

Downward Transfer of Probabilities in AI

By Paul Almond, 15 October 2006

Website: www.paul-almond.com
Email: info@paul-almond.com

© Copyright Paul Almond, 2006. All Rights Reserved.

Downward Transfer of Probabilities in AI

By Paul Almond, 15 October 2006

Introduction

In previous articles [1,2,3,4,5,6] I proposed an approach to artificial intelligence (AI) in which probabilities of meanings obtained from probabilistic interpretation of partial models are stored in a hierarchy. Information on the bottom level of the hierarchy, where input and output events occur, is abstracted on higher levels.

Downward transfer of probabilities involves information flowing from higher levels of the hierarchy back down to lower levels, with high level meaning probabilities influencing low level probabilities. This turns the high level meanings into predictions of future inputs and outputs. It can be viewed as “filling in of details” from high level meanings.

It is proposed that planning is a special case of modelling. Planning is based on the system’s predictions of its own future outputs, which are used to constrain a search for optimum behaviour. The planning process really occurs within the probabilistic hierarchy of meanings. This requires abstracted meanings at higher levels to be realized as future input and output predictions. Downward transfer of probabilities is therefore needed to allow the AI system to plan.

Downward transfer of probabilities is an important process in the AI system, but has not yet been described in any detail. This article will give an idea of how downward transfer of probabilities could work.

As before, I point out that readers may want to compare the hierarchical system discussed here with that proposed by Jeff Hawkins [7,8] with which it has some superficial similarity.

Purpose of Downward Transfer of Probabilities

Downward transfer of probabilities “fills in details” at lower levels of the hierarchy, using meanings that probabilistic interpretation of partial models generates at higher levels.

This is needed because it is at the bottom level of the hierarchy that the system’s predictions about its future inputs and outputs are made. The probability values stored at the bottom level of the hierarchy indicate the probability of future input or output events involving values of “1” or “0”. The AI system uses modelling of its own future behaviour for planning. The system’s predictions of its own future outputs are used to constrain a search for optimum outputs. These predictions are probability values of future output events that come from downward transfer of probabilities, and predictions of future inputs are probabilities of future input events that also come from downward transfer probabilities. This is the only reason for any level of the hierarchy above the bottom level

of input and output events to exist – to generate high level meaning probabilities and then pass information about these back down to the bottom level of the hierarchy where it can be used to constrain the search for optimum outputs and indicate the probabilities of different results of making various outputs.

An Informal Description of the Process

Low level observations or meanings suggest higher level meanings, which in turn suggest further low level meanings.

As an example, suppose that we see part of a cat – its head and two legs – the rest of it being hidden by long grass. Probabilistic interpretation of partial models (or the nearest thing to it in humans) would cause a probability for the high level meaning for a cat to emerge.

This is where downward transfer of probabilities comes in. Suppose there are other objects, obscured to some degree, that might be the other two legs. Considering them in isolation we may think the probability that these are a cat's legs is low. We may not even be able to see them, if they are completely obscured, giving a very low probability that cat parts are there. The higher level probability of the cat, however, increases the probability that some of these lower level objects are also a cat's legs.

A Formal Description of the Process

We have a high level meaning bit *Bit H* with some probability $P(H)$. $P(H)$ has been determined by probabilistic interpretation of a partial model algorithm (meaning extraction algorithm) for a particular index.

We cannot be sure whether the partial model algorithm should really be returning “0” or “1”. The “real” value in the conceptual hierarchy is hidden from us. We have only the probability in the functional hierarchy. We consider the cases for the partial model algorithm outputting “1” and “0” separately, using the different paths through the partial model algorithm's decision tree.

List Paths Through Decision Tree

Any path through the decision tree represents a particular sequence of bits being read from the lower levels of the hierarchy and involves the partial model algorithm giving a result of “0” or “1”. The bits that are read may be at different locations in different paths because the reading is controlled by the partial model algorithm and the value that a bit is found to have could change the algorithm's subsequent behaviour in terms of the positions from which it reads bits.

We list every path through the partial model algorithm's decision tree. One of these must be the actual path that the partial model algorithm follows in the conceptual hierarchy.

Compute Naïve Object Probabilities for Paths

For each of these paths through the decision tree we compute a *naïve object probability* [4]. This involves obtaining the low level probability for each instance of a bit being read in this path through the decision tree. If the decision tree path involves a particular low level bit being read and found to have a value of “1” then this value is simply the probability stored for that bit in the functional hierarchy. If the decision tree path involves a low level bit being read and found to have a value of “0” then this value is 1 minus the probability stored for that bit in the functional hierarchy. All of these values are multiplied together to give the naïve object probability for the path through the decision tree.

Determine the Probability for a Low Level Bit

We select some bit *Bit L* at a lower level in the hierarchy. Bit L has the probability $P(L)$ stored for it in the functional hierarchy. We want to know the new value to assign - to *transfer down* - to $P(L)$.

First, we consider the case for the high level meaning bit being “1”. Our consideration is restricted to those paths through the decision tree involving the partial model algorithm returning a result of “1”.

We then consider each such path (for which the resultant meaning bit is “1”) through the decision tree where Bit L has a value of “1”. For each of these paths we obtain the naïve object probability previously computed. These probabilities are summed to give P_{11} , a total naïve object probability for Bit L being “1” and the high level meaning bit being “1”.

We then consider each such path (for which the resultant meaning bit is “1”) through the decision tree where Bit L has a value of “0”. For each of these paths we obtain the naïve object probability previously computed. These probabilities are summed to give P_{10} , a total naïve object probability for Bit L being “0” and the high level meaning bit being “1”.

We then evaluate a resultant naïve object probability, N_1 , for the low level bit being “1” and the high level meaning bit being “1” as follows:

$$N_1 = P_{11} / (P_{11} + P_{10})$$

It may be asked why we need to do this to get a new probability of the low level bit being “1”, when we already have P_{11} . This is done as a kind of normalisation, to eliminate any influence from partial model decision tree paths that do not involve reading of the low level bit that interests us.

This total naïve object probability is then multiplied by $P(H)$, the probability stored in the functional hierarchy for this high level meaning. The resultant probability, R_1 , is the probability that the high level meaning bit is “1” and Bit L is “1”:

$$R_1=N_1 \times P(H)$$

Next, we consider the case for the high level meaning bit being “0”. This means that our consideration is restricted to those paths through the decision tree which involve the partial model algorithm returning a result of “0”.

As before, we then consider each such path (for which the resultant meaning bit is “0”) through the decision tree where Bit L has a value of “1”, obtaining the previously computed naïve object probability for each. These naïve object probabilities are summed to give P_{01} , the total naïve object probability for Bit L being “1” and the high level meaning bit being “0”.

We then consider each such path (for which the resultant meaning bit is “0”) through the decision tree where Bit L has a value of “0”. For each of these paths we obtain the naïve object probability previously computed. These probabilities are summed to give P_{00} , a total naïve object probability for Bit L being “0” and the high level meaning bit being “0”.

We then evaluate a resultant naïve object probability, N_0 , for the low level bit being “1” and the high level meaning bit being “0” as follows:

$$N_0=P_{01}/(P_{01}+P_{00})$$

N_0 is then multiplied by $1-P(H)$ (that is to say, the probability that the partial model algorithm will return a result of “0”). The result, R_0 , is the probability that the high level meaning bit is “0” and Bit L is “1”:

$$R_0=N_0 \times (1-P(H))$$

We can now compute a new value for the probability of Bit L being “1” by adding these two results:

$$P(L)=R_1+R_0$$

That is to say:

Probability that Bit L is “1” = Probability that the high level meaning bit is “1” and Bit “L” is “1” + Probability that the high level meaning bit is “0” and Bit L is “1”

This new probability value, $P(L)$, is used to replace the value currently stored for Bit L in the functional hierarchy. The probability value has been *transferred down* to Bit L.

Multiple High Level Meanings

This article has only described how downward transfer of probabilities could work from a single high level meaning probability. The approach would need further development to

be useful. Real situations will involve many high level meaning probabilities and downward transfer of probabilities cannot occur from all of them separately because each low level bit has a single probability value.

One way of dealing with this would be to give each higher level meaning probability a “vote” in any given low level probability, but this would not be entirely satisfactory. It is a rather arbitrary method without any rigorous reasoning to support it. Furthermore, most partial model algorithms of any practical use will not read most low level bits, so how should this be dealt with? Should we just use those instances where a particular low level bit is actually read in a path through a partial model algorithm decision tree?

Another approach would be to combine all the decision tree paths, so that we have every possible combination of paths through the decision trees of every partial model algorithm being applied with every index. Some of these paths could be eliminated because they involve the same bit being read with different values by two different partial model algorithms, or by the same partial model algorithm with different indices. The higher level probability values stored in the hierarchy could be combined for each of these paths, and the probabilities stored for the low level bits similarly combined to generate a naïve object probability.

Such an approach would be on firmer ground, but the NP-hardness aspect of this makes all this computation impractical. This idea useful, however, because it represents the ideal situation for downward transfer of probabilities and allows us to describe what the “pure” process is. This also relates to the issue of avoiding contradictions between high level probabilities: some kind of interaction between high level probability values is required for this.

Some compromise is needed between just trying to apply downward transfer of probabilities from different high level meanings in isolation and combining all the decision tree paths in a way that NP-hardness makes impractical to process. This would give results which have imperfections, but would provide a process which can be practically executed.

Issues from Previous Articles

An important issue in applying a process like this is prevention of convectional delusion. Convectional delusion was described previously [5] as a situation in which a high level meaning reinforces probabilities that then increase its own probability. This suggests that a change in a low level probability should not be allowed to increase the probability of a higher level meaning probability that caused it, but it is not as simple as saying that, once information is propagated down by downward transfer of probabilities, it should never be allowed to propagate back up. A high level meaning probability may influence a low level probability which should then influence a different high level meaning probability by probabilistic interpretation of a partial model algorithm. This must be done without the original high level meaning being reinforced, or convectional delusion will start.

In a previous article [5] I stated that downward transfer of probabilities would need to maintain consistency and that any low level probabilities produced by downward transfer of probabilities from a high level probability must merely lead to probabilistic interpretation producing that same high level probability. This, however, cannot be true and needs correcting. Low level probabilities are consistent with high level probabilities that were used to transfer changes down on the assumption that information will not be propagated back up the hierarchy to the original high level probabilities.

I also stated in the same article that propagation would occur repeatedly, with information being transferred up the hierarchy by probabilistic interpretation of partial models, then back down by downward of probabilities, then back up, and so on – presumably until stability results. This also needs revision. Such a process would not lead to stability but to convectional delusion. This raises the issue of what the limits on propagation should be.

Limits On Propagation

This article has only discussed the basic idea of downward transfer of probabilities. The propagation process would need to be limited in some way to prevent convectional delusion, but not limited so much that lower level probability changes resulting from downward transfer of probabilities are not available at all for higher level meaning extraction. This requires more consideration beyond the preliminary treatment in this article.

Conclusion

Downward transfer of probabilities is an important process in the probabilistic hierarchy of the AI system that I have been discussing in previous articles.

This article has described a basic way of implementing downward transfer of probabilities. The approach presented here needs further development, but this article gives an idea of what downward transfer of probabilities means in a mathematical sense, rather than in the vague, informal sense in which the process was described in previous articles. The emphasis has been on properly describing downward transfer of probabilities is: efficiency and practicality can come later.

Downward transfer of probabilities involves consideration of the possible decision tree paths associated with a partial model algorithm for a particular index. In the conceptual hierarchy only one of these decision tree paths is valid. In the functional hierarchy we do not know which one it is. We therefore examine the set of all possible decision tree paths for a particular partial model output, the probability of which has been computed in the functional hierarchy, and count the paths for which a particular low level bit (for which we want to compute a transferred-down probability) is found to have a value of “1” and those for which it is found to have a value of “0”.

The approach as it is described now has a limitation: it only deals with downward transfer of probabilities from a single high level extracted meaning probability. In reality, many such high level meanings would need to be used and a working process would need to combine these. This may involve a compromise between mathematical rigour and practicality.

References

- [1] Web Reference: Almond, P. (2006). *How AI Would Work*. Retrieved 4 September 2006 from <http://www.paul-almond.com/HowAIWouldWork.pdf>.
- [2] Web Reference: Almond, P. (2006). *Occam's Razor Part 6: Partial Models as "Envelopes"*. Retrieved 1 March 2006 from <http://www.paul-almond.com/OccamsRazorPart06.htm>.
- [3] Web Reference: Almond, P. (2006). *Occam's Razor Part 7: Hierarchy and Ontology*. Retrieved 30 April 2006 from <http://www.paul-almond.com/OccamsRazorPart07.htm>.
- [4] Web Reference: Almond, P. (2006). *Occam's Razor Part 8: Modelling in Artificial Intelligence*. Retrieved 9 June 2006 from <http://www.paul-almond.com/OccamsRazorPart08.pdf>.
- [5] Web Reference: Almond, P. (2006). *Occam's Razor Part 9: Representation and Planning of Actions in Artificial Intelligence*. Retrieved 29 July 2006 from <http://www.paul-almond.com/OccamsRazorPart09.pdf>.
- [6] Web Reference: Almond, P. (2006). *AI as a Boundary System*. Retrieved 17 September 2006 from <http://www.paul-almond.com/AIAsABoundarySystem.pdf>.
- [7] Hawkins, J., Blakeslee, S. (2004). *On Intelligence*. New York: Henry Holt.
- [8] Web Reference: George, D., Hawkins, J. (?). *Belief Propagation and Wiring Length Optimization as Organizing Principles for Cortical Microcircuits*. Retrieved 24 April 2006 from <http://www.stanford.edu/~dil/invariance/Download/CorticalCircuits.pdf>.