ELSEVIER

International Journal of Forecasting 12 (1996) 57–71

international journal of forecasting

Hailfinder: A Bayesian system for forecasting severe weather[1]

Bruce Abramson[a,b,*], John Brown[c], Ward Edwards[d,e], Allan Murphy[f], Robert L. Winkler[g]

[a]*Cambridge Research Associates, 1430 Spring Hill Road, Suite 200, McLean, VA 22102, USA*
[b]*Carnegie Mellon University, Dept. of Engineering and Public Policy, Pittsburgh, PA, USA*
[c]*National Oceanic and Atmospheric Administration/Forecast Systems Laboratory, MAIL Code R/E/FS 1, 325 Broadway, Boulder, CO 80303, USA*
[d]*Wise Decisions, Inc. 11466 Laurelcrest Road, Studio City, CA 91604, USA*
[e]*University of Southern California Social Science Research Institute, Los Angeles, CA, USA*
[f]*Prediction & Evaluation Systems, 3115 NW McKinley Dr., Corvallis, OR 97330-1139, USA*
[g]*Fuqua School of Business, Duke University, Durham, NC 27708, USA*

Abstract

Hailfinder is a Bayesian system that combines meteorological data and models with expert judgment, based on both experience and physical understanding, to forecast severe weather in Northeastern Colorado. The system is based on a model, known as a *belief network* (BN), that has recently emerged as the basis of some powerful intelligent systems. Hailfinder is the first such system to apply these Bayesian models in the realm of meteorology, a field that has served as the basis of many past investigations of probabilistic forecasting. The design of Hailfinder provides a variety of insights to designers of other BN-based systems, regardless of their fields of application.

Keywords: Bayesian; Belief networks; Meteorology; System design; Weather forecasting; Intelligent systems; Elicitation

1. Introduction

Forecasters must boldly predict the future. The sheer difficulty of this task suggests that a wise forecaster should use as much information as possible before issuing predictions. Available past and current data regarding what is being forecast – and other variables thought to be related to what is being forecast – are certainly helpful. Expert judgment can also be a valuable source of information. Wise forecasters further recognize that the future is uncertain, and report their forecasts in probabilistic terms. Data, judgment, and uncertainty are thus relevant to most forecasting tasks, and should be incorporated into forecasting techniques.

In real-world forecasting problems, there are often large numbers of variables and relationships among them that may be viewed as potentially important, leading to complex knowledge structures. Most traditional forecasting tech-

* Corresponding author.

[1] This work was supported in party by the National Science Foundation under grants SES-9106440 and IRI-9424378, in part by the Forecast Systems Laboratory of the National Oceanic and Atmospheric Administration, and in part by the Wood Kalb Foundation.

SSDI 0169-2070(95)00664-8

niques apply sophisticated mathematics to relatively simple knowledge structures and may impose restrictions on the types of inputs that can be used. In contrast, modern computing makes it possible to unlock the intricacies of complex knowledge structures using only simple mathematics. Furthermore, the simplicity of individual relationships within these complex networks allows the model to capture the multiple types of data that populate many domains.

Forecasting of severe local storms is one such domain. Like any aspect of weather forecasting, severe storm forecasting requires a solid background in atmospheric science. The ability to recognize short-fused localized weather phenomena, however, also requires a good deal of location-specific knowledge. A model capable of forecasting these storms should thus be able to combine 'hard' meteorological data and mathematical equations that describe the environment with 'soft' judgments made by a local expert.

This article describes the design of Hailfinder, an explicitly Bayesian system that forecasts severe local weather in Northeastern Colorado. Hailfinder uses a *belief network* (BN) to combine atmospheric science and local judgments using conditional probabilities; BN-specific inference algorithms are used to unlock the network's implicit relationships (Jensen, 1996). Discussions about Hailfinder and its underlying theory were initiated in mid-1989, and led to a proposal that fall. Funding was obtained, and serious system-development work began in September 1991. The system's structure (i.e. the variable set and the direct interrelationships captured by the BN) was essentially set by mid-1992, and the probabilities necessary for the system to run were specified by mid-1993.

Evaluating Hailfinder would require a large, complex, and costly set of observations of local storms in the plains of Eastern Colorado in the summer months, and of the local, national, and global meteorological data that it needs to make forecasts. Neither already collected data sets from past summer months nor funding to collect such information about present ones have been available. These logistical difficulties have made it difficult to test and/or to fine-tune the system on real data, and stalled Hailfinder's development at the end of its first design stage. As a result, this article must be read as a case study in system design, rather than as a report about the performance of our system, or even about its performance relative to other systems designed to address the same task. We hope that such a paper will be forthcoming. In the meantime, however, the development of Hailfinder did provide us with a variety of insights about the elicitation of large data sets comprised entirely of expert judgments and about the design of probabilistic intelligent systems.

The remainder of this article is organized as follows: Section 2 provides background information about two key topics, the specific task that Hailfinder was designed to address and our reasons for choosing a BN as the knowledge base format with which to best address that task. Section 3 describes some of the key issues that we had to consider while designing Hailfinder, and reviews the insights that we gained while considering them. Section 4 provides a fairly high-level overview of the system; the Appendix contains a more detailed system description. Section 5 combines a brief summary with some concluding remarks.

2. Background

2.1. Task selection

In 1987, the Forecast Systems Laboratory of the National Oceanic and Atmospheric Administration (NOAA/FSL) announced a forecasting experiment called Shootout-89; a follow-up experiment, Shootout-91, was announced 2 years later (Moninger et al., 1991). The Shootouts were intended to allow comparative evaluations of the forecasting performance of human experts, knowledge-based systems, and conventional statistical systems at a fixed task. The task selected for the Shootouts was the forecast of thunderstorm-related weather over northeastern

Colorado. The nature (format, lead time, etc.) of the forecast and the precise geographic region involved in the forecast were both specified, and all automated systems were run by an impartial operational meteorologist at NOAA. Shootout entrants (human or automated) were allowed to draw their inputs from the set of meteorological data collected and disseminated daily by NOAA. Their output had to be a probability distribution across three mutually exclusive and exhaustive weather categories:

Severe: Occurrence at some point within a specified area, during a specified time period, of at least one of (i) hail with a diameter greater than or equal to 0.75 inches, (ii) surface winds of 50 knots or greater, or (iii) a tornado.

Significant: Occurrence at some point within a specified area, during a specified time period, of at least one of (i) hail with a diameter between 0.25 and 0.74 inches, (ii) surface winds between 35 and 49 knots, (iii) rainfall of at least 2 inches h^{-1}; or (iv) a funnel cloud.

Nil: The absence of significant or severe weather.

The results of Shootout-89 were somewhat disappointing. Not one of the six automated systems was able to outperform a simple base-case climatological forecast in a statistically significant manner (Moninger et al., 1991). The results of Shootout-91 were only mildly better (Walker et al., 1992).

Hailfinder was initially conceived as an entrant in Shootout-91. We noted that every one of the knowledge-based systems entered in Shootout-89 was based on the same modeling technology:production rules. Rules are one of the three classical representations popularized by the developers of first-generation expert systems (the other two are logic and frames) (Steels, 1990). We believe that Hailfinder's underlying BN model constitutes a much more powerful approach to system design (Abramson and Ng, 1993), and one that is particularly well suited to the task of forecasting severe weather phenomena. Unfortunately, logistic difficulties made it impossible for system design to begin before late 1991. This delay, and NOAA's subsequent decision *not* to sponsor a Shootout-93, have so far precluded a rigorous test of Hailfinder.

2.2. Technology

The definition of the Shootout task required every entrant to output five discrete probability distributions per day, each defined across three (mutually exclusive and exhaustive) weather categories, and tied to a well-defined geographic region drawn on a map. It placed no restrictions, however, on the systems' internal workings used to generate those probabilities. Systems that forecast probabilities can be categorized on the basis of how the probabilities are produced:

1. The output of a deterministic forecasting model that yields an exact, categorical forecast can be post-processed. For example, a human expert may turn a categorical forecast into a distribution by saying 'that model is right only 80% of the time when it makes this specific forecast'. Historical data on the model's forecasts and the corresponding outcomes make statistical versions of post-processing possible.

2. Inputs to deterministic forecasting models may be treated as uncertain. If the current distribution of one or more input variables is known, then this distributional information may be propagated through the model to yield a distribution of possible outputs.

3. The forecasting models themselves may be inherently probabilistic.

Combinations of these approaches are also possible.

Belief networks offer one appealing technology to use in building inherently probabilistic models. BNs are modeling tools, recently popularized among parts of the decision analysis and artificial intelligence communities, that allow probabilistic representations of uncertainty to be captured in model input, internal processing, and output. Stated simply, a BN models a domain's relevant variables as *nodes* in a graph. Each node is described by a distinct set of *states* that it may assume. *Directed arcs* between nodes are drawn to indicate direct influence among variables. Each node then specifies an algorithm that

converts inputs to outputs. Examples of these conversion algorithms include (but are not restricted to) algebraic formulas and conditional probability distributions. Node outputs are then expressed as either a single *instantiated* state (i.e. the member of the state-set that the node assumed) or as a probability distribution across its state set. Nodes and arcs are typically referred to as a BN's *structure*, and mathematical/probabilistic relations as its *parameters*. For reviews of the BN literature, see Henrion et al. (1991) and Matzkevich and Abramson (1995).

Relatively few accounts of BN-based systems have been published. Successful systems in the literature include (but are not limited to) Pathfinder for medical diagnosis (Heckerman et al., 1992) Rachel for advising couples with fertility problems (Holtzman, 1989), and the ARCO-BN systems for forecasting crude oil prices (Abramson and Finizza, 1991, 1995). This paper introduces another example, named Hailfinder, that forecasts severe weather on the plains of Northeastern Colorado in the months of June, July, and August.

One common assumption among BN-based systems is that domains should be described as collections of interrelated objects, and that non-definitional relationships are often best described as conditional probability distributions. The domain of severe weather fits these assertions nicely. Meteorological data are abundant, regularly available, public, and subject to uncertainty of interpretation. Weather systems are huge, complex, and chaotic. As a result, meteorological forecasts are frequently presented probabilistically, in some cases even to the general public. Much experience with and thought about probability forecasts in meteorology have led to a good understanding of how to evaluate them (Murphy and Winkler, 1987, 1992). Hailfinder's BNs apply state-of-the-art probabilistic modeling techniques to a longstanding and familiar area for probabilistic forecasting.

Both the structure and the parameters of Hailfinder (and of the other BN-based systems mentioned above) came from the judgments of a domain expert. In principle this feature is not necessary. Structure could come from standard models of the physical processes leading to the predicted event, and probabilities could be assessed by collecting historical relative frequencies. But by making complex probabilistic models easy to describe and to use, BNs permit users to avoid some the simplifying assumptions normally necessary for successful modelling. BNs are at their best in dealing with complex phenomena, not simple ones. Complex phenomena seldom have standard models, and the relevant probabilities are typically neither collected nor recorded. Models based on BNs are likely to continue to depend on, and so to represent, judgments by experts.

Dependence on judged probabilities is only one of the links between BN models and the Bayesian position. Another has to do with the nature of the processes represented by the model. For a person with a non-Bayesian outlook, use of an inherently probabilistic model to represent a complex deterministic physical process is at least uncomfortable and probably wrong. Better to tease out the full detail of the physical process. For a Bayesian, the probabilistic model is a model, not of the physical process itself, but rather of a coherent set of beliefs about it. Those beliefs incorporate knowledge of the physical process when that knowledge exists, but focus more on linkages among physical processes about which deterministic knowledge does not exist, but experience and opinion do. Such models therefore represent a sort of halfway house between pure judgment in probabilistic form and deterministic modelling.

The Bayesian view of inference (Edwards et al., 1963; Winkler, 1972; von Winterfeldt and Edwards, 1986) is well-adapted to the needs of the Shootout-specified forecasting task. It combines the formal properties of probability theory (e.g. rules for combining evidence) with the view that probabilities are orderly opinions. The introduction of BNs has made work with complex probabilistic models far easier than it used to be. Furthermore, past research has shown that weather forecasters do quite well at making explicitly probabilistic judgmental predictions (Murphy and Winkler, 1984, 1992). Our adoption of the BN as the model underlying Hailfin-

der thus emerged from several beliefs about elicitation, system design, and the weather forecasting problem.

3. Key issues

This article is intended as a case study in system design. This section describes some of the key intellectual advances, heuristics, and tricks that we learned during Hailfinder's design. Would-be designers of comparable systems may find some of these hints useful.

3.1. Basic concepts

3.1.1. Location

Perhaps our most severe intellectual challenge was to create a Bayesian knowledge-based system of acceptably small size. The problem was simply that all hypotheses about severe weather must be specific about their location, and all data are location specific. At the same time, the atmospheric constructs that affect local weather are potentially hemispheric or global in scope and the interactions among them may be extremely complex. As a result, the forecast task began by implying a potentially huge variable set and a potentially vast hypothesis set (e.g. by treating each possible location in which a thunderstorm might develop as a separate hypothesis). Fortunately, a reduction of the hypothesis set was provided by the Shootout ground rules: Northeastern Colorado was divided into four smaller regions (labelled Regions 1 through 4), and forecasts were required for all four, as well as for their union (labelled Region 5). Also, although all inputs and all intermediate events have locations, the location of a specific phenomenon can generally be specified as either in the mountains or in the plains. When discrimination among regions was important and location information was relevant, we preserved it by making it a part of state specification for relevant nodes.

3.1.2. Scenarios

Among the most significant steps taken towards managing the complexity of the observation set was John Brown's introduction of *scenarios*. (Brown, one of the authors of this article, was the domain expert for Hailfinder.) Roughly speaking, a scenario is a description of a typical day. The recognition and interpretation of these 'typical days', of course, requires a substantial amount of local expertise. Scenario names like 'Denver Cyclone', 'Longmont Anticyclone', or 'Front Hung up on Palmer Ridge', are unlikely to impart much information to most readers. To a meteorologist with local expertise in NE Colorado, however, they are quite informative; they describe different local weather regimes, and provide different frameworks within which clouds, moisture, temperature, and other meteorological variables may be interpreted.

Brown was able to describe ten scenarios (i.e. day-types) that capture better than 80% of the days during the summer severe weather season; an eleventh, 'other' scenario captures the rest. Each scenario imparts different expectations about the weather and provides different interpretations of meteorological data. The scenario concept thus recasts the forecast as a two-step process. In the first step, certain cues (including personal observations of the weather outside, outputs of meteorological models, and readings of data available in the forecast center) are used to 'diagnose' the scenario, or to determine what type of day it is. In the second step, additional data are collected and interpreted within the scenario to generate the forecast. This two-step process embodies the type of hierarchical Bayesian inference and probabilistic conditioning formalized in a BN, but it in no way constrains software implementations. In fact, the current working version of Hailfinder collapses the two conceptual calculations into a single BN, and merges scenario diagnosis and variable interpretation into a single Bayesian-updating process. The impact of the scenario node, however, was more than just conceptual – the management of a central eleven-state node played an important role in elicitation and system specification. This topic is discussed in greater detail in the Appendix.

3.1.3. System boundary

When Hailfinder is actually used in a forecast center, day-specific inputs will be input by an operational meteorologist. The system's inputs must thus be judgments of the sort with which competent meteorologists are comfortable. This requirement casts the system as a complex bootstrap; simple judgments are input and complex one are output. It also forced us to focus on the role of the human user in an interactive system. Within the system, all processing is done by computer; the *system boundary* separates what the computer does from what people do.

A great deal of information is available at a meteorologist's computer-based workstation. Any or all of it could be relevant to Hailfinder. We did not wish to spend the time and modeling effort required to reduce the size of the potential input set through a series of complex sensor readings and interpretation algorithms. We chose instead to let the user summarize the information as a relatively coarse judgment. This approach dramatized the concept of system boundary. The distinction between nodes inside the system boundary and the input nodes outside the boundary is that all processing inside the system boundary is algorithmic and so is done by the computer; processing outside the system boundary is judgmental, and may be done by people. The states that the nodes immediately outside the system boundary may assume, however, are specified by the system (i.e. free-form judgments are disallowed). All inputs must be specified as either a selection of or a distribution across the state-sets of nodes at the system boundary.

Our conclusion that most inputs had to be judgmental establishes an implicit research agenda for the future: the automation of (at least some) judgment formation by moving the system boundary. As a general rule, every formal model and/or intelligent system requires that some work be done by a human user before processing may begin. In 'interactive' systems, the exchange between human data processing and automated data processing is iterative. In 'stand-alone' systems, humans begin the work and computers finish it. The system boundary provides a useful conceptualization of the seam between the portion of a task that machines can handle and those that remain in the domain of human intelligence. During the design of Hailfinder, the concept of the system boundary helped us clarify and standardize the system's input needs and the types of information processing and automated aggregation that it could perform.

3.2. Elicitation strategy and tactics

Our elicitation strategy progressed through three phases. The first phase was basically educational; the non-meteorologists learned about meteorology in general and the specifics of forecasting in NE Colorado in particular, and those unfamiliar with BNs learned about them. The second phase was essentially a translation of Brown's meteorological knowledge relevant to this problem into BN form. Its result was, to some extent, the equivalent of trying to embody a textbook of meteorology in a BN, and was much too complicated for the purpose at hand. This phase produced a structural model that, if fully specified, would have required about 66 000 assessed probabilities. As intended, the thought of assessing that many probabilities encouraged the expert to rethink parts of the model. In the third phase, the textbook model was streamlined to address the specific purpose of producing the outputs required by the Shootout experiment, rather than the more general task of forecasting local weather.

All phases of the elicitation required enormous patience and careful bookkeeping. Patient tutoring is required in the educational phase (and beyond), and patient deliberation is required in all subsequent phases. Bookkeeping was achieved by maintaining a sequence of state-of-the-system documents as a project memory; they allowed us to reconstruct the model's evolution, as well as its state at a given point in time.

3.2.1. Structure

The elicitor's first task is to understand both the task that the system being designed must perform and the expert's normal way of performing it. Local weather forecasting was the focus of Phase 1. The actual BN model began to emerge

during Phase 2. BN elicitation consists of two distinct tasks: building a graphical structure and assessing probabilities. Phase 2 consisted entirely of structural discussions. One key issue addressed early in this phase was the ordering of work on the parts of a vaguely conceived whole. Since the Shootout task provided a very explicit definition of required system output, it was natural to begin by designing output nodes and nodes immediately antecedent to them, then nodes antecedent to them, and so on, back towards the inputs. When that process seemed to be diminishing in value, we switched to developing structures starting from the inputs. We also spent time on intermediate fragments.

This bidirectional elicitation procedure led to the development of nine fragments: (i) Scenario Diagnosis, (ii) External, (iii) Other Inputs, (iv) Vertical Motion, (v) Moisture, (vi) Clouds, (vii) Intermediate Processes, (viii) Interpretation within Scenario, and (ix) Forecasts. (These fragments were modified during Phase 3 of the elicitation. Structural details of the final fragment set are shown in the Appendix.) The next step consisted of linking them together to yield a single BN instead of a set of fragments. The final step consisted of specifying, for each node, the number and labels of the states it could be in.

Though some modest efforts at simplification were made in Phase 2, it became the focus of effort in Phase 3. A theme emphasized from the beginning was that all structure was tentative and subject to revision. The emphasis during Phase 2 had been on 'getting it right' (i.e. finding a representation of Brown's meteorological expertise that he felt was appropriate and relevant to the task at hand).

The emphasis in Phase 3 was on determining whether specific elements of structure were really necessary. Though we recognized that removal of nodes could help, we did not focus on them. Instead we focused on removing arcs. The complexity of both probability assessment and computation is largely a function of a BN's bushiness, or the number of arcs that point to each node. The design of the most parsimonious yet realistic model is thus the key to the success of the entire system-design effort. One tactical trick that helped pare down the original BN was the reconsideration of generalized triangles: instances in which more than one directed path connected two nodes. Triangles often imply redundancy. After all, if information flows directly from A to C, and *also* from A to B to C, the information captured in the second path may be subsumed by the information passed along the first. Each triangle was the occasion for discussion of the domain issues it addressed. Sometimes those discussions led to the deletion of an arc, sometimes to the redefinition of one or more nodes, sometimes to the removal of a node, and sometimes to the conclusion that the structure should remain as it was.

When all arcs had been thus reviewed, the next step was to look inside each node to see if the number of states it could be in could be reduced. The focus was explicitly on whether the fineness of grain produced by, say, four states was needed for integrity of the output forecasts. If fewer states in that node would serve, the node was thus redefined. Again, discussions in the course of doing so led to further simplifications.

More specifically, state sets enumerate the values that variables may assume. State-set specification can lead to structural modifications in three ways. All three proved to be useful in designing Hailfinder.

First, it sometimes becomes evident that nodes do not capture precisely defined variables, and that they must be rethought before suitable variables may be defined and states may be assigned. One example of this type of shift occurred when a single node, Temperature and Moisture Stratification, was split into four components, (i) Mid-level Lapse Rate, (ii) Low-level Lapse Rate, (iii) Mean RH (relative humidity), (iv) RH Ratio. Each of these variables retained some of the discriminatory properties of the original variable while introducing the degree of specificity necessary to define state sets.

Second, even when variables are sufficiently precise, the assignment of state sets is often obscure, particularly when a continuous variable is discretized. (Note that continuous variables are precluded by available software, and com-

plexity argues for state-sets that are as small as possible for virtually the same reasons that bushy BNs are problematic.) Since the specific task for which Hailfinder was designed is considerably narrower than general meteorological forecasting, rather coarse differentiations are often possible.

Third, some variables can be grouped together logically and routed through a 'collector' node before impacting the rest of the system. One example of collection can be seen in the Vertical Motion fragment (see the Appendix). Vertical motion in the atmosphere may be extracted from (at least) three different sources: (10.7μ) satellite data, quasigeostrophic theory, and subjective expert judgments. Each of these sources is represented as a distinct node, and each one discriminates among four grades of vertical motion (strong up, weak up, neutral, and down). Any node to which all three of them pointed would thus have the size of its conditional probability table multiplied by a factor of 64 (4^3). They are summarized, however, by the singe collector node, Combined Vertical Motion, which reduces the 64 possible outputs back to the original four. Several of these collector nodes were introduced to help manage the BN's complexity.

These simplification strategies were highly effective. By the end of Phase 3 structuring, most of Hailfinder's variables had fewer than five states; some had as many as seven. One central node, however, continued to stand out. The Scenario node presented a special problem because it has eleven states – a large number. The elicitation of structure surrounding the scenario node was facilitated by our use of a discrimination table, or a table that listed the inputs necessary to discriminate among each pair of scenarios. This discrimination table is similar in spirit to (but considerably simpler than) the *similarity networks* introduced in the development of Pathfinder (Heckerman, 1991). The purpose of both of these tools was to focus the elicitee's attention on the relatively simple task of discriminating between pairs of variables, rather than being overwhelmed with the challenge of global discrimination. Hailfinder's discrimination table is shown in the Appendix.

3.2.2. Probability

3.2.2.1. Elicitation

The simplifications described above led to a solid structure within which probabilities could be assessed. Even after all of the streamlining and simplifying was done, however, the model required over 3957 probabilities – a number that is quite large, but certainly approachable. By this time the domain expert know as much about how to work with BNs as did any member of the team. After some practice and some discussion of sample sets of assessments, he was left with the task of making all of these assessments himself, alone.

In retrospect, this was a mistake. Brown had other work to do, unrelated to Hailfinder. This remarkably tedious and endless task was easy to postpone. And it got postponed. Fortunately, we were able to find compromises that combined with Brown's natural diligence to enable him to finish the job of probability assessment. Our advice to designers of similar systems, however, is: no matter how expert the domain expert becomes, you will stress him or her beyond endurance if you allow probability assessment to be solely his or her responsibility.

3.2.2.2. Consistency checks

One of the reasons (perhaps *the* reason) that probability specification became the sole task of the domain expert was that the system designers could think of relatively few ways in which they could help. Relatively few, however, is not none. We were able to exploit two structural aspects of conditional probability assessment to provide two strong internal consistency checks on judged probabilities.

The first is the sum check. For each mutually exclusive and exhaustive set of events, probabilities must sum to exactly 1.0. No automatic normalization was allowed. Moreover, Brown was urged to treat each event in a partition as a topic of thought independent of the other events in that partition. This treatment often led to sums other than 1.0. Whenever that occurred, Brown was urged to think carefully about each element of the partition. Was its probability too high or too low?

A far more important consistency check arises because most of the probabilities assessed were conditional. Table 1 presents an example: the conditional probability table relating Scenarios to Wind Fields, Plains. Wind fields in the plains are among the factors used to diagnose scenarios. The entry in cell (i,j) of Table 1 thus indicates the expert's judgment of the probability that Wind Field i will be present, given that the day is, in fact, of Scenario type j. The natural approach to filling this table is to assess the probabilities in the first column, making sure that they sum to one. Once that has been done, focus can shift to the second column, and so on, until the table is complete.

When the table contains a full specification, the elicitee can be asked questions about the relationship between probabilities in one column and those in another column. For example, is a widespread downslope more likely if today is a Dry Microburst (G) day than it is if today is a Cheyenne Ridge (D) day? If the answer indicated by the conditional distribution table appears counterintuitive to the expert, at least one of the two assessments must be changed. But an elicitee cannot change just one number in a column; the sum check compels such changes to be of at least two numbers. New numbers are thus introduced, and similar comparison of them with numbers in other columns become appropriate. Only when the entire system of numbers is coherent is the process of assessment complete.

If done conscientiously, this kind of check is so demanding that it leads to assessment, reassessment, and re-reassessment of virtually every number in the table. The process is fascinating to watch. The elicitee, from an initial feeling that the numbers being assessed are vague and hard to specify, quickly comes to a resentment of the demands for consistency implicit in the checking process – which can easily lead to needs for assessments to three decimal places. After much practice, Brown discovered that failure of such a cross-check should lead to a rethinking of an entire distribution, not just of one number.

We speculated that the strong determination of probability assessments engendered by the cross-checking described above would lead to highly reproducible assessments. As an informal check on that hypothesis, we obtained test–retest judgments of probabilities in one table from Brown. The retest, obtained more than a month after the original judgments, was a complete surprise to Brown. He made no effort to remember his original judgments; he went through the process of making and then checking all judgments as he had done all along. The result is presented in Table 2. Informally, this is remarkably high test–retest reliability. If this kind of finding can be replicated more formally, it will constitute strong evidence that probability as-

Table 1
An example of a conditional probability table. Table 1 is representative of Hailfinder's conditional probability tables. Note that columns contain probability distributions, but that rows do not. The rows are headed by the six members of the Wind Field, Plains' state set. Columns are headed by the eleven-member state set of scenario

Scenario	A	B	C	D	E	F	G	H	I	J	K
Light Variable											
Denver Cyclone											
Longmont Anticyclone											
E–NE											
SE Quad											
Widespread Downslope											

The eleven types of local weather days that served as the states of the Scenario node: A, Denver Cyclone (DCVZ) w/through, SW flow; B, DCVZ; C, Longmont Anticyclone; D, Cheyenne (CYS) Ridge; E, Monsoon w/SW Flow Aloft; F, Indonesian Monsoon; G, Dry Microburst; H, Stable, Post-Frontal Upslope; I, Front Hung Up on Palmer Ridge; J, Ridge Aloft, F/V flow W of Dry Line; K, Other.

Table 2
Two full specifications of a conditional probability table, generated independently by the same expert, roughly 1 month apart. The top entries in each cell belong to the first full specification, the bottom entries to the second. All numbers refer to chances out of 100. The scenarios are coded according to the key given with Table 1

Scenario	A	B	C	D	E	F	G	H	I	J	K
Light Variable	3	3	5	25	55	35	30	10	10	65	5
	5	5	10	20	40	30	35	7	20	65	10
Denver Cyclone	80	80	0	3	5	20	15	5	35	5	15
	77	75	0	10	17	20	5	5	20	10	5
Longmont Anticyclone	0	0	90	30	15	5	25	10	5	20	25
	0	0	80	15	13	5	10	2	10	10	20
E–NE	5	5	0	2	10	30	5	75	30	0	15
	3	5	0	10	10	25	5	78	25	0	20
SE QUAD	12	12	0	20	10	10	5	0	15	0	10
	15	15	0	25	15	20	5	8	25	10	15
Widespread Downslope	0	0	5	20	5	0	20	0	5	10	30
	0	0	10	20	5	0	40	0	0	5	30

sessments linked by this kind of internal consistency check are not at all vague, variable, imprecise numbers that some parts of the probability assessment literature might lead us to fear.

4. System overview

Most knowledge-based systems contain components (often the entire system) that are either proprietary or classified. Because Hailfinder is entirely in the public domain, we can specify the system at arbitrary levels of detail. This section presents a high-level overview of the interactions among Hailfinder's conceptual fragments and of the flow of information through the system.

Fig. 1 shows Hailfinder's schematic. Each node in the schematic represents a network fragment of several nodes. Note that these fragments are similar, but not identical, to the set developed in Phase 2 of the elicitation (as described in Section 3).

The three nodes in the upper left-hand corner diagnose the scenario relevant to the day being forecast. Date helps calibrate the forecaster's expectations about scenarios; not all scenarios are equally likely throughout the severe weather season. The Inputs to Scenario fragment captures the judgments and data readings that the user inputs into the system. These entries are combined with the assessed prior and conditional probabilities to yield a distribution across the (eleven) Scenarios.

The scenario probabilities are then combined with a set of judgments made by the operational meteorologist (user) outside the system boundary – judgments about Vertical Motion, Moisture, Clouds, and a few Other Atmospheric Conditions. As noted earlier, these judgments are used to determine whether the day is expected to have a greater or lesser probability of severe or significant weather than a typical day

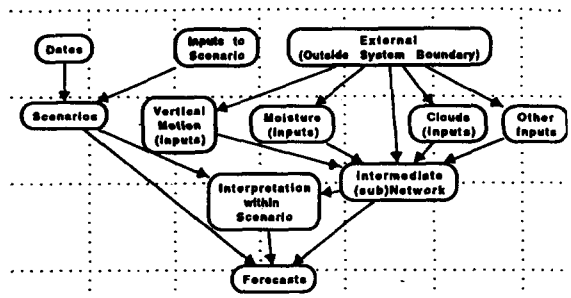


Fig. 1. Schematic of Hailfinder.

with the already-determined distribution of scenario probabilities. Specifically, these judgments are combined through a set of intermediate variables having states corresponding to forecast-relevant atmospheric conditions and probability distributions across these states determined by these judgments. This process is executed by both the Intermediate (sub)Network and the Interpretation within Scenarios fragments. Finally, the atmospheric variables, their scenario-dependent interpretations, and the diagnosis of scenarios itself combine to yield a Forecast.

5. Summary

Hailfinder is a member of a growing class of intelligent systems based on BNs. So far, the performance of members of this class has been highly encouraging. Hailfinder's BN has five major sectors: inputs, scenario determination, physics, interpretation, and forecast generation. The interplay among these components of the BN casts Hailfinder as a complex bootstrap that generates complex judgments from simple ones.

The conceptual simplifications to structure introduced during our consideration of location, scenarios, and system boundaries helped us reduce an unmanageably complex task to a challenging but achievable one. They also, however, added to our collection of general modeling and system-design tools. In particular, we believe that scenarios and system boundaries are useful ways to think about many forecast tasks and about the translation of information from human-accessible form to formal model-accessible form. Scenarios proved to be invaluable in conveying information about meteorological forecasting from experts to meteorological novices (Abramson, Edwards, and Winkler), as well as an important computational simplification. System boundaries, defined so that inputs are easily observable by users, were equally useful in discussing abstractions about system design with meteorologists (Brown, Murphy). The contribution of the one team member with a background in both meteorology and modeling (Murphy) proved to be invaluable in aiding the mutual education process. Scenarios and system boundaries are general concepts that promise to be applicable in many domains and that should facilitate the design of further inherently probabilistic forecasting systems.

Two further techniques emerged after the initial structure was set. First, a judicious reduction in the number of parents and the number of states they can attain reduced the probability assessment burden without unduly compromising the model. Second, checking qualitative relationships in rows while maintaining column sums in nodes' conditional probability tables supported coherent probability elicitation.

One issue that remains unaddressed, of course, is the system's performance. Hailfinder was motivated by the existence of a level playing field for system evaluation, the Shootouts. Unfortunately, that evaluation mechanism expired before the system was fielded. Attempts to assemble a suitable data set are ongoing.

Appendix A. Model specification

This Appendix describes some particularly interesting aspects of Hailfinder's structure. It does not, however, purport to show all of the system's details. Readers interested in a full specification should contact the authors. What this Appendix does provide is: (i) additional detail about the design of Hailfinder's single most interesting subtask, the diagnosis of a day's scenario, and (ii) structural diagram of all fragments and of the entire system. Versions of Hailfinder were implemented using Demos (Drake and Arnold, 1991) and Ergo (Noetic Systems, 1991), two software packages for BN-based system design.

A.1. Input to Scenario fragment

The network fragment of Fig. A1 represents the Input to Scenario fragment. The task of the

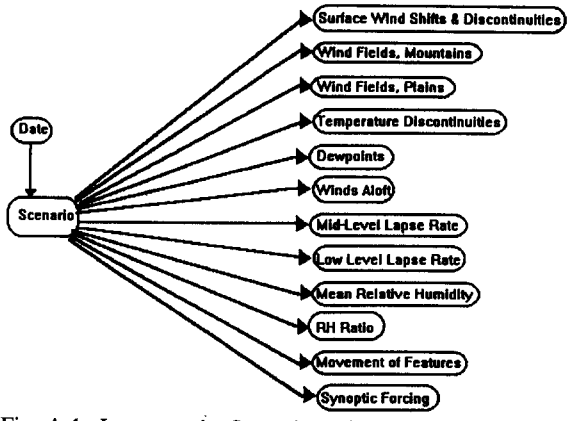


Fig. A.1. Inputs to the Scenario node. Though only Date is drawn as an input, in fact all nodes listed here are inputs to the Scenario node, assumed to be conditionally independent of one another in order to reduce the number of conditional probabilities to be assessed. No node (other than Scenario) in this figure links to any other node in Hailfinder. Thus Hailfinder's inputs are sharply segregated. The ones specified here, all available early in the day, serve only to diagnose today's scenario. Those specified in Figs. A.3 and A.4 available by late morning or early afternoon, are interpreted to produce the required forecasts on the basis of that diagnosis.

nodes in this fragment is to diagnose the day's Scenario. It is important to note that even though these nodes are considered to be 'inputs', arcs are drawn *to* them *from* Scenario, rather than in the other, more natural direction. Although this reversal of directionality introduces spurious independencies, it gains a tremendous amount in tractability. As the fragment stands, the number of probability assessments needed is 11 (the number of states of scenario) times the number of the state-set sizes of the 12 input variables; if they were drawn the other way, the number of necessary assessment would be eleven times their *product*. In the opinion of the domain expert, the fidelity lost by introducing these independencies was negligible, and certainly worth the increase in both assessment and computational efficiency. When the network is used, inputs are observed and arcs reversed using Bayes' rule.

Arc reversal provided a substantial reduction in the fragment's complexity. Nevertheless,

Table A.1
Scenario discrimination table. Letters refer to scenarios, numbers to input variables. The numbers in cell (i,j) enumerate the variables necessary to differentiate between scenario i and scenario j. Key discriminators are shown in bold

	A	B	C	D	E	F	G	H	I	J	K
A		10,15	**5**	**5**,6	5	5,6,**10**	9,11	**5**	5	5	**5**,**6**
B			**5**	**5**,6	5	5,6	**11**	**5**,**6**	5,6,8	5,9,**10**	**5**,**6**
C				4,5,6	**5**,**10**	**5**,9,11	9,11	**5**	5,6,8	4,5,6,9,10	5,10,12
D					10,11	4,11	5,9,11	5	5,8,11	5,6,9	5,6,10,12
E						9,**11**	**9**,**11**	**5**,11	5,6,8,9,10,11	9,**10**	**10**,**11**
F							**9**,**11**	5,9,11	6,8,9,**11**	**9**,**11**	9,10,**11**
G								5,**11**	6,8,**9**	6,9	4,5,6,8,9,**11**,12
H									5,8	**5**,**6**,8,9,10	5,6,**11**,12,15
I										5,8,9,10	5,**6**,**8**
J											9,**10**,**15**
K											

Discriminatory variables: 4, Wind Fields, Mountains; 5, Wind Fields, Plains; 6, Wind Shifts and Discontinuities; 8, Temperature Discontinuities; 9, Dewpoints; 10, Winds aloft; 11, Temp. and Moist. Stratification; 12, Movement of Features; 15, QG Vertical Motion.

Scenarios: A, DCVZ w/trough, SW flow; B, DCVZ; C, Longmont Anticyclone; D, CYS Ridge; E, Monsoon w/SW Flow Aloft; F, Indonesian Monsoon; G, Dry Microburst; H, Stable, Post-Frontal Upslope; I, Front Hung up on Palmer Ridge; J, Ridge Aloft, L/V flow W of Dry Line; K, other.

Note that some of the terminology is held over from earlier versions of the system (i.e. Phase 2 elicitation). This convention was maintained to stress the incremental nature of system development. The variable *Temperature and Moisture Stratification* was later split into four components; (i) *Mid-level Lapse Rate*, (ii) *Low-level Lapse Rate*, (iii) *Mean RH*, (iv) *RH Ratio*. Each of these variables may assume different sets of labels. As far as the triangle table is concerned, however, no distinctions were made among them.

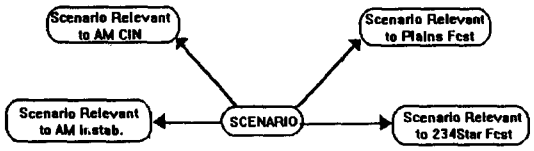
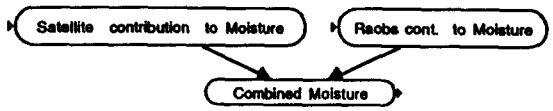
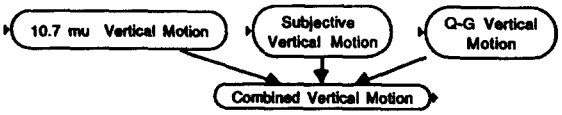


Fig. A.2. The Scenario node and its collector nodes. The state set for each collector node is defined by combining the scenarios that are equivalent for the specified purpose. For example, Scenario Relevant to AM CIN has two states. One combines the two states in the Scenario node that refer to Denver Convergence Vorticity Zone; the other combines all 9 other states of the Scenario node. The probability of each state in a collector node is the sum of the probabilities of the appropriate states in the Scenario node.

eleven times the sum of the state-set sizes remains a large number. In order to reduce assessment complexity even further, Brown recognized that although every variable in this fragment plays a role in diagnosing the scenario, they are not equally important to all diagnoses. The discrimination table of Table A.1 specifies the



(a) Moisture Fragment



(b) Vertical Motion Fragment

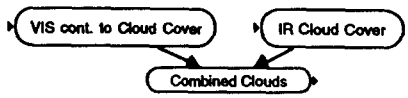


Fig. A.3. Three fragments with 'collector' variables. These fragments summarize information from different sources about moisture, vertical motion, and clouds, respectively.

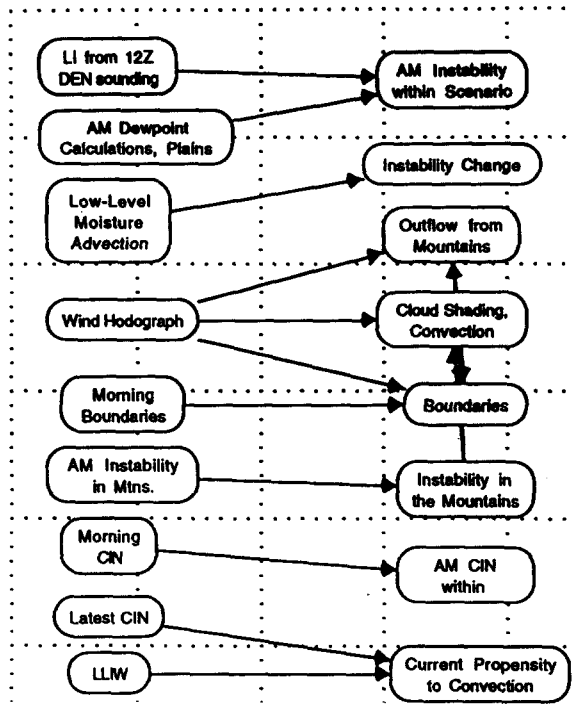


Fig. A.4. The 'Other Inputs' fragment and the intermediate nodes that they influence. Nodes in this fragment are shown in the left column. Nodes in the right column are in other fragments; they are included here simply to show the immediate influences of the input nodes. The variables in this fragment are entered directly by a user. They are 'other' only in the sense that they were not originally elicited as part of the Vertical Motion, Moisture, or Clouds fragments. Some of the input variables in this fragment are judgment calls.

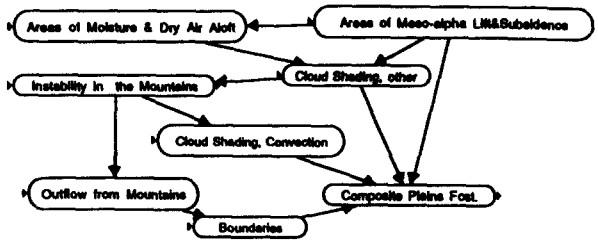


Fig. A.5. The Intermediate (sub)Network. This fragment consists of the nodes that we originally labelled 'intermediate' because they lie somewhere between the inputs and the actual forecast (a fine reason!). Note that they are no longer the only nodes with that characteristic; the Interpretation fragment is intermediate, as well.

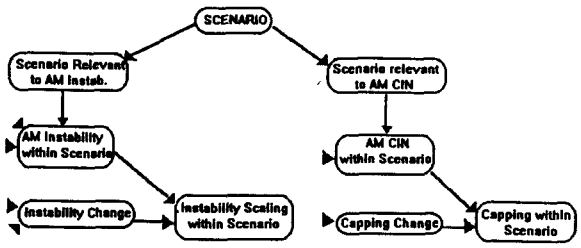


Fig. A.6. Interpretation of Scenario node outputs. Scenario Relevant to AM Instability and Scenario Relevant to AM CIN are collector nodes. The states of both nodes that have the word "Change" in their names are: Less than average, Average, and More than average. The four nodes that have the words "within Scenario" in their names have three states apiece: Less than average for scenario, Average for scenario, and More than average for scenario. Thus the physical measurements and judgement calls that are the inputs to Hailfinder have been coarsely recoded as deviations from values that today's scenario would lead a meteorologist to expect.

variables appropriate to each potential differentiation. Key observations, or those that make it easy to discriminate, are in boldface.

A.2. Other fragments

This Appendix contains the structural components of the other network fragments. The entire system, which contains 56 nodes and 68 arcs, is too large to lay out neatly on a single page.

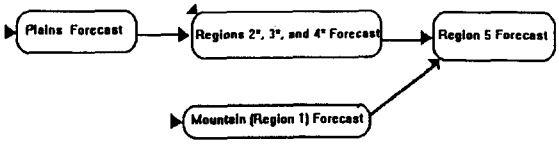


Fig. A 7. Output nodes. The states of these four nodes, by the rules of the Shootouts, are Nil, Significant, and Severe. Regions 1 through 4 are the small, specific regions used in Shootout-89. Region 5 is the entire Denver warning area; it was added to the list of areas for which forecasts were required by Shootout-91. Regions 2*, 3*, and 4* represent Regions 2, 3, and 4 plus or minus the geographic changes needed to make them fill Region 5 without overlap. They are combined because of a tentative conclusion, reached late in the development of Hailfinder, that we would not be able to make forecasts that would differentiate them from one another.

References

Abramson, B. and A.J. Finizza, 1991, Using belief networks to forecast oil prices, *International Journal of Forecasting*, 7(3): 299–316.

Abramson, B. and A.J. Finizza, 1995, Probabilistic forecasts from probabilistic models: A case study in the oil market, *International Journal of Forecasting*, 11(1), 63–72.

Abramson, B. and K.-C. Ng, 1993, Towards an art and science of knowledge engineering: A case for belief networks, *IEEE Transactions on Knowledge and Data Engineering*, 5(4), 705–712.

Drake, K. and B. Arnold, 1991, *Demos Tutorial – An Introduction to Demos* (Rockwell International, Palo Alto, CA).

Edwards W., H. Lindman and L.J. Savage, 1963, Bayesian statistical inference for psychological research, *Psychological Review*, 70(3), 193–242.

Heckerman, D.E., 1991, *Probabilistic Similarity Networks* (MIT Press, Cambridge MA).

Heckerman, D.E., E.J. Horvitz and B.N. Nathwani, 1992, Toward normative expert systems: Part I. The pathfinder project, *Methods of Information in Medicine*, 31, 90–105.

Henrion, M., J.S. Breese and E.J. Horvitz, 1991, Decision analysis and expert systems, *AI Magazine*, 12(4), 64–91.

Holtzman, S., 1989, *Intelligent Decision Systems* (Addison-Wesley, Menlo Park, CA).

Jensen, F.V., 1996, *An Introduction to Bayesian Networks* (UCL Press, London) in press.

Matzkevich, I. and B. Abramson, 1995, Decision analytic networks in artificial intelligence, *Management Science*, 41(1), 1–22.

Moninger, W.R., J. Bullas, B. de Lorenzis, E. Ellison, J. Flueck, J.C. McLeod, C. Lust, P.D. Lampru, R.S. Phillips, W.F. Roberts, R. Shaw, T.R. Stewart, J. Weaver, K.C. Young and S.M. Zubrick, 1991, Shootout-89, a comparative evaluation of knowledge-based systems that forecast severe weather, *Bulletin of the American Meteorological Society*, 72(9), 1339–1354.

Murphy, A.H. and R.L. Winkler, 1984, Probability forecasting in meteorology, *Journal of the American Statistical Association*, 79, 489–500.

Murphy, A.H. and R.L. Winkler, 1987, A general framework for forecast verification, *Monthly Weather Review*, 115, 1330–1338.

Murphy, A.H. and R.L. Winkler, 1992, Diagnostic verification of probability forecasts, *International Journal of Forecasting*, 7, 435–445.

Noetic Systems, 1991, *Ergo Reference Manual* (Noetic Systems, Inc., Redwood City, CA).

Steels, L., 1990, Components of expertise, *AI Magazine*, 11(2), 28–49.

von Winterfeldt, D. and W. Edwards, 1986, *Decision Analysis and Behavioral Research* (Cambridge University Press, Cambridge, UK).

Walker, D.C., W.R. Moninger and T.R. Stewart, 1992, Shootout-91: A strategy for integrating computer assistance into the operational environment, in: *Preprints, Fourth Workshop on Operational Meteorology* (Whistler, BC, Canada), 49–58.

Winkler, R.L., 1972, *Introduction to Bayesian Inference and Decision* (Hold, Rinehart, and Winston, Inc.).