Probabilistic Semantics for Cost Based Abduction

Technical Report CS-90-02

Eugene Charniak and Solomon E. Shimony

February 1990

Probabilistic Semantics for Cost Based Abduction*

Technical Report CS-90-02

Eugene Charniak
Solomon E. Shimony

Computer Science Department
Box 1910, Brown University
Providence, RI 02906

February 22, 1990

Abstract

Cost-based abduction attempts to find the best explanation for a set of facts by finding a minimal cost proof for the facts. The costs are computed by summing the costs of the assumptions necessary for the proof plus the cost of the rules. We examine existing methods for constructing explanations (proofs), as a minimization problem on a DAG. We then define a probabilistic semantics for the costs, and prove the equivalence of the cost minimization problem to the Bayesian network MAP solution of the system.

*This work has been supported in part by the National Science Foundation under grants IST 8416034 and IST 8515005 and Office of Naval Research under grant N00014-79-C-0529. We also wish to thank Robert Goldman for helpful comments about the semantics, and Randy Calistri for proof reading earlier versions of this paper.

1 Introduction

The deductive nomological theory of explanation has it that an explanation is a proof of what is to be explained from knowledge of the world plus a set of assumptions. While there are well known problems with the theory [7], it is nevertheless an attractive one for people in AI, since it ties something we know little about (explanation) to something we as a community know quite a bit more about (theorem proving).

From an AI viewpoint the *real* problems with the deductive nomological theory are not the abstract ones of the philosopher, but rather the immediate one that there are many possible sets of assumptions, which together with our knowledge of the world would serve to explain (prove) the desired fact. Somehow, a choice between the sets must be made. Several researchers ([6], [4]), have used the above technique and graded assumption sets by a) only allowing some formulas to be assumed, and b) preferring sets with the minimum number of assumptions. That these simplifying assumptions are severely limiting we take as sufficiently obvious as to require no justification.

Cost-based abduction is the obvious generalization of these theorem proving techniques. It allows any formula to be assumed, and assigns all assumed formulas a non-negative real number cost. The best explanation is then the proof with the minimum cost (a formal definition of cost-based abduction will be given in the next section). Something very similar to what we are calling "cost-based abduction" has been proposed and implemented by Hobbs and Stickel [5], and it appears to be a promising method for handling abductive problems. However, their scheme has one immediate drawback; at the present moment the "costs" have no adequate semantics: for Hobbs and Stickel they are simply numbers pulled out of a hat.

In what follows we will provide a probabilistic semantics for cost-based abduction. We will do so in two steps. Section 2 will first formalize a cost-based proof (explanation) of some facts as an augmented DAG. Section 3 will exploit the similarity of the DAG to a belief network (Bayesian network) [8] to define a probability distribution using the topology of the DAG plus the costs. A major theorem of the paper will show that the maximum a-posteriori (MAP) assignment of truth values to the network corresponds to the minimal cost proof[1]. Section 4 will discuss these results in a more intuitive fashion.

[1]A MAP assignment is the way to set the values of all random variables such that their

Appelt, in [1], has attempted to give a semantics for the Hobbs-Stickel cost scheme. In Section 5 we will discuss the Hobbs-Stickel approach in more detail, and show why Appelt's semantics is not adequate. Lastly, in section 6 we will briefly mention our practical experience with an implementation of cost-based abduction.

2 DAG Representation for Rule Systems

A rule based system with assumability costs has rules of the form:

$$R: \quad p_1 \wedge p_2 \wedge \ldots \wedge p_n \quad \rightarrow \quad q$$

with costs $c(p_i)$ for each conjunct, and a cost $c(R)$ for applying the rule. A conjunct has the same cost in all the rules where it appears on the left hand side (LHS). The cost of proving q with this rule is the cost of all the conjuncts assumed, plus the cost of the rule. For the rest of sections 2 and 3, we assume without loss of generality that all rule costs are 0. We can do this by adding a p_0 (that appears nowhere else) to the LHS, with a cost $c(p_0) = c(R)$. We want to find a minimal cost proof for some fact set $\mathcal{E}$ (called "the evidence").

We now formalize the minimum cost proof problem as a minimization problem on a weighted AND/OR DAG (acronym WAODAG). We use a three valued logic (values $\{T, F, U\}$), augmented by symbols for keeping track of assumed nodes, versus implied nodes. The values we actually use are $Q = \{T^A, T, U, F^A, F\}$, where U stands for undetermined (intuitively: either true or false), T for true, F for false, and the A superscript stands for "assumed". We use $u \searrow v$ to say that u is an immediate parent of v.

Definition 1 *A WAODAG is a 4-tuple* $\{G, c, r, s\}$*, where:*

1. G is a connected directed acyclic graph, G = (V, E).

2. c is a function from {V, Q} to the non-negative reals, called the *cost function.* For values T, F, U, we have zero cost. $c(v) \equiv c(v, T^A)$.

3. r is a function from V to {AND, OR}, called the *label.* A node labeled AND is called an AND node, etc.

---

joint probability is highest.

4. s is an AND node with outdegree 0 (the *evidence* node).

Definition 2 *A truth assignment for a WAODAG is a function f from V to Q. A truth assignment is a (possibly partial)* model *iff the following conditions hold:*

1. If v is a root node then $f(v) \in \{T^A, U, F^A\}$.
2. If v is a non-root OR node then one of the following statements holds:
   (a) $f(v) = T^A$ and $\exists u,\ u \searrow v \wedge f(u) \in \{T^A, T, U\}$[2].
   (b) $f(v) = T$ and $\exists u,\ u \searrow v \wedge f(u) \in \{T^A,\ T\}$.
   (c) $f(v) = F^A$ and $\exists u,\ u \searrow v \wedge f(u) \in \{U, F^A, F\}$.
   (d) $f(v) = U$ and $\exists u,\ u \searrow v \wedge f(u) = U$ and $\neg\exists u,\ u \searrow v \wedge f(u) \in \{T,\ T^A\}$.
   (e) $f(v) = F$ and $\forall u, u \searrow v \Rightarrow f(u) \in \{F^A, F\}$.
3. If v is a non-root AND node then one of the following statements holds:
   (a) $f(v) = F^A$, and $\exists u,\ u \searrow v \wedge f(u) \in \{F^A, F, U\}$.
   (b) $f(v) = F$ and $\exists u,\ u \searrow v \wedge f(u) \in \{F^A, F\}$.
   (c) $f(v) = T^A$, and $\forall u,\ u \searrow v \Rightarrow f(u) \in \{U, T^A, T\}$.
   (d) $f(v) = U$ and $\exists u,\ u \searrow v \wedge f(u) = U$ and $\neg\exists u,\ u \searrow v \wedge f(u) \in \{F,\ F^A\}$.
   (e) $f(v) = T$ and $\forall u,\ u \searrow v \Rightarrow f(u) \in \{T, T^A\}$.

Intuitively, an assignment is a model if the AND/OR constraints are obeyed. If a node's parents do not uniquely determine the node's truth value, it can have any value in $\{T^A,\ F^A,\ U\}$[3]. A node v where $f(v) = T^A$ in an assignment, is called an *assumed true* node relative to the assignment. Likewise for other values of $f(v)$.

Definition 3 *A model for a WAODAG is* satisfying *iff $f(s) \in \{T^A,\ T\}$.*

[2]Note that in our DAG, an OR node is true if at least one of its *parents* is true, as in belief networks, but *not* as commonly used for search AND/OR trees.

[3]A node may still be assumed true if its parents determine that it has to be true.

Definition 4 *The cost of an assignment $\mathcal{A}$ for a WAODAG is the sum*

$$C = \sum_{v \in V} c(v, f(v))$$

The Best Selection Problem is the problem of finding a minimal cost (not necessarily unique) satisfying model for a given WAODAG. The Given Cost Selection Problem is that of finding a satisfying model with cost less than or equal to a given cost. Note that in a partial model, assuming a node false is useless, as such an assumption cannot contribute towards a satisfying model.

Theorem 1 *The Given Cost Selection Problem is NP-complete.*

The theorem is easily proved via a reduction to the Vertex Cover problem (see appendix A). The Best Selection Problem is clearly at least as hard as the Given Cost Selection Problem, because if we had a minimal cost satisfying model, we can find its cost in $O(|V|)$, and give an answer to the Given Cost Selection Problem. Thus, the Best Selection Problem is NP-hard.

Theorem 2 *The Best Selection Problem subsumes the problem of finding a minimal cost proof for the rule-based system with assumability costs, assuming that the rule based system is acyclic.*

Informal proof: by constructing a WAODAG for the rule set, as follows:

1. For each literal appearing in any rule[4] R's LHS, construct an OR node v, and set $c(v, T^A)$ to the cost of the literal in the rule-based system (i.e. in R). For literals appearing only on the RHS of rules, construct an OR node v, with $c(v, T^A) = \infty$.

2. For each LHS of a rule R, construct and AND node v with $c(v, T^A) = \infty$, make it a parent of the node constructed for the literal on the RHS of R (in step 1), and make it a child of all the nodes constructed for the literals in the LHS of R.

3. Construct an AND node s, with parent nodes corresponding to the facts to be proved.

[4]We mean any *relevant rule instance*. We assume that exactly all relevant instances are given, and that literals with the same name in different rules are the same literal.

Definition 5 *A root-only assignment for an WAODAG is an assignment where only root nodes may be assumed (i.e. have values in $\{F^A, T^A\}$).*

It is possible to force root-only assignments for a WAODAG to be globally minimal by setting the cost of all non-root node to infinity (in practice, it suffices to set a cost greater than the sum of the root costs). We now show that given an WAODAG D, we can create another WAODAG D', such that the semantics of minimal models is not changed, i.e. if a non-root node is selected in D, some corresponding new root node is selected in D'.

Proof: by construction, as follows (with $D' = D$ initially):

1. For each AND node v in D', with cost $c(v,\ T^A)\ <\ \infty$, construct a new OR node w in D', and a new root node u, where $c(u, T^A)\ =\ c(v, T^A)$, and make both $v \searrow w$ and $u \searrow w$. Transfer all the children[5] of v to w.

2. For each OR node v with cost $c(v,\ T^A)\ <\ \infty$, create a new root-node u, with $c(u,\ T^A)\ =\ c(v, T^A)$, and make $u \searrow v$.

3. For all non-root nodes v in D', set $c(v, T^A)\ =\ \infty$.

It is clear that each time a node is selected in a minimal cost model of D to be assumed true, the node constructed from it in D' will be assumed true in some root-only minimal cost model for D'.

Definition 6 *An assignment (or model) is* complete *iff* $\forall v \in V,\ \ f(v) \neq U$.

A variant of the Best Selection Problem is one of selecting a minimal cost *complete* model. Clearly, if the cost of assuming a node false is 0 for all nodes, the solution will be exactly the same as for the partial model Best Selection Problem. However, as we intend to treat assumability costs as negative logarithms of probabilities, for purposes of the probabilistic semantics, and we want $P(f(v) = F^A)\ =\ 1 - P(f(v) = T^A)$ to hold for all root nodes, the cost of assuming a node v false is:

$$c(v,\ F^A)\ =\ -log(1 - e^{-c(v,\ T^A)})$$

[5]If the AND node is s, create a new sink AND node, s', and make $w \searrow s'$.

3 Probabilistic Semantics for AOWDAGs

We now provide a probabilistic semantics for the cost based abduction system. We construct a boolean belief network out of the weighted AND/OR DAG, and show the correspondence between the solution to the Best Selection Problem and finding the most likely explanation for a given fact (or set of facts).

We assume that the rule based system has already been converted to the WAODAG format with root-only assignment, as per the previous section. We now construct a belief network from the given WAODAG, and show that a minimal cost satisfying complete model for the WAODAG corresponds to a maximum-probability assignment of root-nodes given the evidence in the belief network (where the evidence is exactly the set of facts to be proved using the rule system).

From a WAODAG D we construct a belief network B as follows:

1. B has exactly the nodes and arcs of D[6]. Nodes retain their labels[7].

2. Each root node v in B has a prior probability of $e^{-c(v,\ T^A)}$.

3. The node s is the "evidence node", i.e. the event of node s being true is the evidence $\mathcal{E}$.

Defining an assignment for the network analogously with the WAODAG assignment, we assume, without loss of generality, that we are only interested in assignment to the set of root nodes[8] We want to find the "best" satisfying model $\mathcal{A}$, which assigns values from $\{T^A,\ F^A,\ U\}$ to the set of all root nodes, i.e. the assignment that maximizes $P(\mathcal{A} \mid \mathcal{E})$. An assignment of U to a root node means that it is omitted from the calculation of joint probabilities, as $P(v_i = U) = 1$. Intuitively, we are searching for the most probable explanation for the given evidence. This can be done by running a Bayesian network algorithm for finding Bel* on the root nodes, as defined in [8]. We now show the following result:

[6]Thus, we use the same name for a node of B and the corresponding node of D.

[7]A belief network AND node has a 1 in its conditional distribution array for the case of all parents being true, and 0 elsewhere. An OR node is defined analogously.

[8]Maximizing the probability over assignments to root nodes is equivalent to finding the MAP, when we allow only *complete* models. See appendix B for a proof.

Theorem 3 *In a boolean belief network B constructed as above, a satisfying complete model $\mathcal{A}$ that maximizes $P(\mathcal{A} \mid \mathcal{E})$ will also be a minimal cost satisfying complete model for D.*

Proof: In a belief network, all root nodes (given no evidence) are mutually independent. Thus, for any assignment of values to root nodes, $\mathcal{A} = (a_1, a_2, ..., a_n)$, where $a_i = (v_i, q_i)$ and $q_i \in \{F^A, T^A\}$[9].

$$P(a_1, a_2, ..., a_n) = P(a_1)\, P(a_2) \ldots P(a_n)$$

However, we also have (by definition of conditional probabilities):

$$P(\mathcal{A} \mid \mathcal{E}) = \frac{P(\mathcal{E} \mid \mathcal{A}) P(\mathcal{A})}{P(\mathcal{E})}$$

But as $P(\mathcal{E} \mid \mathcal{A}) = 1$ when the assignment is a satisfying model (because all nodes are strict OR and AND nodes), and 0 otherwise, and $P(\mathcal{E})$ is a constant, we can eliminate everything but $P(\mathcal{A})$ from the maximization. Also, we have:

$$P(\mathcal{A}) = \prod_{i=1}^{n} P(a_i) = \prod_{i=1}^{n} e^{-c(a_i)} = e^{-\sum_{i=1}^{n} c(a_i)}$$

Since e^x is monotonically increasing in x, we see that maximizing $P(\mathcal{A} \mid \mathcal{E})$ is equivalent to minimizing the cost of the assignment, Q.E.D.

We now generalize the DAG so that nodes can have any gating function[10]. The definition of a model is extended in the obvious way.

Theorem 4 *Given a gate-only belief network, with a single evidence node, the problem of finding the most probable complete satisfying model given the evidence is equivalent to finding a minimal cost complete model for the weighted gated DAG.*

Proof: The proof of theorem 3 relies only on the fact that the probability of the evidence given a satisfying model is 1, and 0 given any other complete model. Thus, we can use exactly the same proof here[11].

---

[9] Additionally, we use the a_i's to denote the event of node v_i having value q_i.

[10] Gate nodes are any probabilistic nodes which have only entries of 1 and 0 in their conditional distribution arrays.

[11] The above result can be generalized even further, see appendix C

If we want to find the best *partial* model, and are only interested in satisfying models, the above theorem still holds. It is no longer true, however, that finding the minimum cost model it is equivalent to finding the MAP over the entire belief net.

4 Evaluation of Cost Based Abduction

In essence, the theorems of the last two section serve to define not one scheme of cost-based abduction, but four. The most obvious distinction is between Theorems 3 and 4. The first restricts itself to WAODAGs, while the second generalizes to arbitrary gating functions. There are two things to keep in mind about this distinction. First, if one is using a standard theorem prover, as found in typical rule-based systems, the theorem prover will itself only generate AND/OR DAGS. So unless one is willing to add capabilities to the theorem prover, arbitrary gates are pointless. Also, since AND/OR gates do not require the labels F^A and F, they are marginally simplier to implement. (However, as we will note in the section on implementation, we found that we needed the capabilities of arbitrary gates for our domain.).

The second distinction is between partial and complete models. The distinction here is whether we simply assign costs to those facts which we must assume true (or false) to make the proof work, or go on to make decisions about every fact in the domain, whether or not it plays a role in the proof. Intuitively the former makes more sense. Unfortunately, our theorem (that we would get the MAP assignment) is only true for complete models, and counterexamples exist for non-complete models. Also, the minimal cost complete model will *not* agree, in general, with the minimal cost partial model, even if we compare only the sets of nodes assumed. However, the minimal solution of the complete model problem will be a nearly minimal solution for the partial model solution, provided a) there is no other complete model with nearly the same cost as the minimum, and b) the cost for assuming nodes false is low. These are reasonable assumptions in many cases. For example, the probabilistic semantics presented in [2] is characterized by low prior probabilities, thus low costs for assuming nodes false. It is not hard to add completeness, but, as you would imagine, it is computationally expensive, and probably not worth the computational effort.

Lastly, a further word is required about the relation between the costs and the probabilities. In our definition of cost-based abduction there were three sorts of entities which received costs: rules, root nodes, and interior nodes. By the time we reached theorems 3 and 4, however, we had reduced this to one, by showing that rule and interior node costs could be replaced by added root node costs. But how do these transformations affect our interpretation of what the costs mean in terms of probabilities?

Things are most obvious for root nodes. As stated earlier, the cost of assuming a root node must be $-log(P(node))$. For rules and interior nodes, however, things are slightly more complex. The cost of a rule got moved to the cost of a new root node. Suppose for rule R we add the new root node R'. We can, of course, say that the cost of a rule must be $-log(P(R'))$. However, R' does not correspond to anything in our model of the world (it is merely a mathematical fiction designed to make the proof simpler). We need to define the cost in terms of elements of our world model. Thus, suppose R is the rule:

$$p_1 \ \wedge \ \ldots \ \wedge \ p_n \ \rightarrow \ q$$

and we will denote the AND node corresponding to its left-hand side as $A_R{}'$. We will refer to the AND node without the added cost root attached as A_R. Suppose that the other rules which can prove q are R_1, ..., R_n with the corresponding AND nodes (without attached cost roots) $A_{R_1}, \ldots, A_{R_n}$. It is easy to see that $P(q \mid A_R \wedge \neg A_{R_1} \wedge \ \ldots \ \wedge \neg A_{R_n}) = P(R')$. That is, the cost of a rule is minus log probability of its consequent being true given that a) its antecedent is true, and b) none of the other ways of proving (or assuming) the consequent are true.

Analogously, we can show that the cost of assuming an interior OR node v is $-log(P(v \mid \text{all the ways of proving it are false}))$. Finally, since in practice there is never a need for assuming an AND node, we will ignore it here.

5 Appelt's Semantics for Hobbs-Stickel

In [5], Hobbs and Stickel have proposed a scheme very similar to what we have proposed. In their scheme, the initial facts to be explained are each assigned an assumption cost c_i. All inference rules are of the form:

$$p_1^{w_1} \ \wedge \ \ldots \ \wedge \ p_n^{w_n} \ \rightarrow \ q$$

The cost of the p_i's are then given by $cost(p_i) = w_i \ cost(q)$.

However, if two separate portions of the proof-tree require the same assumption, the proof is only charged once for the assumption (and is charged the minimum of the two costs being charged for the fact).

Hobbs and Stickel point out that that $C = \sum_{i=1}^{n} w_i$ does not have to equal 1. If $C < 1$ then the system will prefer to assume $p_1, \ ..., \ p_n$, since that will be less expensive than assuming q. They refer to this as *most-specific abduction.* On the other hand, if $C > 1$, then, everything else being equal, the system will tend to just assume q (*least-specific abduction*). Note, however, that even with least-specific abduction, cost sharing on common assumptions can make a more specific scenario cost less. Hobbs and Stickel believe that least-specific abduction (with cost sharing) is the way to go, at least for the abductive problems they are concerned with (natural language comprehension). In general we agree with this assessment.

As we have noted, Hobbs and Stickel did not give a semantics for their weights, and Appelt in [1] is concerned with overcoming this deficit. Appelt takes as his starting point Selman and Kautz's theory of default reasoning [9] called *model preference theory.* In this theory a default rule $p \rightarrow q$ is interpreted as meaning that in all models in which p is true, the models in which q is also true are to be preferred. Appelt, in the spirit of abduction, reverses this by saying that a rule $p^{w_p} \rightarrow q$ where $w_p < 1$ is to be interpreted as a model preference among those models which have q for those which have p was well. However, it is possible to have another rule $r^{w_r} \rightarrow s$ (where $w_r < 1$), but where p and r are not compatible. Thus if s is also in our model we must choose which rules to use. Appelt specifies that if $w_p < w_r$ then use $p \rightarrow q$, and vice versa. Appelt calls this scheme *weighted abduction.*

The most obvious difference between Hobbs-Stickel, and weighted abduction is that the former use rules of the form:

$$p_1^{w_1} \wedge \ ... \ \wedge \ p_n^{w_n} \rightarrow q$$

where weighted abduction only has rules of the form $p^{w_p} \rightarrow q$. We assume that what Appelt has in mind is recasting the Hobbs Stickel rules as $(p_1 \ \wedge \ ... \ \wedge \ p_n)^{w_p} \rightarrow q$. With $w_p = \sum_i w_i$. Assuming this is correct, there are two immediate problems with Appelt's semantics. First, it only handles the case where the sum of the w_i's is less than 1. This is what Hobbs and Stickel call more-specific abduction. But as they note, less-specific abduction seems to be the more important case, and Appelt says nothing about it.

Secondly, Appelt can give no obvious guidance to how to apportion w_p into the individual w_i's required by Hobbs-Stickel.

But even if we restrict ourselves to finding the single w_p, and also restrict ourselves to more-specific abduction, it would seem that Appelt's semantics gives little guidance when trying to judge if a number is "right". Suppose we have a system with a group of w's for various rules $\{w_a, \ w_b \ ... \ w_y\}$ and we now want to add a rule with the number w_z, and we want to know what w_z should be. Following Appelt we will look for models in which the rules $w_a, \ w_b, \ ... \ w_y$ are used but where the rule Z conflicts with their use. In each case we see which rule takes precedence. But this is not sufficient. Suppose we have a model in which A and B are used but where Z makes both unusable. Since costs are additive we have several possibilities: $\{A + B > Z$, but $A < Z\}$, $\{B < Z$ but $A + B < Z\}$, etc. Depending on the numbers, in some cases we should prefer using rules A and B together over Z, in other cases not. But this is not sufficient either, what about $A+C$, and $A+B+C$, and $B+C$, etc. In fact, the number of models which need to be checked grows exponentially with the number of costs in the system.

6 Implementation

As we have showed in this paper, anything that can be done with cost-based abduction can also be done by probabilistic methods. One might then ask, why not use the more standard probability theory? The reason would have to be that the standard probabilistic methods are computationally expensive. Evaluating general belief nets is NP-hard.

Unfortunately, as we saw in Theorem 1, so is the minimal cost proof problem. However, as the mathematics is quite a bit simpler for minimal cost proofs, one might hope that the constant in front is a lot less. Also, thinking in terms of minimal-cost proofs suggests different ways of looking for solutions. There are well known techniques for doing best first search on AND/OR search spaces, and this looks like an obvious application.

Thus we have implemented a best-first search scheme for finding minimal cost proofs. We have applied it to the work we have described in [2], which uses belief nets to make abductive decisions on problems that come up in natural language understanding, such as noun-phrase reference, or plan recognition. Thus we had a ready-made source of networks and probabilities upon which we could test the system, and a benchmark (the speed of our current probabilistic methods).

The approach seems to have some promise[12]. We were able to adapt our networks to the new scheme with little difficulty. (Of the four versions of cost-based abduction mentioned above, we found that complete models did not seem to be required for our networks, but we did need general gates, and not just WAODAGs.). Because we did not use MAP labelings in our earlier work, but rather decided what to believe on the basis of the probability of individual statements, we simulated this by looking for multiple MAP solutions when there were several within some epsilon of each other, and only believing the statements which were in all of them. For the examples we have tried it on, this gave the same results as our belief network calculations.

Furthermore, after minor tuning, our cost algorithm seems to be running significantly faster than the belief-network updating scheme we were using. However, we have not yet done formal timing comparisons between them, and at the moment both the cost algorithm, and the belief-network updating scheme, are very sensitive to efficiency measures. Thus it is not clear how much the timing will prove. It does, however, seem safe to say that cost-based abduction deserves serious consideration.

We should note that this speed-up occurs despite the fact that we do not have a good admissible heuristic cost estimator for our best-first search. We are using the simplest heuristic imaginable – at any point in a partial proof, we assume that the final cost of the proof will be the costs incurred to date. This is a very poor estimator because the bulk of a proof's cost comes from the root assumptions, and they are not found until the end. We are currently exploring other estimator possibilities, such as logic minimization approaches and factoring in assumption costs earlier in the search.

7 Conclusion

We have shown that cost-based abduction can be given an adequate semantics based upon probability theory, and that under the appropriate circum-

[12]An earlier draft of this paper was much more downbeat on the implementation. The difference is due to a minor efficiency measure which turned out to have a large effect on the run time. (One nice thing about NP-hard problems is that it is always possible to save exponential amounts of time.)

stances it is guaranteed to find the best MAP assignment of truth values to the propositions in our theory. Initial experiments with the model show that it does produce results consistent with full blown probability theory, and does so quickly compared to our current probabilistic methods. However, further experimentation is clearly required.

A Proof: Best Selection Problem

Proof of NP-hardness: Using reduction from the Vertex Cover Problem (VC), which is a known NP-complete problem [3], defined as follows: Given a graph $G = \{V, E\}$, and an integer K, is there a subset $S \in V$ such that $\mid S \mid \leq K$, and for all edges $e = (v_1, v_2)$ where $e \in E$, at least one of v_1, v_2 are in S ?

The reduction is as follows: given a graph $G = \{V, E\}$ and an integer K, construct an weighted AND/OR DAG, $T = \{G', c, f, s\}$ as follows:

- The nodes of G' are: an AND node s with cost $\mid V \mid +1$, for every $v \in V$ a root node v' with cost 1, and for every edge $e \in E$ an OR node v_e with cost $\mid V \mid +1$.
- The parents of s are exactly all the OR nodes constructed above.
- The parents of each OR node, v_e are exactly nodes v' where where in G the edge e corresponding to v_e is incedent on the node v corresponding to v' .

Clearly, the construction process is of complexity $O(\mid V \mid + \mid E \mid)$, and is thus polynomial in complexity.

Claim: The VC problem on G with cost K has an affirmative answer iff there exists a minimal cost satisfying truth assignment for G' with cost at most K.

Proof: We show that a satisfying truth assignment for G' with cost C induces a vertex cover for G with cost C, and vice versa.

- The cost of an assignment is exactly the total cost of the vertices assumed true. Obviously, only cost 1 vertices will be assumed true in the minimal cost assignment, because assuming all the root vertices true induces a satisfying truth assignment with cost $\mid V \mid$ which is less than the cost of assuming a non-root vertex true. But when we have a satisfying cost assignment where only root vertices are assumed, with a cost C, then there is a vertex cover of cost C for the VC problem, consisting exactly of the set of vertices corresponding to the set of assumed vertices, because each "edge" v_e is covered by at least one "vertex" v' .

- A subset S that solves the VC problem with cost C induces a satisfying truth assignment on G' , consisting of assuming true all root vertices corresponding to nodes in S, setting all other root vertices to F^A, and assigning all other vertices to T. Clearly, this assignment has cost C, and is consistent because each vertex corresponding to an edge has at least one parent assumed true (solution to VC), and the sink node has all parents with value T, hence it has value T.

The Given Cost Selection Problem with cost K is in NP, as the following non-deterministic polynomial time algorithm, solves it:

1. Propose an assignment for the WAODAG.
2. If the assignment is consistent, (takes $O(\mid V + E \mid)$ steps to check), continue.
3. If the assignment has cost at most K (takes $O(\mid V \mid)$ steps to check), halt with an affirmative answer.

B Proof: Equivalence to MAP

Given a boolean belief network B, as in section 3, We define the following nomenclature. R is the set of all root nodes, and I is the set of all non-root nodes. $\mathcal{A}$ is an assignment to a set of nodes, and we also use it to denote the event of the assignment holding, when unambiguous. We use a subscript of $\mathcal{A}$ to denote an assignment to particular set of nodes (or the event of the set of nodes having a particular set of values). For example $\mathcal{A}^R$ is an assignment to the set of root nodes. We now show that finding the MAP model for the belief network is equivalent to finding a maximum probability a-priory assignment for R.

Theorem 5 *Given a boolean gate-only belief network B, minimizing $P(\mathcal{A}^V|\mathcal{E})$ over complete models*[13] *is equivalent to minimizing $P(\mathcal{A}^R)$.*

[13]That is, finding the MAP model.

Proof: By using the definition of conditional probabilities, we have:

$$P(\mathcal{A}^V|\mathcal{E}) = \frac{P(\mathcal{E}|\mathcal{A}^V)P(\mathcal{A}^V)}{P(\mathcal{E})}$$

But $P(\mathcal{E})$ is constant for this maximization, and $P(\mathcal{E}|\mathcal{A}^V)$ is 1 for all satisfying models, and 0 for all all other models. Thus, maximizing over $P(\mathcal{A}^V)$ is equivalent to MAP. But we also have (by theorem on the semantics of belief networks, in [8]):

$$P(\mathcal{A}^V) \equiv P(\mathcal{A}^I\mathcal{A}^R) = P(\mathcal{A}^I|\mathcal{A}^R)P(\mathcal{A}^R)$$

But $P(\mathcal{A}^I|\mathcal{A}^R)$ is 1 for all complete models and 0 for all other complete assignments, because the belief net is connected, and is gate-only. Thus, maximizing over all $P(\mathcal{A}^R)$ is equivalent to MAP, Q.E.D.

C Generalization to Non-Boolean Networks

We now generalize theorem 4 to cover non-boolean belief networks. A complete Multiple Value Best Selection Problem with root-only assignment is defined in a way similar to the complete Best Selection Problem, except that we use an n valued logic with values $L_1, \ldots, L_n$, and similarly using the superscript A for values assumed, we have $Q = \{L_1, \ldots, L_n, U, L_1^A, \ldots, L_n^A\}$. Our DAG, D, is a generalized gated DAG, i.e. each node's label is also a function on Q. Assume, without loss of generality, that the "satisfying" value for node s is L_1 (or, of course, L_1^A). The costs for assuming values on interior nodes are set at ∞, and costs for assuming any root node v obey the following constraint:

$$\sum_{i=1}^{n} e^{-c(v,\, L_i^A)} = 1$$

The above amounts to the standard probabilistic constraint that the probability of the entire sample space has to equal 1.

Now, we construct the belief net B in a way analogous to that of section 3, and define an assignment as a natural extension of $\mathcal{A}$ of section 3, where we do not allow U values to be assigned.

Theorem 6 *Finding a MAP assignment (given the evidence) for the generalized gate-only belief network B, where only complete satisfying models are considered, is equivalent to finding a complete minimal cost satisfying model for the augmented DAG, D.*

Proof: again we use the proof of theorem 3, where we only need the fact that the probability of the evidence given a complete satisfying model is 1, and 0 for all other models.

References

[1] Douglas E. Appelt. A theory of abduction based on model preference. In *Proceedings of the AAAI Symposium on Abduction*, 1990.

[2] Eugene Charniak and Robert Goldman. A logic for semantic interpretation. In *Proceedings of the AAAI Conference*, 1988.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractibility, A Guide to the Theory of NP-completeness*, page 190. W. H. Freeman and Co., 1979.

[4] Michael R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, pages 411–436, 1984.

[5] Jerry R. Hobbs and Mark Stickel et. al. Interpretation as abduction. In *Proceedings of the 26th Conference of the ACL*, 1988.

[6] Henry A. Kautz and James F. Allen. Generalized plan recognotion. In *Proceedings of the Fifth Conference of AAAI*, August 1986.

[7] Drew V. McDermott. Critique of pure reason. *Computational Intelligence*, 3:151–60, November 1987.

[8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[9] Bart Selman and Henry Kautz. The complexity of model-preference default theories. In Reinfrank et. al., editor, *Non-Monotonic Reasoning*, pages 115–130. Springer Verlag, Berlin, 1989.