Toward Detecting Deception in Intelligent Systems

Eugene Santos Jr. and Gregory Johnson Jr.

University of Connecticut, 371 Fairfield Road, Unit 2031, Storrs, CT 06269-2031

ABSTRACT

Contemporary decision makers often must choose a course of action using knowledge from several sources. Knowledge may be provided from many diverse sources including electronic sources such as knowledge-based diagnostic or decision support systems or through data mining techniques. As the decision maker becomes more dependent on these electronic information sources, detecting deceptive information from these sources becomes vital to making a correct, or at least more informed, decision. This applies to unintentional disinformation as well as intentional misinformation. Our ongoing research focuses on employing models of deception and deception detection from the fields of psychology and cognitive science to these systems as well as implementing deception detection algorithms for probabilistic intelligent systems. The deception detection algorithms are used to detect, classify and correct attempts at deception. Algorithms for detecting unexpected information rely upon a prediction algorithm from the collaborative filtering domain to predict agent responses in a multi-agent system.

Keywords: intelligent agents, expert systems, deception, deception detection

1. INTRODUCTION

Intelligent systems are increasingly relied upon as sources of information. The results offered by these systems are often used as a basis for action by human decision makers. There is a current trend toward distributed systems, usually in the form of multi-agent systems, in contemporary AI which poses a possible security risk. In critical systems there must be some monitoring for deceptive information, either intentional or unintentional. Sources of deceptive information may originate from malicious trusted experts who are allowed access to these systems or from systems which have been compromised from outside attackers. Intentional deception is often referred to as disinformation. Other, less sinister, sources of deception may originate from incomplete knowledge on the part of the expert involved in constructing the system or an incomplete knowledge-base. These sources of deception would be considered misinformation. Particularly with the interaction of multiple intelligent agents in multi-agent systems, it becomes increasingly difficult to verify information offered by these systems. This creates a scenario in which incorrect information can be introduced into the system causing the system to reach incorrect conclusions which, in turn, mislead the decision maker. Historically, this has been ignored by the developers of intelligent systems by assuming that all contributing experts operate in good faith toward a common goal. Thus, there arises a need to monitor these knowledge-based systems for deceptive information. In this paper we focus on detecting intentional deceptions in intelligent multi-agent systems.

Detecting deception is a difficult task. In fact, humans can detect when someone is being deceptive only a little better than fifty percent of the time (45-65%).[1] Research suggests that when people use electronic media to interact, the deceived is even less effective at detecting when another is perpetrating a deception.[1] Fortunately, there are some constraints on the systems at the focus of our work. In Section 2 we introduce some characteristics of the types of systems in which we are interested. In Section 3, we identify a theory of deception and types of deception due to Bell and Whaley,[2–4] perhaps the only work to date on a general theory of deception. We also identify a model of deception detection, based on work by Johnson et. al.,[5] which is applicable to intelligent systems and not only attempts to detect deception but identify the deception tactic as well as the goal of the deception. This model of deception detection consists of four steps. In Section 4, we identify techniques for implementing each of the steps in intelligent systems. Next, we summarize our current results in implementing the first step, Activation in Section 5. Finally, in Section 6 we outline some directions for future work.

2. SYSTEM CHARACTERISTICS

The current trend in intelligent systems is toward distributed AI in the form of intelligent agents. Therefore, we initially focus on systems comprised of multiple intelligent agents. A decision maker, either human or another type of agent, provides the agents with a set of observations which the agents use as evidence, perform reasoning over their knowledge and return their opinions for the remaining random variables. All agents are given the same observations. Once the decision maker has received all of the opinions from all the agents, it can use belief aggregation techniques to fuse the opinions of all the experts and use the opinions to decide on a course of action. We assume these agents possess some shared knowledge within a particular domain and this knowledge is stored in a probabilistic form which can be reasoned over given evidence. For simplicity, we require the knowledge bases of all agents to share a specific sets of random variables. The expert constructing the knowledge base may introduce intermediate random variables during the construction of the knowledge-base. However, a set of evidence and a set of hypothesis random variables are fixed at system design time and are not changed. Given a set of observations or instantiations of some subset of the evidence random variables, the agent knowledge-base produces a posterior probability distribution over the hypothesis random variables and, if desired, the evidence random variables which were not observed. A posterior probability distribution will be referred to as an agent opinion. Early knowledge-based systems encoded expert knowledge in simple rules. These representations proved to be too brittle for real-world domains which are often riddled with uncertainty or for large domains in which exceptions to the rule cause the number of rules required to grow quickly.[6] Using probabilistic knowledge representations allows one to handle this uncertainty. Knowledge is represented by random variables with discrete states and conditional (in)dependence relationships between the random variables.

Multiple agents can be used to effectively leverage information from heterogeneous information sources and their use distributes the computational burden among several agents, most often executing on separate servers in a distributed environment. The incorporation of several information sources is expected to produce more accurate results, mimicking the real-world practice of group decision making. For example, several specialist medical doctors may collaborate on a difficult diagnosis or the formation of a treatment plan in complex cases. Deception in classical, single-agent "expert systems" can be detected with much the same approach we describe for multi-agent systems. The largest modifications necessary are the construction of models of the expert system using Bayesian learning algorithms on a history of results from that system. These models are used to predict agent opinions as described in Section 4. This technique can also be used to create agent knowledge-bases from heterogeneous information sources in the multi-agent environment. We focus our discussion on multi-agent systems due to their wide use and flexibility. However, we endeavor to develop general methods for deception detection which may be applied to many system architectures.

3. DECEPTION THEORY

In this section we present an overview of the theories of deception and deception detection which has been developed by researchers in cognitive science and psychology. These theories appear to hold promise for implementation in knowledge-based intelligent systems. Deception is defined by Burgoon and Buller to be a "deliberate act perpetrated by a sender to engender in a receiver beliefs contrary to what the sender believes is true to put the receiver at a disadvantage".[7] This definition, and most other accepted definitions of deception, clearly include the intent of the deceiver to achieve some goal as a requirement for deception. The conveyance of misleading information is not sufficient to conclude deception. Therefore, we adopt an intentional stance[8] when attempting to detect deception. That is, we presume that all agents have particular beliefs about the environment and perform actions (e.g. deception) to achieve one or more goals. We unconsciously use this simple yet powerful presumption everyday. Intentionality allows us to determine the goals of others and therefore predict their behavior. In fact, to succeed, deceivers must also rely upon intentionality when crafting their deceptions.

3.1. Theory of Deception

Perhaps the only work in studying deception itself and forming a general theory of deception has been done by Bell and Whaley.[2–4] They have not only formed a theory of deception and general classification of deception tactics, but also a theory of how to deceive. For this work, we are focused on detecting and classifying deception.

Bell and Whaley[2–4] identify two broad categories of deception, dissimulative and simulative. Dissimulative deceptions attempt to 'hide the real' whereas simulative deceptions attempt to 'show the false.' Each may be further divided into three categories and each category is referred to as a deception tactic. The attributes in the environment on which the deception is focused is referred to as the 'deception core'.[5]

3.1.1. Dissimulation

Dissimulation is characterized by an attempt to 'hide the real'. Simple examples include the use of camouflage, the practice of making mobile missile launchers appear to be cargo trucks and jamming RADAR. Deceptions of this type fall into three categories[2–4] masking, repackaging and dazzling as described below.

- **Masking** attempts to make a feature of the environment invisible either by blending with the background or avoiding detection. In our intelligent system this would be accomplished by modifying the posterior probability distribution to be closer to the prior probabilities for that random variable. Paul Johnson et al.[5, 9] propose an additional deception tactic in which the deceiver attempts to cause the target to overlook the deception core by providing weak evidence for it. They term this tactic **double play**. This tactic, however, seems to be a special case of Masking in which providing weak evidence is considered an attempt at blending a feature in with the background of the environment.
- **Repackaging** attempts to hide the real by making something appear to be what it is not. This may be achieved by making something appear dangerous, harmless or irrelevant. In intelligent systems this is difficult, if not impossible, to achieve. The semantics of the random variables included in the system can not be changed once agreed upon at system design. Furthermore, agents are generally not permitted to add or remove random variables in the system.
- **Dazzling** is employed when the feature in the environment is known to exist. It is used to confuse the target of the deception. This can be done by randomizing or partially obscuring the features in the environment to reduce the certainty of the nature of objects or events. A specific example is when a military commander sends multiple movement orders for troops via radio, only one of them being the real orders, knowing the enemy will hear all of the messages. In intelligent systems, this may be encountered when a diagnosis includes too many faults for a human decision maker to respond. Dazzling could also be achieved by an agent who sends several conflicting opinions for the same random variable or an invalid probability distribution as an opinion.

3.1.2. Simulation

Simulation is characterized by an attempt to 'show the false' and may be used in conjunction with hiding. Examples of showing include using a duck call and duck decoys by a hunter to attract real ducks, using false identification and the D-Day invasion in which General Patton was used as a decoy to lead enemy forces from the real invasion force. Three types of simulative deception include[2–4] mimicking, inventing and decoying as described below.

- **Mimicking** is achieved when something is made to appear to be what it is not. This can be employed not only to hide but also to achieve an advantage. This tactic can be used to lure prey or elicit additional actions in which the target unwittingly assists the deceiver achieve a goal. In intelligent systems this tactic may be employed to hide the source of a fault but not the fault itself. It will usually involve hiding characteristics which are inconsistent while showing characteristics which are consistent with what one is trying to mimic.
- **Inventing** a new reality is another tactic for showing the false as in the case of the wolf in sheep's clothing. Inventing in intelligent systems is difficult to achieve since introducing new random variables into the system is easy to detect with some constraints in the design of the system.

- **Decoying** is another simulative tactic which is aimed at luring the target of the deception away from discovering the real. For example, some birds will lure predators away from their nest of babies. One phrase which describes this tactic is 'creating a diversion.' This tactic is often used when the perpetrator can leave several courses of action open, one of which to be chosen at the last moment. For example, in intelligent systems, this can be achieved by reporting an additional fault to draw attention away from the true fault.

Since the categories of Simulation and Dissimulation are opposites of each other, by natural extension so are Masking and Mimicking, Repackaging and Inventing, as well as Dazzling and Decoying.[2] In perpetrated deceptions, several tactics may, and often are, used in conjunction. Deceptions often include Simulation and Dissimulation tactics together.

3.2. Detecting Deception

Paul Johnson et al.[5] identify several strategies for detecting deception generalizing from work by Mawby and Mitchell.[10] These strategies can be divided into two broad groups. The first group consists of strategies which are based on detecting evidence in the environment that a deception has been formed. The second group of strategies are based on inspecting information within the environment for signs of deception.

3.2.1. Detecting the Process of Deception

Although not necessarily indicative of deception, detecting evidence of deception is often easier than detecting the deceptive information in the environment. Detecting evidence of deception may be difficult in intelligent systems. However, security measures coded into a system could limit or indicate where manipulations in the system have occurred.

Motivation Based This strategy for detecting deception is achieved by searching the environment for indications that an agent has the motivation to deceive. Attributes indicating the motivation to deceive may not be connected to the deception core but may be easier to detect than the deceptions. In intelligent systems, this approach seems to be of limited use and relies heavily on an accurate model of the agent.

Process Based Process based strategies focus on detecting attributes in the environment that indicate a deception is being perpetrated. This focus is on detecting signs of manipulation and not the manipulations themselves. For instance, intrusion detection measures may indicate where possible manipulations have occurred.

3.2.2. Detecting Deceptive Information

The following are strategies focused on detecting deception by analyzing information in the environment. Intuitively, strategies of this type are the most effective and simplest to implement in intelligent systems. Therefore, we concentrate on strategies of this type.

Recognition Based This detection strategy focuses on detecting common deceptions or deceptions encountered in the past. This strategy can identify additional cues to look for when detecting a deception. However, often there are many possible deceptions or variations which may not be detected by this approach. Nonetheless, an approach based on case-based reasoning is a possible solution to detecting deception.

Conservative Based This strategy prefers an agent detecting deception to remain open-minded. That is, the agent should infer less, or infer more soundly or delay inference until more data has been collected. This detection tactic is only useful when incomplete information is available from a system. Most often, the system should be able to give a complete diagnosis over all known faults or recommendations over all known actions to a decision maker. Therefore, with respect to intelligent systems, we follow this strategy by default.

Preemption Based This strategy seeks to limit the amount of deception that may be practiced by an agent by limiting the degrees of freedom an agent has in manipulating the environment. This includes limiting any changes to the knowledge-base once it has passed verification and validation. We assume agents in the system are not expected to add or remove random variables in their opinions. The agent may add or remove intermediate random variables to form its opinion, but these are not reported to the decision maker.

Intentionality Based This deception detection strategy acknowledges that agents have beliefs and knowledge of their environment as well as goals. Deceptions practiced are done so to achieve one or more of those goals. This strategy focuses on an agent's goals and the actions that agent can take. The knowledge of an agent's beliefs and goals are also powerful in predicting the actions that agent will take.

3.3. A Model for Detecting Deception

Johnson and Grazioli[5] propose the following general model for detecting deception. It consists of four domain independent steps and is not only concerned with the detection but also the classification and correction of deceptive information. It's purpose is to guide the construction of deception detection models and the analysis of existing models. Our intended implementation of deception detection is based on identifying tools and techniques for implementing each step in multi-agent systems. The four steps are:

1. **Activation** The target of the deception observes an attribute in the environment which is not what was expected. The results of each of the agents are monitored for inconsistencies.
2. **Detection** The target of the deception employs detection strategies to produce hypotheses about suspected manipulations in the environment. Several detection strategies are to be employed in identifying the deception tactic and the deception core.
3. **Editing** The target of the deception edits the representation of the environment which is observed to one which is consistent with the hypothesized manipulations. The decision maker bases it's analysis only on 'trusted' information therefore agent opinions which are suspect are removed from the information available.
4. **Reevaluation** The target of the manipulation chooses an action based on the revised knowledge of the environment. The decision maker fuses the trusted information from the trusted agents and uses the information as the basis for a diagnosis and/or action.

We show how these steps may be implemented in an intelligent system in the following section. Activation techniques have been implemented and tested. The results are offered in Section 5.

4. DETECTING DECEPTION IN INTELLIGENT SYSTEMS

With the increased acceptance and implementation of Distributed Artificial Intelligence (DAI) in the form of Multi-Agent systems and intelligent agents, an interest in detecting deceptive opinions from agents has developed. We would like to apply the deception detection tactics and models to the Multi-Agent System framework. Specifically, we are interested in methods for detecting intentional deceptions aimed at causing the system to fail. The following section outlines possible methods for implementing the phases in the model of fraud detection developed by Johnson et al..[5]

4.1. Activation

The availability of the opinions of other agents in multi-agent systems allows deception to be detected through the comparison of agent opinions when those agents have shared knowledge over the same domain. In fielded systems, one would expect a high degree of correlation between agents with similar knowledge. Therefore, the comparison of agent posterior distributions, given the same observations, appears to be a promising method for the detection of deceptive information. Furthermore, this approach is independent of the chosen knowledge representation or aggregation scheme. It may be applied to any probabilistic knowledge representation. In order

to detect deceptive information in multi-agent systems, one must have some expected value for a particular expert's opinion given some evidence. That is, we need some way of predicting what an agent will offer as an opinion. To achieve this goal, we employ techniques developed from research in collaborative filtering systems.

4.1.1. Process Based Techniques

Process based techniques focus on attributes in the environment which indicate a deception has occurred. In probabilistic systems, we can screen incoming agent opinions for some signs of tampering. First, we can check that the posterior probability distribution for a random variable is indeed a valid probability distribution. All probabilities must be real numbers in the range [0,1] and must sum to 1. We can also confirm that only valid random variables are included in an agents response and that all expected random variables are included in that response. This may also be considered a Preemption based strategy since we are limiting the actions an agent can perform in the system.

4.1.2. Preemption Based Techniques

Inventing or repackaging tactics can be detected by a Preemption based strategy. The architecture of the multi-agent system allows for a design decision between two alternatives which limit the freedom of an agent to *invent* new features in the environment or remove (i.e. *repackage*) features which the agent would like to keep hidden. First, the system designer may require that all agents offer an opinion for all hypotheses. In this case, deception is detected simply by ensuring each agent offers a probability distribution that includes all hypothesis random variables. Alternately, if the problem domain is sufficiently large, some agents may not have the required knowledge to evaluate all hypotheses included in the system. The next section introduces a method for comparing agent opinions which requires us to store a history of agent posterior distributions. With this information, *repackaging* may be suspected when the posterior distribution offered by an agent does not contain a hypothesis (or hypotheses) included in previous posterior distributions produced by that agent. In both situations an agent is prevented from inventing a feature in the environment since all random variables in the system are agreed upon at system design. If an agent includes an unknown random variable in their posterior distribution, this is easily detected as *inventing*. The system also requires that all agents produce all opinions at the same time. This requirement also allows us to follow a Conservative Based strategy in that we are ensured that we are evaluating the validity of an agent's opinion with all the available information.

4.1.3. Intentionality Based Techniques

Agents are certainly presumed to possess knowledge of their environment. That is, after all, their purpose in the system. This knowledge is the source of the opinions they produce. Therefore, we hypothesize, agent opinions should be consistent and therefore predictable. We propose the use of algorithms from the collaborative filtering domain to predict expert opinions. Collaborative filtering techniques are most often used in recommendation systems. These recommendation systems use the past behavior or opinions of other users of the system with similar tastes or behaviors to make recommendations for a particular user. For example, consider a movie recommendation system. A group of users of a movie recommendation system rate their opinion of movies which they have viewed in the past on a five point scale. The movie recommendation system uses these ratings to find correlations between users who have seen some of the same movies in the past. From these correlations, the system attempts to select movies a particular user has not viewed (or rated) and is likely to enjoy. Just as it seems reasonable, in the context of collaborative filtering, that a good way to find interesting content is to find people with similar interests, it seems reasonable to assume we may anticipate an agent's opinion from the opinions of agents with similar knowledge. With this assumption, we may use the opinions of several agents to verify each other. In the case of the stand-alone knowledge-based system, models learned from the responses of the system can be used to predict agent opinions. One approach is to use a distance metric to assist in deception detection in the multi-agent framework. An elegant solution is obtained through the use of the Pearson correlation coefficient and the GroupLens[11] prediction equation as shown in Figure (1) which is used to predict the utility of a news item from the opinions of other users in a newsgroup. This metric can be adopted in a multi-agent system to attempt to predict what an agent will offer for it's opinion.

$$r_{AB} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$$
$$= \frac{\sum (A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_i (A_i - \bar{A})^2} \sqrt{\sum_i (B_i - \bar{B})^2}}$$

Figure 1. Equation to compute the Pearson correlation coefficient of two agents.

$$A_{X_{predicted}} = \bar{A} + \frac{\sum (B_{i_X} - \bar{B}_i) r_{AB_i}}{\sum_i |r_{AB_i}|}$$

Figure 2. Predicting an agent opinion using the GroupLens prediction equation.[11]

Comparing Agent Opinions The Pearson correlation coefficient (Figure (1)) can be applied in the multi-agent framework to obtain a measure of how correlated the opinions of two agents are. We can consider each possible hypothesis individually. For hypothesis $h = true$ we store a probability value from each agent for each set of instantiations of evidence r.v.s. So A_i corresponds to the posterior probability of $h = true$ offered agent A for the i^{th} set of instantiations of evidence r.v.s. Likewise, B_i is the probability agent B assigns to the event $h = true$ for the same instantiation of r.v.s. In the same fashion we consider every possible state for all hypotheses for which the agents offer opinions. The equation returns a weight between -1 and 1. If the two agents agree completely, their correlation coefficient would be 1, if they never agree, it would be -1. Also, $\bar{A}$ is the average of all probabilities agent A has assigned to a hypothesis over all i sets of instantiations of evidence r.v.s and likewise for $\bar{B}$. In the case where an agent does not agree with the majority of other agents, we can use this metric to determine the average correlation of the disagreeing agent from the agreeing agents. If the average coefficient is greater than $1 - \epsilon$, for some small ϵ, we may conclude the difference of opinion is due to differences in experience or training and any disagreement is in good faith.

Expected Agent Opinions The Pearson correlation coefficient is then used in the GroupLens[11] prediction equation shown in Figure (2). Given a sufficient record of posterior distributions over a set of hypotheses returned by all agents and the opinions of all the other agents, we can produce an estimate of what we expect a particular agent to return for each hypothesis. If the agent offers a probability for that hypothesis which is extremely different than the expected probability, the next step in deception detection is carried out to search for patterns in the agent's posterior distribution which match one of the deception tactics. Resnick et al. note that this prediction equation is robust with respect to certain interpretations of the rating scale for the news items. For instance, if one user consistently rates news items in range 1-3 and another rates items in the range 3-5 and the two users would be perfectly correlated otherwise, a score of 5 for the second user would predict a score of 3 for the first user. Likewise for agent opinions, the prediction equation is invariant under shifting and scaling of agent opinions. If one agent consistently assigns probabilities in the range of 0.4 - 0.7 for a hypothesis and another agent assigns probabilities in the range 0.5 - 0.8 and the agents are perfectly correlated otherwise, a probability of 0.8 assigned by the second agent would result in a prediction of 0.7 for the first agent.

Model-Based Approach In the collaborative filtering domain, probabilistic methods attempt to predict the expected value of a vote given what is known about the user.[12] Likewise, a probabilistic model-based approach can be used to predict agent opinions for the purposes of deception detection. Given a history of an agent's prior activity in a domain, a model of that agent can be constructed using data mining algorithms and this model can be used to predict future responses from that agent. This approach is particularly useful in the single-agent case where there are no other agents with which to compare responses. The models obtained through these data-mining learning algorithms may be incomplete however, and may not model the agent perfectly. Therefore, we should only view the models as a guide to detecting deception among agents and not as a replacement for agent knowledge-bases.

Activating Deception Detection We assume the error of the chosen prediction algorithm is normally distributed. We make this assumption to take advantage of a property of normal distributions with respect to the standard deviation of the distribution. This property states that approximately, 68.3% percent of the data samples are within one standard deviation of the mean of a normal distribution. Likewise, approximately 95.4%, 99.7% and 99.99% of samples are within two, three and four times the standard deviation of a normal distribution, respectively. These properties hold for all normal distributions regardless of their particular variance or standard deviation. Therefore, we hypothesize that if the difference between an opinion offered by an agent and the predicted value is greater than a selected multiple of the standard deviation then it is reasonable to proceed to the next step in deception detection. That is, once a prediction has been made, if the prediction error is greater than some threshold the agent's opinion is considered potentially deceptive and the next step in deception detection is employed. This threshold is determined as a multiple of the standard deviation of error for predictions for a particular random variable from a particular agent. This threshold can reasonably range from two to four times the standard deviation of error. Using a threshold of twice the standard deviation of error will likely detect more deceptions than four times the standard deviation of error. However, using twice the standard deviation as a threshold will also produce more false Activations. Using four times the standard deviation as a threshold may detect fewer deceptions but will result in fewer false Activations. Abstractly, selecting twice the standard deviation of error as a threshold is a more paranoid view whereas four times the standard deviation of error is a more trusting view. In implementations, the threshold for different agents, or different random variables from the same agent can use different multiples of the standard deviation of error as a threshold.

Discussion The methods introduced above clearly favor agents who share large portions of similar knowledge and whose knowledge does not change over time. If agents in the system constantly adjust their knowledge, the system can store all agent opinions and use only a recent history of agent responses to compute correlation coefficients. However, this may allow a crafty agent to gradually adjust probabilities over time in order to eventually offer deceptive opinions undetected. If it is necessary to allow agents to modify their knowledge, the system can monitor changes in the correlation coefficient over time for suspect trends. For example, if the average correlation coefficient between one agent and all the others for a particular opinion is monotonically decreasing over time, one might become suspicious. However, the caveat is that if an agent does disagree with the group, it may be due to specialized knowledge and not necessarily due to deception. To differentiate between the two, it seems one must examine all the evidence and consider what deception tactics the system reports and what goals, if any, the system can ascribe to the agent. With all of the evidence it may take a human decision maker to determine if the system's results are consistent and if further action is necessary.

4.2. Detection

The classification step attempts to categorize the suspected deception into one of the six types included in the deception model of Bell and Whaley.[2–4] First of all, Repackaging and Inventing tactics are easily classified. If an unknown random variable, or one not previously reported by a particular agent, is included in a posterior distribution this is classified as Inventing. If this is accompanied by the absence of a posterior distribution for one or more random variables for which the agent has reported on in the past, this is classified as Repackaging. The remaining four tactics are somewhat harder to categorize. First of all, we make the assumption that if more than one random variable is involved in a deception, they are strongly correlated in some way (either positively or negatively). That is, if an agent intends to perpetrate a deception which involves more than one characteristic of the environment, it will either try to make the environment appear to show a similar characteristic or a dissimilar characteristic. For example, if a medical patient is experiencing chest pain, among other symptoms, and the most probable diagnosis is a heart attack, a deceiver may try to assert that the patient is suffering from a pulmonary embolus which shares the symptom of chest pain and is therefore correlated with a heart attack.

To classify the remaining deceptions, we first screen the incoming opinions to ensure they are valid probability distributions. All posterior probability values must sum to 1 and they must all be in the range [0,1]. Any opinions which violate these requirements are considered attempts to confuse and are classified as Dazzling. Next, all random variables which have been identified by Activation as potentially deceptive are separated into groups of random variables which are correlated. For each random variable, we determine the deviation from the predicted

value. The deviation from the prediction (predicted - opinion) is calculated for the random variable state which deviates most from the prediction. A negative value indicates an attempt at "Showing the False" and a positive value indicates an attempt at "Hiding the Real." Within each group of correlated deceptive opinions, if the group consists of all attempts at hiding the real, this is classified as Masking. If a group consists of one attempt at showing the false, this is classified as Decoying. If there are several attempts at showing the false, it is classified as Dazzling an attempt to confuse a decision maker by reporting too many problems to determine the real problem and act accordingly. If there is a mixture of hiding and showing this is classified as Mimicking. In this case, hiding is used to obscure true characteristics of the environment which are inconsistent with the desired deception and showing is used to incorporate false characteristics which are consistent with the deception into the environment.

4.3. Editing

The Editing step when applied to intelligent systems is quite simple. The system can automatically remove or recommend the removal of the links in the decision network relating to detected deceptions in order to remove the deceptive opinion(s) from the decision makers pool of information. The decision maker can now reason with or act on "true" knowledge without being mislead by deceptive information.

4.4. Reevaluation

For the Reevaluation step, the decision maker proceeds with the "Edited" knowledge-base or decision network. This allows the decision maker to act on consistent knowledge. Reevaluation also offers the opportunity to expand our deception detection efforts. Once a deceptive opinion has been detected, the nature of the collaborative relationship between agent and decision maker changes to one which is adversarial in nature. A sensitivity analysis of removing the suspect opinion(s) can allow us to determine what, if any, changes in knowledge or action would have resulted in accepting the deceptive opinion. From this information we may be able to extrapolate the goals of the agent perpetrating the deception. Furthermore, by understanding the goals of a deceptive agent we may be able to predict it's actions. This additional knowledge may assist the decision maker in taking additional action to address the act of deception which has been practiced.

5. CURRENT RESULTS

An empirical evaluation of the intended implementation was conducted to implement and test the activation algorithms to determine if it is possible to detect deception in intelligent systems with our current model. Experiments were conducted in a multi-agent environment using the GroupLens[11] prediction equation. The empirical evaluation was conducted using the Alarm[13] Bayesian network. The Alarm network contains 37 random variables (nodes) and 46 arcs. The Alarm network is clearly divided into 8 diagnoses, 16 findings and 13 intermediate random variables. This network was selected for it's moderate size and clear structure.

5.1. Assumptions

We assume that deceptions occur with a low base-rate. That is, the deceiver acknowledges that to be successful it must be trusted and constant deception is perceived as unreliability. Therefore, the deceiver most often offers good-faith opinions and only offers deceptive opinions only when it is necessary to achieve one or more goals. We assume the opinions given by an agent during system validation are not deceptive and therefore it is reasonable to assume we can collect a substantial number of honest opinions with which we can determine the correlation between multiple agents. We also make the additional assumption that agent opinions will be correlated, at least to some degree. This assumption seems reasonable when we consider that experts in a particular domain are often educated or trained in similar institutions. For example, one would expect a group of medical doctors trained at different medical schools to agree on diagnoses for the vast majority of cases. We also assume the prediction error is normally distributed as presented in Section 4.

5.2. Multi-Agent Detection

5.2.1. Experimental Setup

Multiple agents were simulated by introducing perturbations into the conditional probability tables of the Alarm[13] network. Note, the structure of the Bayesian networks in these agents is not changed and therefore is identical for all agents. Random (selected from a uniform distribution) real numbers from the interval $[-0.1, 0.1]$ (and then a second experiment was conducted with real numbers from the interval $[-0.2, 0.2]$) were added to each conditional probability of each table. The conditional probability tables were then normalized to ensure each represented a valid probability distribution. Ten agents were generated in this fashion. Next, two sets of 1000 observations were generated by randomly selecting a random number (1-10) of random variables and instantiating each with a randomly selected state. All random selections were generated from the uniform distribution. One set of observations served as a training set to compute the Pearson correlation coefficient and prediction error standard deviation while the other set served as a test set to evaluate the algorithm's ability to accurately detect deceptions. Ten simulated agents were generated for this experiment. The Pearson correlation coefficient between the Alarm network and each simulated agent was computed for each state of each random variable using the opinions obtained using the random observations in the training set.

Next, for each of the observations in the test set, the opinions of each of the ten simulated agents were used to predict the opinion obtained from the original Alarm network. A simple deception distribution was simulated by perturbing the posterior probability distribution computed by the Alarm network. That is, for each random variable, the posterior probability for the first state was reported to be that of the second, and so on. If the error in predicting the Alarm network response was greater than three times the standard deviation of the prediction error for that random variable state, this was considered a false activation. Likewise, if the prediction error of the simulated deception distribution was greater than three times the standard deviation for the prediction error for that random variable state, this was considered successful detection of deception.

5.2.2. Multi-Agent Results

Tables 1 and 2 show a summary of the results obtained from our experiments with multi-agent deception detection. Due to the volume of data collected, only this summary is shown here. These tables show the minimum, maximum, average and median absolute value of correlation coefficients between the ten simulated agents and the Alarm network (for all 105 random variable states in the Alarm network) in each of the two experiments. Also in these tables are the minimum, maximum, average and mean values for the standard deviation of prediction error for each experiment. A multiple of this standard deviation is used as a threshold to detect deceptions. The last two rows in these tables show the minimum, maximum, average and mean fraction of deceptions detected and false activations reported.

These results show that there is indeed a high correlation between the simulated agents produced for this experiment. Furthermore, and more importantly, these results show that if there is a high degree of correlation between agents, as we have assumed, it is possible to accurately detect deceptions by individual agents using the GroupLens prediction algorithm. The results for the first multi-agent experiment show the average standard deviation of error for predictions to be approximately 0.07. On average, in this case, we can detect manipulations in the environment as small as 0.22. Noting that all the states of each random variable are inextricably bound to each other, Tables 3 and 4 show a summary of results for the state of each random variable which the algorithm was most successful at detecting deceptions. Clearly, if deception is suspected for one state of a random variable, the posterior probability of all other states of that random variable become suspect. Therefore, one could argue that we need only detect deception in the opinion for one state of a random variable in order to successfully detect a deceptive opinion. In the first experiment, our approach was able to detect an average of 68% of the simulated deceptions. In the second experiment, an average of 59% of the simulated deceptions were detected. In both experiments false activations were reported in approximately 2% of the cases on average. The worst case for false activations in both experiments was approximately 6.5%. The GroupLens prediction algorithm performed well in predicting agent opinions in our experiments. This allowed our deception detection algorithm to detect a respectable number of the simulated deceptions with a reasonable number of false activations.

	Min	Max	Mean	Median
Coefficients	0.389980102	0.998693299	0.911652591	0.93938709
Prediction error σ	0.003411243	0.152786685	0.074216888	0.070682193
Deceptions	0.110352673	1	0.683922583	0.708381171
False Activations	0	0.066746126	0.022965879	0.019953052

Table 1. Summary table of results for agents generated with perturbations in the interval $[-0.1, 0.1]$.

	Min	Max	Mean	Median
Coefficients	0.040879821	0.996061714	0.849596795	0.884381314
Prediction error σ	0.00662076	0.248520976	0.112183338	0.10610505
Deceptions	0.066210046	1	0.592662416	0.575147929
False Activations	0	0.067207416	0.017989745	0.017045455

Table 2. Summary table of results for agents generated with perturbations in the interval $[-0.2, 0.2]$.

6. FUTURE WORK

6.1. Deception Detection

Deception is not merely the transmission of a falsehood or inaccurate fact. Deception must be directed to some goal. Therefore, true deception detection in intelligent systems must adopt an intentional stance.[8] In order to conclude true deception has been practiced, one must determine that the misinformation is actually disinformation and the deception is pursuant to some goal. This attitude makes the detection task more complex. However, adopting this attitude possibly allows us to predict the behavior of the deceiver. The following subsections offer some areas for future development of deception detection in intelligent systems.

6.2. Activation

Further empirical study is necessary to determine the minimum number of experts necessary for accurate detection of deceptive opinions. Further empirical measurement of detection error on crafted deceptions is needed to assist in determining the accuracy of our methods for activation of deception detection tactics. To this end we plan to develop a deception agent capable of crafting deceptions based on the six deception tactics of Bell and Whaley.[2–4] We also would like to investigate the possibility of collaborative deceit by two or more agents. It appears the model may be robust to this type of attack if there are a sufficient number of agents in the system to reduce the impact of each individual agent's opinion. Further empirical analysis is necessary to determine the threshold at which experts are not correlated enough for our method of deception detection. Further work will also consider methods for differentiating false activations from true deceptions.

6.3. Detection

An empirical study is necessary to evaluate our proposed method of Detection. First of all, we must construct an agent capable of crafting non-trivial deceptions. This agent must be able to create deceptions based on the six deception tactics and at a low base rate. Once this has been completed, we may test our assumptions and intended approach by empirical study. This stage will be tested within a multi-agent system with a decision maker which has Activation and Detection algorithms available to verify the opinions offered by agents in the

	Min	Max	Mean	Median
Deceptions	0.309859155	1	0.825790735	0.859931114
False Activations	0.001133787	0.061484919	0.025495806	0.020954598

Table 3. Summary table of results for agents generated with perturbations in the interval $[-0.1, 0.1]$. These results include only the most accurately detected state of each random variable.

	Min	Max	Mean	Median
Deceptions	0.252293578	1	0.768870872	0.802497162
False Activations	0	0.067207416	0.020321088	0.020713464

Table 4. Summary table of results for agents generated with perturbations in the interval $[-0.2, 0.2]$. These results include only the most accurately detected state of each random variable.

system. These Detection algorithms will be implementations of (or based on) the Detection methods described in Section 4.

6.4. Editing

Implementation of Editing is a fairly straightforward step in intelligent systems. Once a deceptive opinion has been detected, the deceptive opinion can be disregarded by removing the links in the Bayesian network which originate from the node representing that agent's deceptive opinion.

6.5. Reevaluation

Reevaluation can be implemented simply at first, merely consisting of reasoning with the edited network. However, we can use the knowledge or belief that deception has occurred as additional information about the suspect agent. However, this requires us to model the agent's beliefs and goals in order to predict it's future actions. This is not a simple task. This remains a possible focus for future research.

ACKNOWLEDGMENTS

This work is funded by the Air Force Office of Scientific Research under Grant No. F49620-03-1-0014.

REFERENCES

1. J. F. George and J. R. Carlson, "Group support systems and deceptive communication," in *Proceedings of the 32nd Hawaii International Conference on System Sciences*, IEEE, January 1999.
2. B. Whaley, "Toward a General Theory of Deception," in *Military Deception and Strategic Surprise*, J. Gooch and A. Perlmutter, eds., Frank Cass, 1982.
3. J. B. Bowyer, *Cheating*, St. Martin's Press, 1982.
4. J. B. Bell and B. Whaley, *Cheating and Deception*, Transaction Publishers, 1991.
5. P. Johnson and S. Grazioli, "Fraud Detection: Intentionality and Deception in Cognition," *Accounting, Organizations and Society* **25**, pp. 355–392, 1993.
6. J. Pearl, *Probablilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, second ed., 1988.
7. J. K. Burgoon and D. B. Buller, "Interpersonal Deception: III. Effects of Deceit on Perceived Communication and Nonverbal Behavior Dynamics," *Journal of Nonverbal Behavior* **18(2)**, pp. 155–184, 1994.
8. D. C. Denett, *The Intentional Stance*, The MIT Press, 1987.
9. P. Johnson, S. Grazioli, K. Jamal, and R. G. Berryman, "Detecting Deception: Adversarial Problem Solving in a Low Base-Rate World," *Cognitive Science* **25**, pp. 355–392, 2001.
10. R. Mawby and R. W. Mitchell, "Feints and Ruses: An Analysis of Deception in Sports," in *Deception: Perspectives on Human and Nonhuman Deceit*, R. W. Mitchell and N. S. Thompson, eds., State University of New York Press, 1986.
11. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the conference on Computer supported cooperative work*, pp. 175–186, IEEE, October 22-26 1994.
12. J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," Tech. Rep. MSR-TR-98-12, Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmond, WA, 98052, October 1998.
13. I. Beinlich, G. Suermondt, R. Chavez, and G. Cooper, "The alarm monitoring system," in *Second European Conference on AI and Medicine*, 1989.