Generating Hypotheses to Explain Prediction Failures

Steven Salzberg
Yale University
Department of Computer Science
New Haven, Connecticut

Abstract

Learning from prediction failures is one of the most important types of human learning from experience. In particular, prediction failures provide a constant source of learning. When people expect some event to take place in a certain way and it does not, they generate an explanation of why the unexpected event occurred [Sussman 1975] [Schank 1982]. This explanation requires hypotheses based on the features of the objects and on causal relations between the events in the domain. In some domains, causal knowledge plays a large role; in some, experience determines behavior almost entirely. This research describes learning in intermediate domains, where causal knowledge is used in conjunction with experience to build new hypotheses and guide behavior. In many cases, causal knowledge of the domain is *essential* in order to create a correct explanation of a failure. The HANDICAPPER program uses domain knowledge to aid it in building hypotheses about why thoroughbred horses win races. As the program processes more races, it builds and modifies its rules, and steadily improves in its ability to pick winning horses.

1. Introduction

This research models a person learning in a new domain, in which he creates rules to explain the events in that domain. When rules succeed, he confirms or strengthens already held beliefs; but when rules fail, he can learn by explaining the failures. For example, a stock market analyst creates new rules about how the market works when an investment fails to yield a profit. The new rules are based on the relevant features of companies in which he is investing. He determines which are the relevant features by querying his knowledge about how each feature affects a company's performance. His knowledge of causality allows him to determine in advance, for some features, whether they will predict improving or declining performance; for example, he knows if an oil company makes a big strike in an offshore well, its stock will probably go up. The causal knowledge involved here is that the earnings of oil companies are directly related to the amount of new oil they discover. New oil leads to increased earnings, which in turn cause stocks to go up. A similar type of hypothesis generation occurs in the domain of horse racing. Here, whenever a horse wins (or loses) contrary to expectations, new rules about why a horse wins or loses are generated by the racing handicapper, who bases his rules on the data available to him about the horse.

*This work was supported in part by the Air Force Office of Scientific Research under contract F49620-82-K-0010.

2. The Effect of Domain Characteristics

Regardless of the domain in which one is learning to be an expert, certain rules about learning apply. One needs to know, among other things:

1. What features exist for the objects in the domain
2. What features are relevant; i.e., which ones affect performance and how strongly
3. How the features interrelate (causal knowledge)

Features will be loosely defined here as anything a person can notice consistently when it is present; for example, a nose is a feature of a face because we have the ability to see it when we see someone's face (even though we don't have to notice it). For some domains, the knowledge of the above items is much easier to obtain than others. For the current project a domain was chosen in which this knowledge is relatively easy to obtain.

2.1. Horse racing

Thoroughbred horse racing is a domain where the relevant features are pretty clear. As for area (1) above, what features exist, most racing experts (according to two experts I consulted) get all their data from the *Daily Racing Form*, a daily newspaper which lists, for each horse in a race, essentially all the possibly relevant data about a horse. Area (2) is a little more difficult. By questioning racing experts, though, it is possible to throw out at least some of the data that appears in the *Form*, because they ignore some of it. As for the third and most difficult area above, the causal knowledge of the domain, again experts provide some clue. (Causal knowledge will be discussed in detail later.) For example, two items of data are how a horse finished in its last race and the claiming price of that race. (The claiming price is the value for which any horse in a given race may be purchased. Higher claiming prices indicate better, or in other words faster, horses.) If a horse won its last race, it is quite likely that the claiming price of that race was lower than the current one, because an owner is likely (indeed, required in some cases) to move his horse up in value when it is doing well. As will be shown later, such causal knowledge will be of use in restricting the possible hypotheses that may be generated to explain a given failure.

3. Causal Knowledge

When a person fails at a prediction, he generates new hypotheses to explain the situation where he failed, and uses the features mentioned in the hypotheses to index the situation [Schank 1982]. A central question of much research on

explanation is how great a role causal knowledge plays [Schank & G. Collins 1982]. Psychological evidence indicates that although people do causal reasoning all the time, they often are not very good at it [A. Collins 1978]. Because of this, Lebowitz (1980) claimed that such analyses should not be used by a computational model. However, although Allan Collins' work shows that people are not very good at causal reasoning about scientific problems, people *are* good at many other kinds of everyday causal reasoning, such as explaining why a waitress acts nice to customers (she wants a good tip), why people go swimming in hot weather (to cool off), and so forth [Schank & Abelson 1977]. The extensive use of causal reasoning, and of previously constructed causal chains in everyday life, must be considered in the construction of any computational model of explanation.

Causal knowledge can be used in at least two distinct ways to aid in building explanations: it could be used as a filter to throw out irrelevant hypotheses after they are generated, or it could be used as a filter on the relevant features that the generation process will use. HANDICAPPER uses the latter method, which saves it the trouble of generating useless hypotheses.

4. Generating Hypotheses

Given any expectation failure, a model of human explanation must decide which features of the situation are appropriate to use in an explanation. Without causal knowledge, the only way a computer can generate hypotheses is to use all the features in every possible combination, since it has no way of deciding which are relevant and which are not. Previous models of learning have had some success with this approach because they pre-selected the relevant features [Lebowitz 1980], or because they only allowed no more than one or two features to be used in hypothesis generation [Winston 1975], thus finessing the combinatorial explosion problem. HANDICAPPER, though, does not know ahead of time which features to use in generating a hypothesis (although the *Daily Racing Form* does provide some constraints). Now it turns out that if one uses causal knowledge to generate an explanation, *the question of which features are relevant is answered automatically.* An explanation, which will rely on basic knowledge of how the domain works, will only include those features which are necessary to make the explanation complete. Irrelevant features will not be used in the explanation, and hence they will never be used as indices for the expectation failure. The following example illustrates the importance of generating hypotheses based on explanations: horse A has just won a race, and a comparison of A with the horse which was predicted to win shows that A has ten features that the other horse lacks. Assume further that one of these features is the ability to run well on muddy tracks. One explanation which someone might offer is that A won because it was a good mud runner. The next thing to ask is, was the race run on a muddy track? If the answer is yes, then the explanation is complete, and the other nine features can be *disregarded.* If the track was not muddy, on the other hand, this feature cannot be part of any explanation in this instance. The important thing is that one does NOT want to use all ten features in every combination to explain the failure: the thousand or more new conjunctions which would be created as a result would be mostly useless and, what's worse, completely absurd as a model of how experts explain their failures. The fact is that people are usually content with only *one* explanation, and if not one then two or three possibilities at most will suffice. Again, the crucial fact to recognize is that, as difficult as explanation is, it solves other difficult problems about hypothesis generation that cannot be handled by simple "inductive generalization," particularly the "what to notice" problem [Schank & G. Collins 1982].

4.1. The necessity for causal knowledge: an example

There are cases when causal reasoning is absolutely essential to explain a failure. The primary example for my purposes here is a thoroughbred race run at Belmont track on September 29, 1982. The HANDICAPPER program and a racing expert whom I asked both agreed that the horse *Well I'll Swan* should win that race, and in fact the expert said that it should "win by a mile." The actual result of the race was that the horse *Decorous* won and *Well I'll Swan* finished second. The most significant features of each horse are summarized in the Table 1*.

One possible hypothesis using simple feature analysis, is to assume that the *dropping down* feature of *Well I'll Swan* is the reason for the loss, since this feature is the only difference between the horses. Anyone who knows about horse racing, however, knows that dropping down is usually a good feature, so this explanation is not adequate. When the expert was told that, in fact, *Decorous* won, he re-examined *Well I'll Swan* to explain why that horse did not win. What he noticed, and what he said he had been suspicious of in the first place, was that *Well I'll Swan* looked *too* good. The reason is that if a horse is winning, as this horse was, then he should be moving

*"Claimed" in the table means the horse was purchased by a new owner just before the race

Horse	Decorous	Well I'll Swan
Features	Finished well in last two races (4th, 1st)	Finished well in last two races (1st, 3rd)
	Claimed in last two races	Claimed in last race
		Dropped down $15,000 from last race

Table 1: Comparison of two horses

up in value, which this horse was not. Furthermore, this horse had just been claimed for $15,000 more than the current race, and it was quite possible that it would be claimed again, which means the owner would take a fast $15,000 loss. If the horse was doing well in more expensive races, it would make no sense to drop him in value. The conclusion this expert made was that there must be something *wrong* with this horse -- it was sick or hurt -- and the owner probably wanted to get rid of it while he could still make some money. Given this conclusion, one would be wise not to bet on the horse, or at least to be wary of it. The rules of the domain which were necessary here included:

1. Higher claiming prices indicate better horses.
2. Races with high claiming prices have larger purses than races with lower claims.
3. The goal of an owner is to make money on his horse, hence to enter the horse in the most expensive claiming races which the horse has a chance to win.
4. If a horse is sick or injured an owner may try to hide this fact until he can sell the horse.

This kind of complex reasoning is the most likely way to arrive at the correct explanation for the win of *Decorous* over *Well I'll Swan*. The good features of the former were shared by the latter, and the only way to predict the results otherwise would be to suppose that moving up in claiming value was better than dropping down, which we have already said is wrong. In order to be suspicious of a horse like *Well I'll Swan*, the racing expert must know a great deal about what claiming a horse means, what changes in claiming price mean, and why owners make these changes. This is an example of why simply knowing what features are involved is not sufficient to explain the results of the race. HANDICAPPER uses the rules enumerated above, in conjunction with the features of the horses in question, in generating its explanation (or new hypothesis) of why *Well I'll Swan* did not win this race.

5. What HANDICAPPER does

The HANDICAPPER program learns the rules of horse racing by predicting winners based on its hypotheses and creating new hypotheses when its predictions fail. The program starts out with the ability to recognize about 30 different features and some simple causal knowledge about each one.

The basic mechanism for generating hypotheses is a function for comparing two horses, the one which actually won a race and the one which was predicted to win (if the horse picked to win does win, nothing new is learned, but old rules are strengthened). The first step in building new hypotheses is to look at all the features that are different for the two horses. Next, the program consults its domain knowledge about racing to ascertain whether each feature might be part of an explanation. HANDICAPPER knows for most features whether they are good or bad, and the few it does not know about will not be eliminated in the initial phase of hypothesis generation. HANDICAPPER also knows some more complex rules about racing, and these rules are used here to notice contradictions. Examples include:

Rule1: Finishing poorly (worse than 4th) in several races in a row should cause an decrease in the claiming value of a horse.

Rule2: Winning a race easily (a 'wire to wire' victory, where a horse is in the lead from start to finish) should cause an increase in the claiming value of the horse.

If these or other causal rules are violated, then the horse is looked upon suspiciously, as possibly not "well meant" (e.g., the race might be fixed). In the example with *Decorous* and *Well I'll Swan*, where a horse was both doing well and dropping down, the simultaneous presence of both of these features caused the program to notice a violation of its causal rules. The explanation it generated has already proven useful in making predictions about later races: in particular, the program processed another race with a horse similar to *Well I'll Swan*, in that it had been claimed recently and dropped down. The program correctly predicted that this later horse would not win (and the horse it actually picked did, in fact, win the race).

When a causal violation occurs, the features responsible for that violation become the sole hypothesis for the prediction failure. Lacking such a violation, the program generates several hypotheses, one for each combination of the feature differences between the predicted winner and the actual winner. These hypotheses are then used to re-organize memory, where the features become indices to a prediction. Similar sets of features on future horses will thereby result in predictions that reflect the performance of horses with those features which the program has seen in the past. The knowledge of which features are good and which are bad is enough to constrain the number of such combinations enormously. Before such knowledge was added to the program, it generated on the order of 5000 new hypotheses for the first three races it predicted, but by adding this knowledge the number of hypotheses was reduced to less than 100 (the number of combinations grows exponentially, and the knowledge of whether features were good or bad reduced the number of feature differences by about half). The addition of causal reasoning rules reduced the number further to only about eight hypotheses per race.

6. Conclusion

HANDICAPPER currently makes predictions on the dozen or so races in its database which agree in general with experts' predictions. After seeing the results of all the races, it modifies its rules so that it predicts every winner correctly if it sees the same races again (without, of course, knowing that it is seeing the same races). The source of all new hypotheses is failure, and in particular the failure of old theories adequately to predict future events. It is this fact which makes failure such an important part of the learning process. Without failures, there never is a need to generate new explanations and new predictions about how the world works. By reorganizing memory after each failure, the HANDICAPPER program makes better and better predictions as it handicaps more races. This memory-based approach to organizing expert

knowledge should prove useful, and in fact indispensable, in a program attempting to become an expert in any domain. The issues raised here demonstrate the increasing importance of causal knowledge in reasoning about complex domains, particularly the need to consult domain knowledge so as to avoid generation of incorrect explanations.

Acknowledgements

Thanks to Larry Birnbaum, Gregg Collins, and Abraham Gutman for many useful comments and help with the ideas in this paper.

Bibliography

Collins, A.
Studies of plausible reasoning. BBN report 3810, Bolt Baranek and Newman, Inc., Cambridge, MA. 1978.

Lebowitz, M.
Generalization and Memory in an Integrated Understanding System. Ph.D. thesis, Yale University. 1980.

Schank, R.
Dynamic Memory: A theory of reminding and learning in computers and people. Cambridge: Cambridge University Press, 1982.

Schank, R. & Abelson, R.
Scripts Plans Goals and Understanding. New Jersey: Lawrence Erlbaum Associates, 1977.

Schank, R. & Collins, G.
Looking at Learning. Proceedings of ECAI-82, pp. 11-18.

Sussman, G.
A Computer Model of Skill Acquisition. New York: American Elsevier, 1975.

Winston, P.
Learning Structural Descriptions from Examples. In P. Winston (Ed.), *The Psychology of Computer Vision*, New York: McGraw-Hill, 1975.

Winston, P.
Learning New Principles from Precedents and Exercises. *Artificial Intelligence* 19:3 (1982), 321-350.