NEURAL UNCENSORING IN HOTEL REVENUE MANAGEMENT

Introduction

Small- to medium-scale resorts strive to differentiate themselves from the rest of the hospitality industry by offering their customers unique experiences beyond a typical set of standard services. Easily recognized examples include ski lodges, golf clubs and scuba-diving resorts. In their reliance on an experience-seeking customer base, they may benefit from the integrated property and revenue management strategies leveraged by major casino empires. This project seeks to study a part of the system that comprehensively serves the unique needs of resorts in handling experience-seeking travellers. These needs range from business basics to advanced marketing, pricing, and revenue optimization.

Specification

A typical approach to building revenue management systems consists of three stages: Descriptive Analytics, Predictive Analytics and Predictive Analytics. This project proposal focuses on the part of the Predictive Analytics stage, more precisely on estimating demand distribution. Although we usually have some information about client arrivals, it is censored. Once capacity is full, the dataset does not track new arrivals. However, they can significantly leverage resort profits via selecting customers of bigger auxiliary spendings and overbooking.

Dataset

This project has started as consulting work with a business partner. Although the dataset they provided is under NDA, we have some other similar datasets publicly available on the Internet. These datasets have been formatted to the initial format. Essentially, for the purpose of uncensoring and estimating demand distribution, the following fields are important: booking date, arrival date, departure date, and property type. We also know room capacities and have data for three years for Portugal Resort and Portugal City datasets.

Approaches

Since NNs are somewhat hidden models, they are usually not very much liked by the business. Therefore this idea has been postponed for a while. However, from the research point of view, there are some analogies with image generation, performed by GANs and Normalizing Flows.

Idea 0. (Baseline without NN). Uncensoring with Generalized Linear Models.

Reference: Kevin P.Murphy. Machine Learning: a Probabilistic Perspective.
http://noiselab.ucsd.edu/ECE228/Murphy_Machine_Learning.pdf
Chapter 9. In R, the most relevant libraries are glm2 and glmnet.

Idea 1 (doable, NN-related baseline). It leverages an intuition of a neural network as a universal approximator.

A. For a two-stage generation graphical model, it can be reasonable to estimate the arrival rate parametrically. In the current implementation we use, the arrival rate is modelled as a linear function of some features. This rate is learnt by computing total log-likelihood now. However, a neural network can be used instead of the function, and stochastic learning in batches can be applied for general information log-likelihood optimization.

B. There is an Expectation-Maximization approach already implemented, which is yet another faster and more theoretically clear way to optimize log-likelihood. The parameters are still modelled as a linear function and are optimized on the full dataset. Again, a neural network can be used instead of the linear function, and stochastic learning in batches can be applied for perfect information log-likelihood optimization.

Pros: I understand how to implement this approach and have a baseline for it.
Cons: It is not very fancy in terms of true deep learning from raw features to sophisticated output. Something like a staged speech recognition model compared to the sequence-to-sequence approach.

Idea 2 (risky). In some sense, demand is an image with patterns. The first axis of the image is the booking date, and the second axis is the week of arrival. Other axes can be room category and stay duration. It somewhat looks like a convolution. This intuition has several problems, however. If to select a wide concept of the image, the amount of non-zero pixels are small. To make it precisely an image, one wants its cells to be not 0-1 encoded. Instead, it is much more useful to have them somewhere inside 0-255. Second, the amount of data can be small. This sometimes is managed by pre-training on another dataset, but we do not have that many similar datasets. On the other side, a scale selection can help. The image does not have to be 2-dimensional for NNs to work. Sometimes 1-dimensional patterns are good enough. Third, some of the “pixels” of the “image” are censored.

Now besides that problems, it still looks like fitting the distribution. Normalizing Flows and GANs are pretty proficient with that. It can be somewhat like fitting a conditional distribution. Condition tells which pixels are censored, as well as some input information about the slices. I feel it can be easier to use Normalizing Flows, and training GANs is tricky.

Set of deep tricks that can help (essentially, augmentation):

- The artificial introduction of the censoring threshold can potentially increase the dataset.
- By learning a simpler model (GLM or NN-baseline) first, one can create a basis for pre-training.

Relevant Tutorial for Normalizing Flows:

- https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial11/NF_image_modeling.html#
- We can modify it by supplying coupling layers with additional features describing booking features and the current censored structure. I think dequantization can be done analogously.

Note. Friends suggested a relevant paper for GANs. I'll upload it once I get access to their response.

Note. It can be reasonable to google similar demand datasets. The case we have is not very standard, i.e. we gave delay between booking and actual service. Therefore, the majority of Kaggle datasets are not relevant. However, it can be useful if we find some relevant sets with more data (e.g. for at least 10 years/periods, but not 3 only). Deep Learning is supposed to work better when the amount of data is larger.

Pros: I do not know many results in this direction, and it studies new and fancy stuff. Moreover, any distribution reconstruction with small data is an interesting result.
Cons: It may be hard to make this approach work. Myself, I do not have much experience with it, and the model is somewhat blurred.

Idea 3 (risky). The idea is to apply a Bayesian Network for the same setting as in Idea 2. It can be tricky since these guys are typically about having weights as normal distribution generators. Without explicit transformation, it can become very much like recovering discrete distribution. I have heard that this is problematic, however maybe not that much. Again, very risky since I have no intuition if it will at all work. The Pros and Cons are the same as in the previous idea.

Metrics

Metrics are somewhat controversial here. Sometimes one uses AIC, BIC, log-likelihood, or KL-divergence. Since the distribution is censored, the recovered result should look different anyway. Usually, it is about "how it looks for the customer" and how much profit the final system brings. I think one can just compare log-likelihoods and visualize results with the time we have. The visualization is relatively easy.

One another idea is to test the result somehow. Of course, uncensored and censored samples are different. However, we can censor our output again, and then the reasonable question is if the censored output and initial input are from the same distribution. I think that the likelihood ratio may have some information, but it likely tests goodness of fit for a particular model but not compares models.