An Attempt to Generalize AI Part 6: Measuring Relevance

By Paul Almond

8 April 2010

Website: http://www.paul-almond.com
E-mail: info@paul-almond.com

This is the sixth in a series of articles attempting an overview of how minds may work and how similar systems could be implemented in computers. Previous articles have described a probabilistic hierarchy based on *patterns*. A pattern has a specification describing a set, or population, of *pattern instances*, distributed throughout a hierarchy containing the pattern instances of all the patterns. Each pattern's set of pattern instances is used to obtain statistical information for probabilistic predictions. Each pattern's population of pattern instances is to be described in a very general way, to provide a very general ontology. The fourth article discussed the need to focus the hierarchy on what is *relevant*, and how this requires the ability to remove pattern instances from the hierarchy, and the fifth article modified the description of the hierarchy, making it completely probabilistic – without reliance on any special case of certainty – to enable this. The hierarchy will be made relevant by an exploratory process that extends it where it is likely to be relevant, and prunes it where it is less relevant. Such a process will need a way of measuring the relevance of individual pattern instances in the hierarchy, or of parts of the hierarchy. This article describes such a measurement process. The measurement process is made possible by the way that the action selection process, described in the second article, requires probabilistic predictions of specific pattern instances corresponding to future evaluation function score values. This enables a measurement process in which these pattern instances are regarded as relevant, relevance then being back-propagated from them through the rest of the hierarchy.

Table of Contents

List of Abbreviations

AI	artificial intelligence
EFS	evaluation function score

1 Introduction

This article is the sixth in a series about artificial intelligence (AI) and how our own minds might work. The first article, *An Attempt to Generalize AI - Part 1: The Modeling System*, is available at http://www.paul-almond.com/AI01.pdf.[1] The second article, *An Attempt to Generalize AI - Part 2: Planning and Actions*, is at http://www/paul-almond.com/AI02.pdf.[2] The third article, *An Attempt to Generalize AI - Part 3: Forgetting*, is at http://www.paul-almond.com/AI03.pdf.[3]

These three articles together described a hierarchy based on *patterns*, which are sets of *pattern instances*, and were intended to give an idea of how humans may model the world, plan actions and discard information from the model when it is no longer useful. However, some way would be needed of ensuring that only *relevant* information about the world would be represented in the AI system or a human mind: Some way would be needed of ensuring that only *relevant* parts of the hierarchy would be represented.

This issue was discussed in the fourth article, *An Attempt to Generalize AI - Part 4: Modeling Efficiency*, which is at http://www.paul-almond.com/AI04.pdf.[4] It was suggested in that article that pattern instances should be allowed to have *incompletely specified pattern inputs.* The hierarchy might contain information about some of the pattern inputs to a pattern instance, while others, for practical purposes, would be non-existent. This would allow the removal of pattern instances from the hierarchy without having to remove what was "above" them, and it could simplify the connection of new pattern instances into the hierarchy. This would be done so that the hierarchy could be "pruned" by some process seeking to maximize its relevance. For this to be practical, the hierarchy needed to be *completely* probabilistic: It had previously relied on the special case of pattern instances known about with certainty (previously referred to as "fixed" pattern instances) and this reliance needed removing. This issue was dealt with in the article immediately before this one, *An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy*, which is at http://www.paul-almond.com/AI05.pdf.[5]

That leaves things ready for this article, which will start to deal with the issue of how to ensure *relevance* in the hierarchy. Relevance will be provided by an exploratory process

[1] Almond, P. (2010). An Attempt to Generalize AI - Part 1: The Modeling System. *paul-almond.com*. http://www.paul-almond.com/AI01.pdf. (Also available at http://www.paul-almond.com/AI01.doc.)
[2] Almond, P. (2010). An Attempt to Generalize AI - Part 2: Planning and Actions. *paul-almond.com*. http://www.paul-almond.com/AI02.pdf. (Also available at http://www.paul-almond.com/AI02.doc.)
[3] Almond, P. (2010). An Attempt to Generalize AI - Part 3: Forgetting. *paul-almond.com*. http://www.paul-almond.com/AI03.pdf. (Also available at http://www.paul-almond.com/AI03.doc.)
[4] Almond, P. (2010). An Attempt to Generalize AI - Part 4: Modeling Efficiency. *paul-almond.com*. http://www.paul-almond.com/AI04.pdf. (Also available at http://www.paul-almond.com/AI04.doc.)
[5] Almond, P. (2010). An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy. *paul-almond.com*. http://www.paul-almond.com/AI05.pdf. (Also available at http://www.paul-almond.com/AI05.doc.)

which extends the hierarchy, by allowing pattern instances to be added, where it appears relevant, and reducing the hierarchy, by removing pattern instances, where it appears insufficiently relevant to justify its existence. In this way, the hierarchy becomes a dynamic structure, continually changing as inputs/outputs occur and relevance is sought.

Such an exploratory process will require a way of measuring the relevance already provided by any part of the actual hierarchy in an AI system. The problem is made tractable because of the way in which the action selection process, described in the second article, *An Attempt to Generalize AI – Part 2: Planning and Actions*, works.[6] The action selection process, which is used to *drive* the system's behavior in a particular direction, relies on predictions of pattern instances which will be used for future input of a continually computed evaluation function score (EFS). The requirement for such specific predictions from the hierarchy provides a natural end-point – what the hierarchy is expected to produce – and a clear goal: The hierarchy needs to minimize the uncertainty in these particular pattern instances. This tells us what is most relevant and allows us to describe a process of back-propagation of relevance from these pattern instances, through the rest of the hierarchy, allowing the relevance of every part of the hierarchy to be measured. This is what will be dealt with in this article. The next article will build on this, showing how such a measuring process can be used as the basis for an exploratory process that continually modifies the structure of the hierarchy so that it generates the required information.

[6] Almond, P. (2010). An Attempt to Generalize AI - Part 2: Planning and Actions. *paul-almond.com*. http://www.paul-almond.com/AI02.pdf. (Also available at http://www.paul-almond.com/AI02.doc.)

2 A Note on Pattern Instances and Probability Values

This article will tend to discuss pattern instance as if each pattern instance in the conceptual hierarchy has one of two possible values – 0 or 1 – as this is the most obvious form for the hierarchy, and the easiest one to describe. With such a hierarchy, any pattern instance in the actual hierarchy in the AI system can be described by a single probability value that indicates the probability that the corresponding pattern instance in the conceptual hierarchy has a value of 1.

As will be discussed in this article, the purpose of the actual hierarchy is to minimize the uncertainty in specific pattern instances. For a pattern instance described by a single probability value, this means getting its probability as far away from 0.5, and as close to 0 or 1, as possible.

It should be noted, however, that it is possible, in principle, for the conceptual hierarchy to take other forms. It could take a form in which each pattern instance has more than two possible values. In such a hierarchy, more than one probability value would be needed to describe a pattern instance. The general approach discussed in this article would still apply, however: It would just involve propagation with multiple probability values for pattern instances, and minimizing the uncertainty associated with a pattern instance would simply mean minimizing it for its multiple probabilities.

For the most part, I suggest that readers ignore this, and just assume we are dealing with a hierarchy of 0/1 pattern instances, each described in the actual hierarchy by a single probability value indicating our state of knowledge about it.

3 How Information Is Propagated Through the Hierarchy

3.1 The Meaning of Probabilistic Propagation

A number of processes have been described in previous articles for pattern instances to affect other pattern instances' probability values, in upwards and downward directions, so that probabilistic information is propagated through the hierarchy.[7] The kind of transmission of "effect" with which we are dealing when we consider such propagation of probability values is not a transmission of *real* effect. The actual hierarchy, with the probability values in the AI system, is a partial representation of the conceptual hierarchy. The pattern instances in the conceptual hierarchy do not pass probability values between each other. The probability values represent what we know about the hierarchy, and when one pattern instance "affects" another through logic application or statistics application, whether in an upward or downward direction, this means that what we know about one pattern instance is being used to tell us something about another.

3.2 The Basic, Probabilistic Propagation Processes

The following processes are used to propagate probabilistic information through the hierarchy: for pattern instances to affect the probabilities of other pattern instances.

3.2.1 Logic Application Upwards

The pattern instances which act as pattern inputs to a particular pattern instance can affect its probability, in accordance with their probabilities and the pattern logic for that pattern.

3.2.2 Statistics Application Upwards

The pattern instances which act as pattern inputs to a particular pattern instance can affect its probability in accordance with their probabilities and the previous statistics for pattern instances for that pattern obtained from logic application.

3.2.3 Statistics Application Downwards

The probability of a pattern instance, with the individual probabilities of its pattern inputs, together with the previous statistics for pattern instances for that pattern,

[7] Almond, P. (2010). An Attempt to Generalize AI - Part 1: The Modeling System. *paul-almond.com*. http://www.paul-almond.com/AI01.pdf. (Also available at http://www.paul-almond.com/AI01.doc.)
Almond, P. (2010). An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy. *paul-almond.com*. http://www.paul-almond.com/AI05.pdf. (Also available at http://www.paul-almond.com/AI05.doc.)

affects the probabilities of particular combinations of its pattern inputs. This, in turn, affects the probabilities of individual pattern inputs. Probabilistic information can therefore be propagated down from a pattern instance to the pattern instances serving as its pattern inputs.

3.2.4 Logic Application Downwards

I have not mentioned logic application *downwards* previously. I only mention it now for completeness. In principle, probabilities can be propagated downwards, from a pattern instance to its pattern inputs, just by looking at the pattern instance's probability and the probabilities of its pattern inputs. This would seem a less important process than the others, because logic application will only be worthwhile in parts of the hierarchy which are highly dependent on previous inputs/outputs – and the problem there will be finding out about the higher-level pattern instances. A system should be able to function just with upwards logic application and upwards and downwards statistics application.

An obvious way to start analyzing the relevance of the hierarchy is to try tracking the propagation of probabilistic information through it. We will now consider the above propagation processes in terms of how they relate pattern instances to each other. We will consider things from the point of view of a pattern instance which is *affecting* other pattern instances, and from the point of view of a pattern instance *being affected* by other pattern instances.

3.3 Information Propagation *From* a Pattern Instance

3.3.1 Logic Application and Statistics Application Upwards

These involve *one to many* relationships, because a single pattern instance can act as a pattern input to multiple pattern instances.[8]

3.3.2 Statistics Application and Logic Application Downwards

These are also one to many, because these processes involve propagating information from a pattern instance to its pattern inputs and a single pattern instance can use multiple pattern instances as its pattern inputs.

[8] They also involve many to one relationships, with regard to an individual pattern instance being changed by its inputs, but that is of less interest to us: What matters is that information from a particular pattern will spread out.

3.3.3 All ways of propagation *from* a pattern instance are one to many.

All the ways in which information can propagate *from* a pattern instance are one to many. When information enters the hierarchy, as a result of an input/output occurring and putting information into a bottom-level pattern instance, this pattern instance will tend to affect multiple other pattern instances. Each of these in turn affects multiple other pattern instances and so on. Information will tend to spread out *exponentially*. If we try to map the relationships by which information spreads out through the hierarchy, we will have a tree that branches out very quickly.

3.4 Information Propagation *Into* a Pattern Instance

3.4.1 Logic Application and Statistics Application Upwards

These are *many to one* relationships, because they involve a pattern instance's probability being affected by its pattern inputs, and a single pattern instance can have multiple pattern inputs. A single pattern instance can therefore be affected by multiple pattern instances in this way. This means that if we start at a single pattern instance and go back to the pattern instances which affect it, we will find multiple pattern instances.

3.4.2 Statistics Application and Logic Application Downwards

These are also many to one, because they involve a pattern instance's probability affecting the probabilities of its pattern inputs, and a single pattern instance can be a pattern input to multiple pattern instances, which all affect it.

3.4.3 All ways of propagation *into* a pattern instance are many to one.

All the ways in which information can propagate *into* a pattern instance involve many to one relationships. Viewed backwards, these are *one to many*. If we work *backwards* through the hierarchy, looking at a pattern instance and then at those that affect it and so on, we will find that each pattern instance tends to lead us back to multiple pattern instances, and so on. As with working forwards, the paths we follow will tend to branch out exponentially.

4 What parts of the hierarchy are relevant?

4.1 Analyzing the Actual Hierarchy

Now that we have discussed what is going on when information is propagating through the hierarchy, we need to look at how we might determine the usefulness of different parts of the hierarchy. We need a way of taking an actual hierarchy, in an AI system, and measuring the usefulness of each pattern instance in it. Before we start this, we should step back and consider what we mean by "useful". This all comes back to what the whole point of the hierarchy is.

4.2 The Purpose of the Hierarchy

The hierarchy's use is in the action selection process described in the second article of this series, *An Attempt to Generalize AI – Part 2: Planning and Actions*.[9] An evaluation function score (EFS) is continually computed, being treated each time as if it had been received as an external input to the AI system, so that EFS values are continually encoded as bottom-level pattern instances. When an output is required, each possible value is tried and the hierarchy updated as if the output had occurred with that value, the relevant changes to probabilities being propagated through the hierarchy. The pattern instance(s) corresponding to a future prediction of the EFS input is/are obtained from the hierarchy, and used to obtain an expected EFS value for each possible output value. This allows the better output value to be selected, and the output actually occurs with this value.

This process might be modified to some degree, and it is not the real planning process: The *real* planning process actually occurs as part of the hierarchy's modeling of its own behavior, as a natural part of the hierarchy's functioning. Nevertheless, this is the process by which the hierarchy is driven to produce a certain type of behavior. Even if it gets modified later to some extent, the important feature of it will still remain: *We are interested in predicting the values of pattern instances corresponding to specific future inputs (the EFS values) on the bottom level of the hierarchy.*

4.3 The Importance of the Action Selection Process

It is no accident that the action selection process works in this way: It gives us a well-defined requirement. We want the hierarchy to predict a specific, future input value and this means that we will want the uncertainty in the pattern instance(s) corresponding to that value to be minimal. We want the probability associated with that pattern instance

---

[9] Almond, P. (2010). An Attempt to Generalize AI - Part 2: Planning and Actions. *paul-almond.com*. http://www.paul-almond.com/AI02.pdf. (Also available at http://www.paul-almond.com/AI02.doc.)

to reflect as little uncertainty as possible – meaning that it should be as close as possible to 0 or 1. This gives us something measurable, and it suggests a possible opening for us to measure the performance of the hierarchy, or parts of the hierarchy, with reference to this objective, and optimize it to reduce the uncertainty in the predicted value.[10]

The action selection process provides a well-defined end-point, which can be regarded as relevant, and which can be us a starting point for a process of back-propagation of relevance.

4.4 Does the propagation of information *from* previous inputs/outputs tell us anything useful?

We might consider using the flow of information from the bottom-level pattern instances to tell us which pattern instances are relevant – viewing the pattern instances which get their probabilities strongly affected by the pattern inputs as particularly relevant, and therefore worthy of representation in the actual hierarchy – but the information spreads out exponentially to so many pattern instances that we could never represent them all.

The relationships themselves will tell us nothing useful, but what about the actual probability values? One pattern instance might affect another in a way that significantly reduces the uncertainty in its probability, while another pattern instance affects it in a way that only slightly reduces its uncertainty. Could we say that it is only interesting when pattern instances *strongly* affect others, and try to follow the “paths of strong effect” through the hierarchy?

This deals with part of the issue. The default probability value for a pattern instance is 0.5, and we know nothing of such a pattern instance. If we represented all of the hierarchy, much of it would remain like this, even after propagation of information through it, because the inputs/outputs would just not tell us much about a lot of the hierarchy. It would not be worth using these pattern instances for anything, which means it would not be worth computing probabilities for them in the first place: We should only be interested in parts of the hierarchy which have probability values that are significantly affected by occurrence of inputs/outputs. If we follow the paths through the hierarchy from the bottom-level pattern instances corresponding to previous inputs/outputs, through such pattern instances, we will be considering all of the hierarchy that could really interest us, and if we stray away from such paths we will just enter regions of the hierarchy where the probability values are all close to 0.5, and therefore useless.

[10] If some more sophisticated version of the action selection process were used, involving multiple EFS predictions, the general idea would be the same – as would the important measurability in this: We would just want to minimize the uncertainty in the predictions of whatever EFS values were involved.

This is only part of the issue, however. A pattern instance needs to be significantly affected by propagation of information from previous inputs/outputs to be of any interest, but being significantly affected does not mean that it *must* be of interest: It is merely the minimum requirement. Many paths through the hierarchy will lead to pattern instances that are strongly affected by previous inputs/outputs, but which have hardly anything to do with the probabilities of the pattern instances that interest us.

As an example, suppose that you are a detective investigating the particularly nasty murder of a celebrity in the USA. You may work out that the story of the murder will appear on the Internet, that people in Switzerland will read about the murder, and therefore that a fan of the celebrity *who works in the Swiss chocolate industry* will be horrified by the murder. You might be following the "flow of strong effect" here, but it would probably be leading you nowhere useful.

4.5 Does the propagation of information *into* pattern instances tell us anything useful?

4.5.1 Let's just consider the structure…

If it is useless to start at pattern instances corresponding to previous inputs/outputs and work our way forwards, what if we start at a pattern instance corresponding to a future input/output which we want to predict and work *backwards*, looking at what pattern instances affect it, and what pattern instances affect those, etc., all the way back to the pattern instances corresponding to previous inputs/outputs, the idea being that the pattern instances through which we passed are the important ones?

To start with, we will just consider the *structure* of the hierarchy – the way pattern instances are connected together – without considering any of the probability values allocated to pattern instances. In other words, we will look at what pattern instances a given pattern instance uses at its pattern inputs, but we will not look at any of the probability values themselves. As previously discussed, multiple pattern instances can directly affect a single pattern instance through logic application or statistics application. Suppose we start at a single, bottom-level pattern instance corresponding to a future pattern instance. We could find that a number of pattern instances directly affect it. For each of these, we could find a number of pattern instances affecting it in turn, and so on. As we work backwards, the number of pattern instances that we have to consider will, once again, increase exponentially. Eventually, we might be considering all of the hierarchy. We would eventually get back to the previous inputs/outputs along some paths, but other paths would lead us into regions of the hierarchy that are useless.

Maybe we should consider some approach that combines working forwards from previous inputs/outputs and backwards from future inputs/outputs? Such an approach will still have issues with exponential increase. Fortunately, there is a better way. I introduced this part of the discussion by saying that we would start by ignoring the probability values. What if we tried working backwards from a pattern instance corresponding to a future input/output, but this time we looked at the actual probability values describing individual pattern instances? We will now consider this kind of approach.

4.5.2 But what if we consider probabilities too?

Suppose we start at a bottom-level pattern instance corresponding to a future input/output which we want to predict, probabilistically. As before, we look at the pattern instances that directly affect the probability of this pattern instance by logic or statistics application. This time, however, we do not just look at whether or not a given pattern instance directly affects the one that interests us. We look at how strongly it affects it.

How can we determine how strongly one pattern instance affects another? This will be discussed in more detail; however for now we will just consider the probability of the pattern instance doing the affecting. All else being equal, if two pattern instances directly affect the pattern instance that interests us, and we know a lot about one of them – it has a probability value close to 1 or 0 – and we know little about the other one – it has a probability value close 0.5 – the one about which we know more will do more affecting.

Why this is the case should be obvious. As stated previously, the kind of transmission of “effect” that we are dealing with when we consider propagation of probability values is not a transmission of *real* effect.[11] The probability values represent what we know about the hierarchy, and when one pattern instance’s probability “affects” another’s probability through logic application or statistics application, what we know about one pattern instance is being used to tell us about another pattern instance. If we know a lot about a pattern instance, we might expect this knowledge to have a big effect on what we know about another pattern instance – to alter its probability value so that we are much less uncertain about it. If, however, we know hardly anything about a pattern instance, we should hardly expect our knowledge to have much effect on anything else. Consider, for example, a pattern instance with a probability of 0.5. We know nothing at all about this pattern instance. Does it have a state of 0 or a state of 1? We have no idea: We do not even know which state is more likely. If we tried to use this pattern instance in logic application or statistics application, this would tell us nothing: It would have no effect.

We could, then, look at all the pattern instances directly affecting the one that interests us through logic application and statistics application, upwards or downwards. We would look at the probability of each such pattern instance and that would give us an idea of how much effect it would have, *all else being equal*. We could store a *relevance*

[11] Section 3.1: The Meaning of Probabilistic Propagation, on page 7.

value for each pattern instance indicating how much effect it had on the pattern instance that interested us. This would give us a list of pattern instances, and how relevant each one is. We could now work back to each of these pattern instances and look at the pattern instances that directly affect it, allocating each one a relevance value, but now, we would not just base the relevance values on the probability values of the pattern instances, but also on the importance of the pattern instance they were affecting. That is to say, a pattern instance gets a lot of relevance by having a high probability and affecting a relevant pattern instance. We could continue to work back like this, propagating relevance backwards. Eventually we would reach the bottom-level pattern instances corresponding to previous inputs/outputs.

We could not use such a process to construct the actual hierarchy by itself. A process like this only tells us how important a particular pattern instance in an existing representation of the hierarchy, so we need the hierarchy there before we can use it. However, a process like this would tell us about which paths through the hierarchy were relevant.

Suppose that no attempt had been made to limit the size of the actual hierarchy in the AI system. Applying the process just described would mean we would have to process all of these pattern instance values, but suppose we prioritized? We could start at a pattern instance corresponding to a future input/output which we wanted to predict and work backwards. As we traced the propagation of information backwards, the paths would branch out, but we could prune branches which led us into irrelevant parts of the hierarchy. Whenever we encountered a pattern instance of low relevance – meaning its relevance would be below some defined level – we would not follow any paths going back further from that pattern instance. We would only be following paths of high enough relevance through the hierarchy. A pattern instance on such a path would be important in affecting the probability of the pattern instance that interested us, or the path would have been pruned earlier on, but to have that much effect it would generally have a relatively high probability value: It would have been strongly affected by propagation of information from pattern instances corresponding to previous inputs/outputs. The pattern instances encountered on such paths would be relevant: They would be highly dependent on previous inputs/outputs *and* important in making predictions of future inputs/outputs. With such pruning, the paths followed back from the predicted pattern instance would automatically "seek out" the previous inputs/outputs, passing through regions of maximum relevance to get there.

Such a pruning process would leave us with the *relevant* parts of the hierarchy. It would be doing what we wanted: focusing the system on the *relevant* and removing the *irrelevant*.

Such a process could not be used, exactly as described, to generate the hierarchy. In this discussion we assumed that we started with all of the hierarchy being represented in the actual hierarchy, and then pruned the less relevant parts. In reality, we would never be able to construct or handle a hierarchy that large in the first place; however, this way of propagating relevance can form the basis of a practical process. The ability to measure the relevance of pattern instances in the hierarchy enables an exploratory process. It allows us to analyze a hierarchy and decide which parts to keep and which to remove, based on relevance, and where we should try connecting new pattern instances to the hierarchy. Such a process can start with a hierarchy of manageable size and grow and prune it, the process being informed by continual measurement of the relevance of pattern instances, so that a relevant hierarchy is grown.

We could use a process of pruning like the one just described, in which high-relevance paths are followed back and low-relevance ones are pruned from the relevance back-propagation process itself, but we do not necessarily have to do that. I described such a process to show how we could start with a hierarchy which was too large for us to process in its entirety, and how we could still prune it, but, if we stop the hierarchy getting too large in the first place, by controlling its growth, informed by continual measurement, we could always keep the actual hierarchy small enough to be managed in its entirety.

This article will not get very involved in a discussion of how we could use a relevance back-propagation process to direct the growth of an actual hierarchy in an AI system. The next article in this series will discuss how we could actually *use* relevance back-propagation in an exploratory process to control the growth of the hierarchy: This article will just discuss the relevance back-propagation process.

4.5.3 The amount of effect is not just determined by probability.

When I said that probability would indicate the extent to which one pattern instance affects another I said "all else being equal". This is a simplification, though it is not a bad one. In reality, when one pattern instance affects another, through logic application or statistics application, the amount of effect depends on its probability value and the way that it interacts. The pattern logic will be a factor, as will the probability values of other pattern instances which are affecting the same pattern instance. We might know a lot about a pattern instance – it may have a value very close to 0 or 1 – but this might not translate into it having a strong effect on another pattern instance, because the pattern logic of that pattern instance may use it as a relatively unimportant pattern input, or it may be made unimportant due to other pattern inputs. Probability matters, but so do other things. On the other hand, a pattern instance with a probability of 0.5 will never propagate any information to any other pattern instance: We know nothing about such a pattern instance, so it can tell us nothing about any other pattern instances.

A process for assigning relevance to pattern instances will now be described.

5 Assigning Relevance to Pattern Instances

The process is one of *back-propagation of relevance* through the hierarchy.

Each pattern instance is going to be assigned a *relevance value*. The relevance values of all the pattern instances in the actual hierarchy are initially set to 0. An exception is the bottom-level pattern instances corresponding to the particular input/output values, required for predictions of future EFS values by the action selection process, that we want to predict, and which are assigned non-zero relevance values; for example, they might each be assigned a relevance of 1.

Relevance is propagated back from the pattern instances about which we want to make predictions, to the pattern instances which affected their predictions, depending on how much they affected them, and relevance values are propagated back still further to the ones which affected them and so on. Every incident of a pattern instance propagating information to another one in setting probabilities now has a corresponding act of relevance being propagated back. In other words, a pattern instance gets relevance by being relevant to one that is already relevant.

The basis of all this is the basic relevance back-propagation procedure.

5.1 Back-Propagating Relevance

The Basic Relevance Back-Propagation Procedure

Suppose we have some pattern instance X, which has a relevance value of R_X. For each pattern instance P_1, P_2 ,…P_n which directly affected X's probability (such as in logic application or in upwards or downwards statistic application), determine the *amount of effect* that that pattern instance had on X's probability value. Sum all the amount of effect values, E_1, E_2, …E_n, giving a total amount of effect, E_T. For each pattern instance P_1, P_2, …P_n, whatever proportion of the total effect, E_T, its own amount of effect is, add that proportion (E_1/E_T, E_2/E_T, … E_n/E_T) of X's relevance value, R_X, to its existing relevance value, R_1, R_2, …R_n. e.g. $(E_1/E_T)R_X$ is added to R_1.

When a pattern instance is given relevance because of its effect on the probability of another pattern instance, that relevance is *added* to any relevance that it already has: It does not replace it. This means that a pattern instance can receive relevance from a number of pattern instances which it affects, all the relevance that it receives combining to give its total relevance.

When relevance is propagated from some pattern instance, X, to those pattern instances which have affected its probability, it does not mean that X's relevance is reduced: X still keeps its own relevance.

It is simplest to imagine this working with the basic action selection process discussed previously[12], with a single EFS value being required, and any pattern instance(s) which are to be used for a future input of this value being assigned an initial, non-zero relevance at the start. With different variations on the basic action selection process, however, a process like this could as easily work with multiple EFS predictions being required from the hierarchy: It would just mean that the corresponding pattern instances would be set with non-zero relevance values at the start of the relevance back-propagation process.

Example

A pattern instance, X, has a relevance of 0.8. It was assigned some probability by being directly affected by three pattern instances, P_1, P_2 and P_3.

P_1 is determined to have had an effect of 3.5 on X's probability. P_2 is determined to have had an effect of 2.6 on X's probability. P_3 is determined to have had an effect of 7.2.

The effects are all summed to obtain E_T.

$E_T = 3.5+2.6+7.2 = 13.3$

P_1's proportion of the total effect is $^{3.5}/_{13.3} = 0.263$, so P_1 gets this proportion of X's relevance. Therefore, $0.263 \times 0.8 = 0.2104$ is added to P_1's relevance.

5.2 Determining the Amount of Effect of a Pattern Instance

The basic relevance back-propagation procedure requires us to have the ability to determine the amount of effect that a given pattern instance has on the prediction for another pattern instance, X. What we are interested in is how much the given pattern instance seems to reduce the uncertainty in prediction of X. Any "effect" means moving

[12] Almond, P. (2010). An Attempt to Generalize AI - Part 2: Planning and Actions. *paul-almond.com*. http://www.paul-almond.com/AI02.pdf. (Also available at http://www.paul-almond.com/AI02.doc.)

the probability value of X away from 0.5 and towards 0 or 1, so that X is known about with less uncertainty.[13]

All else being equal, a pattern instance's probability should indicate the amount of effect it can have on other pattern instances. We can see this by considering the case of a pattern instance with a probability of 0.5. Nothing is known about such a pattern instance. In logic application or statistics application, it will be useless for telling us anything about other pattern instances. As the probability of a pattern instance gets closer to 0 or 1, and we know more about it, the more it should tell us about other pattern instances, and the greater the effect that it should have on their probabilities. A crude way of measuring the amount of effect of a pattern instance on X would simply be to look at how far away its probability value is from 0.5.

This was "all else being equal", however. A pattern instance's effect on some other pattern instance, X, is not just determined by its probability. The way in which the two pattern instances are interacting also matters. For example, a pattern instance may be known about with almost complete certainty (say it has a probability of 0.99) and it may affect X by downward statistics application because X serves as one of its pattern inputs, but the pattern instance's logic may mean that this does not tell us much about X. That said, probability is still important: A probability of 0.5 will always have no effect at all, regardless of the interaction between two pattern instances – and the default probability of a pattern instance is 0.5.

We need to determine the effect of a given pattern instance on X in a way that takes account of the actual effect, without just looking at probability. This could be done in a number of ways. A simple method is to look at how much uncertainty was removed from X's probability value when it was affected by the given pattern instance, and how much would have been removed if the given pattern instance had a probability of 0.5. The more effect the given pattern instance is having, the greater should be the difference between the uncertainty removed when it has its actual probability and the uncertainty removed when we pretend it has a probability of 0.5.

[13] The same would apply if there were more that one probability value associated with a pattern instance: Any "effect" that a pattern instance had on the prediction for some pattern instance X would mean moving its probability values towards 0 or 1 to indicate less uncertainty.

Example

A pattern instance, X, is affected by a number of pattern instances in upward statistics application, as these pattern instances serve as its pattern instances. One of these pattern instances is A, which has a probability of 0.8. After statistics application, X has a probability of 0.65, but if we had set A to 0.5, X's probability would have been 0.75. The effect of the information in A was therefore to increase X's probability from 0.65 to 0.75, because 0.75 was achieved with the process working normally and only 0.65 when A was effectively disabled. This means that A can be considered to be removing 0.75-0.65 =0.1 of "uncertainty" from X, and this can be considered an indication of its amount of effect on X.

(Note: A change in uncertainty is not necessarily the same as a change in the probability value. For example, reducing a probability value from 0.7 to 0.4 is a reduction of 0.3 in the probability value, but this is actually an *increase* of 0.1 in uncertainty, because 0.7 is 0.2 away from 0.5 and 0.4 is only 0.1 away from 0.5.)

We might repeat the process by which pattern instances initially affected X, but with the probability value of the given pattern instance set to 0.5, to see what the effect is on the probability value assigned to X. This might involve "rewinding" the process by which probability values were assigned as we go along, or it might be done at the time that they were assigned in the first place, so that, for each pattern instance, the effect on it of other pattern instances is stored for later relevance propagation. We may need to be careful about how we organize such a process, but the basic idea will always tend to be the same: The effect of a given pattern instance on X is defined in terms of how much more uncertainty it removes from X with it working normally, than when it has a probability of 0.5.

6 Conclusion

Previous articles have described a probabilistic hierarchy of pattern instances as a way for AI to work and as an explanation of how the human mind works. A way is needed of ensuring that the hierarchy only represents relevant features of the world. This need will be met by an exploratory process that will reduce the hierarchy, by removing pattern instances, where it seems least relevant and extends it, by adding pattern instances, where it seems most relevant. Such an exploratory method will be described in the next article. It will require a process which can measure the relevance of pattern instances in the existing hierarchy, so that this information can inform any reduction or extension of the hierarchy. This article has described such a measurement process.

The process is based on *back-propagation of relevance* from particular, bottom-level pattern instances that are regarded as the most relevant, each pattern instance being assessed according to the contribution that it makes to the reduction in the uncertainty of predictions of these bottom-level pattern instances.

Such a back-propagation process is made useful by the use of the action selection process, described in the second article of this series, which drives the system's *behavior* in a particular direction.[14] The action selection process involves continually computing an evaluation function score (EFS) from recent inputs, and treating this as an input itself, continually encoding it as bottom-level pattern instances. Probabilistic predictions for bottom-level pattern instances corresponding to future EFS values are used to assess the merits of different output values. This gives relevance to specific pattern instances, giving a well-defined starting point for back-propagation of relevance.

A measurement process like the one just described allows us to examine the actual hierarchy in an AI system and measure the relevance of any pattern instance in it. This is telling us the usefulness of the pattern instance in reducing the uncertainty in the probabilistic predictions for any bottom-level pattern instances needed in the action selection process. This can be done with the action selection process as previously described, to predict the EFS resulting from a particular output, but the same general approach could also be used with more sophisticated versions of the action selection process, such as some version of it requiring multiple EFS predictions.

As well as measuring the relevance of individual pattern instances in the hierarchy, the process allows us to measure the overall relevance of a "region" of the hierarchy: We could look at the pattern instances which are "logically close" to a particular pattern instance in some way, and apply some kind of averaging process to their relevance.

[14] Almond, P. (2010). An Attempt to Generalize AI - Part 2: Planning and Actions. *paul-almond.com*. http://www.paul-almond.com/AI02.pdf. (Also available at http://www.paul-almond.com/AI02.doc.)

Basically, the kind of measurement process discussed here can be used to overlay a relevance "map" on the hierarchy, at any desired level of detail.

One way of applying the process would involve working back from the pattern instances that interest us, computing the relevance of every pattern instance in the hierarchy. Another way would involve pruning paths which encounter pattern instances of low relevance from the back-propagation process. Such a pruning method in the computation of relevance is not necessarily required: The whole point of the process is to use it as the basis for an *exploratory* process which controls the development of the hierarchy so that it remains at a manageable size. Other methods of prioritizing the computation of relevance might be used, and these will be discussed in the next article of this series, which will describe how we can use this measurement process in the exploratory process to control the development of the hierarchy.

The process described here only gives a snapshot of the hierarchy's relevance at some time. In reality, the situation will be continually changing. As inputs/outputs occur, the corresponding bottom-level pattern instance probabilities will change, in turn changing the probability values propagating through the hierarchy and changing the relevancies of pattern instances. This is not a problem, however: The relevance can be continually recomputed to reflect this. The pattern instances required for the action selection process will also be changing, as the present "catches up" with the existing ones and new ones are needed that are in the future. The occurrence of a single input/output may not alter the relevance values throughout the hierarchy very much, because the relevance values would be based on propagation of probabilities from numerous, previous inputs/outputs. Furthermore, before an input/output occurs, the corresponding pattern instance will itself be predicted with decreasing uncertainty by the hierarchy, and probabilistic information will be propagated from it: There will not be a sudden, important instant at which the pattern instance's conceptual value suddenly becomes known: Instead, its probability will gradually move towards 0 or 1, reaching one or the other at the instant the input/output occurs. Even before the corresponding input/output has occurred, a bottom-level pattern instance can be playing a role in the hierarchy, and the information it causes to be propagated can be determining the relevance of other pattern instances. As the relevancies of different pattern instances in the hierarchy change, the exploratory process that modifies the hierarchy can continually take account of these changes, altering the structure of the actual hierarchy in an attempt to include relevant pattern instances and exclude less relevant ones. It should be noted that altering parts of the hierarchy's structure will affect the relevance of other parts of the hierarchy, because a pattern instance's relevance is only meaningful in the context of a particular representation of the conceptual hierarchy: a particular, actual hierarchy in a computer. The relevance of individual pattern instances or parts of the hierarchy might be averaged over some period of time in some way.

The issue of *forgetting* is not addressed by the process discussed here. The process is about measuring relevance in the hierarchy and nothing more. Relevance, *as defined* *now*, should not be confused with whether or not parts of the hierarchy can or cannot be removed by a forgetting process. Relevance, as defined now, is a measure of the importance of a pattern instance in determining the probability value(s) of one or more important, bottom-level pattern instances, but being relevant just means that the pattern instance has been important in propagating probabilistic information, not that it is needed to deal with *changes* in this information. A basic forgetting process was discussed in a previous article, before a fully probabilistic hierarchy was described.[15] Forgetting will be discussed later, but it will work in a way that is still broadly similar to this.

[15] Almond, P. (2010). An Attempt to Generalize AI - Part 3: Forgetting. *paul-almond.com*. http://www.paul-almond.com/AI03.pdf. (Also available at http://www.paul-almond.com/AI03.doc.)

7 Bibliography

Almond, P. (2010). An Attempt to Generalize AI - Part 1: The Modeling System. *paul-almond.com*. Retrieved 8 April 2010 from http://www.paul-almond.com/AI01.pdf.
(Also available at http://www.paul-almond.com/AI01.doc.)

Almond, P. (2010). An Attempt to Generalize AI - Part 2: Planning and Actions. *paul-almond.com*. Retrieved 8 April 2010 from http://www.paul-almond.com/AI02.pdf.
(Also available at http://www.paul-almond.com/AI02.doc.)

Almond, P. (2010). An Attempt to Generalize AI - Part 3: Forgetting. *paul-almond.com*. Retrieved 8 April 2010 from http://www.paul-almond.com/AI03.pdf.
(Also available at http://www.paul-almond.com/AI03.doc.)

Almond, P. (2010). An Attempt to Generalize AI - Part 4: Modeling Efficiency. *paul-almond.com*. Retrieved 8 April 2010 from http://www.paul-almond.com/AI04.pdf.
(Also available at http://www.paul-almond.com/AI04.doc.)

Almond, P. (2010). An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy. *paul-almond.com*. Retrieved 8 April 2010 from http://www.paul-almond.com/AI05.pdf.
(Also available at http://www.paul-almond.com/AI05.doc.)