

Session 8

There is a report called “Completing the forecast”. When we produce a forecast, of course, there is an error. What we can do is see if we can represent the error in our forecast, in a rigorous way. The weather system is a chaotic system which implies something in feature called sensitive dependence on initial conditions. It means if you mis-specify your initial condition by just a little bit, then as you forecast over time, even if you have the physics completely accurate, that small error in the way you specify the correct state of the system will inevitably lead to larger errors over time. Those errors do not grow linearly with time; they explode. This error is inevitable, since you cannot completely specify the state of the atmosphere. Even though you could do that accurately, there is not enough sensor. Let’s represent our actual uncertainty about the forecasting. With different initial condition, you get different model runs coming out and then you measure the uncertainty in the forecast based on those models.



FIGURE 4.2: Headline from *The Forum* newspaper, April 24, 1997. SOURCE: Forum Communications Company.

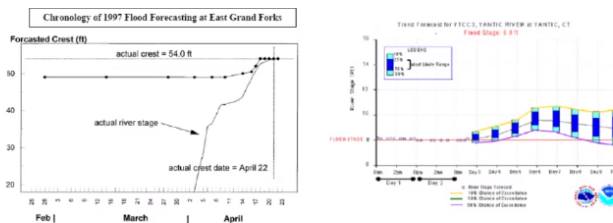


FIGURE 4.3: Left: Deterministic forecasts issued by NWS prior to the Red River flood of 1997. Right: Probabilistic river stage forecast from AHPS. SOURCE: NWS.

Figure 1: 1

There was a flood in the Red River in Grand Forks in 1997. They had prepared for the flood because they were getting forecast from Natural weather service about how high the river would rise. Then, they estimated how height the flood would go and gave the expected value. They forecasted the rivers only got up to “50 ft”. Therefore, they decided to make sandbags and they thought if the height of the sandbags were 50 ft, they would be safe. But, they did not. The forecast and the actual rising were shown in the graph. What happened was it bumped up the forecast very fast and there was not enough time left to increase the height of the sandbags. Therefore, there was a lot of blames that the Natural

weather service did a poor job. It was not actually fair, because they gave the best forecast best on the data. The problem was that the expected value is not the information that they needed. They needed information about how high the sandbag should be. They did not know the best estimation for sandbags, they needed to know what is the probability of getting higher than 50 ft, 51 ft, and so on. When they had that information, they could have themselves calibrated how much risk they were willing to take. It was turned out that the Natural weather service, actually had that kind of the information but nobody asks them to produce that. They found out that the 54 ft was within the reasonable margin of the error given the analytics they were using. The left side graph was the kind of information they have started to produce since that event. In this graph, the green line shows the expected value and the blue boxes show the amount of the error.

The promise: data-driven decision making

Data science hype: “Better information \rightarrow Better decisions!”

Claim: That’s wrong.

Because, more irrelevant information will more distract you away from the points of data you really need for that decision.

Data-driven decision-making also requires complementary analytics and products:

- *Prediction* : calibration of error, \rightarrow *probabilistic* forecasts
- *Optimization* : application-specific models and tools, attention to user objectives
- *Visualization/communication* : intelligence *actionable* in terms of user’s decision problem

In this example, if they built the sandbags higher than the estimation, it would be costly. If they built lower than that, it would be more disaster and the cost of the error is not symmetric here. The cost of the error is greater in one direction. You want the calibration of the error in the form of the probabilistic forecast. You do not want the whole distribution. You have to identify where in that distribution is relevant to the decision you want to make. Then, you do the optimization. In this case, they will solve the risk management problem.

Better information does not make a better decision. We need other tools to convert that better information into a better decision.

Realizing the promise of data-driven decision-making

On prediction, data science gallops along: machine learning, classification, prediction, inference...

Claim: The promise of data science to empower data-driven decision-making lags its potential, because of insufficient focus on *complementary* analytics:

- Rigorous quantification of *uncertainty* and *error* in outputs of predictive models
- Accurate formal representation of decision problems – including decision-maker’s objectives and tolerance for risk
- Appropriate use of optimization techniques
- Visualizations or other products customized to decision contexts

Goal: Identify good *actions*, conditional on predictions made with error

Now, we are going to talk about a particular application.

EX: The Cloud Hunter’s Problem

Meet the Cloud Hunter

The Cloud Hunter is an atmospheric scientist.

The Cloud Hunter wants to collect data from inside *liquid boundary layer clouds*.

That takes aircraft. Which are expensive.

The Cloud Hunter’s Decision Problem

The Cloud Hunter’s Problem concerns how to allocate a fixed budget of flight hours between dates over the course of a field season.

Fly/No-fly decisions must be made 1 day ahead, based on imperfect day-ahead forecasts of whether conditions are good or bad for collecting required data.

Model setup: Conditions for data collection over the course of the field season

D : length of the field season in days

$d = D, \dots, 1$: index of dates

X_d : quality of conditions for data collection:

- $X_d = 1$ if conditions on date d are good, 0 otherwise
- Each X_d a binary random state variable, i.e., a Bernoulli trial

A field season is a particular realization x_D, \dots, x_1 of the stochastic process X_D, \dots, X_1 .

Will assume the X_d are independent and identically distributed (i.i.d.).

- Assumption not actually required, but keeps things simpler and clearer.

Vector notation: $\mathbf{X} = \langle X_D, \dots, X_1 \rangle$ denotes the stochastic process; $\mathbf{x} = \langle x_D, \dots, x_1 \rangle$ denotes a particular realization.

Model setup: Decisions, resource constraints

$F \leq D$: number of flights in the Cloud Hunter's budget.

$f = F, \dots, 1$: index of flights remaining in budget

a_d : binary control variables ("actions")

- $a_d = 1$ iff they opt to fly on date d , 0 otherwise

$\mathbf{a} = \langle a_D, \dots, a_1 \rangle$: actions chosen on dates $d = D, \dots, 1$

Resource constraint: $\sum_d a_d \leq F$.

(Assume flights left over at the end of the season have no residual value.)

Payoffs and objectives

Payoffs: For a given sequence of choices \mathbf{a} and realizations \mathbf{x} , the realized amount of data collected U is given by

$$U = \mathbf{a} \cdot \mathbf{x} = \sum_d a_d x_d$$

.

Decision-maker's objective: Choose a fly/no-fly decision rule to maximize data collected in expectation, subject to the resource constraint on total allowable flights:

Choose \mathbf{a} to $\max_{\mathbf{a}} E[\mathbf{a} \cdot \mathbf{X}]$, subject to $\sum_d a_d \leq F$.

Important: This is a *substantive assumption* about the decision-maker's goals.

- Models a decision-maker with a high tolerance for *risk*.

Forecasts

Decision taken on basis of a day-ahead forecast.

Before taking each decision, decision-maker receives a forecast signal $s_d \in \mathbb{S}$.

Calibration: map this signal to a probability of good conditions:

$$p(s) = \Pr\{X_d = 1 | s_d = s\}$$

.

(Will assume stationarity.)

Distribution of forecast signals

More than one day ahead, don't know which forecast signals $s \in \mathbb{S}$ will be received.

But, *do* know the the likelihood of receiving different signals.

$\pi(s)$: probability that forecasting system will generate signal s .

$\pi(\cdot)$ defines a probability distribution over the set \mathbb{S} of possible forecast signals.

The task of the decision analyst

Given this set-up, the job of the decision analyst is to devise an *optimal decision rule* $a^* = a(d, f|p)$ that delivers a recommended action—fly or no-fly—as a function of

- d the number of days left in the field season,
- f the number of flights left in the budget, and
- $p(s)$ the forecast probability that a flight today would be successful.

A decision rule $a(\cdot)$ is deemed optimal if its consistent application maximizes in expectation the yield of successful flights realized from a given budget F .

Comment

This is a challenge of *intertemporal optimization*: each choice a_d alters the expected payoffs for subsequent decisions.

Need to use tools of optimization that account for this intertemporal structure.

Dynamic programming turns out to be the right tool for the job.

Optimization via dynamic programming

Intertemporal optimization via dynamic programming

Basic idea: break the decision problem into two pieces: (i) the next day's decision, and (ii) the rest of the field season after that.

Assume you've got the right answer (!). Given that this answer is optimal, derive the properties that solution must have.

Solve via backward induction.

The right bottom circle is the end of the season. At that time, we would be out of the season and out of the flight. Therefore, the expected flight for the rest of the season would be zero. Besides, for all of the bottom circles, the expected value again would be zero. On the circle corresponding, one day left and one flight left, the expected value would be 28. What would be the value of being at the circle corresponding to two days left and one flight left? We know there are

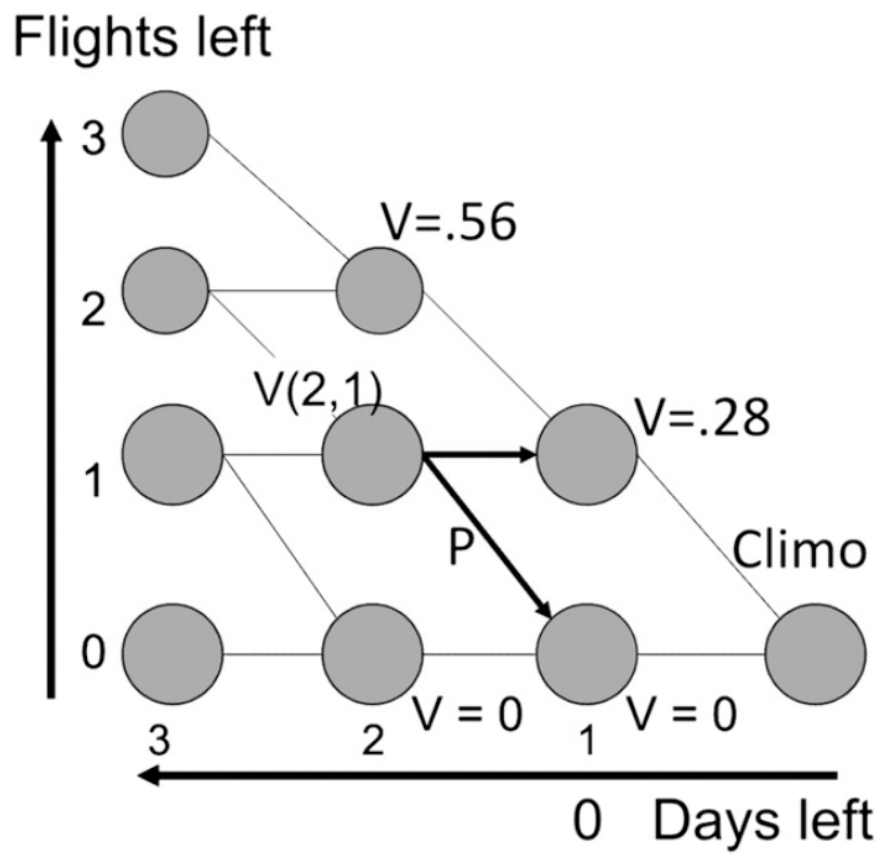


Figure 2: 2

two ways to reach the end of the season. We will choose to go for one day left and no flight if its likelihood is bigger than 0.28.

Suppose an optimal decision rule $a(d, f|s)$ has been found.

Let $V(d, f)$ denote the expected number of successes that would be realized from repeated application of this rule, starting from initial conditions $\langle d, f \rangle$.

By construction, $V(\cdot)$ equals the maximized of the decision maker's objective function, beginning from these initial conditions, under the substitution $a = a(d, f|s)$:

$$V(d, f) = E^\pi \left[\sum_{i=d, \dots, 1} a(i, f_i|s_i) \cdot X_i \right]$$

where the expectation is taken over the probability distribution of all possible sequences of forecast signals (determined by π), and where f_i denotes the number of flights remaining on date i when the optimal program is followed.

$V(d, f)$ called the *value function* for this problem.

Because $a(\cdot)$ is assumed to be optimal, $V(\cdot)$ must satisfy a particular recursive relationship:

$$V(d, f) = E^\pi [a_d^* \cdot X_d + V(d-1, f - a_d^*)]$$

where $a_d^* = a(d, f|s_d)$ is the optimal decision.

Once the forecast signal s_d is received, a^* will be chosen optimally:

- If $a^* = 1$, then $V(d, f|s_d) = E[X_d|s_d] + V(d-1, f-1)$.
- If $a^* = 0$, then $V(d, f|s_d) = V(d-1, f)$.

Since $V(\cdot)$ is by construction a maximum, we must have:

$$V(d, f|s_d) = \max\{E[X_d|s_d] + V(d-1, f-1), V(d-1, f)\}$$

$$= \max\{p(s_d) + V(d-1, f-1), V(d-1, f)\}$$

This formula implies the optimal decision rule: $a^* = 1$ ("fly") only if

$$p(s_d) \geq V(d-1, f) - V(d-1, f-1)$$

i.e., when the expected payoff of the current day's opportunity exceeds the loss in marginal cost associated with arriving tomorrow with one fewer flight in the bank.

Call $p(s_d)$ the *hurdle probability*.

Think of you look at the specific location of the sky. As you look up and go higher, the atmosphere pressure reduces. The scientists instead of measuring the heights in terms of the meters measured the height in terms of the millibars.

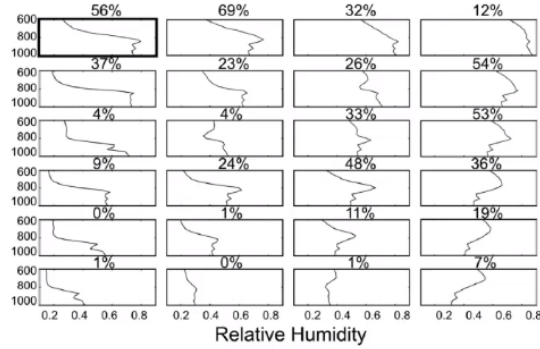


FIGURE 3: 6 X 4 SOM grid for relative humidity profiles

Figure 3: 3

They were investigating what is going on at each different pressure level as you go up to the atmosphere. The basic idea is, the atmospheric pressure of the sea level is 1000 millibars. What you are interested in when you are thinking about forecasting boundaries of clouds is about how relative humidity is changing with pressure. It is not a strictly linear relationship between them. What you do is you would take lots of the snapshots of the atmosphere, which differ according to relative humidity profile. As you go up to the atmosphere, how much the relative humidity is. You will take them and put them in an unsupervised machine learning algorithm (clustering). The formula for distance is the Euclidean distance between these graphs. The algorithm clusters them into 24 groups. The number above each box shows the information about the existence of the boundary clouds. In our historic data set, every time the relative humidity profile end into the first box, 56 percent of them are associated with the presence of the boundary clouds. These things do create that differentiation. We create the differentiation and then each of these classes tells us something informative about the phenomenon role which we are interested in.

These forecasts are not perfect.

We got the American prediction model and it will predict for us, relative humidity at different pressure levels. It will give us the prediction of what tomorrow bring and we can put it in one of the boxes. Now we want to know how good the American prediction system is even if we predict the right box. We got the result which is shown in the above picture; the result is not good. If it anticipates that you land in the first box of the relative humidity profile, Only, 13.3% of times you actually land into the right box and the rest of the time you will land into

13.3%	20.6%	4.4%	3.7%
1.5%	8.1%	1.5%	4.4%
0%	3.7%	4.4%	11.8%
2.9%	3.7%	7.4%	6.6%
0%	0%	0.7%	1.5%
0%	0%	0%	0%

FIGURE 4: Conditional probability distribution of SOM state realizations following a forecast of SOM state 1

Figure 4: 4

the rest of the box. If you are going to link the numerical prediction model to the relative humidity profiles to predict the boundary clouds. Then, we have to deal with the fact that the above prediction system has lots of errors in it.

So, we recalibrate it to give us the real probability distribution. Rather than taking the forecasting literally true, we take it as a signal and recalibrate it.

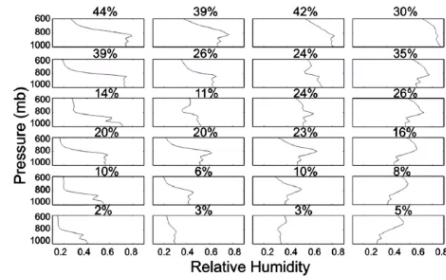


FIGURE 5: Estimated probabilities of good conditions for data collection, as a function of day-ahead SOM forecast

Figure 5: 5

The solid line shows the sequence of the decisions. In the first days, you are more conservative, and there are more days left. Then, you see there are a lot of flights left, and you start dropping the probabilities and having the flight on risky days. In the end, you have a budget left, and you start burning it.

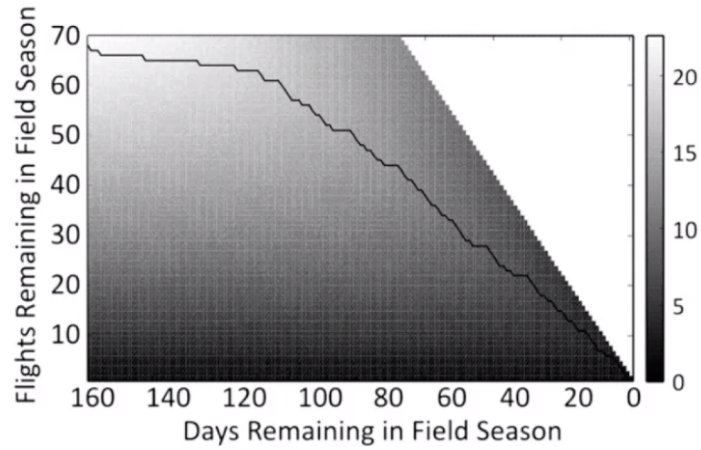


FIGURE 6: Computed values for the value function $V(d, f)$

Figure 6: 6

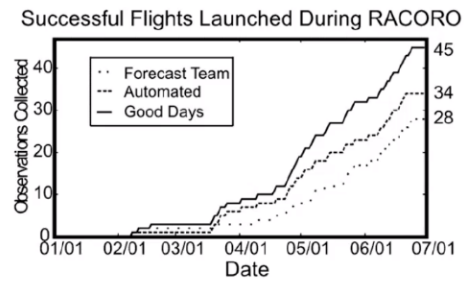


FIGURE 7: Results: the algorithm's simulated performance during 2009 field season, compared with realized performance of heuristic decision procedure.

Figure 7: 7

	Heuristic procedure	Automated algorithm
Flights launched	56	66
Successes	28	34
Type I errors	28	32
Type II errors	17	11

FIGURE 8: Summary of outcomes

Successes are flights launched on days with good conditions.

Type I errors are decisions to fly only to find no clouds.

Type II errors are decisions to stand down only to find that the desired conditions existed.

Figure 8: 8