MASTER'S THESIS | LUND UNIVERSITY 2017

Shopping list generation with machine learning

Daniel Tykesson

Department of Computer Science
Faculty of Engineering LTH

ISSN 1650-2884
LU-CS-EX 2017-31

Shopping list generation with machine learning

Daniel Tykesson
bas11dty@student.lu.se

November 22, 2017

Master's thesis work carried out at
the Department of Computer Science, Lund University.

Supervisors: Milosz Gruzka, milosz@listmi.com
Pierre Nugues, Pierre.Nugues@cs.lth.se

Examiner: Jacek Malec, Jacek.Malec@cs.lth.se

Abstract

When households do their grocery shopping some sort of shopping list is used to make the shopping easier. The lists contain the groceries that are intended for purchase. These lists can be boring to make and is also not free from errors, so an automated way to generate these lists would be practical. This Master's thesis aims to generate these shopping lists with data from past grocery receipts by predicting future receipts. We classified the groceries on the receipts into categories that are organized into two layers of categories, 209 subcategories and 17 main categories. These categories are modeled as time series with an indicator variable that models purchase/no purchase in the category. This indicator variable is estimated by using linear support vector machines in combination with an intensity expectation. The quantity of the groceries uses a Gaussian field as a model and is estimated with ordinary kriging. The data contains 15,969 groceries, from 1,230 receipts and 34 households. The quantity of a grocery on a receipt is measured by using the price paid for the item.

Keywords: Machine learning, consumer prediction, consumer behavior, support vector machine, ordinary kriging

Acknowledgements

I would like to thank my supervisors Milosz Gruzka and Pierre Nugues for providing help when it was asked for. Jacek Malec for examining this thesis, Steve Larkin for helping me to access data in the database and finally Lisa Bryer for allowing me to do this thesis.

Contents

Chapter 1
Introduction

Grocery shopping can easily become a complicated process for many households with many of the problems stemming from a lack of shopping lists. This can be caused by a shopper going straight to the store from their workplace and have not checked what groceries they have at home. It would be very convenient if there existed an automatic way to generate these shopping lists and that is what this Master's thesis aims to provide by using historical data.

The proposed solution is an algorithm that attempts to estimate if a certain product will be bought and then its quantity. It does not predict when the next time a household will go shopping, only what they will shop. The algorithm can be considered a recommendation system by the virtue that it recommends future receipts. It is evaluated on four different timescales, these timescales are days, weeks, two weeks and months. This was done to see the how precise the predictions could be.

The Master's thesis is in the field of machine learning where the developed algorithm uses established machine learning methods in order to estimate the desired parameters.The field of machine learning is a large and varied field where the theory is used to find patterns in data or try to estimate parameters from the data. The parts of machine learning I have worked on in this thesis include classification and regression. Classification can be summarized as generating discrete labels as responses from input data. Examples of this would be predicting if a purchase of an item will be made or not (Duro et al., 2012).

1.1 Previous work

Recommendation system for predicting purchase patterns has been used before in Lu (2014). Her method used discriminative models to find short sequential patterns and the results were presented with a small number of categories. Sequential patterns can be found by using many methods among them by different discriminative methods with varying success. A review of some of these models are given in He et al. (2008).

The approach this thesis presents uses time series analysis and is inspired by hidden Markov models. Their uses for time series are detailed in Zucchini et al. (2016). The state transitions in hidden Markov models that was presented were unsatisfactory in this thesis, therefor classification was used instead. Classification for purchase patterns has been detailed for online shopping in the Master's thesis of Thorrud and Myklatun (2015).

Various regression methods were considered for the quantity predictions, among them were the ARIMA models that are detailed in Jakobsson (2015). The best model found were a multivariate Gaussian model and predictions were done based on ordinary kriging which is detailed in Wackernagel (1998).

Chapter 2
Approach

This chapter is split into two sections, method and theory. The method section contains a description of the data available and the solution, while the theory section contains the necessary theory to implement the solution and to understand why the solution works.

2.1 Method

The aim of this Master's thesis is to present an algorithm that generates shopping list by predicting future receipts of a household. It does this by searching for purchasing patterns in the previous receipts. These patterns include what groceries and their quantities that are going to be purchased within a period. The periods used are day, week, two weeks and month and all of the receipts within a period are considered one receipt. For example, on the week period, all receipts from a single week are added to a single weekly receipt and are considered to have been purchased at a single instance within the week. The time of this instance is not predicted by the algorithm. From now on the periods are referred to as timescales and a point of time on this scale is called an instance. These instances are discrete and every instance has a receipt from the data. This stems from the fact that the algorithm does not attempt to predict when someone will shop for groceries, only what they will purchase. For example on the daily timescale there might be a three day gap between instance 1 and 2 but a four day gap between instance 2 and 3.

The available data consist of 15,969 groceries on 1230 receipts from 34 households. Different stores have different methods of printing their receipts and thus have the consequence that the available information on the receipts may differ. Some receipts for example do not list how many items that is purchased of a grocery. Thus the only consistent information available was the following:

1. The name of the grocery.
2. The price paid for the grocery.
3. The date when the item was grocery.

The data was collected by manually entering these three pieces of information along with which household it came from and which receipt it is. The groceries were categorized into 209 different subcategories; each belonging to one of 17 main categories. This was done in order to make clusters of groceries for easier prediction. Otherwise the predictions would have been made on each unique grocery name, which in total there are 7,661. This is partly due to stores having different names for the same item and misspelling during data entry.

In order to generate the shopping list or rather try to predict the next receipt, the receipts need to be described mathematically. Due to the two layers of categorization and the four timescales there are eight different ways to describe the receipts. However in any of these eight representations, vectors can be used to describe the receipts. Every element in these vectors is the measured quantity for a category, be it a main or sub category. A receipt that uses subcategories is a vector in $\mathbb{R}^{209}$ and for the main categories it is a vector in $\mathbb{R}^{17}$. Instances have their own receipts, thus their own vectors and like mentioned before the instances are discrete points in time, where the household will have a future receipt. The vector elements are modeled as independent time series, one for each element and are called $y^h_{i,j}(k)$. This is the measured quantity that was purchased in the category j at instance k for timescale i and household h and the time series model is given by Eq (2.1). A category can always refer to either a sub or main category, since the algorithm is applied in the same way to both.

$$y^h_{i,j}(k) = \begin{cases} f^h_{i,j}(k) & \text{if} \quad Q^h_{i,j}(k) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

We have $f^h_{i,j}(k)$ describing the quantity from the category at instance k and $Q^h_{i,j}(k)$ is an indicator variable, describing if there will be a purchase in the category at instance k. It is given by Eq (2.2) and can take two values, 1 for purchase or 0 for no purchase. It is mainly used to remove zeros from the data in the quantity prediction. The zeros have a very specific meaning in the model, no purchase, and trying predict these with any regression method is very difficult. However, with the introduction of $Q^h_{i,j}(k)$ the problem of prediction a zero or no purchase is decoupled from the quantity prediction and becomes easier.

$$Q^h_{i,j}(k) = \begin{cases} 1 & \text{if household purchased from category} \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Predictions are thus done in two steps, starting with predicting if the household will purchase from the category, which is done by predicting the value of $Q^h_{i,j}(k)$ and then predicting the amount by estimating $f^h_{i,j}(k)$. The estimation of $Q^h_{i,j}(k)$ uses an accuracy requirement in the estimation, which is used to protect the algorithm from predicting on very random data, more on this in Section 2.1.1.

The subcategories include potato, cucumber, pork and so on. The groceries in a subcategory have to be roughly equivalent to each other for the predictions to be meaningful. The main problem with many of these subcategories is the amount of data available for them and that they might be part of a larger pattern together with other categories, which makes it impossible to estimate $Q^h_{i,j}(k)$ with any satisfying accuracy. The clearest example of this is the various meat and vegetable subcategories, which in themselves do not always have any pattern but their main categories do.

In order to predict the quantity it needs to be measured, which cannot easily be done by only utilizing the information on the receipts. This is because some groceries might be bought in packs of different sizes while the receipt only registers one purchase, for example a customer could buy oranges or other fruits in a net but it will only register as one item and not the weight of the item. The text description on the receipt might also in these instances only contain the type of fruit that the item was. Thus only the amount of money spent on the grocery gives a clue of the true quantity.

A simple way to measure the quantity is using the amount of money spent as an indicator but it also has problems, for example fluctuating prices throughout the year. In spite of these problems, the money spent was used as a measurement for the quantity on the subcategories. The main categories measure the quantity by standardizing each subcategory and adding them together. The standardization is done by taking the ratio of the amount paid and historical mean of the amount paid. For example adding data from the carrot subcategory to the vegetable main category is done in the following way. The first carrot purchase is measured as 1 vegetable unit while the following units are measured by taking the ratio between the price paid and the current mean. This was done in order to make each subcategory somewhat equal to each other.

2.1.1 Predicting purchase event $Q^h_{i,j}$

The first variable that is predicted at instance k for a given category j is the value $Q^h_{i,j}(k)$ at timescale i for household h. This is done by using a model, $\hat{Q}^h_{i,j}(k)$ and with this model predict the next value. In order to avoid notation bloat $Q^h_{i,j}(k)$ is refereed to as $Q_i(k)$ and all categories and household use the same model. Likewise $\hat{Q}^h_{i,j}(k)$ is referred to as $\hat{Q}_i(k)$. This model is given by Eq (2.3).

$$\hat{Q}_i(k) = \begin{cases} C_i(z_i(k)) & \text{if} \quad R_i(k)P_i(k) \geq \alpha \\ L_i(k) & \text{otherwise} \end{cases} \tag{2.3}$$

This model has two main components, the classifier C_i and the extrapolation function L_i. In the next two parts these two and their models will be explained, starting with the classifier.

Modeling the classifier C_i

The classifier $C_i(z_i(k))$ is a binary classification function generated by a support vector machine given by Eq (2.4) with $z_i(k)$ as features at purchase period k for timescale i. These features are a vector of varying size depending which timescale i is used. The classifier will be explained in more detail shortly. We have $R_i(k)$, $P_i(k)$ as the recall and precision, respectively, of the classifier C_i past predictions and not on the training set, with α as a hyperparameter. This hyperparameter sets a minimum for both recall and precision of the classifier and its value is chosen by the user of the algorithm. Recall is defined as the ratio between the number of correct positive (purchases) classifications and the total number of positive responses. The precision is defined as the ratio between the correct positive classification and the total number of positive classifications. Recall can be viewed as a number to gauge how many of the purchases the classifier finds and the precision as how often the classifier is correct when making positive classifications.

$$C_i(z_i(k)) = \begin{cases} 1 & \text{if} \quad \boldsymbol{w}_i \cdot z_i(k) + \beta_i \geq 0 \\ 0 & \text{if} \quad \boldsymbol{w}_i \cdot z_i(k) + \beta_i < 0 \end{cases} \tag{2.4}$$

The condition $R_i(k)P_i(k) \geq \alpha$ is used in order to avoid using bad classifiers. The α parameter sets a minimum value for both the recall and precision due the fact that they are only able to take on values between 0 and 1. Now the classifier C_i given by Eq (2.4) will be explained more.

Here $\boldsymbol{w}_i$ is the inclination coefficients to the hyperplane $\boldsymbol{w}_i \cdot z_i(k) + \beta_i = 0$, this means that $\boldsymbol{w}_i$ have the same dimension as $z_i(k)$ and β_i is an scalar offset. This hyperplane is the support vector machine classification method. The training set for the classifier C_i at instance k consists of the features $z_i(k_{start})$,$z_i(k_{start}+1)$,...$z_i(k-1)$ with the responses $Q_i(k_{start})$, $Q_i(k_{start}+1)$,...$Q_i(k-1)$. This means that the training set grows in size and that the evaluation of the classifier is done by studying the historical accuracy of its prediction at instance k, trained with data up to instance $k-1$ for all $k > k_{start}$. Here k_{start} is the first

instance were features are retrievable from the data. Some features require a previous purchase and can thus not be retrieved from by only having one receipt. For more information about the training see Section 2.2.1.

The features like mentioned before are different between the timescales and are listed at the end on this section in the four Tables 2.1, 2.2, 2.3 and 2.4. The features were selected by picking a combination of features from a set of features that provided the best average precision for the classifier across all households and categories for each timescale. This was done by first picking the feature that gave the best precision, then adding another feature from the feature set that improved the precision the most. Additional features were added as long as they improved the precision and when the precision did not improve the feature selection was completed. The data used to select these features were from a set of households roughly making up 30% of the data; this data is never used again thus not affecting anything in the result section, outside from the choice of features.

Modeling the extrapolation function L_i

If $R_iP_i(k) < \alpha$ then the classifier is deemed too random to be used for timescale i and a higher timescale might be required in order to smooth the data and make it predictable. An example of the smoothing can be seen in Figure 2.1.

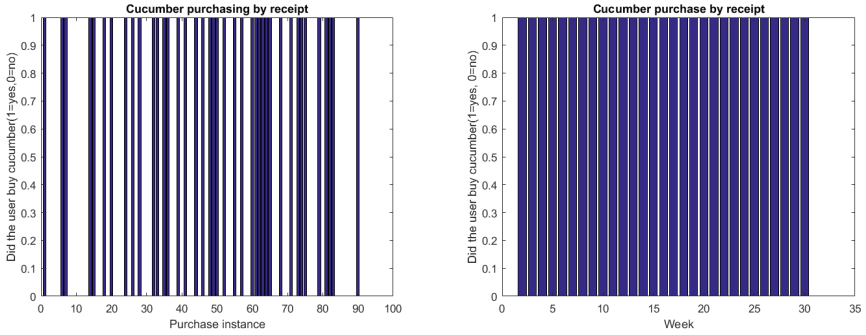

Figure 2.1: The $Q_i(k)$ for category cucumber in different timescales, left is i =days while the right is i =weeks.

A higher timescale is one with a longer purchase period, months>two weeks>weeks>days, the smoothed data can be used for prediction by using $L_i(k)$ to extrapolate from a higher to a lower timescale, and can be seen in Eq (2.5).

$$L_i(k) = \begin{cases} 1 & \text{if } C_l(z_l(g(k))) = 1 \text{ and } R_l(g(k))P_l(g(k)) \geq \alpha \text{ and } t \geq \lambda \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

Here l is the higher timescale and has to be higher than the timescale i, C_l is the binary classifier given by Eq (2.4). Here $g(k)$ is used to map instances between timescale, for example g could be a map from days to weeks. With $R_l(g(k))$,$P_l(g(k))$ as the classifiers historical recall and precision respectively at instance $g(k)$. Here t is the time since the last purchase from this category and λ is the average time between purchases in this category.

The l is chosen as the lowest timescale that has a classifier which satisfies $R_l(k)P_l(k) \geq \alpha$, if there exist no such l then $L_i(k) = 0$. The following four Tables 2.1 ,2.2, 2.3 and 2.4 describe the feature vectors $z^h_{i,j}$ for all timescales i.

Features for timescale: Day
Weekday of purchase instance k (1 for Monday, 7 for Sunday)
The measured quantity of the category on instance $k-1$ (Any positive value)
If there has been a purchase of this category this week (1 or 0)

Table 2.1: The features for timescale day

Features for timescale: Week
How many days ago was the last purchase from this category j (Any positive value)
The quantity measured of the category on instance $k-1$ (Any positive value)
The quantity measured of the category on previous purchase (Any positive value)

Table 2.2: The features for timescale week

Features for timescale: Two Week
The measured quantity of the category on instance $k-1$ (Any positive value)
If there has been a purchase of this category this month (1 or 0)

Table 2.3: The features for timescale two weeks

Features for timescale: Month
How much was spent of the category on instance $k-1$ (Any positive value)

Table 2.4: The features for timescale month

2.1.2 Predicting purchase amount $f_{i,j}^h$

The last part of the algorithm handles the quantity that is predicted for a category. A model is required for $f_{i,j}^h(k)$ where h is the household, j is the category and i is the timescale. These notations will not be showed any further in this section in interest of avoiding notation bloat. It should then be noted that every $f_{i,j}^h(k)$ has its own parameters that need to be estimated, $f_{i,j}^h(k)$ is now referred to simply as $f(k)$. I used a Gaussian stochastic field as a model as can be seen in Eq (2.6), (2.7) and (2.8), and the prediction is calculated with a method called ordinary kriging. It should be noted here that all data used for estimations here are the purchases due to the use of the indicator variable $Q_{i,j}^h$ in Eq (2.1).

$$\boldsymbol{Y} \in N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2.6}$$

Here $\boldsymbol{Y}$ is a stochastic field containing multiple quantities $f(k)$ at different instances k. These stochastic variables in the field $\boldsymbol{Y}$ have an input vector $\boldsymbol{x}_i$ attached to them. This input vector is the coordinate in the field of element i in $\boldsymbol{Y}$. The constant mean of the field is $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ is a covariance matrix that describes the dependency of the quantities in the field, whose elements $\Sigma_{i,j}$ is given by (2.7) and (2.8).

$$\Sigma_{i,j} = \begin{cases} r(\bar{0}) + \sigma_\epsilon^2 & \text{if } i = j \\ r(\boldsymbol{x}_i - \boldsymbol{x}_j) & \text{otherwise} \end{cases} \tag{2.7}$$

$$r(\boldsymbol{h}) = \sigma^2 exp(-\frac{\boldsymbol{h}^T\boldsymbol{h}}{2l^2}) \tag{2.8}$$

We have σ^2 as the output variance and l is called the length scale. The output variance can be interpreted as the amplitude of the covariance, while the length scale is how fast the covariance decay between quantities in the field. There is also σ_ϵ^2, which is the *nugget variance* and is used to model unexpected spikes in the data without affecting the covariance. Under the assumption that Eq (2.6), (2.7), and (2.8), can correctly model the data, ordinary kriging can be used to calculate the prediction. The ordinary kriging method calculates the prediction by using the conditional expectation $E(f_p|\boldsymbol{Y}_N)$. The expectation is given by (2.9), where f_p is the desired quantity that is predicted and $\boldsymbol{Y}_N$ is a vector containing all the previous known quantities. For more information about how ordinary kriging is used for the expectation and the parameter estimations, see Section 2.2.2.

$$E(f_p|\boldsymbol{Y_N}) = \hat{\mu} + \Sigma_{p,N}\Sigma_{N,N}^{-1}(\boldsymbol{Y}_N - \hat{\boldsymbol{\mu}}) \tag{2.9}$$

Here we have $\hat{\mu}$ as the estimation of μ, $\hat{\boldsymbol{\mu}}$ is vector with the estimated constant mean $\hat{\mu}$ as its values and it has the same dimension as $\boldsymbol{Y}_N$. The vector $\Sigma_{p,N}$ is a row vector whose elements are given by Eq (2.7) and (2.8). It contains the estimated covariances between the predicted value and the previously known values and has the same number of elements as $\boldsymbol{Y}_N$. The matrix $\boldsymbol{\Sigma}_{N,N}$ is also given by Eq(2.7) and (2.8). It is the estimated covariance between the observed quantities in the field and is a square matrix with the same number of columns as $\Sigma_{p,N}$.

The input vectors $\boldsymbol{x_i}$ are different between the timescales and they are listed in Tables 2.5, 2.6, 2.7 and 2.8. The inputs were chosen by minimizing the average root mean square error with ordinary kriging as a solution. The amount data used for selection of input vector is roughly 30%, like in the previous section, see Section 2.1.1.

Input vector for timescale: Day
Weekday of purchase instance k (1 for Monday, 7 for Sunday)
Value of the measure of the category on instance $k-1$ (Any positive value)

Table 2.5: The input vector for timescale day

Input vector for timescale: Week
Weekday of purchase instance $k-1$ (1 for Monday, 7 for Sunday)
Value of the measure of the category on instance $k-1$ (Any positive value)
Which number of the week in the current month it is(An integer between 1 and 6)

Table 2.6: The input vector for timescale week

Input vector for timescale: Two Week
How many times the item was purchased at instance $k-1$ (Any positive whole number
How much was measured of the category on instance $k-1$ (Any positive value)
Which number of the week in the current month it is (An integer between 1 and 6)

Table 2.7: The input vector for timescale two week

Input vector for timescale: Month
How much was measured of the category on instance $k-1$ (Any positive value)

Table 2.8: The input vector for timescale month

There are in total 30,736 estimated functions, each having potentially vastly different variances. In order to make the results digestible, three averaged ratios of root mean square errors are presented. The first ratio is $\frac{RMSE_{krig}}{RMSE_{mean}}$ and it shows how the ordinary kriging solution compares to the actual mean as the solution. The second ratio is $\frac{RMSE_{hmean}}{RMSE_{mean}}$ which shows the how the historical mean compares to the actual mean as a solution. Lastly $\frac{RMSE_{krig}}{RMSE_{hmean}}$, shows how the ordinary kriging solution compares to the historical mean as a solution. These three ratios are taken for each category and household and then averaged. The definitions of these root mean square errors can be seen in Eq (2.10), (2.11) and (2.12).

$$RMSE_{krig} = \sqrt{\frac{\sum (E(f(k)|\mathbf{Y}_N) - f(k))^2}{n}} \tag{2.10}$$

$$RMSE_{hmean} = \sqrt{\frac{\sum (\hat{\mu}(k) - f(k))^2}{n}} \tag{2.11}$$

$$RMSE_{mean} = \sqrt{\frac{\sum (\mu - f(k))^2}{n}} \tag{2.12}$$

Here n is the number of data points that is predicted, μ is the actual mean and $\hat{\mu}(k)$ is the historical estimation of μ.

2.2 Theory

In this section the necessary theory is presented for implementation of the algorithm. This includes the classification algorithm support vector machine and the derivation of the conditional expectation Eq (2.9), together with the methods used to estimate the parameters.

2.2.1 Support Vector Machine

The general idea of classification is to categorize data into K discrete classes by utilizing an input vector $z \in \mathbb{R}^D$ which is called a feature vector, these vectors belongs to a feature space. Every feature vector has a corresponding response variable y, which is the class of the feature vector. In this feature space, every feature vector belongs to a decision region, where every point inside a region belongs to the same class, see Bishop (2006) on page 179. These regions are separated by decision boundaries and the task of a classification algorithm is to model and estimate these boundaries. In this thesis, binary classification was used to determine if a category of groceries should be purchased for a given point in time. The training set for the classification grows in time, due to the problem having a time component. Binary classification with a hyperplane as a decision boundary is illustrated in Figure 2.2.

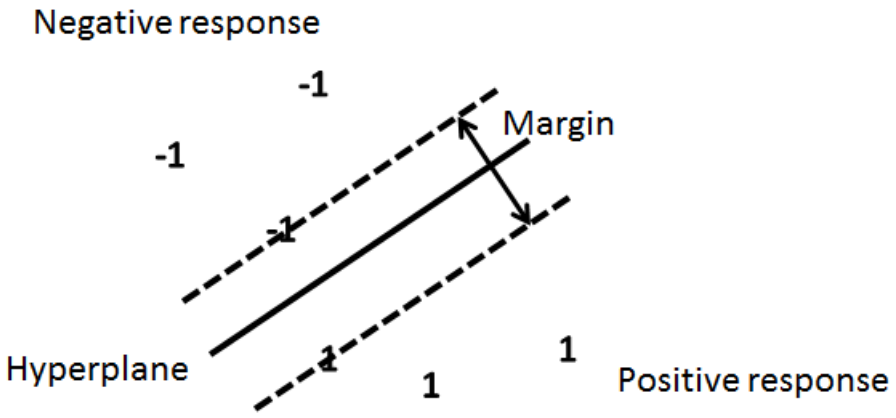

Figure 2.2: This is a illustration of the hyperplane and margin in the feature space.

Linear support vector machines was used as classification algorithm, which models the decision boundary as a linear hyperplane, see Liang et al. (2011) on page 20. The hyperplane is described by a polynomial $P(z)$, which is given by Eq (2.13).

$$P(z) = z^T w + \beta \tag{2.13}$$

The points on the hyperplane can be found by solving $P(\boldsymbol{z}) = 0$. We have $\boldsymbol{w} \in \mathbb{R}^D$ as inclination coefficients with D as the dimension of the feature vector $\boldsymbol{z}$ and $\beta \in \mathbb{R}$ as a scalar offset. In order to estimate these parameters the constraints and conditions for the training need to be defined. The conditions were chosen under the assumption that the data is nonseperable. This means that Eq (2.13) cannot model the decision boundary perfectly and missclassifications are allowed on the training data. The conditions and constraints used for this training were described by Cortes and Vapnik (1995) and can be seen in Eq (2.14), (2.15) and (2.16).

$$\min_{\beta,\boldsymbol{w},\epsilon}(\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + A\sum_j \epsilon_j) \tag{2.14}$$

$$y_j P(\boldsymbol{z}_j) \geq 1 - \epsilon_j \tag{2.15}$$

$$\epsilon_j \geq 0 \tag{2.16}$$

Here A is called the box constraint, which is a penalty variable and it was set to 1. This was because there was no noticible improvement found by changing it. We have ϵ_j, and it is called the slack variable and it allows for missclassification. The response variable y_j can only take on two values since it is binary classification, -1 and 1. This was solved in Matlab using a built-in method called sequential minimal optimization, which can be found in Platt (1998). With the estimated parameters for $P(\boldsymbol{z})$, the binary classifier $C(\boldsymbol{z})$ is given by Eq (2.17).

$$C(\boldsymbol{z}) = \begin{cases} c_1 \text{ if } P(\boldsymbol{z}) \geq 0 \\ c_2 \text{ if } P(\boldsymbol{z}) < 0 \end{cases} \tag{2.17}$$

Here c_1 and c_2 are the classes of the data and in this thesis they are 1 for purchase and 0 for no purchase.

2.2.2 Ordinary Kriging

Oridnary kriging is a statistical regression method and was used to predict the purchased quantity of a category of groceries and is heavily used in geostatistics on spatial data, Baafi et al. (1997). It is used on data that can be modeled by a Gaussian process according to Cressie (1993), and the quantity of a purchased grocery is modeled as Gaussian Random field. Predictor variables $\boldsymbol{x} \in \mathbb{R}^D$ are locations in the field, which thus have D dimensions and every point in the field has a response. The response is simply the quantity of the purchased grocery. In geostatistics $\boldsymbol{x}$ is a location in space but in this thesis the predictor variables use information from the receipts, such as the last purchase amount.

A general definition of a Gaussian field found in Bishop (2006) on page 309. It is a set of stochastic variables y_i (quantity) at the points $x_1, x_2, ...x_N$(predictor variables) that jointly have a Gaussian distribution. By sorting these variables in a stochastic vector, then this vector $\boldsymbol{Y}$ belongs to the following distribution in Eq (2.18).

$$\boldsymbol{Y} \in N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2.18}$$

Here $\boldsymbol{\mu}$ is a vector containing the expectations of y_i, which are the observed values and $\boldsymbol{\Sigma}$ is the fields covariance matrix. In ordinary kriging $\boldsymbol{\mu}$ is a vector with a constant value μ, see Gelfand et al. (2010) in chapter 1 page 8. For modeling $\boldsymbol{\Sigma}$ see Section 2.1.2.

In the field one of these stochastic variables is unknown, the future purchased quantity and needs to be predicted. The method used for predictions with ordinary kriging is described in Gelfand et al. (2010) on chapter 2 page 26 and will be showed in this section. The predictions can be done by splitting the field $\boldsymbol{Y}$ into a vector with known variables $\boldsymbol{Y}_N$ and a scalar y_p, the one in need of prediction. This would require splitting the covariance matrix $\boldsymbol{\Sigma}$ as well, these splits can be seen in Eq (2.19) and (2.20).

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Y}_N \\ y_p \end{bmatrix} \tag{2.19}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{N,N} & \boldsymbol{\Sigma}_{N,p} \\ \boldsymbol{\Sigma}_{p,N} & \boldsymbol{\Sigma}_{p,p} \end{bmatrix} \tag{2.20}$$

Utilizing Eq (2.19) and (2.20) in Eq (2.18) we get Eq (2.21).

$$\begin{bmatrix} \boldsymbol{Y}_N \\ y_p \end{bmatrix} \in N\left(\begin{bmatrix} \boldsymbol{\mu} \\ \mu \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{N,N} & \boldsymbol{\Sigma}_{N,p} \\ \boldsymbol{\Sigma}_{p_N} & \boldsymbol{\Sigma}_{p,p} \end{bmatrix}\right) \tag{2.21}$$

The prediction of y_p is called $\hat{y}_p$ and is given by the conditional expectation in Eq (2.22).

$$\hat{y}_p = E(y_p|\boldsymbol{Y}_N) \tag{2.22}$$

Since the multivariate distribution in Eq (2.21) is Gaussian, there are well known formulas for calculating conditional expectations. This is done by utilizing the conditional distribution of $y_p|Y_N$. This can be seen in Gelfand et al. (2010) and is given by Eq (2.23), (2.24) and (2.25).

$$p(y_p|Y_N) \in N(\bar{\mu}, \bar{\boldsymbol{\Sigma}}) \tag{2.23}$$

$$\bar{\mu} = \mu + \Sigma_{p,N}\Sigma_{N,N}^{-1}(Y_N - \boldsymbol{\mu}) \tag{2.24}$$

$$\bar{\boldsymbol{\Sigma}} = \Sigma_{N,N} - \Sigma_{N,p}\Sigma p, p^{-}1\Sigma_{p,N} \tag{2.25}$$

Thus we have the sought expectation, $\bar{\mu}$ but some parameters are in need of estimation. The first parameter that is in need of estimation is μ which is done with the arithmetic mean and is unbiased, according to Lindgren et al. (2013) on page 42. We also have to find the covariance matrices $\Sigma_{p,N}$ and $\Sigma_{N,N}$, which are assumed to be functions of a set of parameters $\boldsymbol{\theta}$. In order to find $\boldsymbol{\theta}$ the maximum likelihood method was used.

The maximum likelihood method estimates parameters by choosing the parameters that maximize the probability density function (2.26) over the known values $\boldsymbol{Y}_N$ according to Jakobsson (2015) on page 173. The probability density function is a multivariate Gaussian distribution since the observations $\boldsymbol{Y}_N$ comes from that type of distribution.

$$f(\boldsymbol{Y}_N, \boldsymbol{\theta}, \mu) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_{N,N}(\boldsymbol{\theta}))|^{\frac{1}{2}}} exp(-\frac{1}{2}(\boldsymbol{Y}_n - \boldsymbol{\mu})^T\boldsymbol{\Sigma}_{\boldsymbol{N,N}}^{\boldsymbol{-1}}(\boldsymbol{\theta})(\boldsymbol{Y}_N - \boldsymbol{\mu})) \tag{2.26}$$

Utilizing Eq (2.26) the parameters are now given by Eq (2.27).

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{Y}_N|\hat{\mu}, \boldsymbol{\theta}) \tag{2.27}$$

Here $\hat{\mu}$ is the estimate of μ. I used an algorithm called Quasi Newton method in Matlab for solving Eq (2.27), which can be read about in Nocedal and Wright (2006).

Chapter 3
Evaluation

This chapter is split into two parts, results and discussion. In the first part the results are presented with no commentary, in the second part they are discussed and th chapter ends with my overall thoughts on this Master's thesis.

3.1 Results

The results are all averages across categories and household in order to make the results presentable and digestible since there are in total 30,736 categories predicted. The results are presented in three parts, predicting $Q^h_{i,j}$, predicting $f^h_{i,j}$, and finally combining them. It should be noted that this chapter uses the full notation again, where j is a category, i is a timescale, k is an instance of this timescale and lastly h is a household.

3.1.1 Evaluating $\hat{Q}^h_{i,j}$

In this part of the result section the results are provided for $\hat{Q}^h_{i,j}$, which is given by equation (2.3) separately for subcategories and main categories. However in interest of readability it will be shown here again but it uses the full notation.

$$\hat{Q}^h_{i,j}(k) = \begin{cases} C^h_{i,j}(z^h_{i,j}(k)) & \text{if} \quad R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha \\ L^h_{i,j}(k) & \text{otherwise} \end{cases} \tag{2.3}$$

These results for $\hat{Q}^h_{i,j}$ will be presented in three parts, the first part is for $C^h_{i,j}$, the second if for $L^h_{i,j}$ and finally the third part is for $\hat{Q}^h_{i,j}$, and the results are presented by comparing them in various ways with $Q^h_{i,j}$ on eligible data. The results will all be presented for three α values, in order to show the effects of this hyperparameter on the results.

Data for a category (both main and sub) is considered eligible with predictor $\bar{S}^h_{i,j}$ (where $\bar{S}^h_{i,j}$ is either $C^h_{i,j}$, $L^h_{i,j}$ or $\hat{Q}^h_{i,j}$) if conditions (3.1) and (3.2) is satisfied.

$$\sum_{b=1}^{k} Q^h_{i,j}(b) \geq 3 \tag{3.1}$$

$$\sum_{b=k+1}^{N} Q^h_{i,j}(b) \geq 1 \quad \text{or} \quad \sum_{b=k+1}^{N} \bar{S}^h_{i,j}(b) \geq 1 \tag{3.2}$$

The first condition is used to make sure that all features used to train the support vector machines can be calculated since some features requires a previous purchase, an example of such an feature is days since last purchase in a category. The first condition also removes empty subcategories since the majority of them for each household are empty, and thus gives a more representative result. The second condition is used to avoid inflating the accuracy by removing a category if all future predictions are true negative. The reason why this definition is used is explained in the discussion section.

Evaluating $C^h_{i,j}$ on subcategories

The first results for $C^h_{i,j}(z^h_{i,j}(k))$ is the average percentage of how much of a households eligible data satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$, for all four i. All of these results were gathered on the subcategories. This can be seen in Table 3.1. In order to easier relate to these numbers the number of average categories that satisfies the conditions is shown for all household and the three households with the largest data sets. This can be seen in Tables 3.2 and 3.3. The average recall, precision and accuracy by using $C^h_{i,j}(z^h_{i,j}(k))$ as a predictor on the eligible data that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$, for three different α values, can be seen in Tables 3.4, 3.5 and 3.6.

Timescale	$\alpha = 0.25$	$\alpha = 0.4$	$\alpha = 0.6$
Day	4.9%	1.33%	0.113%
Week	20.69%	10.97%	3.03%
Two weeks	46.68%	30.03%	11.77%
Month	59.49%	43.03%	19.17%

Table 3.1: The average percentage of eligible data that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$, for the four different timescales i on the subcategories

Timescale	$\alpha = 0.25$	$\alpha = 0.4$	$\alpha = 0.6$
Day	3.14	1.96	1.13
Week	7.9384	7.6034	3.4268
Two weeks	18.6206	14.6594	8.8897
Month	27.3325	19.3125	16.0612

Table 3.2: The average number of subcategories that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$, for the four different timescales i on the subcategories

Timescale	$\alpha = 0.25$	$\alpha = 0.4$	$\alpha = 0.6$
Day	6.60	3.2425	0.8730
Week	18.6795	10.8946	3.4268
Two weeks	38.9302	26.7594	11.5195
Month	48.3873	34.9604	19.9983

Table 3.3: The average number of subcategories for the three largest households that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$, for the four different timescales i on the subcategories

Timescale	Recall	Precision	Accuracy
Day	0.8260	0.7119	0.7570
Week	0.8300	0.7353	0.7474
Two weeks	0.9064	0.7644	0.7568
Month	0.9531	0.8155	0.7934

Table 3.4: The precision, recall and accuracy over the eligible data that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$ for $\alpha = 0.25$ on the subcategories

Timescale	Recall	Precision	Accuracy
Day	0.9006	0.8274	0.8153
Week	0.8836	0.7551	0.7632
Two weeks	0.9447	0.7892	0.7861
Month	0.9738	0.8509	0.8359

Table 3.5: The precision, recall and accuracy over the eligible data that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$ for $\alpha = 0.4$ on the subcategories

Timescale	Recall	Precision	Accuracy
Day	0.9655	0.9180	0.8986
Week	0.9251	0.8059	0.7940
Two weeks	0.9658	0.8541	0.8447
Month	0.9987	0.9128	0.9121

Table 3.6: The precision, recall and accuracy over the eligible data that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$ for $\alpha = 0.6$ on the subcategories

Evaluating $C^h_{i,j}$ on main categories

The result for $C^h_{i,j}(z_i(k))$ on the main categories are presented identically to the results for the subcategories. The results can be seen in Tables 3.7, 3.8, 3.9, 3.10, 3.11 and 3.12.

Timescale	$\alpha = 0.25$	$\alpha = 0.4$	$\alpha = 0.6$
Day	45.01%	32.84%	13.38%
Week	68.34%	58.91%	37.92%
Two weeks	79.34%	68.08%	51.33%
Month	87.36%	79.05%	59.26%

Table 3.7: The average percentage of eligible data that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$, for the four different timescales i on the main categories

Timescale	$\alpha = 0.25$	$\alpha = 0.4$	$\alpha = 0.6$
Day	4.7486	3.8096	2.7159
Week	6.2427	5.5917	4.2323
Two weeks	7.2723	7.6162	8.5277
Month	8.4896	8.6442	9.7942

Table 3.8: The average number of main categories that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$, for the four different timescales i on the main categories

Timescale	$\alpha = 0.25$	$\alpha = 0.4$	$\alpha = 0.6$
Day	7.7215	5.8243	3.1055
Week	11.6682	10.5201	7.1551
Two weeks	13.0451	12.6373	10.8146
Month	13.4281	12.5266	11.7256

Table 3.9: The average number of main categories for the three largest households that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$, for the four different timescales i on the main categories

Timescale	Recall	Precision	Accuracy
Day	0.9031	0.7451	0.7143
Week	0.9141	0.8623	0.8390
Two weeks	0.9302	0.9092	0.8658
Month	0.9850	0.9521	0.9400

Table 3.10: The precision, recall and accuracy over the eligible data for $\alpha = 0.25$ on the main categories

Timescale	Recall	Precision	Accuracy
Day	0.9481	0.7633	0.7423
Week	0.9239	0.8684	0.8467
Two weeks	0.9484	0.9258	0.8910
Month	0.9918	0.9633	0.9564

Table 3.11: The precision, recall and accuracy over the eligible data for $\alpha = 0.4$ on the main categories

Timescale	Recall	Precision	Accuracy
Day	0.9814	0.8171	0.8080
Week	0.9519	0.8939	0.8753
Two weeks	0.9681	0.9487	0.9253
Month	1	0.9789	0.9789

Table 3.12: The precision, recall and accuracy over the eligible data for $\alpha = 0.6$ on the main categories

Evaluating $L^h_{i,j}$ for subcategories

The next part addresses the results of $L^h_{i,j}$ on the subcategories in a similar manner of how the results of $C^h_{i,j}$ was presented. The formula for $L^h_{i,j}$ is given by Eq (2.5) but is presented below with full notation as a reminder.

$$L^h_{i,j}(k) = \begin{cases} 1 & \text{if } C^h_{l,j}(z^h_{l,j}(g(k))) = 1 \text{ and } R^h_{l,j}(g(k))P^h_{l,j}(g(k)) \geq \alpha \text{ and } t \geq \lambda \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

The next three Tables 3.13, 3.14, 3.15 cover the average precision, recall and accuracy of $L^h_{i,j}$ over the eligible data for all subcategories and households for three different α values. Eligible data is all data that satisfies Eq (3.1) and (3.2).

Timescale	Recall	Precision	Accuracy
Day	0.3609	0.2446	0.6941
Week	0.6338	0.5128	0.6293
Two weeks	0.7655	0.6418	0.6216

Table 3.13: The precision, recall and accuracy over the eligible data for $\alpha = 0.25$ on the subcategories

Timescale	Recall	Precision	Accuracy
Day	0.3627	0.2215	0.6661
Week	0.6103	0.5042	0.6205
Two weeks	0.6723	0.6600	0.6132

Table 3.14: The precision, recall and accuracy over the eligible data for $\alpha = 0.4$ on the subcategories

Timescale	Recall	Precision	Accuracy
Day	0.3260	0.2374	0.6986
Week	0.4959	0.5419	0.6453
Two weeks	0.4742	0.7153	0.5889

Table 3.15: The precision, recall and accuracy over the eligible data for $\alpha = 0.6$ on the subcategories

Evaluating $L^h_{i,j}$ on main categories

The results on main categories are presented in the same fashion as for the subcategories in the three Tables 3.16, 3.17, 3.18.

Timescale	Recall	Precision	Accuracy
Day	0.6504	0.4718	0.5566
Week	0.6367	0.7658	0.6750
Two weeks	0.8503	0.8667	0.7858

Table 3.16: The precision, recall and accuracy over the eligible data for $\alpha = 0.25$ on the main categories

Timescale	Recall	Precision	Accuracy
Day	0.6062	0.4702	0.5569
Week	0.6000	0.7716	0.6634
Two weeks	0.7583	0.8788	0.7345

Table 3.17: The precision, recall and accuracy over the eligible data for $\alpha = 0.4$ on the main categories

Timescale	Recall	Precision	Accuracy
Day	0.5544	0.4779	0.5656
Week	0.5350	0.8020	0.6514
Two weeks	0.6068	0.9141	0.6540

Table 3.18: The precision, recall and accuracy over the eligible data for $\alpha = 0.6$ on the main categories

Evaluating $\hat{Q}^h_{i,j}$ on subcategories

Here are the results for $Q^h_{i,j}(z^h_{i,j}(k))$, which is given by Eq (2.3) on the subcategories. The results will be presented in the same manner as for $C^h_{i,j}$ and $L^h_{i,j}$ with the same definition of eligible data, see Eq (3.1) and (3.2). The results can be seen in these three Tables, 3.19, 3.20 and 3.21.

Timescale	Recall	Precision	Accuracy
Day	0.3392	0.2209	0.7719
Week	0.5437	0.4398	0.6773
Two weeks	0.7140	0.5838	0.6446
Month	0.6690	0.8026	0.6735

Table 3.19: The precision, recall and accuracy over the eligible data for $\alpha = 0.25$ on subcategories

Timescale	Recall	Precision	Accuracy
Day	0.3339	0.1994	0.7537
Week	0.5040	0.4420	0.6820
Two weeks	0.6116	0.6178	0.6574
Month	0.5331	0.8523	0.6314

Table 3.20: The precision, recall and accuracy over the eligible data for $\alpha = 0.4$ on subcategories

Timescale	Recall	Precision	Accuracy
Day	0.2461	0.2222	0.8018
Week	0.3456	0.4984	0.7185
Two weeks	0.3799	0.7044	0.6517
Month	0.2883	0.9361	0.5181

Table 3.21: The precision, recall and accuracy over the eligible data for $\alpha = 0.6$ on subcategories

Evaluating $\hat{Q}^h_{i,j}$ on main categories

Lastly for this section the results for $Q^h_{i,j}(z^h_{i,j}(k))$ on the main categories are presented. The results are presented in a similar manner as for the sub categories and can be seen in these three Tables, 3.22, 3.23 and 3.24.

Timescale	Recall	Precision	Accuracy
Day	0.7208	0.5081	0.5954
Week	0.7444	0.8031	0.7437
Two weeks	0.8522	0.8781	0.7952
Month	0.8992	0.9607	0.8743

Table 3.22: The precision, recall and accuracy over the eligible data for $\alpha = 0.25$ on main categories

Timescale	Recall	Precision	Accuracy
Day	0.7053	0.5109	0.5988
Week	0.6940	0.8069	0.7251
Two weeks	0.6940	0.8068	0.7251
Month	0.8352	0.9685	0.8246

Table 3.23: The precision, recall and accuracy over the eligible data for $\alpha = 0.4$ on main categories

Timescale	Recall	Precision	Accuracy
Day	0.6228	0.5039	0.5907
Week	0.6022	0.8220	0.6924
Two weeks	0.6644	0.9207	0.6981
Month	0.6560	0.9809	0.6740

Table 3.24: The precision, recall and accuracy over the eligible data for $\alpha = 0.6$ on main categories

3.1.2 Evaluating $f_{i,j}^{h}$

The results for the quantity prediction will be presented by comparing the three ratios of different RMSE (root mean square error). These ratios were explained in Section 2.1.2 but will be explained here again. The first ratio is $\frac{RMSE_{krig}}{RMSE_{mean}}$ and it shows how the ordinary kriging solution compares to the actual mean as the solution. The second ratio is $\frac{RMSE_{hmean}}{RMSE_{mean}}$ shows the how the historical mean compares to the actual mean as a solution. Lastly $\frac{RMSE_{krig}}{RMSE_{hmean}}$, shows how the ordinary kriging solution compares to historical mean as a solution.

These three ratios are taken for each category and household and then averaged. The results were collected under the assumption that it is known if a purchase in the category will be made or not. All of these root mean square errors can be seen in Eq (2.10), (2.11) and (2.12) but are given below as a reminder. The results are provided for sub and main categories separately and can be seen in Tables 3.25 and 3.26.

$$RMSE_{krig} = \sqrt{\frac{\sum (E(f(k)|\boldsymbol{Y}_N) - f(k))^2}{n}} \tag{2.10}$$

$$RMSE_{hmean} = \sqrt{\frac{\sum (\hat{\mu}(k) - f(k))^2}{n}} \tag{2.11}$$

$$RMSE_{mean} = \sqrt{\frac{\sum (\mu - f(k))^2}{n}} \tag{2.12}$$

Timescale	Kriging as solution	Historical mean as solution	Result ratios
Day	1.1564	1.1715	0.9871
Week	1.1122	1.1743	0.9471
Two weeks	1.0677	1.1869	0.8996
Month	1.0536	1.2267	0.8589

Table 3.25: The quantity prediction ratios of subcategories

Timescale	Kriging as solution	Historical mean as solution	Result ratios
Day	1.1036	1.1746	0.9396
Week	1.1312	1.1623	0.9732
Two weeks	1.0437	1.1349	0.9197
Month	1.0543	1.1793	0.8940

Table 3.26: The quantity prediction ratios of main categories

3.1.3 Combining $\hat{Q}^h_{i,j}$ and $\hat{f}^h_{i,j}$

In this section the final results of the algorithm will be presented and it is the prediction of (2.1) for both sub and main categories. This will be done by presenting the average RMSE for the following three cases across all categories and households. The first case has $\bar{S}^h_{i,j}(k) = 1$, for all k. Here $\bar{S}^h_{i,j}$ is the estimation of $Q^h_{i,j}$. This is done in order to compare the solution with a straight forward regression model, without any indicator variables. It should be noted that the kriging solution here is allowed to train on zeros due to the lack of indicator variable. The second case is $\bar{S}^h_{i,j} = \hat{Q}^h_{i,j}$ for three different α values and lastly the third case, $\bar{S}^h_{i,j} = Q^h_{i,j}$.

Combining $\hat{Q}^h_{i,j}$ and $\hat{f}^h_{i,j}$ on subcategories

The results for subcategories is presented for all timescales and can be seen in Tables 3.27, 3.28, 3.29, 3.30 and 3.31. As always when using $\bar{S}^h_{i,j}$ only eligible data is considered, and the conditions can be seen in Eq (3.1) and (3.2).

Timescale	Q **is estimated as** 1
Day	4.7543
Week	4.9372
Two weeks	5.2632
Month	5.9437

Table 3.27: The average RMSE of estimating $Q^h_{i,j}$ as 1 on the subcategories

Timescale	$Q^h_{i,j}$ **is estimated as** $\hat{Q}^h_{i,j}$
Day	3.3472
Week	3.967
Two weeks	4.7008
Month	5.7409

Table 3.28: The average RMSE of estimating $Q^h_{i,j}$ as $\hat{Q}^h_{i,j}$ with $\alpha = 0.25$ on the subcategories

Timescale	$Q^h_{i,j}$ **is estimated as** $\hat{Q}^h_{i,j}$
Day	3.3922
Week	3.9842
Two weeks	4.6737
Month	5.8231

Table 3.29: The average RMSE of estimating $Q^h_{i,j}$ as $\hat{Q}^h_{i,j}$ with $\alpha = 0.4$ on the subcategories

Timescale	$Q^h_{i,j}$ is estimated as $\hat{Q}^h_{i,j}$
Day	3.3365
Week	3.9436
Two weeks	4.7056
Month	6.0441

Table 3.30: The average RMSE of estimating $Q^h_{i,j}$ as $\hat{Q}^h_{i,j}$ with $\alpha = 0.6$ on the sub categories

Timescale	$Q^h_{i,j}$ is known
Day	3.2036
Week	3.8117
Two weeks	4.5597
Month	5.6266

Table 3.31: The average RMSE with $Q^h_{i,j}$ known on the sub categories

Combining $\hat{Q}^h_{i,j}$ and $\hat{f}^h_{i,j}$ on main categories

The results for main is presented for all timescales and can be seen in Tables 3.32, 3.33, 3.34, 3.35 and 3.36. As always when using $\bar{S}^h_{i,j}$ as an estimate of $Q^h_{i,j}$ only eligible data is considered, and the conditions can be seen in Eq (3.1) and (3.2).

Timescale	$Q^h_{i,j}$ is estimated as 1
Day	1.3188
Week	1.5164
Two weeks	1.7157
Month	2.3392

Table 3.32: The average RMSE of estimating $Q^h_{i,j}$ as 1 on the main categories

Timescale	$Q^h_{i,j}$ is estimated as $\hat{Q}^h_{i,j}$
Day	1.2782
Week	1.4683
Two weeks	1.7278
Month	2.3201

Table 3.33: The average RMSE of estimating $Q^h_{i,j}$ as $\hat{Q}^h_{i,j}$ with $\alpha = 0.25$ on the main categories

Timescale	$Q^h_{i,j}$ is estimated as $\hat{Q}^h_{i,j}$
Day	1.2877
Week	1.4777
Two weeks	1.7310
Month	2.3286

Table 3.34: The average RMSE of estimating $Q^h_{i,j}$ as $\hat{Q}^h_{i,j}$ with $\alpha = 0.4$

Timescale	$Q^h_{i,j}$ is estimated as $\hat{Q}^h_{i,j}$
Day	1.3028
Week	1.4899
Two weeks	1.7474
Month	2.3750

Table 3.35: The average RMSE of estimating $Q^h_{i,j}$ as $\hat{Q}^h_{i,j}$ with $\alpha = 0.6$ on the main categories

Timescale	$Q^h_{i,j}$ is known
Day	1.2675
Week	1.4533
Two weeks	1.7112
Month	2.3294

Table 3.36: The average RMSE with $Q^h_{i,j}$ known on the main categories

3.2 Discussion

In this section I will assess the results and it will be presented with similar sections as in the result section.

3.2.1 Eligible data

The term eligible data was defined as data that satisfies equations (3.1) and (3.2). These conditions were chosen to allow for the existence of certain features that require a previous purchase and to not inflate the accuracy and make the presented results more representative. The second condition removes the data when there are only true negative predictions left, which are not interesting results. This is because most subcategories are mostly empty with some purchases in the beginning, allowing for many true negative predictions.

The first condition also helps with this problem by removing all empty subcategories; all households had 209 subcategories even if no purchases have been made in them. Each included empty subcategory would highly inflate the accuracy and make it useless as a gauge for performance. The households only have data from about 42% of the subcategories and if these true negatives were included there would be 0.58 accuracy at a minimum.

3.2.2 Interpreting $\hat{C}^h_{i,j}$

Here I will discuss the amount of data covered by $C^h_{i,j}$ for different α values and why I use the condition $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$. Let us start with the condition $Rh_{i,j}(k)Ph_{i,j}(k) \geq \alpha$. When developing this algorithm I favored conservative predictions with an emphasis on the precision of my predictions. This is because it is more important that the positive guesses are correct than the negative predictions being wrong, but this has to be balanced with my predictions recall even if I favor precision.

This led me to use the condition $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$, which combines the two measurements. There is a similar measurement in machine learning for binary classifiers called F1-score, which I did not know about until later during my development. After discovering it I felt that my condition is more intuitive since α sets a minimum requirement for both the recall and precision. The percentage of data that satisfied $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$ can be seen in Table 3.1 for the subcategories and in Table 3.7 for main categories.

Generally we can see that less data is covered by a higher α and more data is covered by higher timescale. This is to be expected since a higher α is makes the condition harder to achieve and the higher timescale smooths the data and makes easier to predict on. The recall, precision and accuracy of $C^h_{i,j}$ on the data that satisfies my condition for different α values can be seen in Tables 3.4, 3.5, 3.6, 3.10, 3.11 and 3.12. The recall, precision and accuracy is quite good on this data and increases with α, which makes sense. This is to be expected due to the condition used to determine the data presented in these tables.

The amount of data that satisfies $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$ increases quite drastically by predicting on main categories instead of the subcategories, especially on the daily timescale. This reminds of the saying; I can not see the forest for the trees, with the tree being the subcategories and the main categories as the forest. The problem I have with predicting on the main categories is that they are very broad and does not require much intelligence to predict on, especially on the higher timescales.

Some of these predictions would be that the household will buy meat on this week's collection of receipts. The subcategories could probably have benefited from being concentrated a bit and perhaps a happy medium could be found between the main and subcategories.

When studying the average amount of categories that are recommended in Tables 3.2, 3.3, 3.8 and 3.9 we can see that there are still quite a few subcategories that are recommended. This is especially true on for the three largest households and low α values. It can be argued that in a practical application one does not want to recommend to many categories at once, thus using only $C^h_{i,j}$ might be suitable.

3.2.3 Interpreting $\hat{L}^h_{i,j}$

By introducing $L^h_{i,j}$, which can be seen in Eq (2.5) I attempted to increase the amount of data covered while avoiding sacrificing too much in accuracy, and utilizing the $C^h_{i,j}$ for the higher timescale. This was done because there might be a very clear pattern on let us say a monthly basis but not which week in the month, thus I attempted to use $L^h_{i,j}$ as an extrapolation from a higher to a lower timescale. The results can be seen in Tables 3.13, 3.14 and 3.15.

We can see there is a clear sacrifice in recall and precision while the accuracy is still acceptable and the results improve when a higher timescale is used. There is an interesting effect with increasing α, the precision and accuracy seems to increase at the cost of the recall. The reason why the day timescale has the highest accuracy is because it has the most zeros which is still the most prevalent prediction and this is supported with the low recall results. Overall I would say my attempt to extrapolate between timescales is useful on the week and two weeks' timescale since the recall and precision is not catastrophic and I do not think one could expect good results on extrapolation between timescales.

3.2.4 Interpreting $\hat{Q}^h_{i,j}$

When combining $C^h_{i,j}$ and $L^h_{i,j}$ in $\hat{Q}h_{i,j}$ the results can be seen in Tables 3.19, 3.20, 3.21, 3.22, 3.23 and 3.23. Generally we can see an improvement in precision for both sub and main categories, compared to only using $L^h_{i,j}$ and an increase in accuracy. Comparing this to $C^h_{i,j}$ we have a significant reduction in precision, recall and accuracy. This is expected since we cover all eligible data with $\hat{Q}^h_{i,j}$ and sometimes less than 1% with $C^h_{i,j}$. I think the trade off is worth the sacrifice since there exists α values such that precision is above 0.5 for all timescales expects timescale days. I have found that trying to predict a receipt on the daily timescale is very hard with 209 different subcategories, since most of them have very sparse data.

3.2.5 Interpreting $\hat{f}^h_{i,j}$

The results of $f^h_{i,j}$ can be seen in Tables 3.25 and 3.26 where we can see that my suggested solution outperforms the historical mean but not the actual mean. The results here were presented as ratios in order for comparison with the actual mean as a solution. Since the actually mean cannot be known at the time of prediction that results has to be considered carefully.

Why $f^h_{i,j}$ outperforms the historical mean most likely due it having an easier time adapting to changing means due to the covariance function (2.8). The previous two prices are included in $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)$ in all timescales. They are generally the largest contributor to the product $(\boldsymbol{x}(k) - \boldsymbol{x}(k'))^T(\boldsymbol{x}(k) - \boldsymbol{x}(k')$, this means if the two previous purchases are close to each other, then this product is generally small and there is a larger covariance.

This allows for local means to be more easily followed compared to using the historical mean. It still leaves a lot to be desired but this was the best method I found for predicting the quantity.

3.2.6 Interpreting $\hat{Q}^h_{i,j}$ with $\hat{f}^h_{i,j}$

The last results I presented was the using $Q^h_{i,j}$ together with $f^h_{i,j}$ in (2.1) and can be seen in Tables 3.27, 3.28, 3.29, 3.30, 3.31, 3.32, 3.33 and 3.34. We can clearly see here that estimating $Q^h_{i,j}$ with my algorithm has a net positive effect on the results, compared to relying on $f^h_{i,j}$ to predict zeros. There might be better regression models to use without an indicator variable $Q^h_{i,j}$ but I have not encountered them. When I started this Master's thesis I initially had no $Q^h_{i,j}$ and only used regression models, none of which were satisfactory.

A comparison can also be made if the actual $Q^h_{i,j}$ is known and we can see that generally there are room for improvement, especially on the sub categories. The main categories seem to have very similar average RMSE when comparing the real $Q^h_{i,j}$ to its estimation $\hat{Q}^h_{i,j}$ on the higher timescales. This can be explained by the data being very smooth and $Q^h_{i,j}(k) = 1$ for virtually all k. The model probably does not need the indicator variable on the higher timescales for the main categories. On those timescales the algorithm basically predicts if a household for example will purchase meat this month, which is overwhelmingly yes on my data.

3.2.7 Overall thoughts

My overall thoughts on this thesis are that I have presented a useful solution that could be improved upon in both steps. One glaring omission from my models is that they take no consideration of how the categories interact with each other when making the predictions, for example a household might be more likely to purchase soda when purchasing candy/snacks. I made a few attempts to model this with clustering, but found none satisfactory. I believe much can be explored by introducing dependency between the categories.

I also believe that this algorithm can be used in practical applications by introducing a maximum cap of recommended categories. With this maximum cap perhaps one could substitute $\hat{Q}^h_{i,j}$ with $C^h_{i,j}$ and only predict on categories that satisfies the condition $R^h_{i,j}(k)P^h_{i,j}(k) \geq \alpha$. One could perhaps also device a way to automatically categorize the data in a way that makes it more predictable.

Since I could not outperform the actual mean when predicting the quantity, there probably is much to be explored there, and how the number of recommended categories affect the quantity prediction. When I studied this I found that when groceries on a receipt came from many different subcategories, the amount of money spent on these subcategories was also generally higher than the average. I believe the reason for this is that when a household is having a party they have a long and varied receipt together with purchasing much

of each grocery, in order to feed their guests. The last thing I will mention is that I was unable to explore seasonal behavior due to no household having data over a three year period, and there is probably much to be explored here, such as how to model holidays and seasons.

It would have been interesting to approach this thesis in a new way, by attempting to solve the problem with clustering or unsupervised learning methods. From these a couple of basic underlying receipts could be found and one of these would have recommended instead of relying on time series.

3.3 Conclusion

In this thesis I have found that using an indicator variable $Q^h_{l,j}$ in my regression model for predicting future receipts based on old ones, is beneficial when ordinary kriging is used as a regression model. I found that it is very hard to get good prediction on a daily basis by representing the receipts as 209 time series, one for each subcategory; however on a weekly basis the predictions seem usable. Utilizing only 17 main categories allows the data to be much more predictable for $Q^h_{l,j}$ but provide less interesting results.

This algorithm could be improved by introducing a method to label the data into predictable categories while the groceries in them still being roughly equivalent. In a practical application, I would recommend using a maximum cap of number of categories recommended and only predict from the series where the best accuracy has been attained. There is room for improving in both $\hat{Q}^h_{l,j}$ and $f^h_{l,j}$, and I believe there is much to be explored by approaching this problem with clustering techniques. Seasonal and holiday modeling could improve the model provided that there is enough data available to support it.

Bibliography

Baafi, E. Y., Schofield, N. A., and Congress, I. G. (1997). *Geostatistics Wollongong '96 / edited by E.Y. Baafi and N.A. Schofield*. Kluwer Academic Dordrecht ; Boston.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, 20(3):280–282.

Cressie, N. (1993). *Statistics for spatial data*, page 10. Wiley series in probability and mathematical statistics: Applied probability and statistics. J. Wiley.

Duro, D. C., Franklin, S. E., and Dubé, M. G. (2012). A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using spot-5 hrg imagery. *Remote Sensing of Environment*, 118:259–272.

Gelfand, A., Fuentes, M., Guttorp, P., and Diggle, P. (2010). *Handbook of Spatial Statistics*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Taylor & Francis.

He, X., Deng, L., and Chou, W. (2008). Discriminative learning in sequential pattern recognition. *IEEE Signal Processing Magazine*, 25(5):14–36.

Jakobsson, A. (2015). *An Introduction to Time Series Modeling*. Studentlitteratur AB.

Liang, Y., Xu, Q.-S., Li, H.-D., and Cao, D.-S. (2011). *Support Vector Machines and Their Application in Chemistry and Biotechnology*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition.

Lindgren, G., Rootzen, H., and Sandsten, M. (2013). *Stationary Stochastic Processes for Scientists and Engineers*. Chapman and Hall/CRC.

Lu, H. (2014). Recommendations based on purchase patterns. *International Journal of Machine Learning and Computing*, 4(6):501–504.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization (Springer Series in Operations Research and Financial Engineering)*. Springer.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Thorrud, T. K. and Myklatun, Ø. (2015). Predicting e-commerce consumer behaviour using sparse session data. Master's thesis, NTNU.

Wackernagel, H. (1998). *Ordinary Kriging*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Zucchini, W., MacDonald, I., and Langrock, R. (2016). *Hidden Markov Models for Time Series: An Introduction Using R, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press.

Appendices

Appendix A
Sub category List

1. Avocado
2. Fruit, other
3. Cooking cheese
4. Spices
5. Lemon and Lime
6. Tomato
7. Orange
8. Plastic bag
9. White bread
10. Exotic fruit
11. Beans
12. Other
13. Egg
14. Baking
15. Radish
16. Mushroom
17. Pear
18. Sour cream
19. Celery
20. Juice
21. Cucumber
22. Paprika
23. Snacks, other
24. Frozen fruit
25. Appetizer
26. Cleanig atricles
27. Banana
28. Hygien
29. Spread
30. Seed and nuts
31. Bread cheese
32. Meatballs
33. Pasta, other
34. Marmalade
35. Berry
36. Peanut butter
37. Chicken
38. Milk
39. Muesli
40. Dried fruit
41. Potato
42. Gum
43. Rice
44. Vinegar
45. Ham
46. Salad
47. Spaghetti
48. Dog treats
49. Canned tomatoes
50. Soda
51. Child articles
52. Yoghurt

53. Apple products
54. Apple
55. Potato chips
56. Household articles
57. Fresh herbs
58. Canned food
59. Crisp bread
60. Detergent
61. Cottage cheese
62. Cream
63. Butter
64. Beef
65. Penne
66. Dark bread
67. Onions
68. Pre cooked meals
69. Cereal
70. Creme fraiche
71. Asparagus
72. Sausage
73. Buns
74. Pork
75. Sourbough bread
76. Vegetables, other
77. Melon
78. Ice cream
79. Flour
80. Fresh cheese
81. Zucchini
82. Cat food
83. Candy, other
84. Coffee
85. Washing up liquid
86. Drink, other
87. Tea
88. Salami
89. Chocolate
90. Delicacy meat
91. Sous
92. Keso
93. Couscous
94. Dental hygien
95. Grounded meat
96. Grape fruit
97. Thick yoghurt
98. Carrots
99. Cooing oil
100. Maccaroni
101. Toilet paper
102. Nature candy
103. Turkey
104. Fruit candy
105. Household paper
106. Kitchen utilities
107. Oatmeal
108. Grapes
109. Fish
110. Garlic
111. Olives
112. Processed tomato
113. Spinach
114. Baking accessories
115. Nuts
116. Mix package
117. Processed meat
118. Energy bar
119. Cracker
120. Biscuits
121. Tagliaelle
122. Pastry
123. Green beans
124. Noddles
125. Food container
126. Potato products
127. Lemonjuice
128. Pork loin
129. Lentils
130. Flowers
131. Kitchen cleaning articles
132. Leek
133. Liver patée
134. Dry child food
135. Quinoa
136. Alcohol
137. Risoni

138. Fresh pasta
139. Soy sous
140. Broccoli
141. Suger
142. Twisted pasta
143. Grain
144. Root vegetable
145. Rigatoni
146. Cabbage
147. Auberine
148. Clothes
149. Frozen Vegetable
150. Hamburger bread
151. Coffee accessories
152. Child food
153. Cough drops
154. Sandwich spread
155. Caviar
156. Salsa
157. Batteries
158. Veal
159. Tortilla
160. Fruit yoghurt
161. Must
162. Water
163. Throat medicine
164. Nut butter
165. Basic goods, other
166. Vegan pasta
167. Sour milk
168. Jam
169. Bars
170. Cleaning supplies
171. Cooking accessories
172. Dog food
173. Frozen berries
174. Ketchup
175. Books
176. Smoothie
177. Kitchen articles
178. Olive oil
179. Vitamine
180. Cookies
181. Soup
182. Beauty products
183. Honey
184. Bulgur
185. Crustacean, other
186. Processed vegetarian products
187. Creme Fraice, vegan
188. Acetum
189. Broth
190. Chocolate powder
191. Chocolate drink
192. Freezing bags
193. Popcorn
194. Hot dog buns
195. Shrimps
196. Liquid food extract
197. Dip
198. Pesto
199. Branded restaurant food
200. Pudding
201. Scratching tickets
202. Tobacco
203. Flat bread
204. Crawl fish
205. School supplies
206. Aspirin
207. Garden supplies
208. Rodent food
209. News paper

Appendix B
Main category list

1. Drink
2. Snacks
3. Frozen wares
4. Vegetables
5. Pantry
6. Bread
7. Meat
8. Prepared food
9. Sandwich spread
10. Pasta
11. Candy
12. Pet products
13. Medicin
14. Health foods
15. Cheese, other
16. Shellfish
17. Other

EXAMENSARBETE Shopping list generation with machine learning
STUDENT Daniel Tykesson
HANDLEDARE Pierre Nugues (LTH)
EXAMINATOR Jacek Malec (LTH)

Automatisk inköpslista

POPULÄRVETENSKAPLIG SAMMANFATTNING **Daniel Tykesson**

Vid inhandling av livsmedel används ofta någon form av inköpslista. Dessa listor är inte alltid tillgängliga, till exemepel på grund av att inhandlingen sker efter jobbet. Detta examensarbetet presenterar en algorithm för att generera dessa listor med historisk data.

Inköp av livsmedel kan väldigt lätt bli en komplicerad process för många människor i deras vardag. En vanlig orsak till att det blir svårt att handla är att man gör det på vägen hem från jobbet eller annan sysselsättning. Problemet kommer ifrån att man inte har koll på vad man har hemma och kan leda till dåliga inhandlingsvanor. Dåliga inhandlingsvanor kan innebära allvarliga problem på både person- och samhällsnivå. För många kan brist av inköpslistor innebära många impulsköp av varor som man inte hade köpt vanligtvis. Det kan också leda till att man handlar varor som man redan har hemma. I Sverige 2015 kastade vi i snitt 54 kilo mat per person i onödan. Vilket är ungefär 500,000 ton mat för hela befolkningen. Detta har konsekvenser för vår miljö och det mostsvarar mellan 20 och 25% av svenskars totala klimatpåverkan, och att få bättre inhandlingsvanor är då väldigt viktigt. Ett sätt att få bättre inhandlingsvanor är att alltid ha tillgång till en inköpslista. En praktisk lösning på detta vore att man automatiskt kunde genera dessa inköpslistor, i t.ex. mobilen.

Jag har i mitt examensarbete utvecklat en algoritm som automatiskt genererar inköpslistor. Den använder data från gamla kvitton där alla varor på kvittona kategoriseras. Kategorierna består av två lager, 209 underkategorier och 17 huvudkategorier. Underkategorierna är t.ex. fläsk, gurka, banan, o.s.v. medans huvudkategorierna är t.ex. kött, grönsaker, frukt, o.s.v. Underkategorierna valdes så att alla varor som finns i dem ska vara ganska utbytbara med varandra medans huvudkategorierna valdes så att deras varor har liknande syfte. Förutom vad som har inhandlats så måste mängden bestämmas och det gjordes med hur mycket pengar som har spenderats i en kategori. Folk har olika mönster när de handlar, vissa handlar flera gånger i veckan medans andra som bor lite avlägset handlar bara en gång i veckan. På grund av detta så kan algoritmen genera olika typer av inköpslistor som täcker olika mycket tid, t.ex kan den generera en inköpslista för en hel vecka.

Algoritmen kan kort sammanfattas med två steg. Först försöker den gissa ifall det kommer handlas i en kategori (huvud eller under) och sedan gissa hur mycket som kommer handlas. Resultaten i mitt examensarbete visar att algoritmen kan med hög precision gissa rätt på ungefär 18 underkategorier och 11 huvudkategorier för veckoinköpslistorna. Typiska underkategorier som den har hög precision på är mjölk, bröd och gurka medans för huvudkategorier är det typiskt grönsaker, kött och dryck. Med hög precision så gissar den rätt över 70% av gångerna på dessa kategorier. Jag tror att min algoritm kan användas i praktisk applikation, t.ex. som en app för mobiltelefoner.