IKEA CASE

Mike Riess*

January 8, 2018

* Aarhus, Denmark. E-mail: Mikeriess@me.com

1 Brief from IKEA

This case is adapted from an answer by iKnow solutions to a tender from IKEA. In the tender, IKEA wants to optimize its waiting time in key areas of the store, which will eventually increase sales per working hour. It is the main goal for IKEA to reduce waiting time in key areas by using predictive analytics on internal, as well as external data to optimize its staff-planning.

The key areas/departments in which IKEA wants to improve its staff-planning is:

- Cash-out
- Return
- Kitchen
- Closet area (PAX)

The effectiveness of the effort will be measured by two KPI's:

- Reduction in *waiting time*
- Increase in *sales per worked hour*

The proposed solution is expected to be a dynamic staff planning model that utilizes knowledge about future customer flows in order to minimize waiting time, and increase sales per hour.

2 Approach to solving the case

In the following, the proposed solution and the approach to solving the problem will be explained in further detail. The solution will be based entirely simulated data, which is again based on assumptions, since no internal data has been provided by IKEA. This paper will strive to answer the following questions, by focusing only on the cash-out area:

1. Details of the predictive model/solution (analytics method, data needed)
2. How to predict the flow of incoming customers
3. How to identify bottle-neck areas and loops in the store
4. What infrastructure is needed for this solution to work? (i.e. Big Data, Hadoop)
5. How to operationalize the insights to reach the KPI and goals

Regarding questions 1 and 2, the prediction model will be described in further detail in section 4.3. Question 4 will be discussed in section 4.5 and question 5 is located in section 4.

2.1 Question 3: How to identify bottlenecks and loops in the store

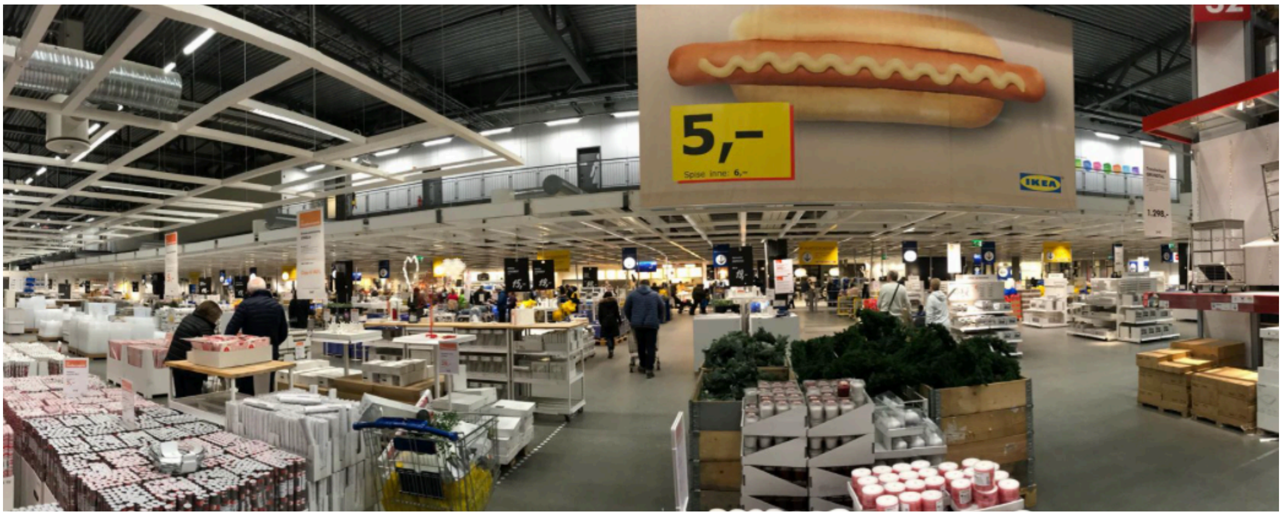
For any queuing-system, a bottleneck is found when the amount of new customers per time unit, exceeds the amount of customers processed per time unit. If the queuing-system is dynamic, and more service-points can be opened to adapt to the demand, a tool such as linear programming is excellent for optimizing the costs while fulfilling demand and other possible constraints.

3 Simulated data

It is assumed that a set of internal metrics/operational statistics is available for IKEA, for each of the areas in which IKEA wants to optimize its staff planning. The time-varying metrics will be simulated using assumed statistical distributions, whereas characteristics such as amount of cash-out desks and capacity in que will be based on assumptions.
The operational data will as mentioned earlier be simulated using the Monte Carlo method, where the final data will be a time-series of values that will be used for modelling.

3.1 Cash-out area

The Cash-out area is assumed to have 15 different stations, with a max capacity of 20 people in line (the point to which the picture below is taken). The que-system for cashing out is first-come-first-served, and each station has its own line, as seen below.



The processing time of the customers are entirely based on the assumption that the average serve time is 4 minutes. The customer arrivals are assumed to be Poisson-distributed, and will differ by day of week. In total, there are 4 different distribution settings for the customer arrivals. To make the simulation more realistic, a discrete stochastic term is added to the customer arrivals. This term is randomly distributed U(0, 5).

No autocorrelation or trend components is added to customer arrivals, since it is assumed that the amount of customers at day(t) will not be influenced by the amount of customers at the previous day (t-1).

- Customer arrivals:
    - Monday to Thursday:
        - Poisson (lambda = 12)
    - Friday
        - Poisson (lambda = 24)
    - Saturday
        - Poisson (lambda = 32)
    - Sunday
        - Closed
    - Stochastic component
        - Uniform (0, 5)

This distribution could have been including the processing time of the customers, as well as fluctuations during the day. However, since the purpose of this case is not to demonstrate complexity in the simulation of data, but rather to solve a business case using predictive analytics, this have not been included.

4 Proposed solution

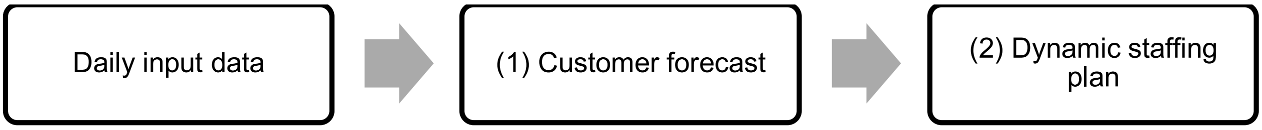
The main motivation for IKEA is to 1) *Reduce waiting time* and 2) *Increase sales per worked hour*. These goals are a function of two factors in the cash-out area:

1) Optimal combination of open service stations at all times
2) Minimal use of working hours

Since it is not the purpose of this case to discuss methods for up-selling, minimizing the use of working hours, is the best choice to increase sales per working hours. As discussed in section 2.1, bottlenecks can be avoided by opening the optimal combination of service stations at all times, subject to the demand and minimal use of working hours (costs in our model). In order to plan ahead, a prediction model is used for forecasting the expected amount of customers, alongside linear programming module for finding the optimal combination of.

4.1 Methodology

To solve the problem, a two-step model is proposed, consisting of the following components: 1) A LSTM Neural Network for forecasting the amount of customers 2) An integer linear programming module (ILP), that acts on the forecast and suggests a staffing plan.



The daily input data consist of internal as well as external data sources (discussed in section 4.2), which is then fed into the forecasting module (1). The forecasting module then makes a prediction for the next 120 minutes. This predicted flow of customers are then used in an ILP module (2), in order to determine how many service points that needs to be opened for the next 120 minutes. These staffing plans are made with 10-minute intervals (e.g. total customers the next 10 minutes).

<u>Overall procedure:</u>
In pseudocode, the overall procedure can be expressed by the algorithm below. The end result is a 15x12 matrix M, with the staffing plan for the next 120 minutes.

For every 120^{th} minute:

1. Load Daily input data
    a. Total customers last 120 minutes.
    b. Daily meta-inputs (Calendar, Marketing, Weather).
2. Make prediction $\widehat{Y}$ (next 120 minutes) from model(1).
3. Generate a 15x12 null-matrix denoted M:
4. *For every j in 1 to 12 elements of* $\widehat{Y}$*:*
    a. Store value *j* in vector $\widehat{Y}$ as Z
    b. Execute ILP model with Z as input
    c. Store vector with results as θ.
    d. Replace column *j* in M with values from θ.

4.2 Input data

Since no data has been provided by IKEA, this section only suggest data sources as well as how to obtain these for predictive modelling. The suggested data sources are listed below, where the ones marked with red have not been simulated for the demo example:

- Internal data:
  - Customers every 10^{th} minute last 5 years
  - Marketing calendar of IKEA campaigns
- External data:
  - Weather forecast (Very good and very bad weather)
  - Seasonality (Fall = Higher demand from students)
  - Google trends (IKEA products and brand mentions)
  - Macro indicators (Customer trust/spending trends)

The forecasting module is assumed to be trained on a dataset containing the amount of customers in the cash-out area ques for the past 24 months. This data can be obtained from surveillance cameras in the cash-out area.
To be more specific, a Convolutional Neural Network, trained to detect the number of persons in an area of the image could utilized to count the number of customers in the que through a period of time. In a sequence
The day of week, as well as the weather forecast and input from the marketing plan of IKEA campaigns could be used as inputs, alongside development in macro indicators as well as mentions of IKEA products in Google trends.

4.3 Model 1: Customer Forecast

The model chosen for predicting the future flow of customers is a LSTM (Long short-term memory) Neural network. The model has a simple architecture with 1 recurrent layer of 60 units in each. The predicted sequence from this model $\widehat{\mathrm{Y}}$ is a vector of length 12, where each point is the expected number of customers arriving in the que every 10^{th} minute, for the next 120 minutes.

Training & Evaluation:
This model was trained on a dataset of 102570 observations (10-minute-interval of all opening hours from 01/01/2016 to 01/06/2016, excluding Sundays). This has been divided into a training set of 70% (71799) and a test set of 30% (30771). The model was able to obtain a Mean absolute error: $MAE = \frac{\sum_{i=1}^{n}|Y_i - \widehat{\mathrm{Y}}_i|}{n}$ of 14.7, meaning that the model on average is 14.7 from the correct amount of customers in que over the 10-minute interval.

4.4 Model 2: Dynamic staffing plan

When the expected number of customers are predicted from model 1, the output will be used as an input in model 2, which is a Integer Linear Programming model (ILP). The ILP model is set to find a combination of employees that minimizes staffing costs, subject to the amount of service stations needed (which is again based on expected demand).

The procedure can be described through the following pseudo-code:

1. Generate a 15x12 null-matrix denoted M:
2. *For every j in 1 to 12 elements of* $\widehat{Y}$*:*
    a. Store value *j* from vector $\widehat{Y}$ as scalar Z
    b. Execute ILP model with Z as input
    c. Store vector with results as θ.
    d. Replace column *j* in M with values from θ.

The service points in IKEA's cash-out area is assumed to be different in terms of their capacity and speed for serving customers. For instance, the self-service points are assumed to be slower than the manned service points. However, these are cheaper to open, since 1 person can supervise 4 customers at the time. The coefficients and constraints for the model are illustrated in the Python-syntax located on Github (https://github.com/Mikeriess/IkeaCase/).

4.5 Infrastructure and software

As can be seen by the source code in the Github repository the solution relies on a set of open-source libraries built for Python. Since the simulation section will not be needed for a real-world implementation, Python and the associated libraries is the only critical components. Since the volume of the data will increase rapidly, a high-performance environment for storage and queries such as Hadoop or Spark is recommended. The suggested solution is supposed to be connected to HR-modules in IKEA's ERP-systems which will act on the staffing plans.

5 Conclusion

It is highly likely that IKEA can benefit by using deep learning alongside traditional optimization methods for generating their staffing plans. As have been pointed out earlier, real data from IKEA would be needed in order to assess to which degree a deep learning framework would be able to increase the sales per working hour, and decrease the waiting time. Since the suggested algorithm is based entirely on simulated data, it is hard to draw any conclusions about the performance compared to the current solutions chosen by IKEA.