This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 777533.

PROviding Computing solutions for ExaScale ChallengeS

D9.4	**Whitepaper to explain general algorithmic concepts of the airline ancillaries use case**		
Project :	PROCESS H2020 – 777533	**Start / Duration:**	01 November 2017 36 Months
Dissemination[1]:	PU	**Nature[2]:**	**R**
Due Date:	**Month 27**	**Work Package:**	**WP 9**
Filename[3]	PROCESS_D9.4_WhitepaperGeneralAlgorithmicConcepts_v1.0.docx		

ABSTRACT

Airline revenue consists not only of ticket sales but also of ancillary services. There has been a lot of research on maximizing the ticket revenue by choosing optimal prices. This paper describes an optimziation method to increase an airline's ancillary revenue by dynamically computing optimal prices for each customer who buys a ticket. In particular, the focus of this paper will be the first bag ancillary, i.e., the first bag each customer books in addition to their ticket. Here it is important to find out how the decision of the customer to book an ancillary depends on the ancillary's price. A customer's willingness to pay for a first bag is modelled as random variable of which the random distribution is unknown. We use machine learning algorithms to estimate this distribution based on given data.

1 PU = Public; CO = Confidential, only for members of the Consortium (including the EC services).

2 R = Report; R+O = Report plus Other. Note: all "O" deliverables must be accompanied by a deliverable report.

3 eg DX.Y_name to the deliverable_v0xx. v1 corresponds to the final release submitted to the EC.

Deliverable Contributors:	Name	Organization	Role / Title
Deliverable Leader[4]	Pancake-Steeg, Jörg	LSY	Coordinator
Contributing Authors[5]	Reichardt, Janek	LSY	Writer
	Pancake-Steeg, Jörg	LSY	Writer
Reviewer(s)[6]	Minich, Tanja	LSY	Reviewer
	Grundner-Culemann, Sophia	LMU	Reviewer
Final review and approval	Höb, Maximilian	LMU	Coordinator

Document History

Release	Date	Reasons for Change	Status[7]	Distribution
0.1	2019-11-12	Initial version	Draft	LSY
0.2	2020-01-08	Final Draft	In Review	Consortium
1.0	2020-01-31	Final version	Released	Public

[4] Person from the lead beneficiary that is responsible for the deliverable.

[5] Person(s) from contributing partners for the deliverable.

[6] Typically person(s) with appropriate expertise to assess the deliverable quality.

7 Status = “Draft”; “In Review”; “Released”.

Executive Summary

This deliverable gives insights to the algorithmic of UC#4. As a whitepaper, this document aims to inform the reader about the mathematical background of the airline ancillary use case.

The document is structured as follows:

- Section 1 serves as a mathematical proplem description and as introduction to later sections.
- Section 2 shows how we model a customer's willingness to pay for an ancillary and how we can estimate it using machine learning procedures.
- Section 3 proposes algorithms for optimizing the ancillary revenue.
- Section 4 explains how we implement the machine learning models and the optimization seen in previous sections.
- Section 5 shows the results of our computational experiments.
- Section 6 wraps up the whitepaper and gives an outlook on potential further research.

Contents

1 Introduction

Airline revenue consists mainly of ticket prices. However, ancillaries still play a big role. Therefore finding an optimal price for ancillaries comes with a significant increase in overall revenue. Our goal is to calculate optimal service prices for each ticket booking. In the case where the customer did not book the considered service at the time of booking, the service price may change after the ticket booking is made. More specifically we work on the following setting. Let X be a set of bookings. For each booking $x \in X$ the variable $t_{x,0} \in \mathbb{N}$ denotes the time of ticket booking of x on a discretized timeline and $d_x \in \mathbb{N}$ denotes the date of departure on that same timeline. We further model the willingness to pay for the considered ancillary service as a real valued random variable $W_{x,t}$ with unspecified distribution function for each booking x and time $t \in [t_{x,0}, d_x]$. In this setting the probability of a customer booking the ancilllary at a given service price $s_t \in \mathbb{R}^+$ at time $t \in [t_{x,0}, d_x]$ is equal to the probability $P(W_{x,t} > s_t)$ that the willingness to pay is larger than the current service price. Note that using this model, the ancillary service can only booked once. Now we can define the expected service revenue for one booking x as a function of a price vector

$$
\begin{aligned}
Rev_x &: \mathbb{R}^{+d_x} \to \mathbb{R}^+ \\
Rev_x(s) &= \sum_{i=t_{x,0}}^{d_x} \left[s_i \cdot P(W_{x,i} \leq s_i) \prod_{j=t_{x,0}}^{i-1} (1 - P(W_{x,j} \leq s_j)) \right]
\end{aligned}
$$

This function still depends on the probability distributions of the willingness to pay. In the coming chapter we propose a method to estimate the distributions based on real data.

2 Estimating the willingness to pay

Consider the following setting: Given is a set X of bookings as well as the information if, and if yes when an ancillary service was booked. Note that this information can be stored in a one-hot-vector. If an ancillary was booked on the i-th day of the booking horizon we denote this with the vector Y' containing $i-1$ zeros followed by a one. We call the resulting data set X'. This notation is advantageous when using the data to train machine learning models. Each record corresponds to one day on which the customer could have bought the ancillary. Out of these days only on the last, that is i-th day, did the customer actually buy the ancillary. In this context it makes sense to not include any more zeros afterwards, as the customer does not get the chance to book an ancillary a second time.

Based on this information we would like to find $P(W_{x,t} > s)$ that fits the data best out of a given class of distributions. This is a regression problem where the dependent variable y' takes one of two values, zero or one, in the observations. By choosing a regression function $f : (s,t,x) \mapsto f(s,t,x) \in [0,1]$ through ones and zeros we estimate the purchase probability. Here $s \in \mathbb{R}^+$ is the ancillary price, $t \in [t_{x,0}, d_x]$ is the time for which the probability is requested, the booking x contains further characteristics, such as the customer's age or the route of the flight. However, another problem lies in the data presented. Since in the airline business ancillary prices are fixed in almost all cases, we need to think about how we deal with data that only contains a single value for ancillary prices. Consider the additional assumption that the ancillary price is s' for each data record in X. In this case a regression through the data would only encounter a single price, and therefore the impact of the price on the willingness to pay can not be estimated. Therefore regression only helps us to explain the dependency of the probability on the time t and the characteristics of x for a fixed price s', in other words we only achieve a regression function $f' : (t,x) \mapsto f(s',t,x)$. As there is no information on the relation between the price and the willingness to pay, we add an additional assumption on this dependency. At this point we also require f to be monotonically decreasing in s.

One type of functions that satisfies the requirements is the class of logistic functions $l(s) = \dfrac{a}{1+e^{b\cdot(s-s_0)}}$. As long as all parameters are nonnegative, a is the supremum of the function values $l(s)$, the slope of the curve depends on b while s_0 is the s-coordinate where the curvature changes. Note that these parameters $a, b, s_0 \in \mathbb{R}^+$ are uniquely determined by three points on the curve.

booking id	...	customer age	date of ticket booking	departure date	date of ancillary booking
123	...	24	23.11.2019	27.11.2019	26.11.2019
124	...	48	24.11.2019	27.11.2019	
...	...	...	...	...	...

booking id	customer age	date of ticket booking	departure date	date	booked ancillary?
123	24	23.11.2019	27.11.2019	23.11.2019	0
123	24	23.11.2019	27.11.2019	24.11.2019	0
123	24	23.11.2019	27.11.2019	25.11.2019	0
123	24	23.11.2019	27.11.2019	26.11.2019	1
124	48	24.11.2019	27.11.2019	24.11.2019	0
124	48	24.11.2019	27.11.2019	25.11.2019	0
124	48	24.11.2019	27.11.2019	26.11.2019	0
124	48	24.11.2019	27.11.2019	27.11.2019	0
	...	...	...	...	...

Figure 2.1: Example of the described data transformation.

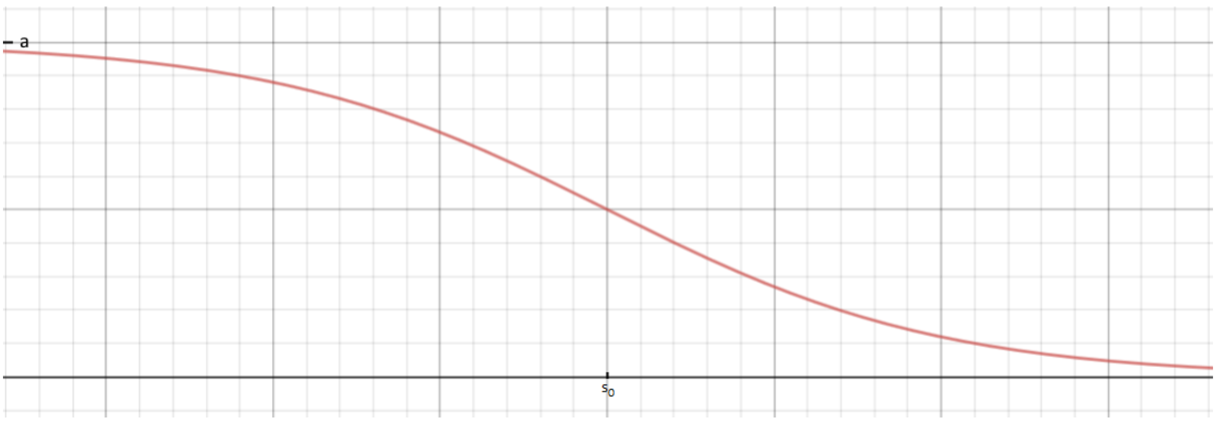

Figure 2.2: Example of a logistic function with parameters s_0 and a drawn.

The derivative of l with respect to s can be computed as $\frac{\partial l}{\partial s}(s) = ab \cdot \frac{e^{b(s-s_0)}}{(1+e^{b(s-s_0)})^2}$ by substituting $z = e^{b(s-s_0)}$. It can be used for optimization later.

3 Optimization of the revenue

As mentioned in the introduction we consider the revenue formula

$$Rev_x(s) = \sum_{i=t_{x,0}}^{d_x} \left[s_i \cdot P(W_{x,i} > s_i) \prod_{j=t_{x,0}}^{i-1} (1 - P(W_{x,j} > s_j)) \right]$$

Inserting the formula for the probability under the assumption that the willingness to pay follows a logistic distribution, we get

$$Rev_x(s) = \sum_{i=t_{x,0}}^{d_x} \left[s_i \cdot \frac{a_i}{1 + e^{b_i \cdot (s_i - s_{0_i})}} \prod_{j=t_{x,0}}^{i-1} (1 - \frac{a_j}{1 + e^{b_j \cdot (s_j - s_{0_j})}}) \right]$$

From this follows a nonlinear optimization problem.

$$\begin{aligned} \max \quad & Rev_x(s) \\ \text{s.t.} \quad & s_l < s_i < s_u, \quad i = t_{x,0}, \dots, d_x \end{aligned}$$

for some lower and upper bounds on the price s_l and s_u. There are different ways to solve this problem which are explained in the next sections.

3.1 Derivative-free optimization with the BOBYQA optimizer

The BOBYQA algorithm is an optimization algorithm proposed by Powell in [Pow09] that uses quadratic approximation of the objective function Rev_x without a need to compute the derivative function. It considers optimization problems with simple box constraints as opposed to the NEWUOA algorithm ([Pow04]) from which it is developed. The algorithm approximates Rev_x by a quadratic function Q around a starting point in each step, computes the optimum of Q with respect to the box constraints and repeats from this optimum. A quadratic function in n dimensions has $\frac{1}{2}(n+1)(n+2)$ degrees of freedom, however BOBYQA only uses $m \in [n+2, \frac{1}{2}(n+1)(n+2)]$ interpolation points which still leaves some freedom for the quadratic function. The quadratic objective Q is instead chosen by both the m interpolation points and by minimizing the Frobenius matrix norm of the change to the second derivative of Q. This is all possible without computing any derivatives of the objective function explicitly.

3.2 Optimization using a conjugate gradient method

Another option to find the minimum of a nonlinear function are nonlinear conjugate gradient methods. Those methods implicitly approximate the objective function by a quadratic function using the functions gradient. The gradient ∇Rev_x can be computed using the derivative of the logistic functions seen before in chapter 2. As opposed to other gradient optimization methods like steepest gradient descent, conjugate gradient method make use of so called conjugate directions which reduces the number of iteration steps. However these methods don't allow bounds on the variables by themselves. Therefore we make due with an unbounded version of the optimization problem: $\max_{s \in \mathbb{R}^n} Rev_x(s)$. In most cases this computation is much faster than using BOBYQA. And depending on the concrete data the unbounded optimum may still fulfil the box constraints. Therefore it is often worthwile to first compute the unbounded solution.

4 Implementation

4.1 Implementing machine learning algorithms

We use the machine learning algorithms offered by the open source project H2O.ai [h2o]. H2O implements many machine learning algorithms, in particular distributed random forests (further denoted as DRF) and deep neural networks (DNN). The software can be accessed by REST-api. There are also interfaces for python, R, as well as Spark available. We use H2O from R to generate both a DRF as well as a DNN model.

4.2 Implementing the optimization

The machine learning models give us an estimation of the purchase probability for the given booking x, time $t \in [t_{x,0}, d_x]$ and the observed service price s' present in the data. Using the notation of chapter 2, the machine learning model returns $P(W_{x,t} > s')$. As we assume f to be a logistic function in s, two additional points are needed to compute the necessary function parameters. Consider the following list of assumptions:

- f is a logistic function in s: $f(s) = l(s) = \dfrac{a}{1 + e^{b \cdot (s - s_0)}}$
- $f(s') = P(W_{x,t} > s')$
- no matter the price, the purchase probability is never higher than 95% for any booking and time. Therefore $a = 0.95$
- tripling the observed price reduces the probability to a fifth of the estimated probability, i.e. $f(3 \cdot s') = \dfrac{f(s')}{5}$

Note that the last two assumptions are chosen more or less arbitrarily. This is because the data set is expected to contain only one distinct price. If instead there are two, three, or even more distinct ancillary prices, the last assumptions should be replaced by assumptions of the form $f(s^*) = P(W_{x,t} > s^*)$, where s^* is the additional price point and $P(W_{x,t} > s^*)$ the corresponding probability. The original assumptions determine the parameters of f uniquely:

- $a = 0.95$
- $s_0 = \dfrac{s' \cdot \frac{l_1}{l_2} - 3 \cdot s'}{\frac{l_1}{l_2 - 1}}$
- $b = \dfrac{l_2}{s' - s_0}$

with $l_1 = \log(\dfrac{a}{P(W_{x,t} > s')} - 1)$ and $l_2 = \log(\dfrac{a \cdot 5}{P(W_{x,t} > s')} - 1)$. Now that the objective function is defined we use the classes implemented in the `apache.commons.math3` packages for java to solve the optimization problems. In particular we make use of the `MultiStartMultivariateOptimizer` class, which allows us to start the optimization multiple times with different starting points. This decreases the risk of finding a local minimum only instead of a global minimum. This optimizer uses a random generator to generate new starting points s in each iteration. The first starting point is chosen by us, the package user. In the next iterations starting points are chosen by adding random vectors to previous starting points.

5 Experimental results

As mentioned before, there have been no experiments with real data done prior to writing this. Instead the described method has been tested with artificially created data.

The generated data consists of records containing information on the bookings ticket price, the time of ticket booking, the customers age, the travel type (i.e. if the customer travels for business or for leisure) as well as the channel of the booking and the point of sale, that is the country in which the booking was done. In addition the data contains information on the first bag ancillary, that is the first piece of luggage added to a booking. For each booking we know whether a first bag was booked and if yes when and for which price.

The testing is then done as follows. After the machine learning models are trained, a simulation starts. This simulation process generates a number of ticket bookings. For each of them revenue optimal prices are computed and compared to the fixed price present in the data. Such a simulation on 10.000 simulated ticket bookings results in an increase in expected service revenue of $63,9\%$. Note that such a drastic increase is not realistic and results from the use of artificial data

During the project there have been tests on the running time. Note that these times also strongly depend on parameters of the models. During training both models achieve similar error ratings with an average root mean square error of 0.0787 for the distributed random forest and 0.0780 for the deep neural network.

Additional tests regarding the performance of the optimization are planned.

Table 5.1: Example of training durations for different models and different amounts of data in hours

count of records	distributed random forest	deep neural network model
10,000,000	0:00:05	0:01:47
20,000,000	0:00:07	0:06:22
50,000,000	0:00:08	0:14:38
100,000,000	0:00:08	0:31:32
200,000,000	0:00:09	1:04:48
250,000,000	0:00:12	1:13:48

6 Conclusion

Ancillaries are an important source of revenue in the airline business. Our research presents a feasible method to find revenue-optimal prices for those ancillaries that can be booked once per booking. Depending on the data available, additional assumptions have to be made on the relation between the price and the customers booking decision. In this case, the result strongly depends on these assumptions. In concrete applications it is therefore advised to further research this dependency.

Listings

Glossary

X the set of bookings.

$t_{x,0}$ the time of ticket booking for a booking $x \in X$.

d_x the date of departure for a booking $x \in X$.

$W_{x,t}$ the customer's willingness to pay for the ancillary corresponding to booking $x \in X$ at time $t \in [t_{x,0}, d_x]$.

s_t the service price of an ancillary at time $t \in [t_{x,0}, d_x]$.

$P(W_{x,t} > s_t)$ The probability that a customer's willingness to pay is bigger than the service price s_t at time $t \in [t_{x,0}, d_x]$.

Rev_x real-valued function describing the expected ancillary revenue for a given booking x as dependent of a service price vector.

List of figures

Bibliography

[h2o] h2o.ai. *h2o.ai homepage*. URL: https://www.h2o.ai/ (visited on 08/01/2020).

[Pow04] M.J.D. Powell. *The NEWUOA software for unconstraind optimization without derivatives*. website. http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2004_08.pdf; last accessed 29.11.2019. 2004.

[Pow09] M.J.D. Powell. *The BOBYQA algorithm for bound constrainedoptimization without derivatives*. website. http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf; last accessed 29.11.2019. 2009.