Washio Science Challenge

Damon Alexander

January 13, 2015

1 Introduction

The Washio Science challenge consists of a programming exercise, meant to test the candidate's ability when dealing with a situation similar to those faced at Washio. The project should take somewhere between 2 and 6 hours. We consider the problem faced by a hypothetical advice-by-phone service called Callio that considers themselves like Washio, but for phone calls. They have customers who place phone calls seeking advice, and operators who answer the phones. We schedule operators so that a fixed supply of operators are always working. See Figure 1 for the call volume in a typical day at Callio.

The company is struggling because there is a huge surge of calls at 6 PM, with lulls at 4 PM and 8 PM, so scheduling operators to handle the peak means we have too many scheduled during the lulls. The proposed solution is to implement splurge pricing, which seeks to modulate the prices for callers so that demand is smooth.

2 The model

There are many ways to model this problem. We use the following one, which has the advantage that it does not require estimating the demand curve.

At time t, we let $x(t)$ denote the number of ongoing calls. The objective is to ensure $x(t)$ never reaches beyond a given safety threshold, because we never want to risk having to turn away callers. This is expressed with the desire to make it so that $a \leq x \leq b$. Then we use the model

$$x(t) \approx x(t - t_0) + (t - t_0)\frac{dx}{dt} + \frac{(t - t_0)^2}{2}\frac{d^2x}{dt^2}.$$

Using this to forecast x, we can solve for the future time t_1 where $x \notin (a, b)$, that is, where x escapes our target threshold for number of operators. You may assume that t_1 exists, that is, the number of calls always does stay in the desired range.

In practice, $x(t)$ only changes at discrete time points when calls occur or end, which makes the calculus messy. We provide functions to approximate the derivatives, get_first_derivative and get_second_derivative, provided in the file "get_derivatives.py", which take the most recent two and three points of $(t, x(t))$.

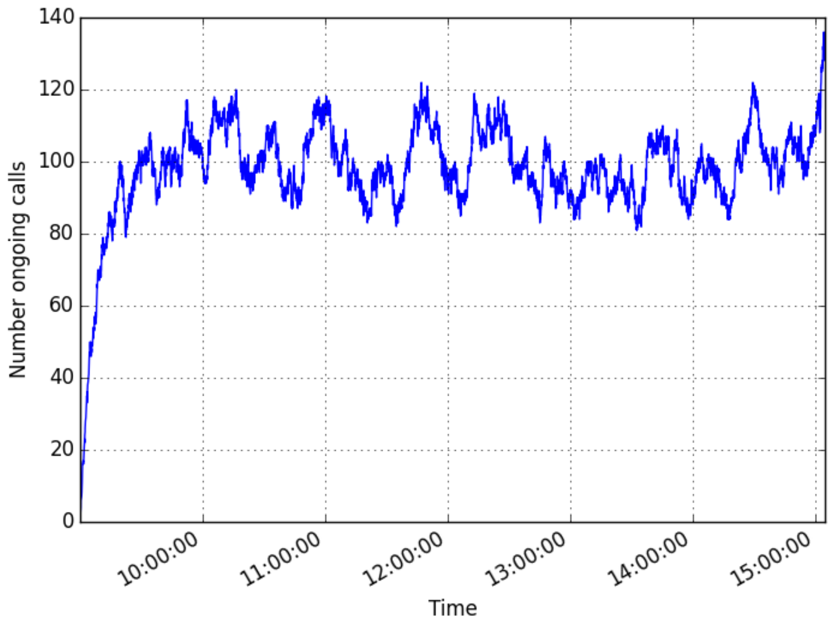

Figure 1: A sample graph of calls against time

At this point, given an old price p, we wish to modify the price by the percentage $\frac{\alpha}{t_1 - t_0}$, and so our new price is

$$p \cdot \left(1 + \frac{\alpha}{t_1 - t_0}\right).$$

3 Provided materials

Provided are two files: "calls.json" and "system_state.json"". The calls file contains a json array of calls, which have a datetime, and if they are completed, they have a duration, expressed in minutes. The system_state file contains the following fields:

- current_datetime
- α, the percent scaling coefficient
- "call_lower_bound" and "call_upper_bound", the desired lower and upper bounds on ongoing calls (these correspond to a and b)
- "last_price", the last known price (in dollars per hour)

4 Objective

Write a software solution that reads in the data, and return the current price in the file “file price.txt”. Also, answer the following questions, in under a page:

- What is the most direct thing that can go wrong with this model?
- How would you improve or extend the software solution built so far?
- What features could you build in to the software solution to be confident putting it into production?