EVIDENTIAL REASONING WITH TEMPORAL ASPECTS

Dr. Thomas C. Fall

Advanced Decision Systems
201 San Antonio Circle, Suite 286
Mountain View, California 94040-1289

ABSTRACT

In the real world, one usually cannot gather enough evidence to completely determine the activity of a system. Typically, different pieces of evidence tell you about different aspects of the system with different certainties. Correlating these at a given point of time is a much studied problem. The problem becomes even less tractable when the evidence is acquired at different points in time. The system described in this paper uses frame-like objects called "models" that propagate the effects of a piece of evidence through time and uses Gordon and Shortliffe's theory to combine the effects of the active models. These models do not require either a great deal of storage or that the evidence be processed in temporal order. Further, they seem to be a construct that the experts in our problems easily relate to. Results with test problems are consistent with the estimates of experts and run in not unreasonable time.

I WHAT THE SYSTEM NEEDS TO ACCOMPLISH

A. An Example Problem

1. Problem Background

Several systems,currently under development, address problems which share three characteristics: 1) In each problem, the goal is to determine and/or predict the behaviors of a class of objects. 2) The sources of information available vary in accuracy and descriptive power. 3) Each object can be ultimately classified as being in one of a fixed number of states. Further these states can be arranged hierarchically. Unfortunately, the problems these systems are designed to solve cannot be publicly discussed. However we developed the core processing system to be domain independent. So I have made up an example problem that will show the kinds of issues the processing part of the system addresses but avoids the real life knowledge content.

2. The Example

Let's say we're in a town with three competing taxicab companies: the Blue Cabs, the Green Cabs and the Orange Cabs. The owner of the Blue Cab Company feels that if he can set up a system to determine the whereabouts of all the competing taxis, he can get a competitive advantage by dispatching his cabs to areas where the competitors are underrepresented and keeping his cabs out of areas where they are overrepresented. He gets his information from reports from his drivers or from listening in to his competitors' dispatches. His drivers might only be able to identify which company the cab was with, but would be fairly precise as to activity. On the other hand, when overhearing the competitors' dispatches, the identity of the cab might be precise, but the activity might not. The knowledge of the activity of each cab can be diagramed using a hierarchy as illustrated in Figure 1.

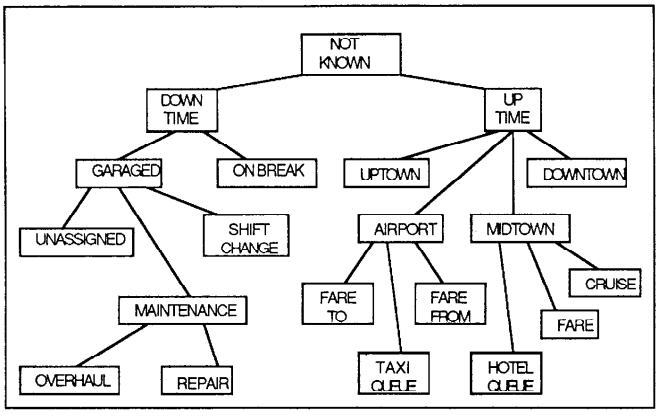

Figure 1. Activity Hierarchy for a Taxi Cab

Numbers are assigned to each slot of the diagram according to the certainty we have that the cab is

engaged in that activity. If we had absolutely no knowledge, then we would put 1.0 in the "not known" slot and 0.0 elsewhere. If we were sure that it was garaged, but could not come to any more definite conclusion than that, then we would put 1.0 in the "garaged" slot and 0.0 everywhere else. If we were 100% sure that the cab was garaged, and we were 50% sure that he was garaged because of shift change, then we would put 0.50 in "garaged", 0.50 in "shift change" and 0.0 everywhere else. The total certainty in the hierarchy is always 1.0 and as we get more and more specific knowledge, that amount will be pushed lower into the hierarchy. Each piece of evidence yields an assignment to the diagram and Gordon and Shortliffe's work [1] allows us to combine the assignments from several pieces of evidence.

3. A Naive Method for Prediction

These hierarchy diagrams are, unfortunately, only a snapshot at a particular instant. Our pieces of evidence come in at various times and all need to be combined to get an evaluation for a particular point of time. Further, that point of time will be in the future if we want to use our system to predict activities. One way is to use our knowledge of probable future behavior to project the effects of evidence through time. For instance, say we know that usually a pickup from the airport would be going to midtown hotel after which he would wait in the hotel's cab queue for another fare. One would predict then that if a cab were in the airport queue, then in 30 minutes there would be a 30% likelihood that the cab would have picked up a fare and a 25% likelihood that he'd picked up a fare and had already dropped him off and was waiting in the hotel queue for another fare. The remaining 45% would be relegated to not known. One of our drivers reports at 1:30 that he is 80% sure that cab Green-4 was in the airport taxi queue. Using our knowledge, we would say that in 30 minutes, there is a 24% (0.8 x 0.3) likelihood that Green-4 will be cruising to the hotel, a 20% (0.8 x 0.25) chance that he will already be in the hotel queue and there would be 56% uncertainty. We could fill this into the hierarchy diagram for 2:00 as one piece of evidence.

If there were two Blue cabs at the airport and each made a separate report, the combination of the two pieces of evidence would yield a 36.11% certainty the cab would be carrying a fare to the hotel at 2:00, a 29.20% chance that the cab was in the hotel queue and now only a 34.69 % uncertainty. Which is sort of questionable. If there are 10 Blue cabs who each reported separately into the system, then the combination of evidence yields 61.88% for the cab having a fare, 36.32% for the cab being in the hotel queue and only 1.80% for not-known. In repeated additions of similar pieces of evidence, the slot which is lowest in the hierarchy eventually dominates. If there are two competing slots that do not have an ancestral relationship, the larger will dominate. So, even though there is a difference of only 4% between "fare" and "hotel queue" in the report (24% vs. 20%), after 60 applications there would be 95.60% in the "fare" slot, 4.40% in the "hotel queue" and a negligible quantity in "not-known". This conclusion is clearly counterintuitive.

4. Deficiencies of the Naive Approach

This procedure for propagating the effects of evidence has two main deficiencies:

a. The procedure for projecting effects through time is inflexible. If a taxi was doing one activity now, it would project a probability distribution of activities for each time in the future. Unfortunately, this does not take into account other history for the taxi. For example, though we know it is now in the airport queue, it may be that 5 minutes ago it was still driving the fare to the airport. This would imply that it is still fairly far back in the queue.

b. The Gordon and Shortliffe combination method assumes that the pieces of evidence are independent. The example above shows what can happen when you treat several reports which are all just separate reports of one activity as if they were independent.

These deficiencies are closely related and can be to a large extent solved by the introduction of an additional mechanism, the "model".

B. <u>The "Model" Mechanism</u>

1. The Model Instantiated

The "model" is a frame-like object whose slots are activities that a taxi would engage in along with the minimum and maximum residency time in each of them. For instance, a model might have "fare to airport" which would take the cab a minimum of 20 to a maximum of 50 minutes followed by "wait in airport taxi queue" for 30 to 60 minutes, "fare from airport" for 20 to 50 minutes and finally "wait in hotel queue" for 20 to 40 minutes. If a report showed that the taxi was in a state with some certainty and that state was one of these four or a child of them, the model would be instantiated with the given certainty. The residency times of the instantiation would be fixed according to the report time. For instance, say at 1:00 we hear the Orange dispatcher tell Orange-7 to go ahead and join the airport queue after dropping off his fare. This data only tells us that Orange-7 is

probably driving out to the airport (say 60% sure), not what his ETA is. The following diagram shows how the other states would be predicted.

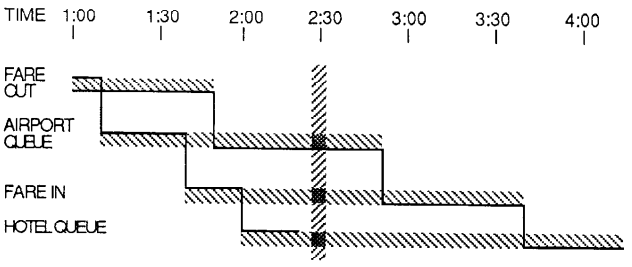

Figure 2. A Model as Instantiated by a Report at 1:00

Assuming our finest time resolution is to 10 minutes, the earliest the taxi could be at the airport is at 1:10. If the taxi had just started, it could take him 50 minutes. If he had a minimum wait at the airport, 30 minutes, and a minimum ride back, he could be at the hotel at 2:00. However, if he had a long ride out and a long wait, he could still be in the airport queue till almost 3:00. The hatched lines show the possible times the taxi could be in each state.

If we were trying to predict what the taxi was doing at 2:30, this model would say that he could be waiting in the airport queue, driving to the hotel or in the hotel queue. If we divided the certainty of being in this model by the number of possible states at 2:30, we would get a certainty of 0.2 for each.

2. The Model Updated

Now, say a Blue driver reports at 2:10 that he's 70% sure that he sees Orange-7 in the airport taxi queue. Combining the two certainties using the Gordon Shortliffe theory gives us a new certainty for the model of 0.88 = (0.7 x 0.6) + (0.7 x (1-0.6)) + ((1-0.7) x 0.6). Also, we now know that he cannot start driving back until 2:20 and even if he has the shortest drive, he won't be in the hotel queue until 2:40. Thus, at 2:30, Orange-7 could only be in one of two states. If we divide the 0.88 between these two states, we get 0.44 apiece as compared to the 0.2 we had from just the original piece of evidence.

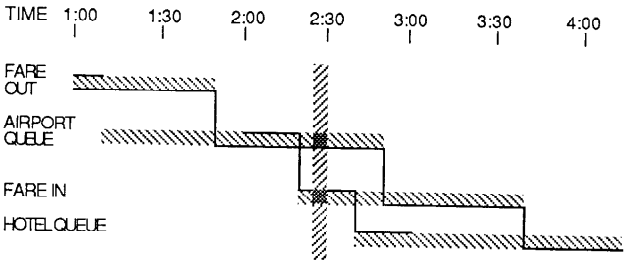

Figure 3. The Model Updated by a Report at 2:10

Earlier, we had said there were two problems with our original system. This model mechanism addresses the first in that it clearly is flexible and does account for history from other evidence. For instance, the second report in our example had an effect on the minimum residency for the subsequent states, but had no effect on the maximum, since the first piece was still the constraint for it.

As for the deficiency due to dependency, the second of the two, this mechanism at least partially addresses it. For instance, repeated reports early in the model would be combined as if they were independent to give the net effect of increasing the certainty of being in that model. However, when we look into the future, that model may predict several possible activities, so that certainty would be split among them. Only a piece of evidence temporally distinct from the first flurry can have the effect of reducing the number of states.

Note that the evidence could have been processed in the reverse order with the same results

II HOW THE SYSTEM ACCOMPLISHES IT

The systems developed consist of four major parts: 1) the hierarchy of states, 2) the set of models that describe the patterns of behavior, 3) a report conditioner that takes the reports in the form they are received and converts them to the form (state object-id time certainty) and 4) the engine that performs the procedures. The first three are domain dependent and the last is domain independent. This section will go a little more deeply into the architecture of the engine.

A report is processed by the report conditioner to produce the data item (maintenance Orange-7 4:00 0.50). Let's say we have a model where the state "garaged" appears and another where the state "overhaul" appears. Referring back to the state hierarchy, illustrated in Figure 1, we see that

"garaged" is a parent of maintenance. That is, if Orange-7 is in "maintenance" then it is surely "garaged", so we would want to call that model with the full 0.50 certainty. The other model with the state "overhaul" is more problematic. In the current system we make the ad hoc valuation by calling the model with half the certainty of the parent state (since in this case there are two parents). This could be changed to reflect greater knowledge, using countings for instance. This process is handled by a function (state-relation state-1 state-2) that returns the value 0 if state-1 and state-2 are not related, 1 if state-1 is above state-2 and 1/n if state-1 is one of n children of state-2. If state-1 is further below state-2, it returns the product of the intermediate steps.

Once it is determined that there is a non-zero state-relation between the event state and a state on a model, another module is brought into play that adjusts the timings of that instance of the model. This mechanism is what implements the timing adjustments discussed in the example of Section I.B, "The 'Model' Mechanism".

A couple of additional comments are in order about the time adjustment function. First is that if a state occurs several times on a model, the model will be instantiated for each occurence. The second is that there are two phantom states: "begin" and "end". Every instance of a model has the "begin" at its start and "end" at its finish where both have a zero time duration. If an event state does not match any states on the instance, a match to "begin" or "end" is attempted. In the case of the "maintenance" report at 4:00 from above, it would have the effect on the example model of Section I.B of closing it off at 4:00 and incrementing its certainty. If there were no match on these two phantom states either, then the certainty of the model would be decremented. An incoming piece of evidence will be used to adjust timings of instances already on the blackboard. Only if it cannot will it be used to instantiate new models.

Typically, the question the system is supposed to answer is what does the accumulated evidence tell us about the object's activities at some particular point in time. Depending on the evidence, there might be many models which are active at that time. This is the realm of the combining function which uses the methods of Gordon and Shortliffe. Each model on the blackboard active is treated as a piece of evidence. For instance, in Section I.B we discussed how the first piece of data would instantiate a model and how we would use that to make a prediction for 2:30. We use that prediction to assign a valuation of 0.2 to each of "airport queue", "fare in" and "hotel queue", 0.4 to "not known" and 0.0 elsewhere. The combining function uses the Gordon and Shortliffe techniques to combine this valuation with valuations derived from other active models to arrive at the assessment of likely activities of Orange-7 at 2:30.

III DISCUSSION OF THE SYSTEM IN OPERATION

A common activity can be on several models. A report of this activity would call all of them. Of course, when we looked into the future where these models would be predicting activities that conflict with activities predicted by other models also called, when we combined them, they would cancel each other out so that we wouldn't get too much information out. But that is intuitive; a common activity should not have a lot of predictive power.

In the real world. we can also get contradictory evidence. To incorporate this type of evidence in our implementation, each piece of evidence either combines and enhances models already on the blackboard if it is compatible, or it decrements the certainty of the existing models and adds new models that it is compatible with (these new ones have their certainty decremented by any evidence already posted and in conflict). Again, the effect of these models will tend to cancel one another out, but a preponderance of complementary evidence will push the prediction in its direction.

This system has been implemented in ZetaLisp on a Symbolics 3670 for a problem which has about 30 states and 10 models. In this problem, we receive a report that could correspond to one of several candidates and we would like to narrow the field. The system projects the the effect of known evidence about each candidate to the time of the report of the unknown vehicle for comparison. The candidates are ranked by their degree of similarity to the unknown. In a typical test problem involving two candidates, the experts felt the evidence indicated one was about 60 to 70 percent likely to be the correct one. Our system required about 5 minutes to analyze the evidence, about 50 reports, to arrive at a 64% likelihood for that candidate. Further, the experts can relate to the system's reasoning process, since they use similar internal constructs.

As has been pointed out by several authors, some systems of handling uncertainty have regions of parameter values where they are unstable. For instance, Lesmo, Saitta and Torasso [2] cite a case in PROSPECTOR where a 10% relative error in parameters causes a 50% relative error in the output

certainty, i.e. 0.53±0.29. Our experience with models is that they tend to make the system stable. That is changes in input parameters tend to only make proportional changes in output certainties. About the only exception is a change in residency time which might add or detract states from a prediction.

Another issue is control. Martin-Clouaire & Prade [3] discuss the need for trying to use the inferences which will have the greatest effect in order to prune the number of inferences that have to be processed. The models effectively replace large numbers of rules, but pruning is still necessary. If too many models get on the blackboard and/or an activity calls too many (certain common activities may be on numerous models) the machinery that determines compatibility can be seriously slowed. Our implementation has a "global rejection value" (GRV). If evidence calls a model or decrements a model to below this global rejection value, that model will be rejected. This test is done at many points in the implementation. For instance, the rejection test is done before the compatibilty test, since compatibility chews up CPU cycles. In fact the evidence is tested right at the front. Usually we run with the GRV set at 0.01 which seems to allow weak evidence to have some effect without slowing the system unreasonably. At this setting, a piece of conditioned evidence will process in about 5 seconds. In contrast, when set at 0.0, some pieces of evidence required 10 minutes for processing, since they called up a mammoth number of models.

In sum, using the Gordon and Shortliffe theory in combination with the model concept to temporally propagate the effect of evidence mimics much of the expert's reasoning and achieves results that are intuitive to the expert.

REFERENCES

[1] Gordon, J. & Shortliffe, E.H., "A Method for Managing Evidential Reasoning in a Hierarchical Hypothesis Space" Artificial Intelligence 26 (1985) 323-357

[2] Lesmo, L., Saitta, L. & Torasso P., "Evidence Combination in Expert Systems" International Journal of Man-Machine Studies 22:3 (1985) 307-326

[3] Martin-Clouaire, R. & Prade, H., "On the Problem of Representation and Propogation of Uncertainty in Expert Systems" International Journal of Man-Machine Studies 22:3 (1985) 251-264