Learning Whom to Trust: Using Graphical Models for Learning about Information Providers

(Extended Abstract)

Philip Hendrix and Ya'akov Gal and Avi Pfeffer
Harvard University
33 Oxford St.
Cambridge, MA 02143 USA
{phendrix, gal, avi}@eecs.harvard.edu

Categories and Subject Descriptors

I.2.6 [**Learning**]: Parameter learning; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

Artificial Intelligence, Trust, Graphical Models, Bayesian Networks

1. INTRODUCTION

In many multi-agent systems, information is distributed among potential providers that vary in their capability to report useful information and in the extent to which their reports may be biased. This abstract shows that graphical models can be used to simultaneously learn complex reporting policies that agents use and learn their capabilities; weigh the benefits of different combinations of information providers; and optimally choose a combination of information providers to minimize error. An agent's *policy* refers to the way in which the agent reports information. We show that these models are able to capture agents that vary in their capabilities and reporting policies. Agents using these graphical models outperformed the top contestants of the recent international Agent Reputation and Trust testbed competition. Further experiments show that graphical models can accurately model agents that use complex policies to decide how to report information, and determine how to combine these reports to minimize error.

2. MODELING AGENTS' POLICIES

This abstract addresses the problem of learning about information providers in settings in which agents need to make decisions, and the outcome of these decisions is directly affected by the quality of the information they can obtain. The information reported by provider agents may not be truthful. Providers may attempt to deceive others by reporting inaccurate or biased information. We model such agents using a mixture of several possible reporting policies: overestimating (or underestimating) the estimate obtained from the system by an unknown constant, reporting the true estimate, or reporting an estimate from a normal distribution whose mean and variance is unknown. This is not meant to be an exhaustive set of all possible policies, but rather a meaningful set of some policies that might be seen in the real world. More complex policies may be captured using this model by creating a distribution over the policies. The probability that the mixture assigns to each individual policy is also unknown. A Bayesian network describing this model is shown in Figure 1, from the point of view of a principal agent. For expository convenience we only include the nodes relating to a single interaction and provider agent. Dashed nodes represent variables whose distribution is unknown.

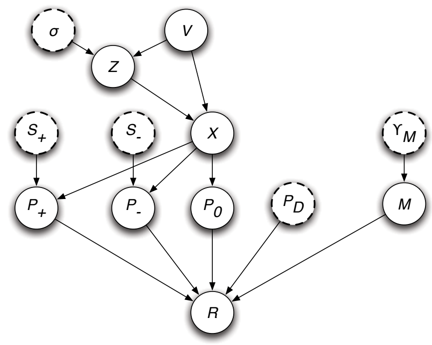

Figure 1: A Bayesian network of a provider agent using a mixture policy. Dashed nodes represent variables whose distribution is unknown.

The node σ represents the competence of the provider agent, and is modeled using an Inverse-Gamma distribution. The node V represents the true value of a partner, and is a sampled from a known uniform distribution. The node Z represents the error rate of the provider agent, which is sampled from the normal distribution $N(0, \sigma^2 \cdot V)$. The node X represents the provider agent's true estimate, and is deterministically computed as $(V + Z)$. The node P_0 represents a truthful report by the provider agent of the estimate X. This policy is represented by a deterministic probability distribution encoded in P_0 that assigns value 1 to X. The node P_+ represents a report by the provider agent that multiplies its estimate of X by a positive constant value represented by the node S_+. This policy is specified by assigning a deterministic distribution over P_+ that assigns probability 1 to the value of $(X_i \cdot S_+)$. Similarly, the node P_- represents a report by provider agent that multiplies its estimate of X by a negative constant value represented by the node S_-. The distribution of P_- is assigned in a similar fashion to P_+. The node P_D represents a policy that reports a random appraisal.

The node R represents the report that the provider agent submits to the principal. The parents of R are the report nodes P_0, P_+, P_-, P_D and the node M. The domain of M ranges over the possible report nodes. The distribution of M is a multinomial, and the node

Cite as: Learning Whom to Trust: Using Graphical Models for Learning about Information Providers, (Extended Abstract), Philip Hendrix, Ya'akov Gal, Avi Pfeffer, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1261–1262

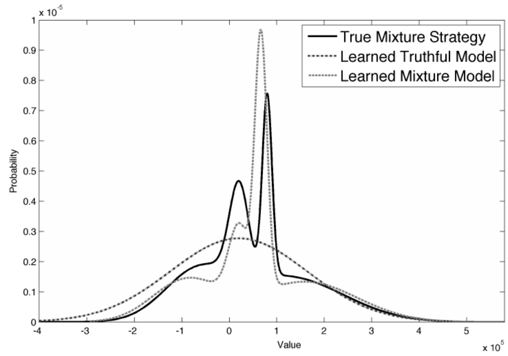

Figure 2: Learned distributions for one provider agent.

γ_M specifies the parameters for this distribution. These parameters represent the beliefs of the principal over how the provider agent chooses between its individual reporting policies.

The complexity of the network of Figure 1 means that computing the posterior distribution over the unknown parameters cannot be done in closed form. Instead, we employ Gibbs sampling to compute these quantities, through the WINBUGS system, a publicly available inference toolkit (http://www.mrc-bsu.cam.ac.uk/bugs/).

We first demonstrate the use of the mixture model for learning the policy of a provider agent that employs the following mixture policy: With probability 0.45 the value of X is multiplied by 6. With probability 0.25 the value of X is multiplied by -3.5. With probability 0.15 the value of X (reported truthfully). With probability 0.15 a normal distribution with mean 80,000 and standard deviation 10,000. The estimation error for this agent is distributed normally with a mean of zero and a standard deviation of 1.0.

Figure 2 compares the true mixture policy of a provider agent to the policy learned by two possible models. One in which the provider agent is assumed to be truthful (does not distort its estimate), and one which uses the network of Figure 1 to model a mixture policy. The figure is presented for a single value of 20,000.

The true mixture policy for the provider agent is shown as the black, solid curve in the figure. The modes that can be seen in the distribution represent the different individual policies that are used by the agent. The dotted curve shows the mixture policy that is learned by the principal that models agents modeling a mixture policy. The darker, dashed curve shows the truthful model that is learned when assuming only truthful reports.

3. COMBINING INFORMATION FROM PROVIDERS

In this section we show how to use the model of agents' policies to make decisions. To make good decisions we need to determine an appropriate way to aggregate information from multiple information providers to create an appraisal that is more accurate than any single opinion from any individual information provider. The model must estimate the opinions of the information providers and determine the estimated error of asking any set of information providers for their opinions. We first show how an agent can directly use the Bayesian network to weigh the potential contribution of the reports of different information provider agents, and then show how the network can inform a process for optimally choosing between different providers.

Let D denote a set of provider agents chosen by the principal for interaction i, and let W_d^i denote the weights for an individual provider $d \in D$ for interaction i. The principal's combined estimate of a value at i, denoted A_D^i , is the weighted average of the opinions of the provider agents in D.

Computing the optimal weights for information providers using a mixture model is a highly non-linear optimization problem. We cannot solve this problem analytically. Instead, we approximate the weight values by envisioning what the ideal weight values would look like. To this end, we extend the Bayesian network of Figure 1 to explicitly represent the weights attributed to provider agents, and perform inference on the network to approximate their values from a sampled set of future interactions. Weighing the potential contributions of the provider agents in D corresponds to solving a constrained optimization problem for finding the set of weights W_D^i that minimize the expected error.

The Bayesian network of Figure 1 was used to choose among different provider agents. A soft-max function was used to allow the principal agent to explore different combinations of information providers. The likelihood of selecting a subset of agents is correlated with their expected error.

3.1 Demonstration of Model

We now demonstrate how to use this technique to learn how to combine the reports of three provider agents $\{a_1, a_2, a_3\}$, with the following parameters. A provider a_1 with low error of estimation ($Z_1 \sim N(0, 0.01) \cdot V$), that consistently multiplies its estimate by a factor of 4. A provider a_2 with low error of estimation ($Z_2 \sim N(0, 0.01) \cdot V$), that consistently multiplies its estimate by a factor of (-4). A provider a_3 with high error of estimation ($Z_3 \sim N(0, 4) \cdot V$), that reports its estimate truthfully. We learned from 10,000 interactions with these providers. The following table lists the learned weights for each possible set of providers and combined error rate for estimating a held out test-set of values. The last line in the table is a baseline for the case in which no information was obtained from any of the three information providers.

Providers	w_1	w_2	w_3	Error
$\{a_1, a_2, a_3\}$	0.5133	0.4867	0.0	13350
$\{a_1, a_2\}$	0.5133	0.4867		**13350**
$\{a_1, a_3\}$	0.2245		0.7755	79470
$\{a_2, a_3\}$		0.2178	0.7822	79420
$\{a_1\}$	1.0			205500
$\{a_2\}$		1.0		209800
$\{a_3\}$			1.0	87440
$\emptyset$				51490

As these results show, each provider agent is useless when chosen in isolation. (The error obtained from any single provider was always greater than selecting none of the providers). However, the model was able to learn that combining information from both providers a_1 and a_2 resulted in the lowest amount of error. The model was also able to learn that there is no benefit to adding the report from a_3 to a combination that includes the joint reports of a_2 and a_1.

Experiments show that an agent using these graphical models outperformed contestants from the Agent Reputation and Trust (ART) testbed using the identical competition parameters to those in the AAMAS 2009 competition. In addition, we compared our model to synthetic agents that added bias to their reports. After an acclamation period, the agent was able to outperform the previous finalists of the ART competition.

4. ACKNOWLEDGMENTS

The work reported in this paper was supported in party by NSF grant IIS-0705406.