Retail Store Workforce Forecasting With Aggregated Output Regression

Author: Tim Janssen

Supervisors Radboud University:
Elena Marchiori
Twan van Laarhoven

Supervisors Info Support:
Maarten van Duren
Hylke Peek

July 2, 2016
Abstract

In this research we use previously seen data and weekly forecasts of sales and customers, to predict the amount of workforce required in a retail store. The data set contains a difficulty, the target values for observations are latent. We do however have aggregated target values and know that these targets are a sum of the latent values.

We assume retailers are cost minimizing agents and approach the problem by reducing labor costs through more accurately forecasting of the needed labor hours. We compare three different type of approaches: 1) scaling 2) linear regression and 3) time series. We show how these methods perform when estimating the mean and a low quantile.

The results show that all methods outperform forecasts based upon stopwatch estimates. The time series approach gives the best results on our test set. It reduces our error metrics, the root mean squared error and pinball loss, by at least 50%.

We think that the results for the time series approach can further improve by increasing the size of the training set, effectively allowing for seasonality correction. We expect a large increase in accuracy and the usability of models by gathering and analysing a smaller granularity of data.

Further research should focus on approaching retailers as profit maximization agents, to research the relation between a larger workforce and an increase in profit.
# Contents

1 Introduction ........................................... 2  
  1.1 Task Definition ..................................... 6  

2 Background ............................................. 8  

3 Data Preprocessing ...................................... 10  
  3.1 Sales & Labor ......................................... 10  
  3.2 Forecast ............................................... 12  

4 Analysis: Time-study .................................... 13  
  4.1 Scaling ................................................ 14  
  4.2 Results ............................................... 16  
  4.3 Conclusion ............................................ 17  

5 Linear Regression Model ............................... 18  
  5.1 Aggregated Regression .............................. 18  
  5.2 Separation Into Clusters ........................... 19  
  5.3 Results ............................................... 20  
  5.4 Conclusion ............................................ 21  

6 Time Series Model ...................................... 22  
  6.1 Results ............................................... 23  
  6.2 Conclusion ............................................ 24  

7 Conclusion & Discussion .............................. 25  

8 Bibliography ............................................ 29
1 | Introduction

Retail is a sector which is full of competition, retailers try to differentiate their shop from other retailers to attract more customers and to make them want to buy more products. Instead of trying to increase sales, a shop manager can also strive to decrease his operational costs to gain the competitive advantage. Employee cost is one of the largest expenses of a retailer and thus is a logical place to start reducing cost. Reducing employee costs can be achieved by increasing efficiency in the business processes. Meaning that we have to schedule less, but more productive, working hours. Even a small increase in productivity can give a large cost reduction over the many working hours of many employees.

However, it is not a trivial problem for a shop manager to assign a reasonable estimate of the working hours for a department, due to the fact it is unknown how many (and which) products will be sold next week and how long it will take to restock them. That is the reason that a shop manager distributes so called budget hours over departments. This simplifies the shop manager’s task to distribute preferred amount of working hours over the different departments and transfers the scheduling task to the department manager.

The budget hours assigned to the departments are a limit to the amount of working hours a department manager can assign to employees within a weekly department schedule. The key idea behind this transfer is that a department manager has a better understanding of productivity in their department and can, within the budget hours restriction, create a better schedule.

While this simplifies the scheduling problem for a shop manager, it transfers the problem to the department manager. Namely a department usually consist of multiple clusters itself, see figure [1.1] for a simple shop structure. Clusters are groupings of related products, like vegetables or dairy products. This basically creates the same non-trivial problem of scheduling the budget hours but now over the different clusters. Within each cluster clerks have a
set of tasks to perform, such as restocking the shelves, cleaning and helping the customers. Each task takes a different amount of time to complete.

Figure 1.1: Example shop with two departments. One department has three clusters, the other has two clusters. Each cluster has a list of possible tasks.

A department manager can take several approaches when creating a work schedule for his/her department. The 'best' approach to take is dependent on the goal the manager wants to achieve. We will discuss three of these goals, which we shall refer to as DM (department manager) goals:

DM goal 1) The department manager has a certain preference of understaffing over over-staffing. He/She wants to know the minimum amount of budget hours needed for his employees to complete a set of tasks. This results in strict but reasonable time limits for the employees.

DM goal 2) The department manager wants to know the expected amount of working hours, meaning that the actual amount of working hours can be either higher or lower than the expected amount. This goal is interesting if the manager does not have a particular preference on under- or overestimate of the amount of work.

DM goal 3) The shop manager wants to increase the quality of service and therefore schedules more budget hours. This will increase costs but can improve sales by presenting a better quality store to the customers.

In this thesis we will tackle DM goal 1 and 2.
Currently, only DM goal 1 is tackled. A core step for solving it is that each task of the different clusters has underwent an extensive time-study. The goal of this study is to determine the time it takes to complete a single repetition of the task. The time study is performed as follows:

1. A measurer follows clerks of single shop throughout the course of several days. He/She times every type of task the employees performs with a stopwatch and note the time.

Note, that a new clerk will probably not work as fast as one with years of experience. Productivity of a clerk can also vary throughout the day and between days or even between shops. Therefore the timings the measurer makes will differ from shop to shop and time to time.

2) If the measurer notices a clerk is taking longer than usual to complete the task, the measurer weighs the timing to more closely match the regular time. In business terms, such a biased measurement is called a quality measurement. Because a domain expert uses his domain knowledge to most accurately bias the estimate.

The biasing of timings is done to give a department manager estimates about how fast a task could be completed. The department manager can then use this information to determine the minimum budget hours needed to schedule. Also this biased estimate gives the clerks a goal time to complete their tasks, this to boost their productivity.

3) Based upon the measurements of each type of task, an average is computed which are then called the **norm hours** of a single task, this is the (biased) average time a task will take.

4) We multiply this average with the number of tasks that are expected for next week. This gives the expected minimal time that clerks are busy with this task in the upcoming week, or the total norm hours for a task for that week.

5) We sum the expected norm hours over all tasks within a cluster, we now basically have a forecasting model of the budget hours for next week for that particular cluster.
However, the time-study has three different drawbacks. Firstly it is inaccurate for three reasons:

- The time bias is based upon measurer subjectiveness, which makes the task completion time estimate inaccurate.
- Timings can also become inaccurate because the knowledge of being timed influences the productivity of the worker.
- Since worker productivity differs from shop to shop, from time to time they do not generalize very well to other shops when measurements are taken in only one or two shops and on only one or two occasions.

The second drawback is that by biasing the measurements we get an estimate of how fast a task could have been completed but we lose information about the true (mean) time it takes to complete the task. Therefore it can not be used for department managers who are interested in creating a schedule based upon the mean norm hours.

The third drawback is that this method does not transfer well with new tasks, new retail formulas and new workers. That is why it is necessary to regularly remeasure the norm hours for every single task. This method is therefore also a very expensive method as every timing event again needs persons who perform quality time estimates of a task and this has to be repeated for every possible task in the different types of retailing shops on a regular basis.

Therefore Info Support has asked us to investigate the possibilities of replacing the manual time study with Machine Learning (ML) methods as ML methods are being increasingly used for predictive analysis [Witten and Frank, 2005].

The time study method is an inaccurate and an expensive method, which can only satisfy DM goal 1. The goal of this research is to at least improve on these two aspects, while satisfying DM goal 1. Next to goal 1 we want extend the current possibilities of a department manager by proposing models that satisfy DM goal 2. These models will give the department manager a new valuable insight in his department.

Luckily, information about previous product sales and working hours are stored in a database. Thus creating the possibility for ML algorithms to learn from it. However there exists a problem with the data available. While budget hours are available on a department level they are missing on cluster level, see table [1.1] for an example of the data structure.
### Table 1.1: Example training data

<table>
<thead>
<tr>
<th>Shop</th>
<th>Department</th>
<th>Cluster</th>
<th>Products</th>
<th>Sales</th>
<th>Customers</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>1</td>
<td>945</td>
<td>1,205</td>
<td>407</td>
<td>105</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>2</td>
<td>539</td>
<td>327</td>
<td>213</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>3</td>
<td>382</td>
<td>1,542</td>
<td>271</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>B</td>
<td>4</td>
<td>258</td>
<td>449</td>
<td>41</td>
<td>55</td>
</tr>
<tr>
<td>1</td>
<td>B</td>
<td>5</td>
<td>535</td>
<td>392</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>1</td>
<td>1,750</td>
<td>3,002</td>
<td>567</td>
<td>16</td>
</tr>
</tbody>
</table>

### Table 1.2: Example test data

<table>
<thead>
<tr>
<th>Shop</th>
<th>Department</th>
<th>Cluster</th>
<th>Products</th>
<th>Sales</th>
<th>Customers</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>1</td>
<td>415</td>
<td>1,754</td>
<td>290</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>B</td>
<td>4</td>
<td>870</td>
<td>1,053</td>
<td>367</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>4</td>
<td>134</td>
<td>2,500</td>
<td>750</td>
<td></td>
</tr>
</tbody>
</table>

#### 1.1 Task Definition

Say we have $M$ shops, $N$ different types of departments and $K$ different types of clusters. Each department $j$ of shop $i$ has a cluster vector $U_{ij}$, the elements $1 \ldots K$ indicate if cluster $k$ is present. The elements of the cluster vector are $U_{ijk} \in \{0, 1\}$ where an 1 indicates that a cluster $k$ is present in department $j$ of shop $i$ and a 0 if it is not. Since clusters can only be present in one department of a shop, we have $\sum_{j=1}^{N} U_{ijk} \leq 1$

$T$ is the vector containing the observed weeks and $t \in T$. $X_{tik}$ is the triplet $<P, S, C>$ of products (P), sales (S) and customers (C) at time $t$ for shop $i$ in cluster $k$. If an observation is missing at a given time or of a cluster type, $X$ is a vector containing all zeros. The true hours $y_{tij}$ for department $j$ is available but the true hours on cluster level $z_{tik}$ is missing. We know that the real valued targets of a department $j$ is the sum of the latent targets within that cluster or mathematically $y_{tij} = \sum_{k=1}^{K} z_{tik}$.

Now consider our task of replacing the time study with ML methods, making it more accurate and scalable, and incorporate the goals for a department manager. Our task is now twofold:

1. Develop models which outperform, in terms of accuracy and scalability, the currently used method of a time-study. Here we make a difference between the two earlier mentioned goals of a department manager:
**DM goal 1**

We create a model that estimates the minimum amount of budget hours a department manager needs for his department per week. The minimum amount of budget hours needed, depends on the optimization function of the department manager. We assume that the department manager is a profit maximization agent, now let $C_-$ be the costs of having too much scheduled hours and $C_+$ be the missed sales due to under-scheduling of labor. Let $m$ be the true hours needed and $n$ the amount of scheduled hours.

Define $\tau$ as the ratio we value under-scheduling over over-scheduling:

$$\tau = \frac{C_-}{C_- + C_+} \quad (1.1)$$

We define the department managers loss function as the forecasting error between $m$ and $n$, times how we value that error (is it over or under scheduled):

$$L(m,n) = \begin{cases} (m - n)\tau & \text{if } m \geq n \\ (n - m)(1 - \tau) & \text{if } m < n \end{cases} \quad (1.2)$$

This loss function is also known as the Pinball (Pb) loss and $\tau$ as the target quantile. We can now interpret a better model as a model with a lower pinball loss between the true budget hours of a department $y_{tij}$ and its estimate $\hat{y}_{tij}$ at a specific quantile $\tau$. This task ensures the same functionality as currently available.

**DM goal 2**

We extend the functionality provided to the department manager, by answering the question of what amount of budget hours do I expect to need this week? This information is currently not available and gives the department manager a valuable insight.

We use the Root Mean Squared Error (RMSE) between true en predicted values as our evaluation metric for model selection. The RMSE is a frequently used error metric in numerical predictions, we prefer models that have a lower RMSE. Thus we can interpret DM goal 2 as a minimization problem of the RMSE.

2. Estimate the unknown values for $z_{tik}$ such that we can not only estimate budget hours on department level but also on the cluster level and therefore no longer need the time-study to estimate these values.
2 | Background

In this chapter we will elaborate on prior work related to this thesis. Since we do only have target labels for departments not of clusters our problem is not a supervised learning problem where every instance has an label. It also differs from a fully unsupervised problem where we do not have any labels at all. In semi-supervised learning [Zhu, 2005] we have sometimes labels for instances and sometimes we do not, clearly this is also not applicable to our problem as our instances do not have labels.

The paradigm Multiple Instance Learning has the most resemblance to our problem. MIL was first proposed under this name in [Dietterich et al., 1997]. It is mostly used in a classification context, see [Amores, 2013] for an extensive review. However our problem is not a classification task but a regression task. There have been work done in this area to formulate MIL as regression problem named multiple instance regression [Ray and Page, 2001], they assume that every labeled collection has a prime instance that determines the target label which is not the case in our problem since we have aggregated target labels. Very similar to our problem is [Musicant et al., 2007] where they propose, next to classification, a regression framework with aggregated target labels for other ML methods such as K nearest neighbours, Random Forests, Neural Networks and Support Vector Machines. However their problem differs from ours since their collection of instances has no structure, they assume the instances in the collections are drawn from a single distribution while in our collections the data is drawn from different distributions.

Linear regression [Neter et al., 1996] is a basic form of regression where we try to explain a real valued dependant variable by a linearly combination of the independent variables. Linear regression minimizes the sum of squared errors (SSE) between the predicted and the true value of the dependant variable. The standard approach is the ordinary least squares (OLS) method which solves the minimization of the SSE analytically by setting the derivative to zero. The RMSE can be interpret as the standard deviation of the unexplained variance of the OLS fit. Since we assumed that the amount of time a clerk spends is linearly dependant on the amount of products to refill,
it matches our problem. However linear regression is a supervised learning problem, it can not handle the aggregated labels.

It makes sense to incorporate a time element into our model since forecasts of sales are made every week and these sales are strongly correlated with the previous week. A typical method for time-series forecasting is Moving Average (MA) [Hamilton, 1994] where we take the working hours of the previous \(n\) weeks and take the mean of this. Another model is the Autoregressive moving average model (ARMA) [Hannan, 2009], this models the time-series as a stochastic process.

Hierarchical models (or multilevel models) [Gelman and Hill, 2006] are particularly suitable to this type of data since we have different clusters within departments within retailing shops. With hierarchical models we can estimate the varying effects of each ‘level’ to better suit the data than with OLS regression. Different types of hierarchical models are possible: random intercepts model, random slopes model or a combination. According to the domain expert every cluster of the same type should be assumed to behave the same e.g. the slopes are equal for all (same type) clusters. If a cluster can not meet predicted budget hours the retail manager should reconsider their specific setting to adjust it accordingly. However every shop is different in size which influences start up and closing times by the time it takes to clean it. A random intercept model could therefore be very well applied on our problem.

The OLS fitted model explains the mean of the data, there are cases where one might not be interested in the mean of the data. For example when you are interested in predicting the lowest prices of a product when prices fluctuate. In our case we made the assumption that clusters should be identical over different retailers. Meaning that the lowest quantile of our data is of high interest, since these are probably the most efficient retailers and indicate that every other retailer could reach this quota as well. To cite [Mosteller and Tukey, 1977]:

> What the regression curve does is a grand summary for the averages of the distributions corresponding to the set of x’s. We could go further and compute several different regression curves corresponding to the various percentage points of the distribution and thus get a more complete picture.

Quantile regression [Koenker, 2005] is therefore very well suited for our problem, not only can it gives us insides into the lowest quantile it also tells us how the highest quantiles behave.
The final data, as seen in table 1.1, is constructed from two different data sets: Sales and Labor. The Labor data contains information about the working hours of the employees. The Sales data set is an hourly aggregated collection of the sales in a specific cluster of a shop. The pre-processing steps for these data sets will be discussed in the sections below. Both data sets contain observations over 24 weeks in the time span of week 29 until week 52 of 2015. The observations are generated over these weeks by 229 shops of one single retail formula, which we shall not mention due to confidentiality.

To allow for validation of our methods we split the final data into two sets, a training and a test set. The first weeks (week 29 - 42) are used for training while all testing is done on the last 10 weeks (week 43-52). These last 10 weeks will be used for 10-fold cross validation, where we train for example on weeks 42 and test on 43, train on 43 test on 44 etc.

Since we want to predict the amount of budget hours of a specific department with the sales information available, we will bring these two data sets into the proper format such that a merge of the two data sets on the variables week, shop and department will give us the weekly sales and labor information per shop per department.

### 3.1 Sales & Labor

**Sales Data**
A single observation of sales information is a hourly aggregated vector of:

- the week of the year
- date and time
- shop id
- department id
- cluster type
- number of products sold \((P)\)
- sales \((S)\)
- number of customers \((C)\)
The first pre-processing step is to aggregate the hourly information to weeks, by summing the variables $P$, $S$ and $C$ for the same shop id and department id.

The second step is to filter out departments which have too few weekly observations. We removed departments if less than 3 shops have this department.

The third step is to remove departments for which some shops need labor hours whereas others do not as they outsource the work. As this is a preliminary study into replacing norm hours, these departments are excluded from the data set.

The fourth and last step is to go from cluster level to the department level by again summing for a specific week the variables $P$, $S$ and $C$ for each cluster within the department. This step can be omitted to use the variables on cluster level, as we will see later in section 5.

The final result is a triplet per week per shop per department of the features $<P, S, C>$.

**Labor Data**

The labor data set has the following structure:

- the week of the year
- department id
- employee id
- productive
- shop id
- working hours

As we can see labor information is only available on a weekly basis and on the department level. This is the reason that we transform all of our sales data to this granularity. Working hours can now act as our target labels $Y$. Productive indicates whether a paid hour was either a truly worked hour ’$Y$’ or a paid leave ’$N$’.

There are two pre-processing steps in the labor data set. 1) We removed the rows for which productive equals ’$N$’. Since our goals refer to truly worked hours instead of a paid leave.

2) We aggregated working hours over all employees, within a department within a working week. The result is the following vector:
• the week of the year • department id
• shop id • working hours

**Merging**
We now have two preprocessed data sets, sales and labor. From these two we will create the final data set by concatenating the variable working hours from the labor data to the sales data. The concatenation is done on records that have the same variables: week, shop id, department id. We drop the records that are only present in one of the two data sets. We dropped some departments while pre-processing the Sales data, the result of this is that records that do not appear in the Sales data are also not present in the final data set. The distribution of departments over the shops can be seen in table 3.1. The final data set has records structured as follows: \(<\text{week, shop id, department id, P, S, T, working hours}>\).

<table>
<thead>
<tr>
<th>Department</th>
<th>No. shops</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>221</td>
</tr>
<tr>
<td>B</td>
<td>175</td>
</tr>
<tr>
<td>C</td>
<td>225</td>
</tr>
<tr>
<td>D</td>
<td>6</td>
</tr>
<tr>
<td>E</td>
<td>171</td>
</tr>
<tr>
<td>F</td>
<td>39</td>
</tr>
<tr>
<td>G</td>
<td>7</td>
</tr>
<tr>
<td>H</td>
<td>178</td>
</tr>
</tbody>
</table>

Table 3.1: Distribution of the different department types over the different shops.

### 3.2 Forecast

Next to the Sales and Labor data sets there is a third data set, which we call the forecast data. The forecast data is a data set which contains the forecasts of P, S, C. These forecasts are a computed average of at least 4 relevant weeks selected by the department manager. The data looks like the Sales data after pre-processing, except that P, S, C and working hours are forecasts. The working hours forecast is computed by averaging the amount of tasks of the selected weeks and multiply the tasks with their norm hours, then sum the total norm hours per task.

The only pre-processing step we have to do with this data set is to keep only those records (week, department, shop combination) which are also present in the pre-processed Sales & Labor data, the other forecasts are removed.
In this chapter we will analyse the manual time-study method of forecasting budget hours, from now on we will refer to this method as the stopwatch method. First we will show how the stopwatch method performs with respect to the RMSE and the pinball loss. Next we will introduce a scaled version of the stopwatch method and show that it performs better than the original on both DM goals.

For analysis of the stopwatch method we use the pre-processed forecast data, combined with the labor data. The data sets are combined on the same week, shop, department combination, such that we have the predicted budget hours and the true norm hours in the same entry. We will use the training weeks 30-42 to train parameters on and test on week 43-52. This gives us a total of 13,253 training observations and 10,222 test observations.

The estimates for the norm hours of a task are biased to a smaller estimate. We therefore expect the forecast to be correlated with, but below, the true working hours. The influence of the bias will increase as more tasks need to be done, for example in larger shops. We plotted the predicted hours of a department against the true working hours of that department in figure 4.1.

We see that our intuition matches the data, the points are mostly above the diagonal and for larger predicted hours the points deviate further from the diagonal. Here the diagonal represents that predicted budget and true hours are equal, so in the ideal case all points lie on this line.

Since norm hours are used by department managers as goals for the employees to meet, we can check with the data how many goals are actually met on department level. In our training data only 34.4% of goals are met, meaning that these data points are below the diagonal and 65.6% is above.

Whether this percentage is desirable is up to the department managers to interpret. However this percentage is taken over all departments of all shops. The percentage of goals that are met in a specific department is more useful
to know for a department manager such that he/she can take action if not enough goals in his department are met. The overall percentage can now be taken as a baseline for the DM to compare his department with others.

Figure 4.1: Horizontal predicted hours by the stopwatch method, vertical true working hours. The red line represents the diagonal meaning true hours = predicted hours.

4.1 Scaling

From figure 4.1 we can not only see that predictions often are underestimated we can also see that they are correlated, as the points lie along a line. In fact the correlation between the two variables has value 0.956, which suggest a linearly dependence. This brings us to the idea of scaling the forecast with a constant factor such that the prediction of the stopwatch method comes closer to the true working hours. This factor can be seen as a correction operation to undo the bias the measurer introduced. We can either scale towards the mean of the data or to a specific quantile, depending on the DM goal.

Since similar products are clustered in a department, the products within a department are more similar to each other than they are to products outside of the department. The tasks within a departments are dependant on the products of that department, therefore the tasks are also more similar within a department than to tasks of another department. The introduced bias per
task is more likely to look alike in a department than between departments because similar tasks are more likely to have the same bias.

We can therefore take the analysis of the predictions a step further and analyze the percentage of budget hours that are achieved per department type. Achieved means that the true hours are below the predicted budget hours. In Table 4.1 we listed the percentage of the budget hours achieved by department type.

We can see that there is a large difference in percentage of goals achieved between department types. This implies that the stopwatch method has fluctuating performance over the different departments. The performance differences over the different departments have as a consequence that some departments are given a strong under estimation of the amount of labor required (department C & G) while others get a reasonable estimate of labor in which about 50% of budget hours are achieved (department B & E). A scaling factor per department type can straighten the performance and is therefore more logical than one factor for all departments.

<table>
<thead>
<tr>
<th>Department</th>
<th>% Budget hours achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>41.4</td>
</tr>
<tr>
<td>B</td>
<td>53.9</td>
</tr>
<tr>
<td>C</td>
<td>7.8</td>
</tr>
<tr>
<td>D</td>
<td>28.2</td>
</tr>
<tr>
<td>E</td>
<td>52.0</td>
</tr>
<tr>
<td>F</td>
<td>34.5</td>
</tr>
<tr>
<td>G</td>
<td>6.6</td>
</tr>
<tr>
<td>H</td>
<td>24.4</td>
</tr>
</tbody>
</table>

Table 4.1: Norm hours achieved per department type.

We can scale the the forecast of the stopwatch method but we can also shift it, allowing for a better fit of the model to the data. To validate that scaling can improve the forecast, we will run three different experiments for the above mentioned cases.

Experiment 1. The original forecast, this is our baseline. This simply involves calculating the pinball loss and the RMSE on the test weeks.
Experiment 2. Scaling with one factor for all departments, for this method we need to find an $\alpha$ and/or $\beta$ that can express the linear relationship between true working hours and the predicted hours. We use respectively quantile and OLS regression to solve for the scaling and shifting factors that minimize the pinball loss and the RMSE on the training weeks.

$$y_{tij} = \alpha + \hat{y}_{tij}\beta + \epsilon$$

(4.1)

Experiment 3. Scaling with a factor per department. This method is comparable with experiment 2 but we find an $\alpha$ and/or $\beta$ for every department type.

$$y_{tij} = \alpha_{j} + \hat{y}_{tij}\beta_{j} + \epsilon$$

(4.2)

Before we can run these experiments we need to determine the quantile on which we compare performance. Since we want to compare with the original forecast we will set $\tau$ to the quantile of the stopwatch method. We determined the quantile $\tau$ by taking the ratio of true working hours that are $\leq$ the predicted budget hours and found $\tau = 0.344$.

4.2 Results

In total we have done five different versions for the three types of experiments. We differentiated in experiments 2 and 3 between experiments that do not use an intercept to shift the data (a) and those with an intercept (b). For every experiment the performance of the quantile model is noted in the column Pinball Loss and the performance for the linear regression model in the column RMSE.

Table 4.2 shows the results of the experiments on the test data. We can see that for DM goal 1 using a scale factor (and intercept) improves the pinball loss over the original forecast. Introducing multiple scale factors (and intercepts) for each of the different department types further improves performance, with a lower pinball loss.

Since the intent of the time-study originally was to give a department manager insight in the minimum amount of labor hours to schedule, it may seem not fair to compare the RMSE of experiment 1 with 2-3. The result for experiment 1 is however added to give the reader a feeling for performance when it is used in the context of DM goal 2: expected amount of labor. We see that scale and shift per department type (3b) also give the best result for this goal.
<table>
<thead>
<tr>
<th>Method</th>
<th>Pb Loss ± SE</th>
<th>RMSE ± SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Stopwatch</td>
<td>15.42 ± 0.52</td>
<td>62.41 ± 4.26</td>
</tr>
<tr>
<td>2a. Scale</td>
<td>14.39 ± 0.38</td>
<td>48.31 ± 3.03</td>
</tr>
<tr>
<td>2b. Scale and shift</td>
<td>13.60 ± 0.36</td>
<td>47.52 ± 2.86</td>
</tr>
<tr>
<td>3a. Scale dep.</td>
<td>12.79 ± 0.39</td>
<td>45.47 ± 3.00</td>
</tr>
<tr>
<td>3b. Scale and shift dep.</td>
<td>12.73 ± 0.38</td>
<td>44.92 ± 3.02</td>
</tr>
</tbody>
</table>

Table 4.2: Pinball loss ($\tau = 0.344$) and RMSE for different scaling methods versus original forecast.

4.3 Conclusion

The results show that all experiments outperform the original stopwatch method for both goals of the department manager. Scaling and shifting the prediction per department type seems to deliver the best results for both regression models. Scaling is therefore both a simple and effective method to increase prediction performance.

When scaling the department with a factor we are effectively scaling all the clusters in that department with that factor as well. Therefore the scaling method can give estimates for budget hours on clusters.

Next to the performance increase, scaling does provide the department manager a new insight into the expected working hours (DM goal 2). This is valuable knowledge when scheduling workforce next to a minimum, as it gives a more complete picture. Different values for $\tau$ can also be used, allowing the department manager to choose for a high quantile to extend the picture with the maximum expected working hours.

Scaling can make the forecast more reliable and give a more complete picture of the expected working hours but it still requires the time-study to create the original forecast. This makes the method expensive and not scalable over multiple retail formulas. In the next chapter we will propose a model that does not require the stopwatch method to make a forecast.
In this chapter we will propose linear models that have better performance than the original forecast and that no longer require the time study to create the forecast. We will compare two different types of models 1) where we use aggregated variables to the department level 2) where we differentiate between clusters in a department. We show that the variables on cluster level can better express the working hours than their aggregated version.

The time-study makes the assumption that every task needs a certain amount of time to complete. This assumption implies that total time spent on multiple repetitions of the same task is the product of the amount of repetitions and the time for a single execution, or in other words the total time spent on tasks is linear in the amount of tasks. In this chapter we adopt the assumption that time spent on tasks is linear in the amount of tasks.

In addition we will also make three other assumptions:

1. Every product has a task of refilling it, we assume that within a department this task takes the same amount of time to complete.

2. The time spent on customers, is linear in the amount of customers. Having more or less customers in the department does not influence the time spent on a single customer.

3. A product price linearly influences the amount of labor spent on the product. A products prize can be more expensive due to material costs or factory processing, but also because it needs some sort of processing in the shop itself.

5.1 Aggregated Regression

As mentioned in chapter 3 an observation $X_{ij}^{dep}$ is made at time $t$ of department $j$ of shop $i$. The superscript $^{dep}$ indicates that the variables are aggregated to departments. In our case $X_{ij}^{dep}$ is a triplet $<$P, S, C$>$ where P is the amount of products that are sold in this week, S the amount of sales and C the amount of customers that bought a product at department $j$ at
Since we assumed that $y_{tij}$ is linear in terms of $X_{tij}^{dep}$ we can set up an experiment with a linear regression model where:

$$y_{tij} = X_{tij}^{dep} \beta + \epsilon$$

$\epsilon \sim \text{Normal}(0, 1)$

### 5.2 Separation Into Clusters

In the fourth pre-processing step of the sales data, aggregation of clusters to departments, disables any differentiation we can make between clusters. A cluster is a grouping of related products and therefore also has related tasks. Tasks in one cluster may take a different amount of time to complete than tasks in another cluster. For example a task may be easier thus requiring less time. The influence that a variable has, for example P (sold products) on working hours may therefore be different between clusters (thus invalidating our first assumption).

Let $X_{tik}$ be the observation of $<P, S, C>$ at time $t$ of cluster $k$ in shop $i$. Note we do not use the superscript $^{dep}$ to indicate that $X$ is an observation of a cluster. Let $V$ be the concatenation of $X_{tik}$ of the available clusters in a department, defined as:

$$V_{tijk} = X_{tik}U_{ijk}$$

(5.1)

Let $o$ be $<t, i, j>$ such that $V_o$ is a row in table 5.1:

<table>
<thead>
<tr>
<th>$P_1$</th>
<th>$S_1$</th>
<th>$C_1$</th>
<th>...</th>
<th>$P_K$</th>
<th>$S_K$</th>
<th>$C_K$</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>24,337</td>
<td>6,419</td>
<td>36,298</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>130</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>21,980</td>
<td>10,449</td>
<td>41,062</td>
<td>239</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>117</td>
</tr>
<tr>
<td>22934</td>
<td>6113</td>
<td>34572</td>
<td>...</td>
<td>20,303</td>
<td>9,848</td>
<td>38,160</td>
<td>367</td>
</tr>
</tbody>
</table>

Table 5.1: Concatenation of clusters variables $<P,S,C>$ for all present clusters in a department

We now set up a second linear regression experiment where we use $X$ on cluster level to predict the working hours on department level:

$$y_o = V_o\beta + \epsilon$$

$\epsilon \sim \text{Normal}(0, 1)$

It would be ideal if we would know $V_o$ exactly when we predict $y_o$. Our training data does contain this information since those weeks already lie in the
past. However, in production the exact numbers of week \( t \) are only available to us in week \( t + 1 \). What is available to us in week \( t \) is the forecast data, these are generated at week \( t - 1 \). We therefore also use the forecast data, which we refer to as \( V^f \), as our training data for the prediction of \( y \). The forecast data will be processed into clusters in the same way as our training data, table [5.1]. This enables us to still express \( y_t \) at week \( t \). We expect this model to be not as expressive when trained on \( V^f \) then when trained on the \( V \) to predict \( y \). To verify this will we also do an experiment with both \( V \) and \( V^f \).

The way \( V \) is set up, particularly allows for the recovery of the budget hours on cluster level. Since \( P_k, S_k \) and \( C_k \) are variables of cluster \( k \). After we have obtained values of \( \beta \), we can say that the influence of a cluster is:

\[
z_{tik} = (\beta_{3k-2}, \beta_{3k-1}, \beta_{3k}) \ast X_{tik}
\]

In words, the budget hours at in cluster \( k \) is a linear combination of the variables \(<P, S, C>\) of cluster \( k \) times the corresponding \( \beta \)'s.

### 5.3 Results

To summarize we have set up three different experiments:

Experiment 4. Using the sales and labor data, we aggregate \( X \) to department level (\( X^{\text{dep}} \)) to predict \( y \) on department level.

Experiment 5. Using the sales and labor data \( V \), we concatenate \( X \) on cluster level to form \( V \) and predict \( y \) on department level.

Experiment 6. Using the forecast data, we concatenate \( X \) on cluster level to form \( V^f \) and predict \( y \) on department level.

The results of these experiments are shown in table [5.2] for reference we have also included our baseline (experiment 1) and the best scaling result (experiment 3b) from the previous chapter.

We see that aggregated linear regression is close to, but does not improve on the baseline regarding pinball loss, but it does show a decrease in RMSE relative to the baseline. Regression on the clusters however does outperform the baseline method in both metrics. Using the true data (exp. 5) gives a lower losses compared to training on the forecast data (exp. 6). However the difference in loss is very small. Notable is that both regression on clusters and on departments are both outperformed by our scale and shift method (3b).
<table>
<thead>
<tr>
<th>Method</th>
<th>Pb Loss ± SE</th>
<th>RMSE ± SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Stopwatch</td>
<td>15.42 ± 0.52</td>
<td>62.41 ± 4.26</td>
</tr>
<tr>
<td>3b. Scale and shift dep.</td>
<td>12.73 ± 0.38</td>
<td>44.92 ± 3.02</td>
</tr>
<tr>
<td>4. Aggregated</td>
<td>16.19 ± 0.35</td>
<td>55.49 ± 2.01</td>
</tr>
<tr>
<td>5. Cluster - True</td>
<td>13.91 ± 0.46</td>
<td>46.69 ± 2.19</td>
</tr>
<tr>
<td>6. Cluster - Forecast</td>
<td>13.99 ± 0.28</td>
<td>46.45 ± 1.63</td>
</tr>
</tbody>
</table>

Table 5.2: Pinball loss ($\tau = 0.344$) and RMSE for different regression models versus original forecast and scaling.

Furthermore, it was not possible to recover the budget hours for the clusters, as described in equation 5.2. When calculating the hours, we find that several clusters where assigned negative hours.

### 5.4 Conclusion

Aggregated regression on the department level gives slightly worse performance over the baseline forecast regarding DM goal 1 & 2. However this method does not need the time study to work and thus is scalable over multiple retail formulas.

Separating the clusters within a department has better predicting performance than both the aggregated version and the baseline. It also has the advantage of scalability over the baseline. The model that was trained on the true data (exp. 5) does not outperform the model trained on the forecast data (exp. 6). We think that we do not see any performance difference due to that the forecast is already very accurate. Therefore it might not be very rewarding to improve the forecast of $<P, S, C>$.

Unfortunately, we could not recover estimates of budget hours on cluster level with this model. We think that the clusters in a department are too correlated to each other to estimate the influence of a single cluster within a department.

Scale and shift department wise (exp. 3b) still has a better performance than linear regression on the clusters variables, it should therefore be used if one wants to have the best performance. However linear regression can be used for prediction when one does no longer want to use the time-study. Since we want both better performance and no longer use the time-study, we will propose a time-series model in the next chapter that can tackle both requirements.
In retail, it is very likely that week $t$ is similar to week $t-1$, since customer behaviour will not change drastically over one week. In such a way that between any two weeks, $X_t$ and $Y_t$ only differ in a relatively small amount from $X_{t-1}$ and $Y_{t-1}$. Exceptions to this are of course holidays, which give a short spike in sales. However, over longer periods of time customer behaviour will change due to seasonal effects or trends.

We are interested in predicting just the upcoming week. The most basic model we can think of, which we shall use as a baseline, is one in which we set the predicted hours of next week to be the true hours from previous week. This is a basic model but should give a reasonable performance already due to similarity of following weeks.

$$\hat{y}_{tij} = y_{(t-1)ij}$$ (6.1)

We do not have enough data to extend the basic time series model to include seasonal effects as we would then need a full year to train on. We have however enough data to extend the model with trend effects. We see between following weeks $t$ and $t-1$, that $y_{tij}$ and $y_{(t-1)ij}$ have a high correlation of 0.98. We include trend effects by modeling the budget hours as an autoregressive model of order 2, AR(2):

$$\hat{y}_{tij} = \beta_0 + y_{(t-1)ij}\beta_1 + y_{(t-2)ij}\beta_2$$ (6.2)

Still this model does not utilise any changes between a forecast $X_{tij}$ and true $X_{(t-1)ij}$ to predict $y_{tij}$. This is particularly important to deal with the earlier mentioned (holiday) spikes. We include these effects to our model, note that we use the concatenation of cluster variables $V$, as we have seen in the previous chapter they can better explain the variable $y$. $V^f$ indicates that we use the forecast data instead of the true data.

$$\hat{y}_{tij} = \beta_0 + y_{(t-1)ij}\beta_1 + V_{tij}^f\beta_2 + V_{(t-1)ij}\beta_2$$ (6.3)
6.1 Results

In the previous section we have introduced three time series models, for each we set up an experiments which can be used to estimate $y_o$:

Experiment 7. The basic model, the predicted hours of week $t$ are the true hours of the week $t - 1$.

Experiment 8. An autoregressive model of order 2. This is an extension to experiment 7 by adding trend effects to explain the changes between sequential weeks.

Experiment 9. A combination of a time series model and a regression model. We include both the hours and the $V$ from last week but also include the forecast of next week $V_f$ in our model.

The results of these experiments are shown in table 6.1. We can see that all time series models outperform the best model so far (exp. 3b). Interesting to see, is that even the basic time series model where we just take the true hours of the previous week gives a better prediction than all our previously discussed methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pb Loss ± SE</th>
<th>RMSE ± SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Stopwatch</td>
<td>15.42 ± 0.52</td>
<td>62.41 ± 4.26</td>
</tr>
<tr>
<td>3b. Scale and shift dep.</td>
<td>12.73 ± 0.38</td>
<td>44.92 ± 3.02</td>
</tr>
<tr>
<td>5. Cluster - True</td>
<td>13.91 ± 0.46</td>
<td>46.69 ± 2.19</td>
</tr>
<tr>
<td>7. Hours week before</td>
<td>9.28 ± 0.91</td>
<td>34.12 ± 3.04</td>
</tr>
<tr>
<td>8. AR(2)</td>
<td>7.67 ± 0.55</td>
<td>33.26 ± 3.42</td>
</tr>
<tr>
<td>9. TS - cluster</td>
<td>7.68 ± 0.44</td>
<td>28.97 ± 1.91</td>
</tr>
</tbody>
</table>

Table 6.1: Pinball loss ($\tau = 0.344$) and RMSE for the stopwatch method (baseline), scaling and shift per department type, regression on clusters variables and time series models (TS).

The results show that the autoregressive model (exp. 8) has an advantage over the basic time series model (exp. 7), regarding DM goal 1, as it has a lower pinball loss. The time series model combined with regression variables (exp. 9) has similar performance as the AR(2) model regarding DM goal 1 and therefor also outperforms the scaling method (exp.3b). However it does look like a better fit for DM goal 2 than the AR(2) model.
6.2 Conclusion

Time series models are particularly useful for our problem due to high correlation between budget hours in following weeks. The results show that our time series models outperform the scaling and regression models in both loss metrics.

The time series model augmented with a forecast shows the best result, decreasing both loss metrics by at least 50%. We think this is because the forecast already includes seasonal and trend information. We think the model can be further improved when explicitly add seasonality into the model.

Next to the advantage of better performance, this model also eliminates the need of a time-study, as it learns from the data available. Therefor it can scale easily over multiple shops and different retail formulas.
We initially started this research with two goals:

1. Develop methods that could operate without the need of a time-study and at the same time provide better forecasts of the needed budget hours, at department level, than the baseline.

2. Investigate if these models can also provide predictions on the cluster level.

We discussed three different types of methods to approach our goals. Each having its own set of advantages and disadvantages, a summary of each method is shown in table 7.1. The methods we discussed are roughly divided into three groups:

1. Scale and Shift
2. Regression models
3. Time series models

**Scale and Shift**

We find that scaling and shifting is an easy method for boosting performance of the current forecast (DM goal 1). It can also predict the expected amount of working hours, giving the department manager a new view on his/her department (DM goal 2).

However the method is not scalable as it still needs the forecast from the stopwatch method to compute its forecast, this makes it still an expensive method. The scaling method is, under the assumption that all tasks within a department have the same bias, usable to make predictions on cluster level.
<table>
<thead>
<tr>
<th>Method</th>
<th>Pb Loss</th>
<th>RMSE</th>
<th>Scalable</th>
<th>Cluster level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stopwatch</td>
<td>15.42</td>
<td>62.41</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Scale and Shift</td>
<td>12.73</td>
<td>44.92</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Linear Regression models</td>
<td>13.99</td>
<td>46.45</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Time Series</td>
<td>7.68</td>
<td>28.97</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 7.1:

**Regression Models**
The regression models tested have similar, but slightly worse, performance compared to the scaling method. This is still better than the performance of the stopwatch method for both DM goals. We find that for the regression models the variables on cluster level give an increase in performance compared to the aggregation of these variables.

The regression models show similar performance when trained on the true data or the forecast data, this implies that the forecast is already accurate. Increasing performance of the forecast would not significantly benefit the regression models.

The big advantage of regression models over the scaling method is that we do not longer require the stopwatch estimates for tasks, to produce our labor forecast. Thus making this method scalable over multiple retail formulas.

Unfortunately we could not make forecasts for cluster hours with the regression models. We think because the variables within clusters, but also between clusters, are too much correlated and that we did not have enough data to recover them.

**Time Series**
Using time series for the prediction of working hours significantly improved prediction performance for both DM goals. Reducing the error for both the Pinball loss and RMSE by at least 50% relative to the stopwatch method, making it our best model for predictions.

The time series model benefits from the addition of the forecast, we think this is because the forecast includes seasonal and trend information. We think the model can be further approved when explicitly add seasonality into the model, but this required unfortunately more training data than we had available.

Another big advantage of the time series models is that these does not require the time study to make a forecast. This makes the time series models
scalable over different retail formulas.

**DISCUSSION**

We think several steps can be taken to improve on our results. First of all we had a relatively small data sets in terms of weeks. Therefore our results are only applicable to those weeks in the test set. A test data set over an one year period should be used to see if the results generalize to other weeks.

A larger data set would also increase the estimation of parameters, for the time series model we expect this results in an increase in performance. For the regression models we expect that a larger training set could possibly counter the high correlation between the variables. This would allow to make predictions of the weekly amount of labor hours required in clusters.

Secondly, we did not include all departments in our models, as some had too few observations or outsourced their work. This has consequences for the employability of the model. Further research should find ways to incorporate these into the models.

Lastly, we did only look at two DM goals, further research should investigate the relation between increasing the workforce and an increase in profit.
Acknowledgements

First, I would like to thank Info Support for providing me the opportunity to do this research and for their excellent guidance during the process. Specifically, I would like to thank Maarten van Duren en Hylke Peek for their feedback provided on several versions of this research.

Furthermore I would like to thank both my supervisors at the Radboud University, Elena Marchiori and Twan van Laarhoven. Whom guided me through the process of completing this research. Thank you.

Finally, I would like to thank my friends whom kept me sane during the process and give special thanks to my girlfriend Iris who was there for me from the start until the end.
Bibliography


