resets the markings on the vertex
- (void) reset;

@property (readonly) NSSet* children;
@property (readonly) NSSet* parents;

@end
```

Listing 10: P1Vertex.m

```
//
//  P1Vertex.m
//  Ahsum Nimity
//
//  Created by Steven Rekké on 20-03-11.
//  Copyright 2011 Rodo - Intelligent Computing. All rights reserved.
//

#import "P1Vertex.h"
#import "P1Edge.h"

@implementation P1Vertex

@synthesize name;
@synthesize visitFromChild;
@synthesize visitFromParent;
@synthesize isVisited;
@synthesize isMarkedTop;
@synthesize isMarkedBottom;

#pragma mark -
#pragma mark Lifecycle stuff

- (id) initWithName: (NSString*) _name {
    self = [super init];
    if (self){
        name = [_name retain];
        incomingEdges = [[NSMutableArray alloc] init];
        outgoingEdges = [[NSMutableArray alloc] init];

        [self reset];
    }
    return self;
}

- (void) dealloc {
    [name release];
    [incomingEdges release];
    [outgoingEdges release];
    [super dealloc];
}
```

```
#pragma mark -
#pragma mark Graph structure stuff

- (void) addEdge: (P1Edge*) edge {
    //Note: an edge to itself is added to both incoming and outgoing
    if (edge.to == self){
        [incomingEdges addObject:edge];
    }
    if (edge.from == self){
        [outgoingEdges addObject:edge];
    }
}

- (NSSet*) children {
    NSMutableSet* result = [[[NSMutableSet alloc] init] autorelease];
    for (P1Edge* edge in outgoingEdges){
        [result addObject:edge.to];
    }
    return result;
}

- (NSSet*) parents {
    NSMutableSet* result = [[[NSMutableSet alloc] init] autorelease];
    for (P1Edge* edge in incomingEdges){
        [result addObject:edge.from];
    }
    return result;
}

- (void) reset {
    visitFromParent = NO;
    visitFromChild = NO;
    isMarkedTop = NO;
    isMarkedBottom = NO;
    isVisited = NO;
}

@end
```