Multiple Models of Reality and How to Use Them

Wojciech Jamroga

Parlevink Group, University of Twente, Netherlands

Abstract

A virtual agent may obviously benefit from having an up-to-date model of her environment of activity. The model may include actual users' profiles, a dynamic environment characteristic or some assumptions being accepted by default. However, the agent doesn't have to stick to one model only, she can possess a set of complementary beliefs, both learned and assumed. A hierarchy of beliefs for an agent is proposed in this paper, together with a decision making scheme and some experimental results to support the idea. The experiments consisted of the agent's interactions with some simulated 0-level agents, acting as customers of an imaginary Internet banking service.

`Keywords:` autonomous agents, user modeling, machine learning, decision making, beliefs.

1 Virtual Agent's Reality

An agent's knowledge about its environment can be either assumed or acquired through some kind of learning. The first approach dominates most Game Theory solutions, with 'best defense models' defining the expected behavior of the 'opponent' [4]. The opponent is usually implicitly assumed to play his best, even to the extent of making him omniscient, which is seldom the case in the real world. The machine learning approach emphasizes the importance of keeping an accurate and up-to-date model of the world. The agent can learn the policy of its adversary to exploit his weaknesses [3, 11, 13], to converge with dynamic, possibly indifferent environment [12, 13], or to learn trust and cooperation with other agents [1, 12]. The goal of the agent – in both cases – is to maximize her numerical reward (payoff, utility) in the long run. Thus the decision making criterion is in most cases based on maximization of the expected payoff with respect to the agent's current knowledge about the environment of her action.

The agent may model the environment in many different ways, and in most cases it's up to the designer to decide which one will be maintained and used by her before she starts her 'life'. The idea behind this paper is that the agent may be better off keeping several models of the reality at the same time, and switching to the most appropriate at the very moment. Also, the concept of belief hierarchy presented here is aimed to enable using the content-based knowledge (individual user profiles) and the collaborative models (stereotypes) at the same time, especially for quantitative beliefs [18, 7].

The rough idea of multilevel environment modeling for e-commerce agents has been already presented in [5]. Similar intuition underlies a number of recent results:

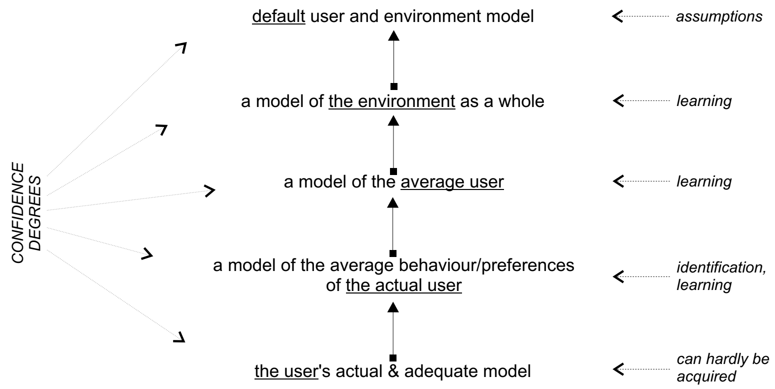

Figure 1: Example belief structure for modeling the reality

an adaptive news agent that keeps two complementary models of the user (long-term preferences + short-term ones) [2], a system that uses alternative Markov models for predicting users' requests on a WWW server [19], etc. In both cases hybrid models are presented that perform better than any of the original models alone. Finally, some papers propose multilevel learning in order to learn user's interest that can possibly drift and recur [8, 17].[1]

2 Hierarchy of Beliefs

An autonomous agent would obviously be interested in possessing an actual and adequate model of the actual user. It may include the user's preferences, his actual strategy, predicted future actions etc. However, such a model can hardly be acquired: the user may dynamically change his profile or even try to cheat the agent about his identity, interests, strategy etc. The agent can only try to build up some model of the average behavior presented so far by this particular user. Otherwise, some model of the 'average user' or a model of the entire reality may prove helpful from time to time. And finally, when we can't trust anything we learned so far, we may need some default assumptions to evaluate possible courses of action and choose among them. All the proposed levels of modeling are shown on figure 1. The more specific the level of knowledge used by the agent is, the more accurate her decisions can be. However, if the agent has little or no confidence in her specific beliefs, she should turn to the more general ones.

2.1 Hierarchies of Quantitative Beliefs

One can imagine expressing the actual agent's beliefs using very different languages; also, the learning may proceed along different methods, and the links in the diagram can represent different relationships (activation or inhibition triggered

[1]big thanks to Ingrid Breymann for her literature overview I dared to use

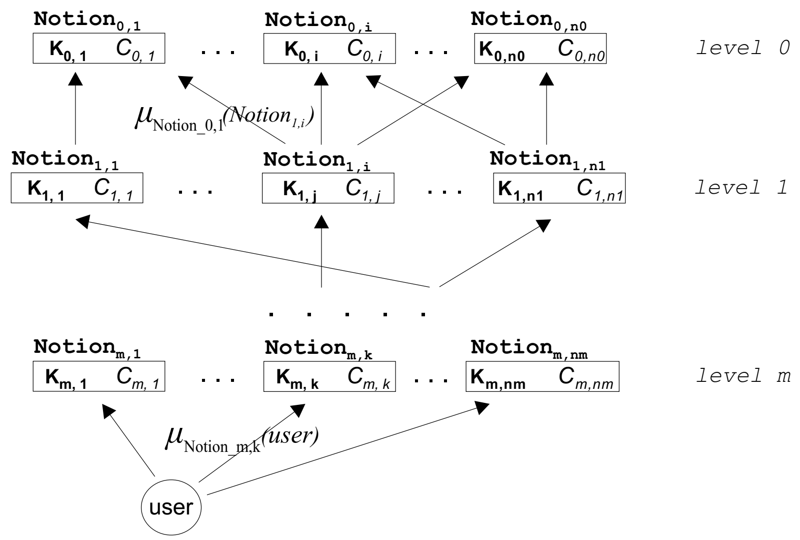

Figure 2: Generalized belief hierarchy for quantitative beliefs

with respect to logical constraints or numerical variables, for instance). For the rest of the paper it is assumed that the agent's beliefs are quantitative in the sense that they imply some numerical evaluation of every action at hand. Moreover, the decision making process can be based on the current evaluation values. A more complex and general hierarchy of beliefs, enabling multiple 'inheritance' and an arbitrary number of concept levels, is presented on figure 2. The agent maintains alternative models (knowledge, beliefs) $\mathbf{K_{l,i}}$ together with some confidence values $C_{l,i}$ for each of them. The links between notions represent the subset/membership relations with explicit weights μ. Figure 3 shows an example belief structure for a banking agent.

The advantage of such a hierarchy is that the final evaluation of choices can be based on a linear combination of the particular evaluations, as proposed in section 2.2. If the agent trusts the most specific model in, say, 70% – the final evaluation may depend on the model in 70%, and the remaining 30% can be derived from the levels above. In consequence, the decision is based on all the relevant models at the same time, although in different proportions – weighting the partial evaluations with the confidence she has in them.

2.2 Decision Making

Let $par(\mathfrak{m})$ denote the set of all the parents of node $\mathfrak{m}$, i.e. all the models exactly one level up. The (multimodel) evaluation of action a with $\mathfrak{m}$ as the most specific model of the environment can be defined recursively:

$$E(\mathfrak{m}, a) \quad = \quad C_{\mathfrak{m}} \cdot eval(K_{\mathfrak{m}}, a) \; + \; (1 - C_{\mathfrak{m}}) \sum_{p \in par(\mathfrak{m})} \mu_p(\mathfrak{m}) E(p, a)$$

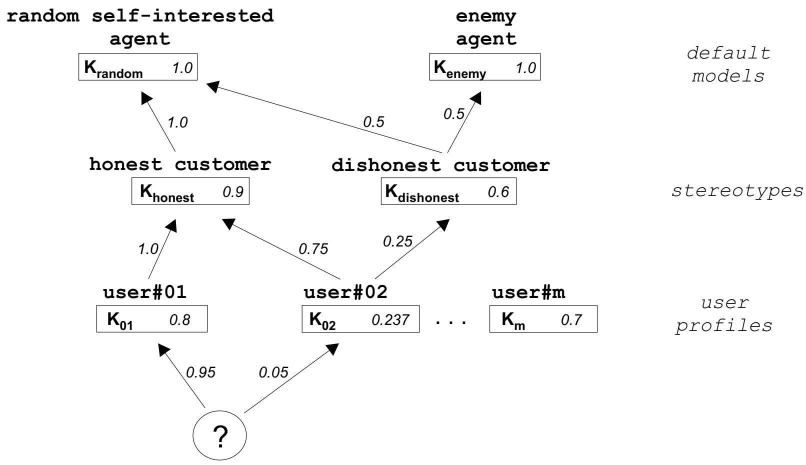

Figure 3: An example: two default models, two stereotypes, m user profiles

where $eval(K_{\mathfrak{m}}, a)$ is a numerical evaluation of a with respect to belief $K_{\mathfrak{m}}$ only – expected payoff estimation, for instance. The final evaluation of alternative decisions can be now calculated as:

$$E(a) \quad = \sum_{p \in par(user)} \mu_p(user) E(p, a)$$

The weights should be nonnegative and sum up to 1 finally [10]; to assure this, the following restrictions on the belief structure are suggested:

- $0 \leq C_{\mathfrak{m}} \leq 1$ and $0 \leq \mu_{\mathfrak{m}}(\mathfrak{m}') \leq 1$ for every node $\mathfrak{m}$ and $\mathfrak{m}'$ (the values are used to represent uncertainty);
- $\sum_{p \in par(\mathfrak{m})} \mu_p(\mathfrak{m}) = 1$ (no relevant notions are omitted in the hierarchy);
- $C_{0,i} = 1$ for every i (the agent is fully committed to her most general assumptions).

Now when the agent is able to compute some rating for every action, she can use any well-established decision-making scheme – like choosing the action with the highest expected payoff.

3 Experiments

Some simple simulations were conducted to verify the idea of keeping and using multiple alternative models of the reality. The essence of the paper is that an agent may build up and use more than one model, so in the experiments the agent uses exactly two models and the environment is static (i.e. the agent's utility function and the customers' policies don't change throughout the interaction) to make things as simple as possible.

The simulations have been inspired by the following scenario: a software agent is designed to interact with users on behalf of an Internet banking service; she can make an offer to a user, and the user's response determines her output. The agent has 3 possible offers at hand: the 'risky', 'normal' and the 'safe' offer, and the customer can respond with: 'accept honestly', 'cheat' or 'skip'. The complete table of payoffs for the game is given below. The 'risky offer', for example, can prove very profitable when accepted honestly by the user, but the agent will lose proportionally more if the customer decides to cheat; as the user skips an offer, the bank still gains some profit from the advertisements etc.

	accept	cheat	skip
risky offer	30	-100	1
normal offer	6	-20	1
safe offer	1.5	-1	1

Of course it isn't essential that the agent is an e-banking broker. What's important is that she should learn users' profiles to approximate the actual preferences of each user. On the other hand, the agent has too much to lose to afford risky decisions when the identity of a user is unknown or the user is completely new to the system. To prevent this, she uses a default user model besides the profiles.

The banking agent is a 1-level agent, i.e. an agent that models other agents as stochastic (0-level) agents. The user is simulated as a random 0-level agent – in other words, his behavior can be described with a random probabilistic policy. To get rid of the exploration/exploitation tradeoff we assume also that the user is rather simple-minded and his response doesn't depend on the actual offer being made: $p(cheat)$, $p(accept)$ and $p(skip)$ are the same regardless of the offer (if he's dishonest, he cheats for a small reward as well as a big one, for instance). The agent estimates the user's policy with a relative frequency distribution, counting the user's responses. The default model is defined in the Game Theory fashion: the user is assumed an enemy who always cheats. There is no uncertainty about the identity of the user – thus $\mu_{profile}(user) = 1$. As there is only one default model, also $C_{default} = 1$ and $\mu_{default}(profile) = 1$.

The aim of the experiments is to compare the efficiency of such agent's behavior with the behavior of a standard learning agent – i.e. the agent who uses only a user profile when making her decisions. 1000000 independent random interactions (a sequence of 100 rounds each) have been simulated; the average results are presented in section 3.1.

3.1 Results

Figure 4 shows the average payoff of the 'banking agent'. 4 different agents were used: a "single-model agent" using only users' profiles (which can be also interpreted as a double-model agent with fixed $C_{profile} = 1$), another single-model agent, using only the default 'best defense' assumptions (or a double-model one with $C_{profile}$ always 0), an agent using both models with fixed $C_{profile} = 0.9$,[2]

[2] in fact, $C_{profile} = 0.5$ and 0.7 were also tried, but the results were virtually the same as for $C_{profile} = 0$

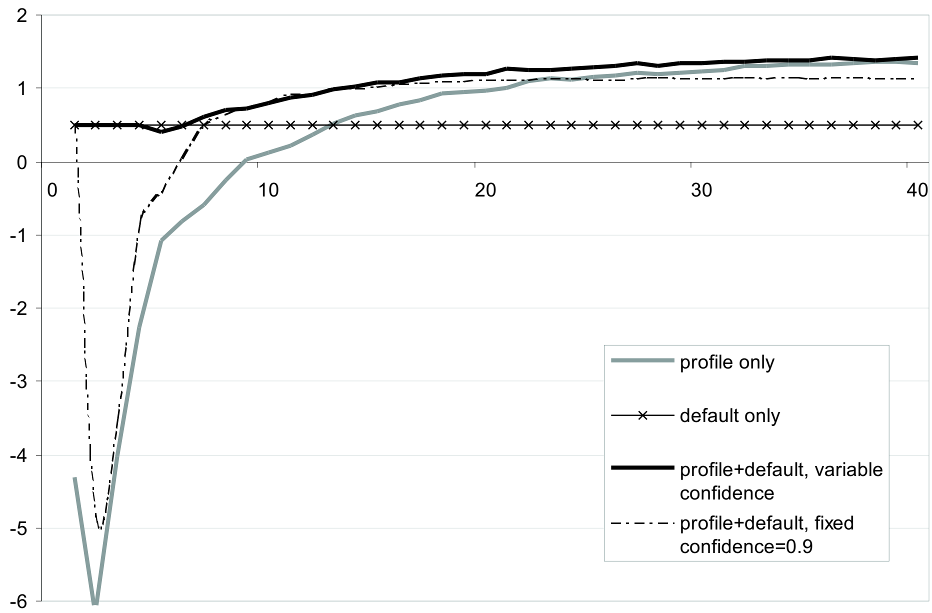

Figure 4: Two-level beliefs vs. simple model: average payoff per round

and a double-model agent with variable confidence values.[3] The output of the simulations shows that the banking agent can indeed benefit from using a default model together with the users' profiles in such setting. The last agent outperforms both single-model agents: she plays much safer in the first 25 rounds (when there is no sufficient data) and after that the payoffs are similar. Only the output of the first 40 rounds is presented on the chart to emphasize the part where the main differences lie. The results for rounds $41-100$ were more or less the same.

4 Conclusions

The concept of the belief hierarchy is aimed to help a virtual agent to behave in a more robust, flexible and consistent way, especially when the agent can't fully trust her beliefs or she can have several competing models of the reality. The experiments showed that an autonomous agent can get more payoff when using multiple models of the environment rather than just one model. The hierarchy

[3]one can suspect problems with obtaining appropriate confidence values (as well as the μ values when they are needed). What we can do at least is to make sure that the confidence is low when the agent has collected few data so far, and that it's close to 1 when the data size is large. Some suggestions can be found in the the literature on statistical inference [15, 14] or higher-order uncertainty [6, 16]. The "variable confidence" agent uses Wang's confidence: $C_{profile} = \frac{n}{n+1}$ as the subsequent confidence values [16], where n is the number of observations (interaction rounds) completed so far.

requires only linear growth of computational power on the agent's part (with respect to the number of models being used), and the particular models can be constructed and updated in parallel since they are independent by definition – they only share the input data.

The experiments were primarily designed to be simple: the user was static (so the confidence could be assumed an increasing function of the data set size), and the agent was using only two alternative models, one of them fixed and both having the same structure. Thus, the agent could arguably use Bayesian updating, for instance [9], to integrate both sub-models in this very case (starting with the default model and updating it sequentially as more user-specific evidence arrives). In consequence the agent would use a single model, and no confidence values would be necessary. However, things are not always like this. If the user isn't static, his behavior may become suspect from time to time, so the agent can be better off turning back to the default model to some extent – but it doesn't seem clever to require that she abandons *all* the knowledge gathered so far, and starts the learning process from the scratch again. If both models are evolving, the agent must keep them anyway to proceed with the updates. Last but not least, the models may be built upon different structures (for example, the default model could be a simple Q-function with no probability at all) or they may represent different entities: conscious beliefs, unconscious beliefs, reflexes – and then it's not clear how they can be integrated at all.

It is worth noting that in the course of the simulations the agent did gain some additional profit when incorporating the 'best defense' model against 0-level *random* agents. Some experiments against 1-level adversary agents may help to make the point even stronger in the future.

References

[1] B. Banerjee, R.Mukherjee, and S. Sen. Learning mutual trust. In *Working Notes of AGENTS-00 Workshop on Deception, Fraud and Trust in Agent Societies*, pages 9–14, 2000.

[2] D. Billsus and M.J. Pazzani. A hybrid user model for news story classification. In J. Kay, editor, *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 99–108. Springer, 1999.

[3] D. Carmel and S. Markovitch. Learning and using opponent models in adversary search. Technical Report CIS9609, Technion, Haifa, 1996.

[4] I. Frank and D. Basin. Search in games with incomplete information : A case study using bridge card play. *Artificial Intelligence*, 100(1-2):87–123, 1998.

[5] W. Jamroga. Multilevel modeling of dialogue environment for e-commerce agents. In D. Waldziński, editor, *Electronic Commerce - gospodarka XXI wieku*, pages 49–53. Gdansk, 2001.

[6] G.J. Klir. Uncertainty and information measures for imprecise probabilities: An overview. In *Proceedings of the First International Symposium on Imprecise Probabilities and Their Applications*, 1999.

[7] A. Kobsa. User modeling: Recent work, prospects and hazards. In M. Schneider-Hufschmidt, T. Kühme, and U. Malinowski, editors, *Adaptive User Interfaces: Principle and Practice*. Elsevier, 1993.

[8] I. Koychev. Learning about user in the presence of hidden context. In *Proceedings of Machine Learning for User Modeling: UM-2001 Workshop*, pages 101–106, 2001. `http://www.dfki.de/~rafer/um01-ml4um-ws/proceedings.html`.

[9] H.E. Kyburg. Bayesian and non-bayesian evidential updating. *Artificial Intelligence*, 31:271–293, 1987.

[10] H.E. Kyburg. Higher order probabilities and intervals. *International Journal of Approximate Reasoning*, 2:195–209, 1988.

[11] S. Sen and N. Arora. Learning to take risks. In *Working Papers of the AAAI-97 Workshop on Multiagent Learning*, pages 59–64, 1997.

[12] S. Sen and M. Sekaran. Individual learning of coordination knowledge. *Journal of Experimental & Theoretical Artificial Intelligence*, 10:333–356, 1998.

[13] S. Sen and G. Weiss. Learning in multiagent systems. In Weiss G., editor, *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, pages 259–298. MIT Press: Cambridge, Mass, 1999.

[14] D.J. Spiegelhalter, N.G. Best, and B.P. Carlin. Bayesian deviance, the effective number of parameters, and the comparison of arbitrarily complex models. Technical report, MRC Biostatistics Unit, Cambridge, UK, 1998.

[15] D.J. Spiegelhalter and E.C. Marshall. Strategies for inference robustness in complex modelling: An application to longitudinal performance measures. Technical report, MRC Biostatistics Unit, Cambridge, UK, 1999.

[16] P. Wang. Confidence as higher-order uncertainty. In *Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications*, pages 352–361, 2001.

[17] G. Widmer. Tracking context changes through meta-learning. *Machine Learning*, 27:256–286, 1997.

[18] I. Zukerman and D.W. Albrecht. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11:5–18, 2001.

[19] I. Zukerman, D.W. Albrecht, and A. Nicholson. Predicting users' requests on the www. In *UM99 – Proceedings of the Seventh International Conference on User Modeling*, 1999.