Fundamentals of Data Science

Fall 2017
Daniel Egger

Final Team Projects

(Nov 16, 2017 draft)

You will be given three .csv files, ordered by time stamp, as a training set.

Each file contains labeled packet input and output data from the 12 machines in the Cisco test network topology (8 leaves and 4 spines). The files record:
(1) Failure in a known leaf,
(2) Failure in a known spine, and
(3) No failure event.
These failures are of the same type as you observed in the earlier assignment. Examining the *full file* based on your knowledge from the earlier assignment, it should be obvious which file is which.

You assignment is to model, from partial information – a limited number of rows of data from the top of the file - the probability that the file contains a failure event, and if so, the probability that a given machine is the one failing.

The telemetry data can be thought of as arriving in real time. Your model should be designed to identify potential failure, and locate which machine may be failing, as quickly as possible, ideally before a failure has reached the point of disrupting or delaying network traffic.

You will start with data at the top rows of the file and add a fixed number of seconds of data.
For example, you may be asked to make a first estimate of probabilities at 45 seconds, a second at 60 seconds, a third at 75 seconds, and so on. What matters is your score on the two test files – *new data*.

Based on your experience with assignment 1, after observing data for a full file - approximately 250 seconds – your model should have almost no doubt about whether the file is the null file or a failure file, and if a failure, which machine is at fault.

Probabilistic scoring will occur earlier, when substantial doubt about the nature of the file is still possible. The earlier you detect an event, the better – so long as you don't make a mistake.

The "ignorance prior" for your earliest forecast will be as follows:

Null File	Failure in Leaf									Failure in Spine		
	1	2	3	4	5	6	7	8	1	2	3	4
.50	1/24	1/24	1/24	1/24	1/24	1/24	1/24	1/24	1/24	1/24	1/24	1/24

Later, you will be given two files as a test set to evaluate your model. One is a null file and one contains a failure. Your forecast at each point must obey the following rules:
Probabilities in each forecast must always sum to 1.
Every one of the outcomes must have a minimum probability assignment of at least 1%.

This means that if you are effectively certain, for example, that leaf 7 is failing, your probabilities would be as follows:

Null File	Failure in Leaf									Failure in Spine		
	1	2	3	4	5	6	7	8	1	2	3	4
.01	.01	.01	.01	.01	.01	.01	.88	.01	.01	.01	.01	.01

Scoring Forecasts for Model Evaluation

One outcome is ultimately correct. Divide your forecast of the true outcome by the ignorance prior probability for the true outcome. Take the log to the base 2 of the ratio. The result is the information gain (or loss!) of that estimate.

You should optimize your model on the three training files, for information gain as calculated above, with the assumption that forecast requests will come early in the file time-series.

Note that the penalty for assigning low probability to the true outcome is severe. This is the reason for the minimum 1% probability rule – if you assign 0% probability to the true outcome, your information gain from that forecast alone will be negative infinity!

Project evaluation will consist of three parts: Model Scoring on Test files, Written Model evaluation, and Class Presentation.

Model Scoring

The first half of the two test files will be divided into approximately 10 sections. The first three sections will be provided on Monday, December 4, the next three on Tuesday, December 5, and the last three on Wednesday, December 6 – along with the remainder of the files. You will provide your probability estimates for the sections – uploaded to Sakai Dropbox – by midnight on the same day they are made available. The reason for this is to prevent being influenced by knowledge of later rows of data in estimated probabilities for the earlier sections.

Written Model Submission

Your model needs to be a mathematical procedure for converting the data in rows 1-n of a file into a probabilistic forecast that meets the requirements above.

for illustration purposes, an example of a very simple, but reasonable, model would be as follows:

(1) Calculate the long-term (top row to present) and short term (most recent five seconds) average values of useable data, ignoring null rows, separately for each of the 12 output columns.

(2) If at the final row of a section, one (or more) columns has a last five second average less than 90% of its long-term rolling average, identify the column with the largest decrease, and add 21% probability to the machine that corresponds to that column, subtracting 1% probability from each of the other 11 machines, and 10% probability from the "null" forecast.

(3) If no short-term average is less than 90% of the long-term average, subtract 1% probability from each of the 12 machines and add 12% probability to the null forecast.

(4) If any machine, or the null forecast, reaches 88% probability, "freeze" the forecast and make no further changes.

As you can see, this model can be calculated and evaluated manually, but it could easily be implemented in software. The goal for this assignment is to develop a model that is complete and unambiguous enough that it could be implemented as a computer program, but simple and transparent enough that it can be described completely in a written document of no more than 2-3 pages. This document will be due December 7 at the start of class.

Class Presentation

Class presentations on December 7 will be *strictly* limited to five minutes. Practice to make absolutely sure you will not go over time. You should divide the time between at least two team members.

In your class presentation, address:

Who is in your team? For what part of the data scrubbing, exploratory work, model development and optimization, testing against the test set, and creation of presentation materials did each team member take lead responsibility?

What are the key features of your model? In simple words - How do you detect leaf failures? Spine failures? Explain how you developed your model. Are any useful features of the model surprising or non-intuitive? Do they reveal anything interesting about the data itself?

What was your model's performance, in terms of information gain, on the two test files at each section? Compare your performance on the training and the test files. If performance was significantly worse on the test files, was this due to over-fitting? Did you change and improve your model between the first batch of sections (Monday) and the last (Wednesday)? If so, how?

Knowing what you know now, are there any aspects of the model you would design differently?

You should have no more than 8 Powerpoint slides to accompany your presentation. The first slide should give full names of your team members. The last slide should thank Cisco for providing open source data for this project. Include at least two graphics. Keep text on each slide to a minimum – do *not* read your slides!

END