Beating the baseline prediction in food sales: How intelligent an intelligent predictor is?

Indrė Žliobaitė[1,2]
[2]Vilnius University
Naugarduko 24
Vilnius, Lithuania
zliobaite@gmail.com

Jorn Bakker[1] and Mykola Pechenizkiy[1]
[1]Eindhoven University of Technology
P.O. Box 513, NL-5600 MB,
Eindhoven, the Netherlands
{j.bakker, m.pechenizkiy}@tue.nl

ABSTRACT

Sales prediction is an essential part of stock planning for the wholesales and retail business. It is a complex task because of the large number of factors affecting the demand. Designing an intelligent predictor that would beat a simple moving average baseline across a number of products appears to be a non-trivial task. We present an intelligent two level sales prediction approach that switches the predictors depending on the properties of the historical sales. First, we learn how to categorize the sales time series into 'predictable' and 'random' based on structural, shape and relational features related to the products and the environment using meta learning approach. Next, for the products identified as 'predictable' we apply an intelligent base predictor, while for 'random' we use a moving average. Using the real data from a food wholesales company we show how the prediction accuracy can be improved using this strategy, as compared to the baseline predictor as well as an ensemble of predictors. In our study we also show that by applying an intelligent predictor for the most 'predictable' products we can control the risk of performing worse than the baseline.

Categories and Subject Descriptors

H.2 [**Database Management**]: Database application - Data mining

Keywords

sales prediction, time series categorization, meta learning

1. INTRODUCTION

Accurate and timely sales prediction is essential part of business planning with a direct impact to stock management and profitability. In food sales, the stock includes a large assortment of goods, some require special storage conditions or are quickly perishable.

The variations in consumer demand may be influenced by price change, promotions, changing consumer preferences or weather [17]. Furthermore, seasonal peaks occur due to different cultural habits, religious holidays, fasting. Thus some types of products have high sales during a limited period of time (e.g. eggs around Easter), some high fluctuations in sales (e.g. beer), while the others have more or less flat sales all the time (e.g. salt).

Food sales prediction raises several challenges. First, seasonal patterns are expected, but the predictive features that define these seasons are not always directly observed. Moreover, for some products variations in sales might be affected by consumer habits, others might be of a more random nature. Besides, for some products the sales is highly imbalanced with only a few peaks per year. Predicting those peaks might be essential from business perspective, as compared to the accuracy for the rest of the season.

State of the art methods for food sales prediction are often moving average with different lag or simple regression models. In such settings these predictions are often overridden by managers using their intuition and expertise. Predictions based on moving averages may work well when sales are flat. But when the sales are fluctuating the reaction of moving average is too slow. Predicting a peak in sales too late is particularly loss bearing. Managers often try to improve the performance in seasonal peak periods by prudently increasing the stock and thus costs.

Another typical approach is to keep reminders that hint about the coming school, national or religious holidays, warm or cold, sunny or rainy weather and other demand triggers. These soft rules vary from manager to manager, they are human labor intensive and often lack of consistency, which may result in mistaken predictions.

A desired food sales prediction system should take into account seasonal triggers and be able to distinguish between seasonal and random fluctuations. It should be able to control the risks in predicting random behavior and to exploit predictable behavior.

We develop a two level prediction approach, where we first identify the environmental state and then apply the predictor that suits best to this state. We presented a general framework to integrate context awareness into the food sales prediction process in the workshop [22]. We showed how to learn to categorize the sales time series into four categories and tried to find a suitable learner for each of the categories. With the experimental study we showed that these distinct categories exist, and that the selection of the learner according to the category of the time series can improve the overall prediction performance given that the categorization itself is accurate enough. However, learning an accurate categorization appeared to be a difficult task on its own.

In this study we focus on practical aspects of designing a two level switch model. Particularly, we show how to outperform a baseline currently employed by the company. Our experimental study on the real food sales data provides empirical evidence favoring the introduced approach. We also analyze the effects of categorization and thresholding on the performance of the prediction system and associated risks of performing worse than the baseline.

The rest of the paper is organized as follows. In Section 2 we present the proposed method. In Section 3 we highlight the evaluation challenges and present the experimental approach. Extensive experimental evaluation using the case of food wholesaler Sligro Food Group N.V. is carried out and results are discussed in Section 4. Section 5 highlights the related work and Section 6 concludes.

2. TWO LEVEL PREDICTION MODEL

In this section we present a two level prediction approach, which we call WHALE (from WHolesALEs). It incorporates a mechanism that *switches* the final predictors depending on what environmental state is observed. By environmental state we mean the context affecting the modeled object (e.g. calendar events, weather, macroeconomic situation, consumer habits for product sales) or the properties (e.g. historical behavior) of the object itself.

The main idea of the two level approach is to identify the state first and then use the predictor linked to this state. For instance, different products have different sales behavior and different dependence on calendar events (seasonality). If we can extract distinct categories of products, specific input data construction procedures and specific predictors could be employed for each category, learning the link between the categories and final predictors.

An alternative is to collect all possible predictive features and then learn a complex model expecting to incorporate the categories and the switch mechanism automatically. The task might be too complex w.r.t. the available data. Besides, not all the environmental features might be observable or measurable directly. We argue that by defining the categories we bring in domain specific assumptions to the model and, as a result, simplify the learning process and reduce the number of degrees of freedom in the decision making.

2.1 WHALE model

Let $\mathbf{y}^j = \{y_1^j, y_2^j, \ldots\}$ be a time series of the object j (e.g. historical sales of a product). The ultimate task is to learn a mapping $\hat{y}_{t+1}^j = \mathcal{L}(\{y_1^j, \ldots, y_t^j\}, F_p^j)$, where F_p^j are additional predictive features (e.g. weather). They can be shared among the objects. Constructing and learning the mapping procedure $\mathcal{L}$ is the *design* part of the WHALE model, application of the model to cast the prediction $\hat{y}_{t+1}^j$ is the *operational* part.

We use the two level approach to restrict the space of search of $\mathcal{L}$ types and parameterizations. For that *categorical features* F_s^j related to the environment the object comes from are constructed. Let $c = (c_1, c_2, \ldots, c_m)$ be a set of *categories*, a procedure for constructing them is defined by a designer. Let $\mathcal{G} : F_s \to c$ be a mapping from F_s to categories. Let $L = (L_1, L_2, \ldots, L_m)$ be a set of individual learners. By *a learner* we mean a fixed set of training instances, input feature space, base classifier and it's parametrization.

The two key ingredients of the WHALE model *design* are: defining the categorical feature space F_s with a mapping $\mathcal{G}$, and fixing the mapping $\mathcal{H} : c \to L$. After the mappings are fixed, the decision for a given product j at time $t+1$ is made as follows:

1. individual learner is selected $L_i^j = \mathcal{H}(\mathcal{G}(F_s^j))$,
2. the decision is made using L_i as the base learner $\hat{y}_{t+1}^j = L_i(\{y_1^j, \ldots, y_t^j\}, F_p^j)$.

Thus having a trained WHALE ready for online operation means:

- a set of categories $(c_1, c_2, \ldots, c_m)$ is defined and fixed;
- a mapping procedure $\mathcal{G}$ from products (time series) to the categories is established;
- 'local' expertise of each predictor is known (mapping $\mathcal{H}$).

Figure 1 presents the online operation of WHALE.

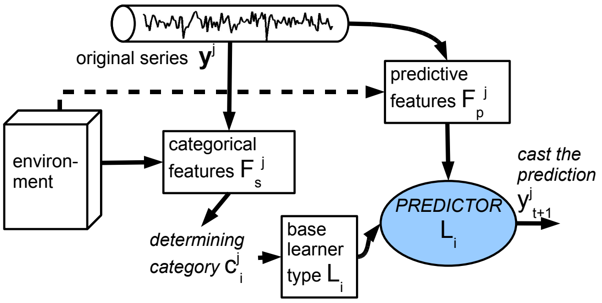

original series $\mathbf{y}^j$
predictive features F_p^j
environment
categorical features F_s^j
cast the prediction
PREDICTOR L_i
y_{t+1}^j
determining category c_i^j
base learner type L_i

Figure 1: Online operation of WHALE.

2.2 Training WHALE

The core part of WHALE is to match the product categories and the base predictors.

Training part. A limited set of base predictors $(L_1, L_2, \ldots, L_m)$ needs to be preselected based on domain knowledge and expectations in order to delimit full state space search. Then m parallel experiments for each object $(y^1, \ldots, y^d)$ need to be carried out using a validation set ($m \times d$ results in total). Next, the objects are grouped into m categories $(c_1, c_2, \ldots, c_m)$ based on the best performing predictor. Each obtained category serves as a basis for constructing categorization rules.

Operational part. The goal of the training process is to learn to assign an object to one of these defined categories online, having only a fragment of the series. When we have the categorization rules $\mathcal{G}$ and mapping $\mathcal{H}$ fixed, an object can be processed as described in Figure 1. First of all the category of the object j is determined (c_i^j). Then the corresponding predictor L_i is used to output the prediction.

2.3 WHALE for Food Sales Prediction

We presented a generic WHALE framework. WHALE is a two level decision making model: first determining the category to an object and then switching the final predictor based on the current category of the object. Next we specify the design and implementation choices accordingly for food sales prediction, where a product is an object.

2.3.1 Categorization and categorical features

We start by distinguishing two intuitive product categories: 'random' and 'predictable'. We assume that for the 'predictable' products the fluctuations in sales should be explainable by external features (e.g. weather, holidays) and using an intelligent base learner (L_{int}), while the sales of the 'random' products in general independent of these explanatory features, thus a moving average of the historical sales would be a better choice (L_{MA}).

We consider two alternative mechanisms to distinguish between the two types of products: cross validation and meta learning approach. Meta learning is aiming to learn the relationship between the performance of base learners and the properties of datasets [2] (in our case - products).

Categorical features. In order to learn a mapping $\mathcal{H}$ between the products and the base predictors, we define higher level features of the datasets, which we refer to as *categorical features* F_s. They should be extractable from the input product historical data online and are desired to be length independent. In addition, we want the categorical features to be related to observable seasonal patterns.

We define three groups of such features: behavior, shape and relational features. Behavior features are expected to give overall information about the behavior of sales time series in terms of peaks, transitions, local and global variation, disregarding the exact configuration of the patterns:

- F_{s1-s2} |mean value - median value|, standard deviation;
- F_{s3} shift: the mean value of the points for which $y_t < mean(y)$ minus the median value of the points for which $y_t < mean(y)$;
- F_{s4-s8} threshold h crossing ratio, where $h = 0.3, 0.4, 0.6, 0.7, 0.8$, scaled to the total length of the signal;
- F_{s9-s10} normalized power of the frequency p in the frequency spectrum $1/52$ and $2/52$ for seasonal patterns (Fast Fourier Transformation, Cooley-Tukey implementatation [3]);
- $F_{s11-s12}$ local variation features: interquartile range (of $y_t - y_{t-1}$) and unequal neighbors (the number of times $y_t \neq y_{t-1}$ to the total length of the series).

Shape features are expected to align the information about the shape patterns of the historical time series. These features depend on the series length:

- $F_{s13-s17}$ quadruple SAX representation of the series [9], which encodes the series into a d-dimensional representation, where each dimension represents the signal level at a given part of the series.

The relational features relate sales to the environment:

- $F_{s18-s26}$ absolute correlations with temperature, rain level, pressure, school holidays, calendar events, seasons (spring, summer, autumn, winter).

We normalize the values of the series to be in a range $(0, 1)$ before extracting structural features.

Learning to categorize using meta-learning approach. In order to learn to assign the category online for a given product, we train a meta classifier $\mathcal{G}$ using categorical features F_s and category labels. We obtain category labels by running both base learners (L_{int}, L_{MA}) for all the products (validation part) and assigning label 1 if L_{int} is more accurate and 0 otherwise. Then for a given product y_i we determine the category using the trained rule $c_i = \mathcal{G}(F_s^i)$.

Categorization by cross validation. An alternative to the meta categorization approach is to assign a category for a given product y^i by directly testing the base predictors on the training part of the series. The training part of the series is divided into k bins, based on temporal order (e.g. one bin is $(y_t^j, y_{t+1}^j, \ldots, y_{t+u})$). Both base learners L_{int} and L_{MA} are tested using k-fold cross validation in time. Here it is assumed that the 'predictable' or 'random' behavior of the series does not change in time (no concept drift). If L_{int} is more accurate than L_{MA} then $c_i \leftarrow 1$, otherwise $c_i \leftarrow 0$.

2.4 Input Features for Final Prediction

Predictive features F_p are used for the chosen final predictor L_{int} or L_{MA} to make the prediction $\hat{y}_{t+1}^j$.

The feature space is formed using internal and external data. Internal data comes from a company sales database. External data (holidays, temperature, seasons) is formed using information from the ministry of culture, meteorological institute and general knowledge. The feature set is specified in Figure 2.

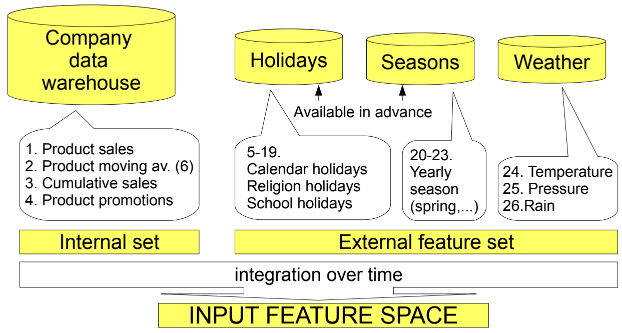


Figure 2: Formation of the predictive feature space.

The internal features are interrelated. The moving average (F_{p2}) is calculated using (F_{p1}). The cumulative sales (F_{p3} include the sales quantity of all the products. Promotions (F_{p4}) for some products are organized and this can be known in advance.

The external features (F_{p5-p23}) are available in advance. Average weekly temperature at a given location (F_{p24}), pressure (F_{p25}) and rain (F_{p26}) can be predicted sufficiently accurately one-two weeks in advance. They are described in a single numerical dimension each. Holidays (F_{p5-p19}) are described in 16 continuous features. If the Seasons holiday happens to be on the week in question, the feature gets a value of 1, if it is in one or two weeks the features get values of 0.6 and 0.2 correspondingly. Seasons ($F_{p20-p23}$) are described in 4 binary features.

3. EXPERIMENTAL EVALUATION

Food sales prediction models are not trivial to evaluate. The evaluation of time series prediction in general is not straightforward [8] due possibly unbounded range and variety of the domains. We identify two additional challenges specific to food sales prediction: comparison across different products and varying impact to inventory management decisions.

Evaluation criterion. We argue that instead of applying absolute error measures it is more reasonable to use relative error measures w.r.t. baseline method. We use the Mean Scaled Absolute Error (MASE) to measure the prediction performance:

$$MASE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{e_t}{\frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t^*|}\right|, \qquad (1)$$

where the error $e_t = y_t - \hat{y}_t$, y_t is the actual value, $\hat{y}_t = L_{model}(F_p)$ is the predicted value at time t, $\hat{y}_t^*$ is the predicted value by the baseline $L_{baseline}$.

This scaled error measure can be used to evaluate predictors regardless of the structure of the time series. Thus, we can aggregate the error measures to obtain a relative performance measure of a predictor over all products.

In food sales prediction we choose L_{MA1} as the evaluation baseline, which is used in as a lower bound for qualitative analysis of predictors. L_{MA1} takes the previous value as the prediction: $\hat{y}_{t+1} = y_t$.

We note that utility based evaluation would make sense in relation to inventory management decision [1]. Though in this study for assessing WHALE we assume equal costs of mistakes in terms of a timeline[1].

Sequential testing. Food sales prediction operates in time. The true value y_{t+1} becomes available after the prediction $\hat{y}_{t+1}$ is casted and before the new prediction decision $\hat{y}_{t+2}$ needs to be made.

[1] More details can be found in the technical report TR-Feb2010.pdf: www.win.tue.nl/~mpechen/projects/sligro/

Thus it makes sense from application perspective to include the newly available point y_{t+1} into the training set for the prediction of $\hat{y}_{t+2}$.Computational complexity related to retraining is not a concern here, since we focus on weekly predictions, which are essential for stock management.

Thus mimicking the operational settings for model evaluation sequential testing should be employed. Sequential testing means the predictor is retrained at each time step including newly available training data.

Label discretization. We argue that from stock management perspective absolute predictions do not matter as much as relative predictions do. The crucial information to be predicted is whether the sales will grow or drop next week and how significant the changes will be. For instance, if 20% increase in milk sales is predicted, the company would stock up more than usual. If 50% decrease is predicted, they will order less than usual from the suppliers.

Thus we use discretized target values (labels) y_t. We operate ordinal levels of sales discretized into 8 bins using the following thresholds: more than 75% decrease in sales w.r.t. to the mean, $75-50\%$ decrease, $50-25\%$ decrease, $25-0\%$ decrease, $0-25\%$ increase, $25-50\%$ increase, $50-75\%$ increase, more than 75% increase. We use these levels as inputs for the MASE (Eq. 1).

3.1 The Data

We study a case of Sligro Food Group N.V. (SLIGRO). The company is engaged in food wholesales. SLIGRO works with corporate clients, mainly food retail and food service companies (restaurants), although there are some direct consumers as well. SLIGRO has around 40 outlets in the Netherlands. The group pursues a multi-channel strategy, covering various forms of sales and distribution (cash-and-carry and delivery service) and using several different distribution channels (retail and wholesale). SLIGRO trades about 60000 products.

Our experimental field consists of a random unstratified sample of 600 products over two years period (from July 2006 to October 2008) at Eindhoven outlet. The sales are aggregated on weekly basis, each series is of 119 weeks length. Each series represent the sales quantities of one product.

Product filtering. The products which have no anomalous sales behavior and have sufficient sales volumes are selected for experiments to fall within the scope of this study. By anomalous behavior we mean discontinued sales volumes.Given a product time series $(y_1, \ldots, y_{119})$ the criteria for insufficient sales volume is: $\max y_i > \theta, \forall i = 1 \ldots, 119$ and $mean(y_i, \ldots, y_{i+m-1}) > \theta^*, \forall i = 1, \ldots, 120 - m$, we use $\theta = \theta^* = 10$ and $m = 4$ weeks.

Experimental set. For experimental evaluation we use 330 products, which we divide into training-validation (220) and testing (110) baskets at random. The testing basket is needed to test for the final prediction accuracy, categorization rules $\mathcal{G}$ and mapping $\mathcal{H}$ are learned on the training-validation basket. In addition, each product series is split into training and testing part in time. The testing part consists of 60 weeks, 59 are left for training. The training part is used for categorization and learning the final predictor. The experimental division of the dataset is illustrated in Figure 3.

We use real valued inputs. The historical sales are normalized to have a mean of 1, obtaining the thresholds from the training part. External features are normalized to be within $(0, 1)$ interval.

3.2 Alternative techniques

The goal of the experiments is to investigate if WHALE can consistently outperform the baseline, currently employed in practice. Since the core part of WHALE is the link between categorization and the final prediction models, the experiments have double focus. First we will analyze the relation between the categorization accuracy and final prediction accuracy. Then we will analyze the final prediction accuracies in relation to parametrization as well as different utility implications.

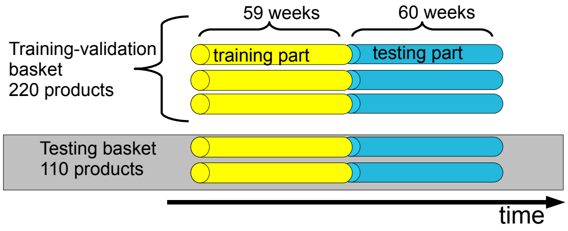


Figure 3: Experimental division of the dataset.

For the *final prediction* we test the following alternative techniques.

1. Moving average of 1 (L_{MA1}) as a naive predictor ($\hat{y}_{t+1} = y_t$).
2. Moving average of 6 (L_{MA6}) as state of the art currently used by the company ($\hat{y}_{t+1} = (y_{t-5} + \ldots + y_t)/6$).
3. Decision tree (L_{tree}) using multidimensional features (F_p: calendar, weather and other external info) as intelligent predictor. We choose the tree expecting to capture non linear relationships between the external features and sales performance, in cases where there is a relation to be captured.

For *categorization* into 'predictable' and 'random' products ($\mathcal{G}$) we test two alternatives:

- cross validation (cross-v) on the training part, and
- meta learning approach (meta) using categorical features F_s of the product sales.

3.3 Experimental set up

The experimental scenario consists of three parts: selecting the base predictors, learning categorization rules and testing the final model accuracies. In the first part we analyze the performance of alternative base predictors ($L_1, \ldots, L_m$) on validation part of all the products without categorization into 'random' and 'predictable' products. We show that there exist product subsets on which it is *possible* to outperform baseline predictor. In the second part we investigate the relation between the product categorization accuracies and the final prediction accuracies. We aim to build the accurate dependencies using two approaches *meta learning* and *cross validation.* In the third part we present and analyze the final prediction accuracies.

4. EXPERIMENTAL RESULTS

4.1 Selection of the base predictors

We select the following intelligent predictors for testing: linear regression (L_{reg}), k Nearest Neighbors (L_{kNN}), regression tree (L_{tree}). We include moving average of 1 (L_{MA1}) as the naive predictor and moving average of 6 (L_{MA6}) as current state of the art (baseline).

We use the predictive features (26) as specified in Figure 2. We do feature selection by calculating the correlations of the features with the labels. The feature selection results will be presented in the following subsection.

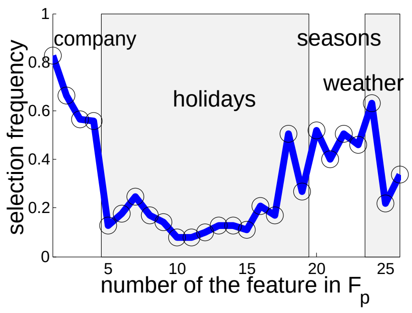


Figure 4: The selected features.

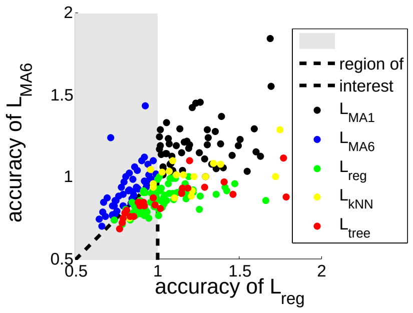


Figure 5: Top performing classifiers.

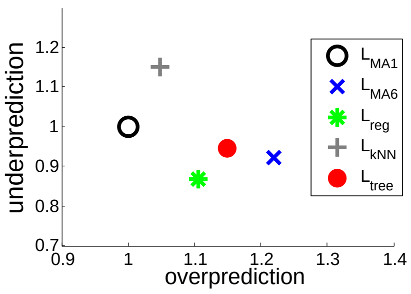


Figure 6: Two types of errors.

Table 1: Results on the training-validation product basket.

	L_{MA1}	L_{MA6}	L_{reg}	L_{kNN}	L_{tree}
MASE	1.00	1.07	0.99	1.10	1.05
ranks	*2.8*	*3.0*	*2.2*	*3.6*	*3.1*

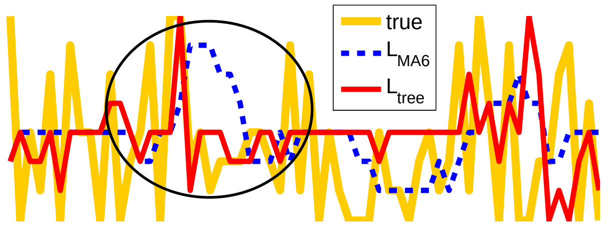


Figure 7: Predictions of L_{tree} versus L_{MA6} for one product.

We run all the predictors on the training basket (220 products). We split each product into two parts along the timeline. 59 weeks are used as a 'warm up' and the remaining 60 weeks are used for sequential testing. In Table 1 we list the average MASEs of all the methods (the lower the better). In the same table we provide the average ranking of all the methods, where the best performing method on a selected product gets the rank '1' and the worst gets the rank '5'.

The regression outperforms other predictors in terms of average rank as well as slightly outperforms the baseline L_{MA6} in terms of the average accuracy. Can we do better? It can be seen that the ranks do not map directly to accuracies. This suggests that some differences between the performances might be of different magnitude and supports the idea of state switching. It is important to be able to identify the products on which the baseline might be outperformed significantly by an intelligent predictor.

4.1.1 Feature selection

We select the features which have the absolute correlation with the class label 0.25 or higher. On average 8.3 features from 26 are selected. In Figure 4 we depict the feature selection counts for the last testing week. It means that complete two years can be used for training. 0 means that the feature is never selected, 1 means that the feature is selected all the time.

The 'ideally' predictable product would depend on external features and the labels would not be correlated with the historical sales. We have quite a large frequency of internal features. It means that often we are operating more sophisticated forms of moving average. However, external features also do contribute, especially seasons and weather. Low frequency of holiday features is not surprising, having in mind that only 2 years of history are available. In addition, intuitively only a small share of products can be strongly related to calendar events, e.g. chocolate eggs to Easter.

4.1.2 'Predictable' products

Let us have a closer look at the distribution of prediction accuracies among the products. In Figure 5 we depict all the products from training-validation basket. The accuracy of an intelligent predictor L_{reg} is depicted against the accuracy of L_{MA6}. The gray area indicates the accuracies of an intelligent predictor below 1, which mean they outperform the naive predictor L_{MA1}. This is the group that we call 'predictable' products. The triangle indicates the cases when the intelligent predictor L_{reg} outperforms both the naive predictor L_{MA1} and the state of the art L_{MA6}. We call this subgroup 'outpredictable' products. These are the products we are interested to learn to categorize.

The relation between the groups is: 'outpredictable' belongs to 'predictable' belongs to all products. The proportions of these groups in the training set are 49%, 65%, 100%.

4.1.3 Advantage of the intelligent predictors

The ultimate goal, from domain perspective, is to project the demand for each product. However, the historical data does not indicate the demand, but sales. The demand might have been higher than the sales if a particular product was out of stock at the time. Thus overprediction errors can be treated 'softer' than underprediction.

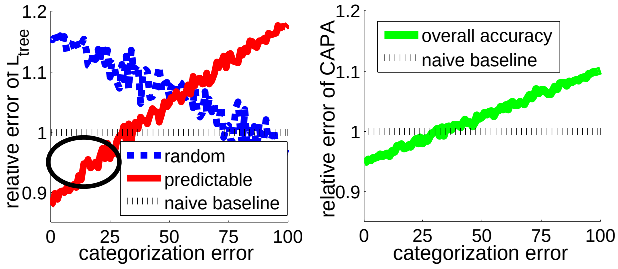
Let us have a look at two different types of mistakes the predictors make. In Figure 6 we depict the overprediction and underprediction MASEs on the training-validation product basket.

It can be seen that the intelligent learners make more overprediction on average than the baseline. Linear regression and regression tree make a bit less overpredictions than L_{MA6}. It suggests an intuition that L_{MA6} is slow at times of rapid changes in sales.

Let us zoom in on a particular case where L_{reg} outperforms L_{MA6}. In Figure 7 the ellipse indicates the area where L_{MA6} is slow to react to the changing sales, while the L_{reg} catches the decrease in sales right away.

4.2 Learning the categorization

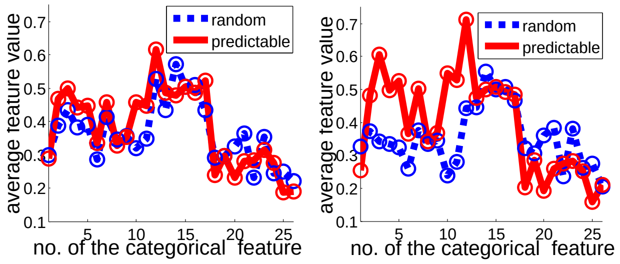
We showed that intelligent and baseline predictors show different performance on different products. We would like to learn how to discriminate between the 'predictable' and 'random' products in advance. We fix an assumption that the sales behavior of a given product is stationary over time. That is why we filter out obvious non stationary cases from the data as described in Section 3.1. In this study we determine the category of a given product once using the training part of the series. Then we employ sequential learning



(a) Within categories.

(b) Overall.

Figure 8: Relation between categorization and final accuracy.



(a) The 'true' categories.

(b) output by Naive Bayes.

Figure 10: Average categorical features.

and prediction procedure, assuming that the category of a given product is fixed.

In order to learn the categorization rules we need the 'true' labels. Given the the intelligent L_{int} and the naive L_{MA} predictors are fixed, we compare their accuracies on a given product. If the L_{int} on average outperforms L_{MA}, the product gets the 'true' label 'predictable' (1), otherwise 'random' (0).

Here we delimit the number of base predictors to two: moving average of 1 L_{MA1} as the naive predictor and a regression tree L_{tree} as the intelligent predictor. This way we get a meta dataset consisting of 220 products having binary labels (validation dataset in Figure 3). The proportion of the 'predictable' class is 43%.

4.2.1 *Target categorization accuracy*

The goal of categorization is to improve the final prediction accuracy. Thus categorization performance shall be evaluated not only based on categorization accuracy, but also based on the resulting final prediction accuracy. We inspect the relationship between the two with the following experiment. We divide the products into 'predictable' and 'random' categories based on the 'true' category label. Then we introduce a categorization error in a controlled way. We start from 0% categorization error (all category labels are correct). Next we increase categorization error by picking e.g. 10% of products at random and swapping their labels. This way we get 10% categorization error. We repeat the steps until 100% categorization error.

In Figure 8(a) we plot the categorization error against the relative error of the final prediction by the intelligent predictor L_{tree} for the two categories. The red curve represents the 'predictable' group. The benchmark at 1 represents the alternative relative error if the naive predictor L_{MA1} would have been used. Thus if we are able to achieve average categorization error less than 40%, we would apply L_{tree} for the 'predictable' category andL_{MA1} for the 'random' category.

The corresponding effect of the categorization error on the overall final prediction is presented in Figure 8(b). The prediction accuracy of WHALE (when switching between the intelligent and naive predictors) is linearly dependent on the categorization accuracy.

4.2.2 *Categorization by meta learning*

The categorization by meta learning uses categorical features, defined in Section 2.3.1. For learning the categorization rules $\mathcal{H}$ we use a Naive Bayes classifier. We pick it after preliminary meta experiments on the training data.

For the meta learning approach we also need to fix the threshold for categorization outputs. The intelligent predictors showing a final relative prediction error around 1 might not be significantly different from the naive predictors. We want to be able to control the risk by regulating the level of certainty in the 'predictable' category. For instance, if the categorizer $\mathcal{H}$ is able to identify only one of the most 'predictable' products, but this product comprises significant amount of total company sales, it is worth doing.

To investigate the effect of thresholding we perform the following sensitivity analysis. We split the training-validation basket to training (60%) and validation (40%) parts. The size of validation set is 88 products. In Figure 9(a) the label threshold options against the categorization error are depicted. The minimum error is achieved between 0.94 and 1.00 label threshold. In Figure 9(b) we depict the size of the 'predictable' class, output by the classifier on the validation set. Naturally, the lower the threshold, the less products are identified as 'predictable'.

In Figure 9(c) we plot the prediction accuracies against categorization thresholds. The red line indicates the accuracies within the 'predictable' group. It can vary a lot in size, as we saw in Figure 9(b). In case of a very low threshold less products are picked as 'predictable', but the classifier is more 'certain' as the prediction error is lower. Even without integrating the costs of mistakes, the overall performance is improved as can be seen in the ellipse area of the green line.

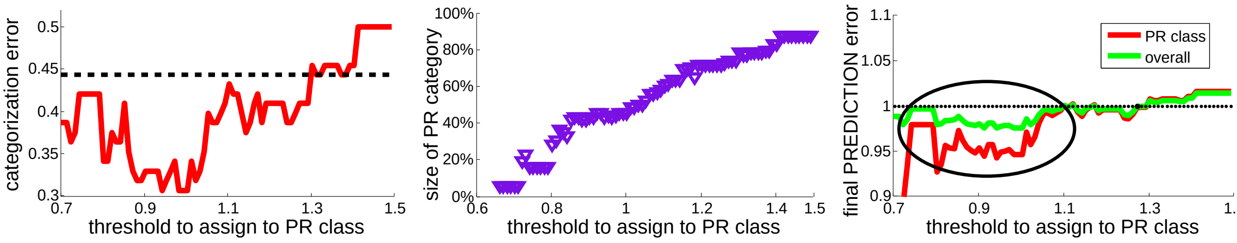
The idea of using a variable label threshold is similar to the concept of the ROC curve. It allows the decision maker to control the risk via rebalancing the output of the model in relation to the costs of mistakes. By moving the label threshold we are moving the decision boundary within the training data and actually train a set of models in order to inspect the sensitivity. While in the ROC curve the decision boundary of the trained model is being moved. A single model is trained and its continuous outputs are used to produce an ROC curve. Let us fix the label threshold at 1.00 which in Figure 9(a) appeared to be one of the optimal thresholds. For the meta learning approach we can employ ROC. For the Naive Bayes categorizer the area under ROC (AUC) in this case is 0.68.

The discrimination between 'predictable' and 'random' products is achieved using categorical features. We look how different these features are within each category. We again fix the category label threshold at 1.00 and we also fix category output threshold at 0.5. In Figure 10(a) the average values of the categorical features for the 'true' 'predictable' and 'random' product categories are depicted. In Figure 10(b) we depict the average features of the categories output by the Naive Bayes categorizer. The plot for the 'true' categories indicates that the categories are on average separable by the chosen categorical features. The plot for the output categories indicates that the categorizer has done good enough.

We showed how the categorization and the prediction accuracy continuously depend on the thresholding decisions.

4.2.3 *Categorization by cross validation*

An alternative for meta learning approach is categorization by cross validation in time on the training part of the series. We use



(a) Variable thresholding.

(b) Relation to the size of the category.

(c) Overall accuracy.

Figure 9: Categorization by meta learning: sensitivity to the training thresholds.

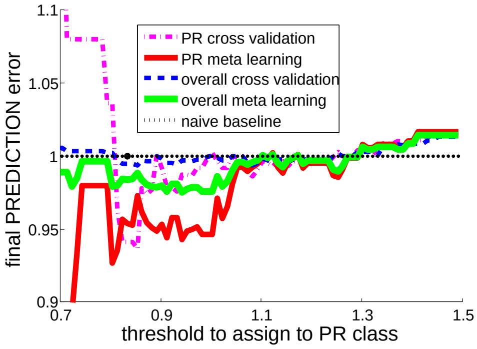


Figure 12: Overall prediction accuracy.

9-fold cross validation and average over the prediction accuracies. Note, that here there is no need for the 'true' labels, the categorization decision is not learnable, it is made directly.

We assign a product to 'predictable' if average cross validation MASE is < 1. For sensitivity analysis we replace 1 by variable threshold, similar to the ones we showed for categorization using meta learning. The results are presented in Figure 11. The ROC curve is not applicable here because there is no categorization model built.

Finally, we overlay the final prediction results with categorization using meta learning and cross validation in Figure 12. Categorization by meta learning outperforms cross validation (green vs. blue lines) in terms of the prediction accuracy averaged over all the products in the validation set.

To focus on the results achieved by cross validation vs. categorization by meta learning, we again fix the label threshold at 1.00. The testing results on the validation set (88 products) are presented in Table 2. 'Prior' means that all the products are considered 'random' and the naive predictor is applied, this is a lower baseline. 'Oracle' assumes the categorization is perfectly correct, so this is the upper baseline, what can be achieved with the correct categorization. It can be seen that categorization by meta learning outperforms both cross validation and 'Prior' on the prediction accuracies using L_{tree}. However, L_{MA6} is still a competitive predictor. It can be seen that final prediction accuracies with categorization using meta learning are better than 'Oracle' when using L_{MA6}. That means the result is achieved due to 'lucky' configuration of categorization errors. This also indicated a potential future research direction. L_{MA6} could be integrated into the switch mechanism in addition to currently used L_{MA1} vs. the intelligent predictor.

4.3 Final prediction accuracies

So far we were presenting the results achieved on the training-validation set. In this section we present the final prediction accuracies on the *reserved* testing basket TE (110) products. Categorization is trained on all the training-validation basket (220 products). In addition we present two more sets of results, which are tested on the training validation basket (110 products each) TV1 and TV2. Recall a scheme of the product division in Figure 3. So far we used training - validation basket for training, tuning the parameters and validation. Testing basket was not used at all so far. Now we present three blocks of final prediction results, using 3-fold cross validation in space. One fold is stratified and consists of the testing basket TE, while the other two are a random division of the training-validation products. See Figure 13 for illustration.

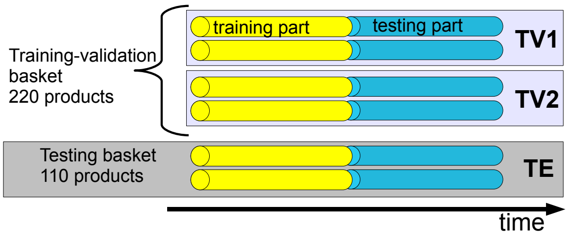


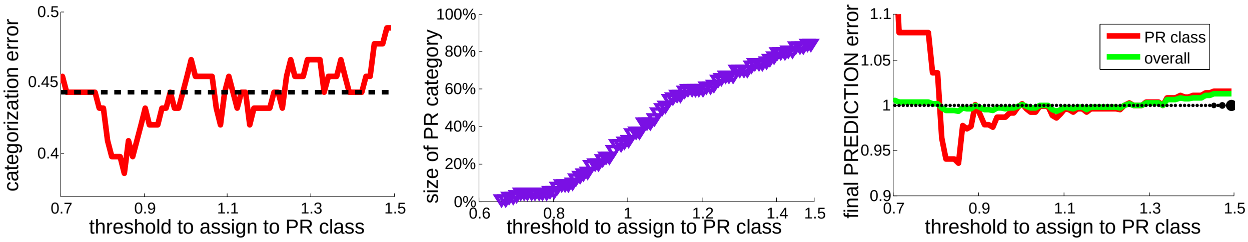
Figure 13: Division of the dataset for final testing.

In Table 3 we provide the categorization and final prediction results. In all the testing cases WHALE outperforms the naive L_{MA1} and the baseline L_{MA6}. It does not outperform L_{MA6} combined with L_{MA1} though. The combination of those two with the intelligent predictor is an interesting direction for future research. We compare two alternative techniques for switching the learner based on the product category: categorization using meta learning and cross validation. In two out of three testing cases, including an unseen testing set, categorization using meta learning outperforms cross validation in terms of the prediction accuracy.

The difference is statistically significant at 5% level of significance in a few cases. It should be taken into account that the length of the testing part for a given product is only 60 points, which constrains the power of the statistical test . We are using prediction errors for discretized labels as specified by the task, not classification accuracies. Different errors are assumed to make equal contribution to the prediction accuracy. The issue of evaluation was discussed in Section 3. We argue that the task requires specific cost based evaluation and conventional statistical techniques do not fully take into account the domain specifics. The cost based evaluation technique is yet to be developed.

5. RELATED WORK

In many real-world domains the situations seen in the past might partially repeat, which is referred as reoccurring contexts [21]. Seasonality comes very close to the concept of reoccurring contexts,



(a) Variable thresholding. (b) Relation to the size of the category. (c) Overall accuracy.

Figure 11: Categorization by cross validation: sensitivity to the training thresholds.

Table 2: Categorization and final prediction results on the training-validation set. 'Prior' means that all the products are considered 'random', 'Oracle' assumes the categorization is perfectly correct. ● indicates the difference from the naive L_{MA1} is statistically significant, ○ that it is not. Mann-Whitney U test [13] is used (5% level of significance).

	Meta	Cross-v	Prior	Oracle
Categorization error	30.68%	45.45%	44.32%	-
Final MASE over all products L_{MA6}	1.0673			
Final MASE within 'predictable' products L_{TREE}	0.9465 ○	0.9922 ○	-	0.8588 ●
Final MASE over all products $L_{TREE} + L_{MA1}$	0.9757	0.9971	1.0000	0.9374
Final MASE within 'predictable' products L_{MA6}	0.9365 ●	1.0328 ○	-	0.9704 ○
Final MASE over all products $L_{MA6} + L_{MA1}$	0.9711	1.0123	1.0000	0.9869

Table 3: Categorization and final prediction results on the testing set. 'Prior' means that all the products are considered 'random', 'Oracle' assumes the categorization is perfectly correct. ● indicates the difference from the naive L_{MA1} is statistically significant, ○ that it is not. Mann-Whitney U test [13] is used (5% level of significance).

		Meta	Cross-v	Prior	Oracle
testing basket TE	Categorization error	40.91%	44.55%	50.91%	-
	Final MASE over all products L_{MA6}	1.0323			
	Final MASE within 'predictable' products L_{TREE}	0.9613 ○	0.9614 ○	-	0.8651 ●
	Final MASE over all products $L_{TREE} + L_{MA6}$	0.9842	0.9849	1.0000	0.9313
	Final MASE within 'predictable' products L_{MA6}	0.9073 ●	0.9616 ○	-	0.9409 ○
	Final MASE over all products $L_{MA6} + L_{MA1}$	0.9621	0.9850	1.0000	0.9699
basket TV1	Categorization error	40.00%	44.55%	42.73%	-
	Final MASE over all products L_{MA6}	1.0801			
	Final MASE within 'predictable' products L_{TREE}	0.9964 ○	0.9990 ○	-	0.8890 ●
	Final MASE over all products $L_{TREE} + L_{MA1}$	0.9984	0.9996	1.0000	0.9526
	Final MASE within 'predictable' products L_{MA6}	0.9733 ○	0.9902 ○	-	0.9551 ○
	Final MASE over all products $L_{MA6} + L_{MA1}$	0.9886	0.9962	1.0000	0.9808
basket TV2	Categorization error	37.27%	38.18%	43.64%	-
	Final MASE over all products L_{MA6}	1.0685			
	Final MASE within 'predictable' products L_{TREE}	0.9821 ○	0.9790 ○	-	0.8692 ●
	Final MASE over all products $L_{TREE} + L_{MA6}$	0.9901	0.9927	1.0000	0.9429
	Final MASE within 'predictable' products L_{MA6}	0.9647 ○	1.0158 ○	-	0.9730 ○
	Final MASE over all products $L_{MA6} + L_{MA1}$	0.9804	1.0054	1.0000	0.9882

with an emphasis that it is not known with certainty, when the contexts will reoccur.

Change detection is the prevailing technique to deal with permanent drifts [6]. After the change is detected, an old portion of the training data is left out. These methods work under assumption that newer examples are always more representative than the old ones.

The methods designed to handle reoccurring contexts can store concept descriptions [21], employ instance [4, 5] or batch [7, 10] selection to look for case bases, or maintain a diverse ensemble of learners [15, 14, 12, 19, 16].

Ensemble learning maintains a set of concept descriptions, predictions of which are combined using a form of voting, or the most relevant description is selected online. An alternative approach is referred as WHALE approach comes close to meta learning [18], where the relevant learners are selected based on offline predefined criteria. A two level learning model is presented by Widmer [20] perceived context changes are used to focus the learner specifically on the information relevant to the current context. Klinkenberg [11] developed an approach, where at each time step not only the training window but also the type of base learner and its parametrization is selected from a fixed set of learners, using cross validation.

In this study we introduced a generic sales prediction approach with state switching. In contrast with the discussed meta learning approaches [20, 11], we incorporate domain expertise and observations in categorization and base predictor selection process.

6. CONCLUSION

We develop an intelligent approach for sales prediction, which uses a mechanism for model switching depending on the sales behavior of a product. For recognizing the behavior we formulate three types of categorical features: behavioral, shape-related, and relational, which allow to categorize the products sufficiently accurate to beat the baseline in the final prediction.

In the SLIGRO case study we show that WHALE consistently outperforms baseline methods (based on moving average) in terms of accuracy. We also demonstrate the tradeoffs between the risk and benefit to the final accuracy while moving the categorization threshold. By varying the threshold, it possible to tradeoff higher accuracy for a smaller set of products against taking more risk on a larger set of products.

In this study we assume that the category of a given product is static over time. The next practical step would be to employ the approach in a dynamic setting, where the structural type might change over time as well. Moving the categorization threshold calls for integrating cost sensitive learning approach.

7. ACKNOWLEDGMENTS

We thank Sligro Food Group N.V. for providing the data and the domain knowledge for the case study. This research has been partly supported by NWO (HaCDAIS Project) and LOIS grant.

8. REFERENCES

[1] J. Bakker and M. Pechenizkiy. Food wholesales prediction: What is your baseline? In *Foundations of Intelligent Systems, Proc. of the 18th Int. Symp., ISMIS 2009*, pages 493–502, 2009.

[2] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta. *Metalearning: Applications to Data Mining*. Springer Publishing Company, Inc., 2008.

[3] J. W. Cooley and J. W. Tukey. An algorithm for the machine computation of the complex fourier series. *Mathematics of Computation*, 19:297–301, 1965.

[4] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle. A case-based technique for tracking concept drift in spam filtering. In *Proc. of the 24th SGAI Int. Conf. on Innovative Techniques and Applications of AI*, pages 3–16, 2004.

[5] W. Fan. Systematic data selection to mine concept-drifting data streams. In *Proc. o the 10th ACM SIGKDD int. conf. on Knowledge discovery and data mining*, 2004.

[6] J. Gama and G. Castillo. Learning with local drift detection. In *Advanced data mining and applications, proc. of the 2nd int. conf., ADMA 2006*, pages 42–55, 2006.

[7] M. B. Harries, C. Sammut, and K. Horn. Extracting hidden context. *Machine Learning*, 32(2):101–126, 1998.

[8] R. Hyndman and A. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[9] J.Lin, E. J. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Min. Knowl. Discov.*, 15(2):107–144, 2007.

[10] I. Katakis, G. Tsoumakas, and I. Vlahavas. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *Knowledge and Information Systems*, 2009.

[11] R. Klinkenberg. Meta-learning, model selection and example selection in machine learning domains with concept drift. In *Ann. Workshop on Machine Learning, Knowledge Discovery, and Data Mining (FGML-2005) Learning - Knowledge Discovery - Adaptivity (LWA-2005)*, pages 164–171, 2005.

[12] J. Z. Kolter and M. A. Maloof. Dynamic weighted majority: An ensemble method for drifting concepts. *J. Mach. Learn. Res.*, 8:2755–2790, 2007.

[13] H. Mann and D. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50–60, 1947.

[14] K. Stanley. Learning concept drift with a committee of decision trees. CS Dept., University of Texas-Austin, 2001.

[15] N. W. Street and Y. S. Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *Proc. of the 7th ACM SIGKDD int. conf. on knowledge discovery and data mining*, pages 377–382, 2001.

[16] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen. Dynamic integration of classifiers for handling concept drift. *Information Fusion*, 9(1):56–68, 2008.

[17] J. van der Vorst, A. Beulens, W. de Wit, and P. van Beek. Supply chain management in food chains: improving performance by reducing uncertainty. *Int. Transactions in Operational Research*, 5(6):487–499, 1998.

[18] R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial Intell. Review*, 18:77–95, 2002.

[19] H. Wang, W. Fan, P. S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proc. of the 9th ACM SIGKDD int. conf. on Knowledge discovery and data mining*, pages 226–235, 2003.

[20] G. Widmer. Tracking context changes through meta-learning. *Machine Learning*, 27(3):259–286, 1997.

[21] G. Widmer and M. Kubat. Effective learning in dynamic environments by explicit context tracking. In *Proc. of the European Conf. on Machine Learning, ECML'93*, pages 227–243. Springer-Verlag, 1993.

[22] I. Zliobaite, J. Bakker, and M. Pechenizkiy. Towards context aware food sales prediction. In *Proc. of the 2009 IEEE Int. Conf. on Data Mining Workshops (DDDM-09)*, pages 94–99, 2009.