Context Aware Sales Prediction: experimental evaluation

Indrė Žliobaitė, Jorn Bakker, Mykola Pechenizkiy

Abstract

Sales prediction is a complex task because of large number of factors affecting the demand. We present a context aware sales prediction approach, which selects the base predictor depending on the structural properties of the historical sales. First, we learn how to categorize the sales time series offline into "predictable" and "non-predictable" based on the structural features of the series. Next, for the products identified as "predictable" we apply an intelligent base predictor, while for "non-predictable" we use a moving average. In the experimental part we show that prediction accuracy can be improved using this strategy, as compared to the base line predictor as well as an ensemble of predictors.

1 Introduction

Demand prediction is essential part of business planning. Accurate and timely sales prediction is essential for stock management and profitability. In food sales the stock includes large assortment of goods, some of them require special storage conditions, some are quickly perishable.

There are general and product specific causes of the demand fluctuation. The variations in consumer demand may be influenced by price (change), promotions, changing (rapid or gradual, global or local) consumer preferences or weather changes [18]. Furthermore, a large share of the products sold in that market is sensitive to some form of a seasonal change. Seasonal changes occur due to different cultural habits, religious holidays, fasting. All these factors imply that some types of products have high sales during a limited period of time.

Although seasonal patterns are expected, the predictive features that define these seasons are not always directly observed. Besides, the historical data is often highly imbalanced with only a few peaks per year. Thus food sales prediction is a challenging task. In addition, some product sales are regular and variations in sales are not directly affected by particular events, for instance salt, coffee. The fluctuations in sales might be of random nature.

A desired food sales prediction system should take into account seasonal triggers as well as be able to distinguish between seasonal and random fluctuations.

In this study we extend our work presented in [9]. We present two level sales prediction approach, where we first identify the nature of a given product sales ("predictable" or "random") and then apply the final predictor accordingly. We analyze the effects of different types of mistakes and report the improvement in final prediction accuracy.

The rest of the paper is organized as follows. In Section 2 we highlight the challenges related to food sales prediction. In Section 3 we present context aware sales prediction approach. In Section 5 we discuss the challenges and alternatives for evaluation of (food) sales prediction. Extensive experimental evaluation using the case of food wholesaler Sligro Food Group N.V. is carried out in Section 6. Section 9 discusses future prospects and concludes.

I. Žliobaitė, J. Bakker, M. Pechenizkiy
Eindhoven University of Technology
P.O. Box 513, NL-5600 MB,
Eindhoven, the Netherlands

I. Žliobaitė
Vilnius University
Naugarduko 24, Vilnius, Lithuania

2 Challenges of sales prediction

In this section we overview the properties of sales data and the nature sales prediction tasks. We address food wholesales prediction. However, the observations and methods discussed here can be generalized to other sales prediction.

Different horizons of the *food* sales predictions are required for business decisions. Variation in sales figures can be classified into short term fluctuations (e.g. party today, shopping tomorrow), medium term seasonal patterns (e.g. June vacations) and long term trends (e.g. economic situation). We focus on medium term seasonal patterns (weekly predictions), which are essential for stock management.

Moving average with different lag or simple regression models are often used as state of the art for food sales prediction. In this setting baseline predictions are often overridden by managers using their intuition and expertise. Predictions based on moving averages may work well when demand is flat. But when the demand follows trends or seasonal patterns the reaction of moving average is too slow. Managers often try to improve the performance in seasonal peak periods by prudently increasing the stock and thus costs.

Another typical approach is to have a number of reminders that should 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 and poor decision making.

We introduce a generic sales prediction approach with context awareness. We incorporate external information and behavioral observations to categorize the target series based on their structural properties and then add relevant external features for prediction.

3 Context aware sales prediction approach

In this section we present the intuition and implementation of context aware sales prediction approach (CAPA). By context we mean a phenomenon affecting sales of a product as a whole. Context awareness here means a mechanism that switches the final prediction models depending on what context is observed. First we give the general overview of CAPA and then explain individual parts of the approach.

The main idea of the context aware approach is to identify the context first and then use the predictor linked to this context. 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 offline.

Why not to collect all possible input features and then learn complex model expecting to incorporate the categories and the switch mechanism automatically. The task might be to complex w.r.t the available data. All the context features might not be feasible to observe or they might be not measurable. We believe 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.

3.1 Decision support with CAPA

In this section we present a snapshot (one time step for a single product) of the decision support approach.

CAPA consists of two blocks - training (offline) and operational (online). First of all we present operational online part and then describe, how the model is trained and parameterized offline. Let us assume that the model has already been trained offline:

1. the categories have been fixed;
2. mapping of time series to the categories is established;
3. “local” expertise of each predictor is known.

Figure 1 presents online operation of CAPA.

Let us take a particular product we are interested to predict online. First of all we extract structural features from the original sales time series of the product. Then we assign the product to one of the categories. We pick a base predictor linked to a particular category. That is the context aware part of the approach. Having the original series, the base learner we can form an input feature space and cast the prediction. Note, that the predictor does not use the same features as were used for assigning the product to a category.

The prediction output now can be used for a decision making in a business process.

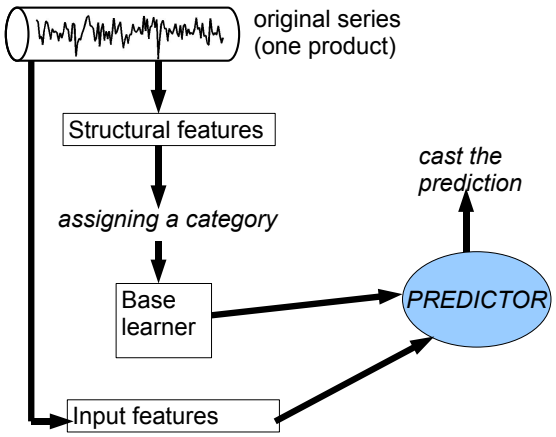


Fig. 1. Online operation of context aware prediction approach (CAPA).

3.2 Building CAPA

The core part of context aware prediction approach is to match the product categories and the base predictors.

Offline part. A limited set of base predictors $(G_1, G_2, \ldots, G_m)$ needs to be preselected based on domain knowledge and expectations in order to delimit full state space search. Then for each product m parallel experiments are carried out using each of the predictors. Next, the products are grouped into m categories based on the best performing predictor. Each obtained category serves as a basis for constructing categorization rules.

Online part. The goal of the training process is to learn to assign a product to one of these defined categories online, having only a fragment of the series. When we have the categorization rules, an unseen product can be processed as described in Figure 1. First of all the category of the product is determined, say C_j. Then the corresponding predictor G_j is used to output the prediction.

4 CAPA Implementation for Food Sales Prediction

In this section we present design choices and implementation of CAPA for weekly food sales prediction.

4.1 Product Categorization

Some products are subject to seasonal or occasional fluctuations in sales, while other products exhibit more or less flat sales or random fluctuations. Thus we distinguish two product categories: "random" and "predictable". We assume that within the "predictable" group the fluctuations in sales should be explainable by external features, such as weather, calendar events, school, religious holidays. This information can be encoded into input features and used as an input for an *intelligent* predictor. On the other hand, we assume that the sales of "random" products are in general independent of these explanatory features, thus a moving average of the historical sales would be an optimal predictor.

The main challenge in this approach is how to distinguish between the "predictable" and "random" products. We consider two alternative categorization mechanisms: cross validation and meta learning approach.

4.1.1 Meta Categorization

Meta learning is aiming to learn the relationship between the performance of base learners and the properties of datasets [1]. In our case historical ales of one product can be regarded as one dataset. In order to learn a mapping between the products and the base predictors, we need to define higher level features of the datasets, which we refer to as *structural features*. They should be extractable from the input products online and are desired to be length independent. In addition, we want the structural features to be related to observable seasonal patterns.

We define three groups of such features: behavior, shape and relational features. We normalize the values of the series to be in a range $(0, 1)$ before extracting structural 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-s3} |mean - median|, standard deviation, shift;
- F_{s4-s8} threshold h crossing ratios;
- F_{s9-s10} normalized power of the frequency p in the frequency spectrum;
- $F_{s11-s12}$ local variation features: interquartile range and unequal neighbors.

On the contrary to the behavior features, shape features are expected to align the information about the shape patterns of the historical time series. These features are dependent on the series length:

- $F_{s13-s17}$ quadruple SAX representation of the series [10].

The relational features capture correlation with the external features:

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

The features F_{s1-s3} capture global characteristics of the series y. Shift is the mean number of points for which $y_t < \mu$ minus the median of the number of points for which $y_t < \mu$, where μ is the mean of the time series.

In order to capture the structural behavior of time series, we define a number of threshold values and note the number of times the signal crosses these thresholds (features F_{s4-s8}). This is done for the threshold values $h = 0.5 \pm 0.1, 0.7, 0.3, 0.8, 0.2$. This number is then scaled to the total length of the signal to obtain the ratio. This ratio is an indicator of the structural nature of the signal.

Seasonal patterns could manifest themselves as, for instance, yearly or bi-yearly changes. This information should appear in the frequency spectrum of the time series as a relative high power in the frequencies $p = 1/52$ and $2/52$. In order to obtain these features we apply Fast Fourier Transformation (Cooley-Tukey implementation [2]) and extract the corresponding frequencies (features F_{s9-s10}).

We aim to capture local variation using features $F_{s11-s12}$. Unequal neighbors is the mean number of times $y_t \neq y_{t-1}$. The interquartile range of $y_t - y_{t-1}$ is a robust measure for the spread of variation in the signal. Any outliers that fall in the upper or lower 25% of the difference distribution will not affect it.

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

By correlations $F_{s18-s26}$ we aim to capture the impact of external factors to fluctuations in the sales.

We do not concentrate on the features related to the seasonality of time series, such as autocorrelation. We believe that in product sales seasonality depends on calendar or weather triggers which are not strictly periodic in terms of time.

4.1.2 Learning Meta Categorization

We would like to learn categorization rules, which would assign a given product to one of the defined categories based on its structural features. We take *bottom up* approach: we use the testing accuracies to label the training products and using these labels try to learn categorization in a supervised manner.

In order to test whether the category assignments are learnable, we train a classifier on the "true" labels of the categories generated using the training dataset. The labels are obtained by running all the classifiers from the pool for all the products and then ranking the accuracies for each product. A product gets the label, corresponding to the best performing classifier.

If the categorization is learnable we should be able to assign an optimal predictor based on the structural features, such that the intelligent predictor methods perform better than the baseline.

We train a classifier in a supervised manner to make the mapping from structural features to the class label. We use the 26 structural features described in previous section.

4.1.3 Categorization by Cross Validation

An alternative to meta categorization approach is to categorize the products by testing the base predictors on the training part of the series. Here it is assumed that the "predictable" or "random" behavior of the series does not change in time. The training part of the series is divided into k bins. Both intelligent and moving average (MA) predictors are tested using k-fold cross validation. If intelligent predictor outperforms MA on average, then the product is categorized as "predictable", otherwise it is categorized as "random". Note that by employing cross validation we mix the time order of the instances, thus stationarity assumption is essential in this approach.

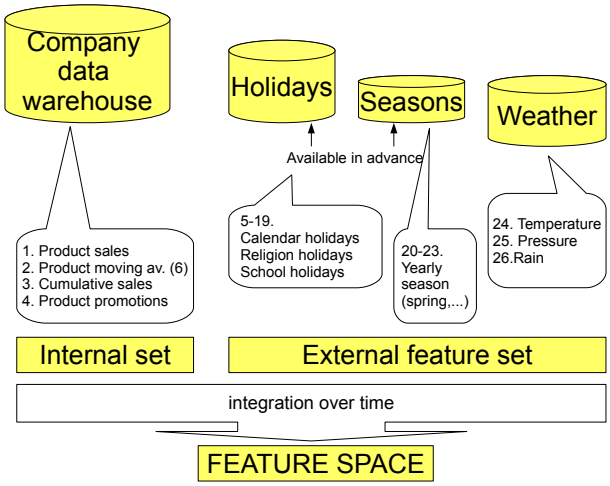


Fig. 2. Formation of the prediction feature space.

4.2 Input Space Construction for Final Prediction

In previous section we discussed extraction of structural features, which are used for time series categorization (see Figure 1). In this section we discuss the input space for final predictions, which are different from the ones, used for categorization.

The feature space is formed using internal an 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.

The internal features are interrelated. Moving average (F_{p2}) is calculated using (F_{p1}). Cumulative sales (F_{p3} include the sales quantity of all the products. We checked corresponding series in monetary sales terms to verify that there is no significant distortion due to different quantity measures. Promotions (F_{p4}) for some products are organized.

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.

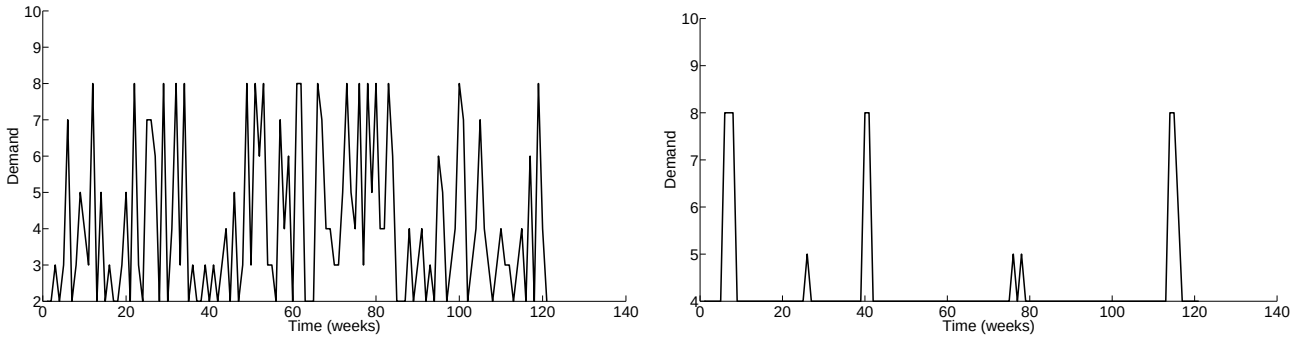
We highlighted CAPA design for food sales prediction. In the next section we will provide experimental evaluation and present the particular choices of the parameters and models.

5 Evaluation of (food) sales prediction

Before turning to the evaluation of the experimental results of the categorization, we need to address the evaluation of food sales prediction. The evaluation of predicted demand is not straightforward. The evaluation of time series in general is not straightforward [7] due to the fact that time series are in principle unbounded (both in range and domain). Widely accepted measures of performance for time series predictions are not robust for all types of time series. The specific case of food sales prediction, however, introduces some additional challenges.

The first challenge is the large number of different sales behaviors of products. While the sales of some products are highly dependent on seasonal influences (e.g. ice cream), other products seem to exhibit no structural pattern at all. And some are only sold at certain occasions. In figure 5 two examples of different sales behavior are given. In order to analyze the performance of a predictor on these products we require the performance measure to be intuitively comparable over all products. Without this constraint, it is not possible to aggregate the performance results over all products. We will argue that performance measures based on unscaled loss functions can not be aggregated over all products without losing information.

The second challenge is is specific to the food sales decision making process. The purpose of making predictions in the food sales industry is to balance the inventory such that the overall profits are maximized. Although the prediction process



(a) Frequent (b) Occasional

Fig. 3. Two different products

itself is studied here as if it where an open loop process, in the wild it is a part of a closed loop process. This means that the outcome of the prediction process has an impact on inventory update policies. This also means that a prediction cast at time t does not only have an influence on the policy for time t or $t+1$ but possibly also for policies in the further future($t' >> t$). In short, the different "types" of error made by the predictors have a different impact on the underlying model.

In order to solve these challenges, we will introduce changes to the standard evaluation measures. First we will argue that the aggregation problem can be solved by using relevant baselines. Second, we illustrate that a possible solution to the time dependency of the predictions is to use utility based measures.

5.1 Performance aggregation

Evaluation using an unscaled error function (e.g. MSE, MAE) is not informative in the case of food sales prediction. To illustrate this, consider the two products given in figure 5. One product has a behavior with high frequency fluctuations(figure 3(a)), where the other has a very regular pattern(3(b)). We could construct a very simple predictor that predicts always the same demand. In the case of the irregular pattern this results in a high MSE, whereas in the regular pattern the predictor has a relatively low MSE with respect to the true time series. The reason for this difference lies in the fact that the regular pattern has constant value for more than 90% of the time.

This difference in evaluation implies that the error function will yield performance values that not directly comparable over different products. If we apply a given predictor method on the two time series in figure 5, the same problems arise. If we then take the average of the two computed errors, the aggregated error loses the information that the good performance on the regular product is actually misleading.

In order to compute a more intuitive performance measure a baseline is needed. For instance, the predictor that always predicts the same value is a bad predictor from domain perspective, but it can serve as a lower bound of performance. In the following sections we will discuss several error measures and investigate useful baselines.

5.2 Error Measures

Error measures that have been proposed in the literature [7] and were commonly applied for evaluation of time series forecast include:

- Mean Squared Error: $\mathit{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2$,
- Root Mean Squared Error: $\mathit{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$,
- Mean Absolute Error: $\mathit{MAE} = \frac{1}{n}\sum_{t=1}^{n} |e_t|$,
- Mean Absolute Percentage Error: $\mathit{MAPE} = \frac{1}{n}\sum_{t=1}^{n} |p_t|$,
- Mean Absolute Scaled Error: $\mathit{MASE} = \frac{1}{n}\sum_{t=1}^{n} \left| \frac{e_t}{\frac{1}{n}\sum_{t=1}^{n} |y_t - h(t)|} \right|$,

where the error $e_t = y_t - f(t)$, y_t is the actual value, $f(t)$ is the predicted value by model f at time t, $h(t)$ is the predicted value by baseline h, and the percentage error $p_t = \frac{e_t}{y_t}$.

Both *MSE* and *RMSE* are well known and widely used to test the accuracy of a model. In the machine learning field these measures are used to evaluate the performance. In the forecasting of time series, however, they are deemed not so suitable because of the aforementioned scaling differences and the sensitivity to outliers.

The *MAPE* has been recommended for measuring accuracy among many different time series. However, it should be noted that in cases where y_t is very close to zero, the resulting *MAPE* will become infinite or invalid. The same holds for the *MASE*, i.e. in case the *MAE(Baseline)* is close to zero, but this case is special in the following sense. The *MASE* uses, in contrast with the other error measures, explicit scaling with respect to some baseline. Notice that if *MAE(Baseline)* is close to zero, the baseline itself is a good predictor. The advantage of *MASE* is that the accuracy of a given model can directly be related to the baseline regardless of scale. What is left is to choose the right baseline.

5.3 Qualitative evaluation

The uninformed (or naive) prediction strategy is a good lower bound baseline for qualitative analysis of predictors. This strategy just takes the current value y_t as the prediction for y_{t+1}. From the domain perspective this is the layman's prediction, as one needs no other information than y_t. Since this predictor will get a relatively high score on the occasional series, it is very suitable to scale the performance of other predictors.

Using the *MASE* and the naive baseline we can construct a ranker as follows:

$$MASE_{naive} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{e_t}{\frac{1}{n}\sum_{t=1}^{n}|y_t - y_{t-1}|}\right| \tag{1}$$

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.

5.4 Quantitative evaluation

The same naive baseline can be used to compute a quantitative aggregated performance measure. Since we are only interested in changing parts of the time series, it is also an option to just disregard the areas in which the demand does not change. In a sense the naive prediction is then filtered out of the signal. Over the remaining part it is possible to calculate an absolute error measure given that the maximal error can be found.

The first part of this measure is a filter based on the naive prediction. The requirements of filtering out certain instances are:

i) the last actual value y_{t-1} is equal to y_t (naive prediction), and
ii) the error $e_t = 0$.

The last requirement is needed to ensure that the error measure remains fair. If it is left out, the performance will increase where in fact the prediction makes an error.

The second part is another baseline that always predicts the worst possible outcome (i.e. maximizes e_t). This is only possible if the labels $\overline{y}$ are bounded. In case of a discrete space, for instance, the maximal error can be computed by:

$$f_{WC}(t) = \frac{1}{n}\sum_{t=1}^{n}(y_t - \max_{i=1}^{\alpha}|i - y_t|)^2, \tag{2}$$

using the MSE as the error measure. Given this baseline and the filtered instances the absolute error, or Scaled Mean Squared Error, can be computed by:

$$SMSE = \frac{\frac{1}{n}\sum_{t=1}^{n}(y_t - h(t))^2}{\frac{1}{n}\sum_{t=1}^{n}(y_t - \max_{i=1}^{\alpha}|i - y_t|)^2} \tag{3}$$

Since this error measure only measures at "interesting" instances and it scales the absolute error to an upper bound, it can be used as an absolute aggregated error measure for predictors over all products.

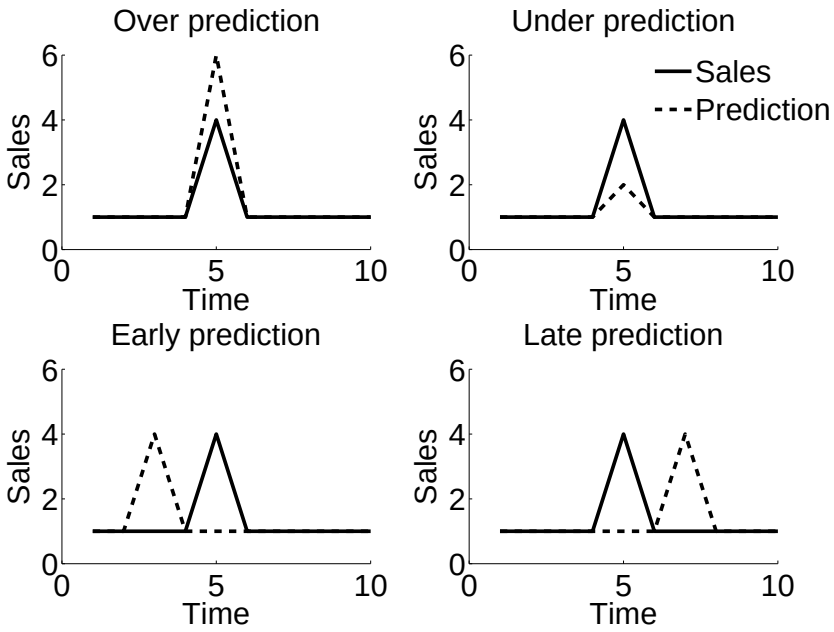


Fig. 4. In the food sales domain the above shown error "types" have very different meaning.

5.5 Error types

The effect of time dependency in the evaluation of predictors has not received a lot of attention. In the general case of a data stream this dependency is not present, since the data generating model is not influenced by observing the data. In the case that the outcome of the predictor is used as a decision support mechanism, the predictor (in)directly influences the outcome of future predictions. Since in that case we can not clearly evaluate the predicted instances individually, the evaluation should be based on historic data.

The problem is best illustrated by a few simple examples. In figure 4 four "types" of errors are shown. The first case is the difference between over(top left) and under(top right) prediction. Although the absolute error ($|y_t - f(t)|$) is in both cases the same, there are different processes associated with these errors. In the case of under prediction, a company might lose profit because of the too conservative estimate. In the case of over prediction, products that are perishable have a higher risk of becoming obsolete. There is a difference in impact on the underlying model, despite the fact that any absolute error measure does not represent this.

There is also a significant difference between early(bottom left) and late(bottom right) prediction. Predicting a rise in demand too late is very bad due to possible overstocking. Predicting the peak too early, however, might not be as bad as suggested by the error function. If an over prediction is immediately followed by a rise in demand, the net result is that the products are still sold. Hence, there is a huge difference between these two "types" of error.

5.6 Time dependent evaluation

The error measures mentioned in the previous sections do not capture these differences. In order to make any claim about the usefulness of a prediction in terms of the impact on the decision making process, we need additional evaluation measures. Measures that are sensitive to the different "types" of error in figure 4. Here, we discuss two quantitative measures that can visualize the occurring "types" of errors.

The main challenge of the decision making process is to balance the costs accompanying a particular decision. For a lot of cases the behavior of the time series is chaotic and, therefore, it is risky to always employ a greedy strategy. In some cases it is better to stick with a predictor like the moving average because of its conservative nature. In light of the costs associated with strategy choice, we take a closer look at how the errors relate to the types of time series.

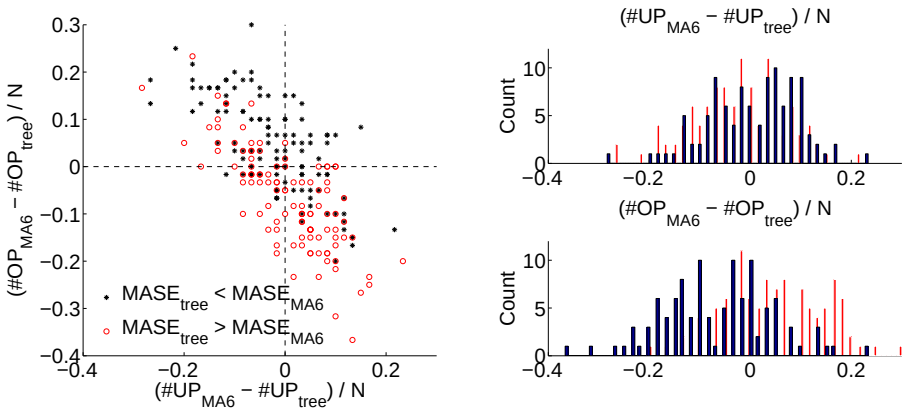


Fig. 5. Difference in Over Predictions (OP) and Under Predictions (UP) of MA6 and the decision tree.

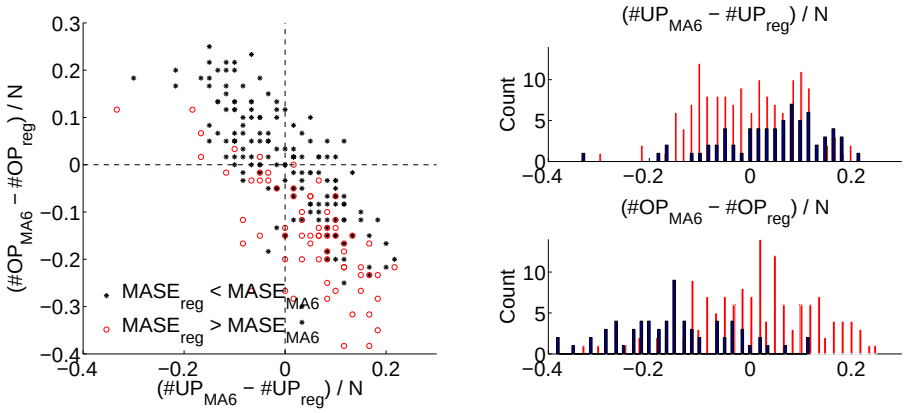


Fig. 6. Difference in Over Predictions (OP) and Under Predictions (UP) of MA6 and the regressor.

5.6.1 Over vs under prediction

The difference between under and over predictions can be visualized by scoring the errors. For each predictor we can score the number of under predictions (or $y_t > f(t)$) and over predictions ($f(t) > y_t$). By comparing these two measures between predictors we can construct a view on the "types" of errors that each predictor is prone to make. This can only be used as a quantitative measure.

In figures 5, 6, and 7, the scores are given for all the products in the training set. For each product, the number of over and under predictions is calculated for both a predictor (decision tree, regressor, and a random predictor) and the moving average of six instances (MA6). In the graph the difference between the MA6 and the predictor is plotted scaled to the number of instances (N). The upper left quadrant represents the area in which the moving average has both more under and over predictions. Consequently, the right lower quadrant signifies that the moving average has both less under and over predictions than the predictor. Since a number of scores are overlapping, on the right side the distributions of the scores along the axes are given. Each star represents a product for which the $MASE_{predictor} < MASE_{MA6}$ and each circle represents the case $MASE_{predictor} > MASE_{MA6}$.

From these results it can be seen that there is a sensitivity of the moving average towards over predictions. The error gradient runs over the diagonal $x = y$, since the scores are directly related to the error. In the lower right quadrant, the number of under predictions are relatively high (above 0 with respect to the predictor), and the number of over predictions low. In this quadrant there are more products for which the moving average has a lower $MASE$. In the upper left corner, however, it is the other way around. If the moving average makes more over predictions than the intelligent predictor, the number of products for which the moving average wins (see table 1) is relatively low. This suggests that the relative sensitivity to each type of error is different.

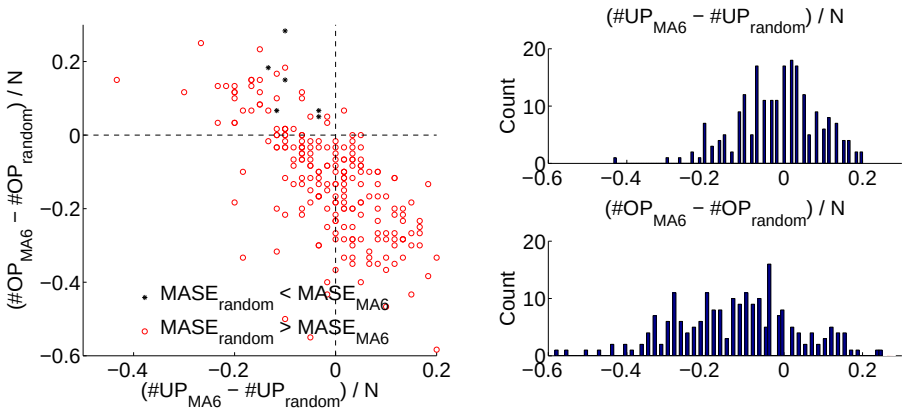


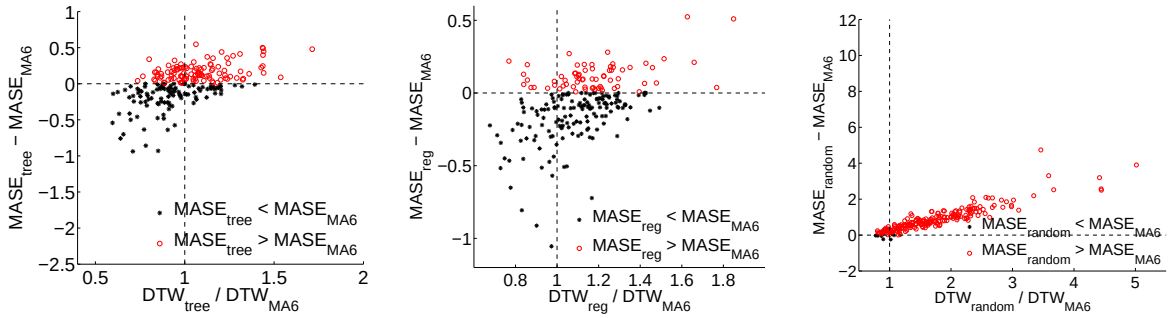
Fig. 7. Difference in Over Predictions (OP) and Under Predictions (UP) of MA6 and random predictions.

Table 1. Number of wins ($MASE_{predictor} > MASE_{MA6}$) for each quadrant in figures 5, 6, and 7.

	Random			Regressor			Decision tree		
	Wins	**Total**	**ratio**	**Wins**	**Total**	**ratio**	**Wins**	**Total**	**ratio**
+, +	0	1	0	16	16	1	15	15	1
+, -	0	91	0	56	103	0.54	23	74	0.31
-, +	6	42	0.14	59	65	0.91	50	71	0.70
-, -	0	61	0	9	20	0.45	11	32	0.34

5.6.2 Late vs early prediction

The difference between early and late predictions can be computed by taking the Dynamic Time Warping distance(DTW) between the true labels and the predicted labels (see figure 8. The DTW computes the minimal amount of shifts needed to align two time series. In other words, if two time series differ significantly in the spaces between peaks, this leads to a high DTW distance. Hence, this measure can be used to visualize the discrepancies over time.



(a) Decision tree (b) Regressor (c) Random predictor

Fig. 8. Difference in $MASE$ against the DTW ratio of predictor to the moving average

This measure can be used to compare the sensitivity of predictors to time related errors as can be seen in figure 8. This figure shows the ratio of the DTW applied to the intelligent predictor with respect to that of the moving average (values below 1 means the DTW distance of the intelligent predictor is lower than that of the moving average). Against the ratio the difference in $MASE$ is shown(f it is negative it means that the intelligent predictor is better). Again, the error gradient should run along the diagonal $x = y$.

There is a lot of variation, however, along the diagonal. This is somewhat expected, since we just argued that the $MASE$ does not necessarily encapsulate the time dependent evaluation. In general, there is an observable trend between the DTW distance and the error measure. In order to make an effective evaluation of the predictions, the accompanying costs need to be evaluated.

5.7 Implications for food sales prediction

We have illustrated that straight forward evaluation in food sales prediction is not possible without loss of information. Due to time series specific issues, predictors should always be compared to a baseline. Due to food sales domain specific issues, time dependent evaluation is paramount. For the food sales domain it means that evaluation can not be seen separately from utility analysis.

This is also relevant to the context aware approach of prediction. For instance, for a chaotic sales behavior the moving average might be the best predictor, since it behaves as a smoothing function. For this to be apparent, the actual impact on the risks and benefits associated with the prediction needs to be quantified.

6 Experimental evaluation

The main goal of the experiments is to investigate the relation between the categorization accuracy and final prediction accuracy.

6.1 Prerequisites

We experimentally analyze the proposed approach using the data from Sligro Food Group N.V. (SLIGRO). The company is engaged into 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.

6.1.1 The Data

Our experimental field consists of 537 products over two years period (from July 2006 to October 2008) at Eindhoven outlet (?). The sales are aggregated on weekly basis, thus each series is of 119 weeks length. Each series represent the sales quantities of one product.

Product filtering. Not all products are equally interesting from both domain and research perspective. Products are subject to changes in policy, changes of supply, and changes of customer interest. These factors might result in a product being discontinued or being introduced to the stock (see Figure 6.1.1b). This can be regarded as a form of concept drift and for the handling of concept drift an additional mechanism is needed. Since this falls outside of the scope of the work presented here, products that exhibit this behavior will not be part of the data set.

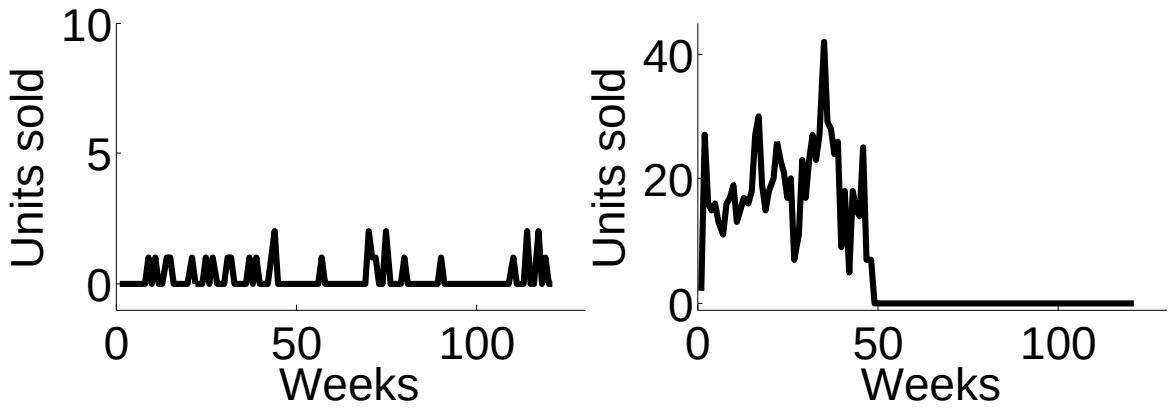
There is also a big variance in the average sales volume between products. Some products are very popular, whereas others are only sold once in a while (see Figure 6.1.1a). The products that have a low sales volume are relatively uninteresting from the domain perspective. Moreover, since low sales volumes also indicate lack of information, they will also not be part of the data set.

In short, products that have an anomalous behavior will not be selected for the experiments. This means we should construct criteria based upon which products will be accepted or not. The products that exhibit sudden shifts in volume are filtered out by visual inspection. Products that have low sales volumes are selected by the following criteria:

- the maximum number of sold items in a week exceeds a given threshold θ and
- the mean number of items sold in m weeks, given by $\frac{\sum_{t=1}^{n} \frac{\sum_{i=t}^{t+m} y_i}{m}}{n}$, exceeds a given threshold θ'.

In the procedure we used $\theta = \theta' = 10$ and $m = 4$. Products that never exceed a sales volume of 10 items and products that have an average sales volume of less than 10 items a month are deemed uninteresting.

Experimental set. For experimental evaluation we have 330 products, which we divide into training-validation (220) and testing (110) baskets at random. Testing basket is needed to test for categorization accuracy, categorization rules are learned



(a) Low volume (b) Concept drift

Fig. 9. Examples of filtered out products.

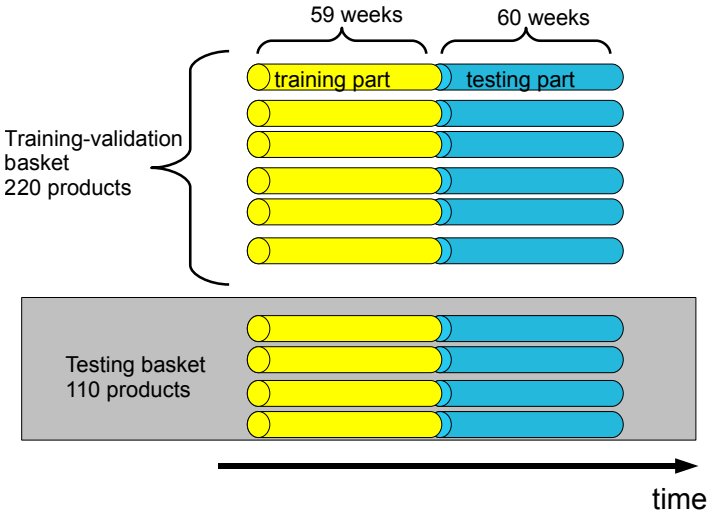


Fig. 10. Experimental division of the dataset.

on training basket. Each individual product is split into training and testing part in time. Testing part consists of 60 weeks, thus 59 are left for training. Training part is used for categorization and learning the final predictor. Then sequential learning procedure is employed to record the final predictions. The experimental division of the dataset is illustrated in Figure 10.

Sequential testing means the predictor is retrained at each time step. At time t we have all the historical sales up to this point available and want to predict sales level at $t+1$. After the prediction is casted, the value from time $t+1$ is included into historical set and then we proceed with predicting the value for time $t+2$. 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.

We use discretized target values (labels). We operate ordinal levels of sales discretized into 8 bins using the following thresholds:

1. more than 75% decrease in sales w.r.t. to the mean,
2. $75 - 50\%$ decrease,
3. $50 - 25\%$ decrease,
4. $25 - 0\%$ decrease,
5. $0 - 25\%$ increase,
6. $25 - 50\%$ increase,

7. $50 - 75\%$ increase,
8. more than 75% increase.

From the domain perspective the interest is in bulk levels of sales rather than very precise quantities.

6.1.2 Evaluation

For model evaluation we use *Mean Absolute Scaled Error* (MASE) [8]:

$$MASE = \frac{1}{n}\Sigma_{t=1}^{n}|\frac{e_t}{MAE(Baseline)}|, \tag{4}$$

where e_t is the prediction error at time t, $MAE(Baseline)$ is the mean absolute error of the baseline method. We use naive one step ahead prediction as the baseline (the prediction for the next week is equal to the factual sales this week). This is a moving average, when the lag is equal to one.

Using MASE as the accuracy measurement, we compare the intelligent classifiers to the baseline method.

6.1.3 Alternative techniques

For final prediction we test the following alternative techniques.

1. Moving average of 1 as a naive predictor (next week sales are predicted to be the same as this week).
2. Moving average of 6 as state of the art currently used by the company.
3. Decision tree using multidimensional features (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 we test two alternatives:

- cross validation on training part of each product, and
- meta learning approach using structural features of the time series.

6.2 Experimental set up

Experimental scenario consists of three parts: selecting the base predictors, learning categorization rules and testing final model accuracies. In the first part we analyze the performance of alternative base predictors on 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 learn the accurate dependencies using two approaches *meta product selection* and *cross validation*. In the third part we present and analyze the final prediction accuracies.

7 Experimental results

In this section we present and discuss experimental results.

7.1 Selection of the base predictors

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

We use the input features (26) as specified in Figure 2. For feature selection we calculate the correlations of the features with the labels. We select the features which have the absolute correlation 0.25 or higher.

We run all the predictors on the testing-validation basket (220 products). We split each product into two parts along the timeline. 59 weeks are used as a "warm up" and then the remaining 60 weeks are used for sequential testing. In Table 2 we list average MASEs of all the methods (the lower the better). In the same table we provide 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".

Table 2. Average MASEs and ranks on the training-validation product basket.

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

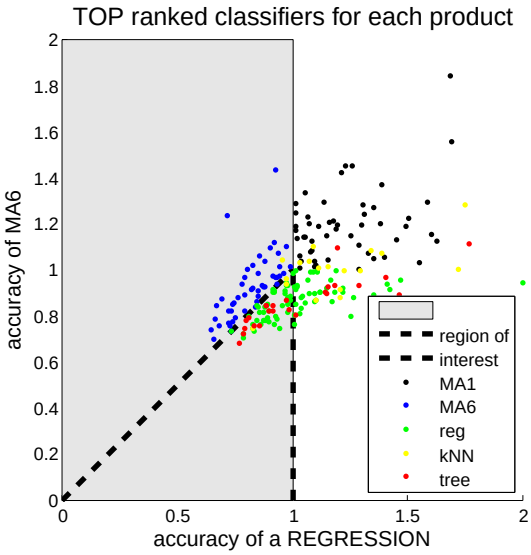


Fig. 11. The prediction accuracy on the training-validation basket.

The regression outperforms other predictors in terms of average rank as well as slightly outperforms the baseline moving average 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 context switching. It is important to be able to identify the products on which the baseline might be outperformed significantly by an intelligent predictor.

7.1.1 "Predictable" products

Let us have a closer look at the distribution of prediction accuracies among the products. In Figure 11 we depict all the products from testing-validation basket. The accuracy of an intelligent predictor is depicted against the accuracy of moving average of 6. The gray area indicates the accuracies of an intelligent predictor below 1, which mean they outperform the baseline predictor MA1. This is the group which we call "predictable" products. The triangle indicates the cases when the intelligent predictor outperforms both the baseline MA1 and the state of the art MA6. Let us call this subgroup "outpredictable" products. These are the products we are interested to learn to categorize.

The relation between the groups is all products $\in$ "predictable" products $\in$ "outpredictable" products. The proportions of these groups in the training set are $100\%, 65\% and 49\%$.

7.1.2 Competitive advantage of the intelligent predictors

The ultimate goal of from the 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 sales if a particular product was out of stock at the time. Thus overprediction errors can be treated "'softer" than underprediction. Because there is a possibility that the model might have been right by overprediction, just there was not enough stock for that to show up in sales figures.

Let us have a look at two different types of mistakes the predictors make. We still keep the accuracies scaled to the baseline (MA1) for comparability reasons. In Figure 12 we depict the overprediction and underprediction errors on the training-validation product basket.

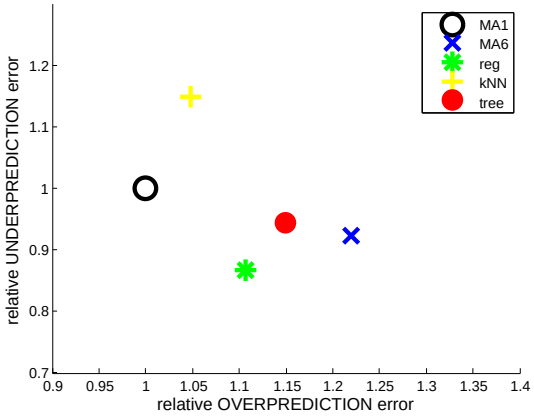


Fig. 12. The prediction accuracy on the training-validation basket.

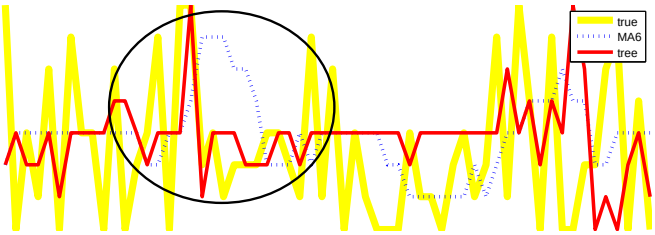


Fig. 13. Predictions of an intelligent predictor versus moving average for a single product.

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 MA6. It suggests an intuition that MA6 is slow at times of rapid changes in sales.

Let us zoom in a particular case where intelligent predictor outperforms the moving average. In Figure 13 the ellipse indicates the area where moving average of 6 is slow to react to the changing demand, while the intelligent regression tree catches the decrease in sales right away.

7.1.3 Selected features

We employ feature selection mechanism based on correlation. On average 8.3 features from 26 are selected. In Figure 14 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 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.

7.2 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 obviously non stationary cases from the data as described in Section 6.1.1. Thus in this study we determine the category of a given product once using the training part of the series. Then we employ sequential learning and prediction procedure, assuming that the category of a given product is fixed.

A product can be assigned to the "predictable" or "random" category depending on the (expected) performance of intelligent and baseline predictor. In order to learn the categorization rules we need to assign the "true" labels to the products. For

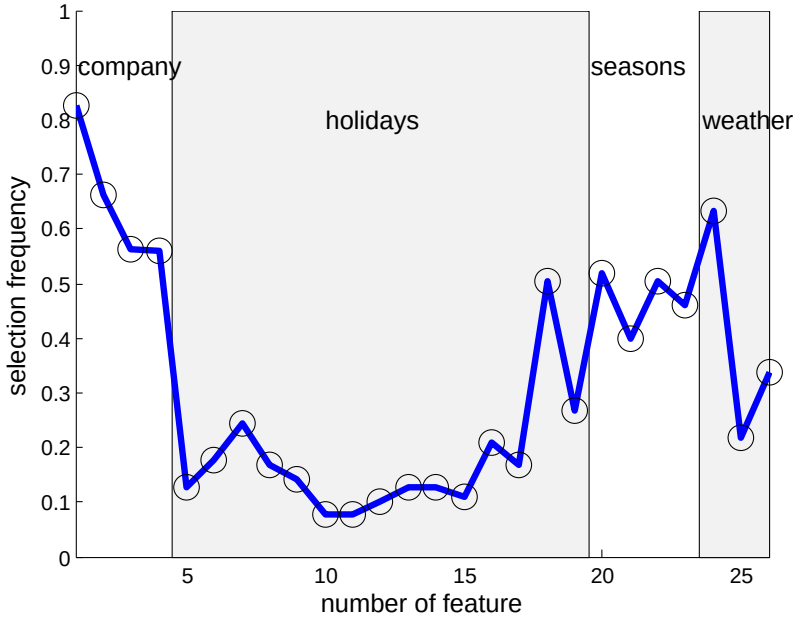


Fig. 14. The selected features.

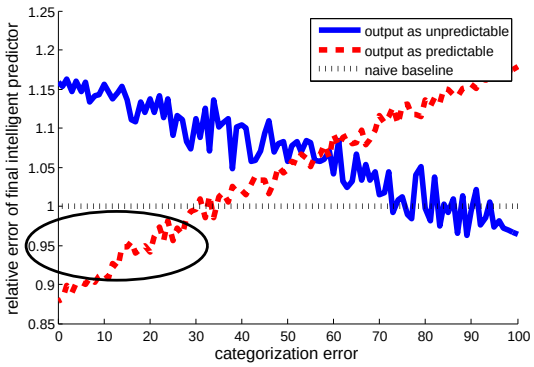


Fig. 15. Relation between categorization accuracy and final prediction accuracy.

that first we need to fix the intelligent and the naive predictors and then compare their accuracies on a given product. If the intelligent predictor on average outperforms the naive, the product gets labeled as "predictable" and it is labeled as "random" if the results are vise versus.

We fix MA1 as the naive predictor and regression tree as the intelligent predictor. This way we get a meta dataset consisting of 220 products having binary labels. The proportion of the "predictable" class is 43%.

7.2.1 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. Let us inspect the relationship between the two with the following experiment. We divide the products into "predictable" and "random" categories, but we control the categorization accuracy. When we use all the true category labels, the categorization error is 0%. Next we increase categorization error, e.g. we select 10% of products at random and swap their labels. This way we get 10% categorization error.

In Figure 15 we plot the categorization error against the relative error of the final prediction by the intelligent predictor (tree) for the two categories. The red curve represents the "predictable" group. The benchmark at 1 represents the alternative

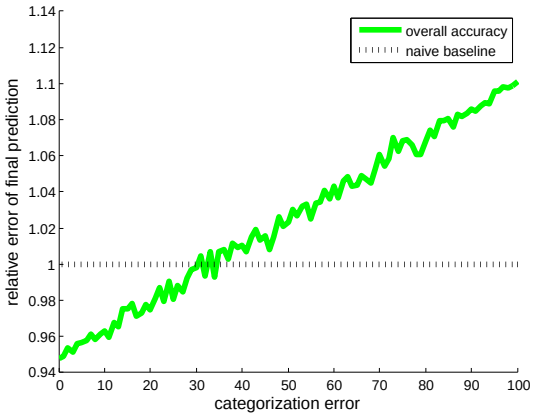


Fig. 16. Overall final prediction accuracy.

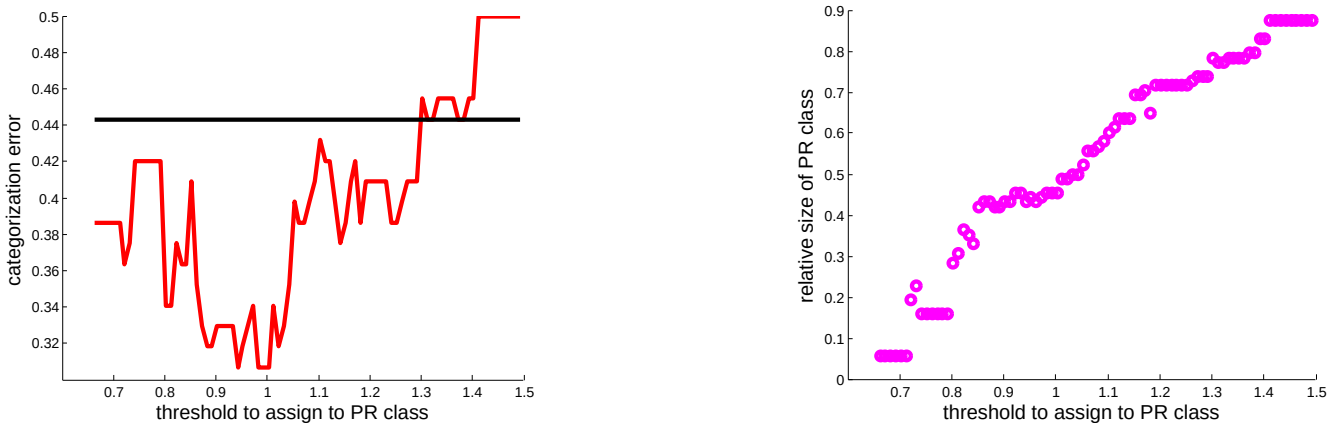


Fig. 17. Meta categorization: sensitivity to the training thresholds.

relative error when using the naive predictor. Thus if we are able to achieve average categorization error less than 40%, we would apply the final intelligent predictor for the "predictable" category and the naive predictor for the "random" category.

Let us see what is the corresponding effect of categorization error on the overall final prediction accuracy. We present it in Figure 16. The final prediction accuracy of CAPA (switching between the intelligent and naive predictors) is linearly dependent on the categorization accuracy.

7.3 Meta - categorization approach

Meta categorization is categorization using structural features, defined in Section 4.1.1. In order to learn the categorization rules, we first need to choose the base classifier. After preliminary experiments on the training data we choose a Naive Bayes classifier, which is not complex and appears to be powerful in this meta learning case.

The second important design choice is to choose the threshold for classification. The intelligent predictors showing final relative prediction error around 1 might not be significantly different from the naive predictors. We want to be able to regulate the level of certainty in the "predictable" category. For instance, if the categorization rule is able to identify only one the most "predictable" product, but this product comprises significant amount of total company sales, it is worth doing.

We do sensitivity analysis on the thresholding of the meta labels (training part). We split the training-validation basket into training (60%) and validation (40%) parts. The size of validation set is 88 products. In Figure 17a we depict the label threshold options against the categorization error.

It can be seen that minimum error is achieved between 0.94 and 1.00 label threshold. In Figure 17b 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".

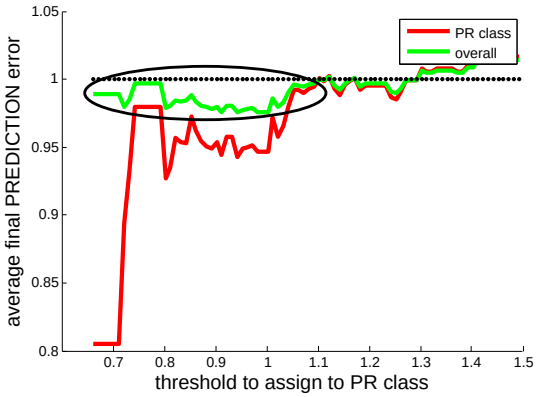


Fig. 18. Meta categorization: overall final prediction accuracy.

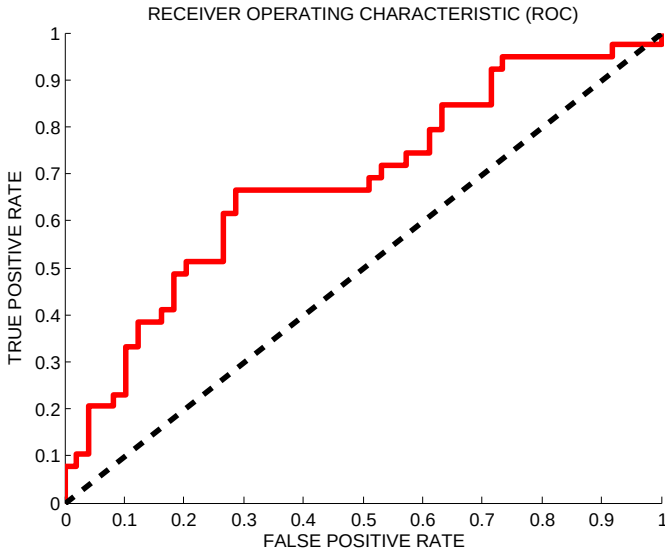


Fig. 19. Meta categorization: ROC curve.

Let us look what are the corresponding final prediction accuracies. In Figure 18 we plot the final 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 17b. The result is intuitive, since in case of very low threshold less products are picked as "predictable", but the classifier is more "certain" as the final 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 variable label threshold is similar to the concept of ROC curve. It allows the decision maker to balance and rebalance 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 17 appeared to be one of the optimal thresholds. Then the ROC curve for Naive Bayes categorizer is plotted in Figure 19. The area under ROC (AUC) is 0.68.

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

The discrimination between "predictable" and "random" products is achieved using structural features. Let us see how different the structural 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 20a average values of the structural features for *true* "predictable" and "random" product categories are depicted. In Figure 20b we depict average features of the categories output by Naive Bayes categorizer.

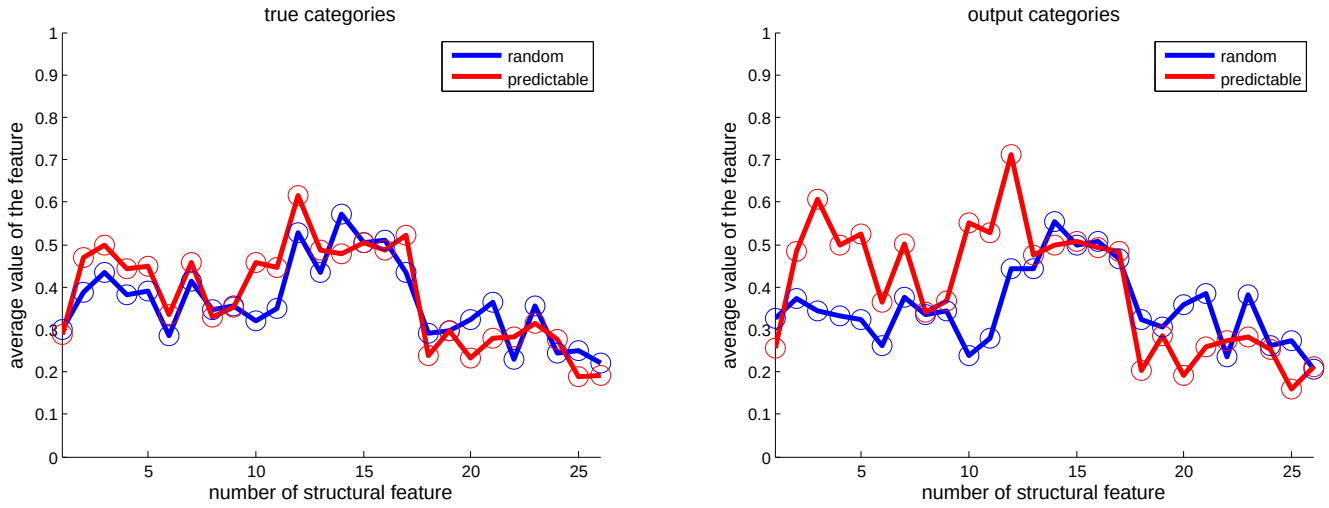


Fig. 20. Average structural features for the "predictable" and "random" product categories: a *true* **categories, b output by Naive Bayes categorizer.**

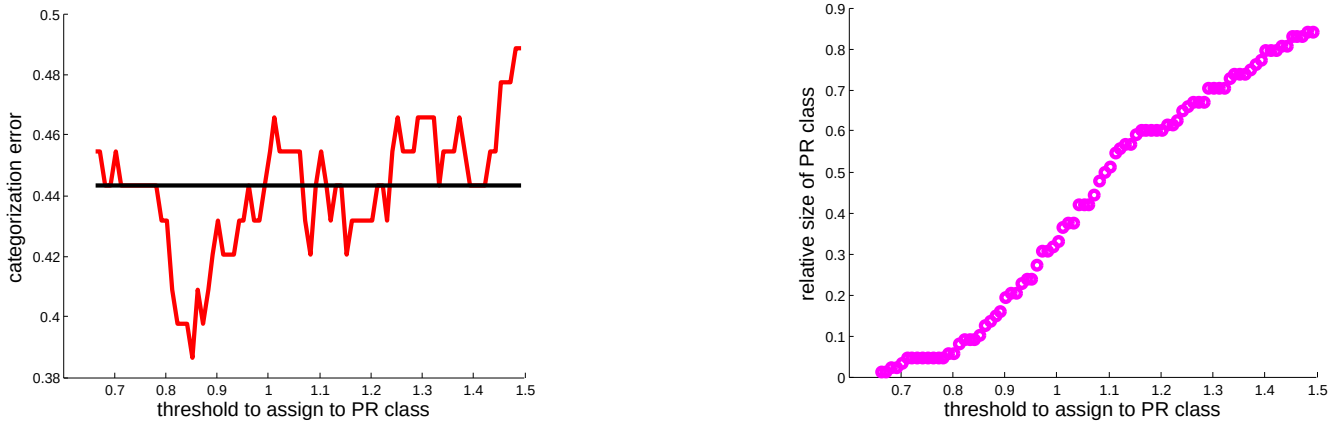


Fig. 21. Cross validation: sensitivity to the training thresholds.

The plot for the *true* categories indicate that the categories are on average separable by the chosen structural features. The plot for the output categories indicate that the categorizer has done good enough.

7.4 Categorization by cross validation

An alternative for meta learning approach is categorization by cross validation on the training part of the series. We use 10 fold cross validation and average over the final accuracies. It is important to observe the order of the instances, since we are measuring MASE, which depends on the order of instances.

We assign a product to "predictable" is average cross validation MASE is < 1. For sensitivity analysis we replace 1 by variable threshold to produce sensitivity analysis, similar to the ones we showed for meta categorization. ROC curve is not applicable here because there is no categorization model built, the categorization process is data driven.

Finally we overlay the final results of meta categorization and cross validation in Figure 23. Meta categorization outperforms categorization by cross validation (green vs. blue lines) in terms of final prediction accuracy averaged over all the products in the validation set (assigned as "predictable" and "random").

Let us have a closer look at the results achieved by cross validation vs. meta categorization. We again fix the label threshold at 1.00. Testing results on the validation set (88 products) are presented in Table 3. Prior means all the products are considered "random", this is a lower baseline. Oracle assumes the categorization is perfectly correct, so this is the upper baseline, what

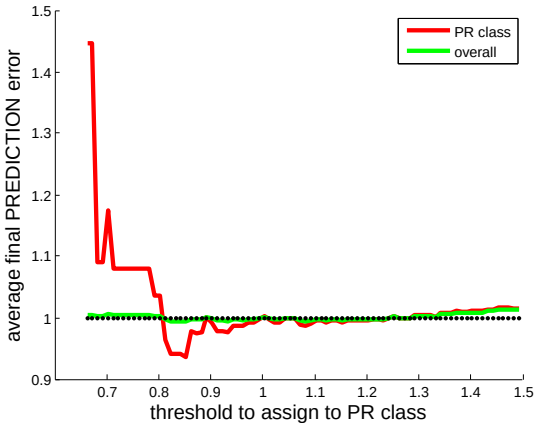


Fig. 22. Cross validation: overall final prediction accuracy.

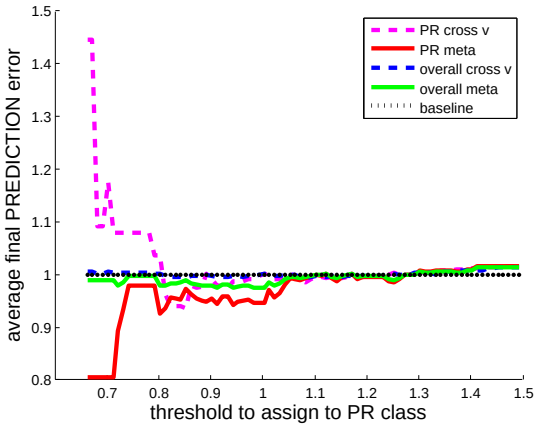


Fig. 23. Overall final prediction accuracy.

Table 3. Categorization results. Prior means all the products are considered "random", Oracle assumes the categorization is perfectly correct. • indicates the difference from the baseline is statistically significant, ○ that it is not. Mann-Whitney U test [14] is used (5% level of significance).

	Meta	Cross-v	Prior	Oracle
Categorization error	30.68%	45.45%	44.32%	-
Final MASE over all products MA6	1.0673			
Final MASE within "predictable" prod. TREE	0.9465 ○	0.9922 ○	-	0.8588 •
Final MASE over all products TREE + MA1	0.9757	0.9971	1.0000	0.9374
Final MASE within "predictable" prod. MA6	0.9365 •	1.0328 ○	-	0.9704 ○
Final MASE over all products MA6 + MA1	0.9711	1.0123	1.0000	0.9869

can be achieved with the correct categorization. It can be seen that meta categorization outperforms both cross validation and prior on the final accuracies using regression tree for the final prediction. However, MA6 (moving average of 6) is still competitive final predictor. It can be seen that final accuracies with meta categorization are better than Oracle when using MA6. That means the result is achieved due to "lucky" configuration of categorization errors. This also indicated a potential future research direction. MA6 could be integrated into the switch mechanism in addition to currently used MA1 vs. intelligent predictor.

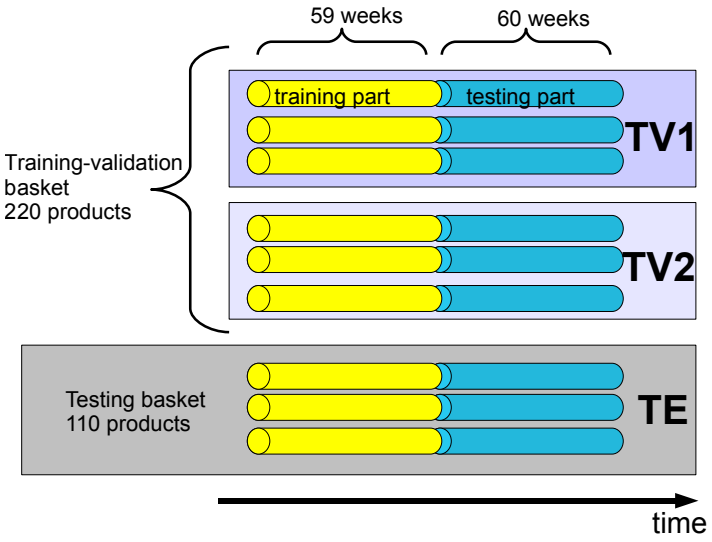


Fig. 24. Experimental division of the dataset.

Table 4. Testing results. • indicates the difference from the baseline is statistically significant, ◦ that it is not. Mann-Whitney U test is used (5% level of significance).

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

7.5 Final prediction accuracies

So far we were showing the results on 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 10. 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. 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 24 for illustration.

In Table 4 we provide the categorization and final prediction results.

In all the testing cases CAPA outperforms the baselines MA1 and MA6. It does not outperform MA6 combined with MA1 though. The combination of those two with the intelligent predictor is an interesting direction for future research. We compare two alternative techniques for identification of the context (product category): meta categorization and cross validation. In

two out of three testing cases, including an unseen testing set, meta categorization outperforms cross validation on the final prediction accuracy.

The results do not show statistically significant difference at 5% level of significance. 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 final accuracy. The issue of evaluation was discussed in Section 5. We argue that the task requires specific cost based evaluation and conventional statistical techniques do not fully take into account the domain specifics. Cost based evaluation technique is yet to be developed, it is another direction for the future research.

8 Discussion of related work

In many real-world domains the situations seen in the past might partially repeat, which is referred as reoccurring contexts [22]. Seasonality comes very close to the concept of reoccurring contexts, 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 [5]. 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 [22], employ instance [3,4] or batch [6,11] selection to look for case bases, or maintain a diverse ensemble of learners [16,15,13,20,17].

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 CAPA approach comes close to meta learning [19], where the relevant learners are selected based on offline predefined criteria. A two level learning model is presented by Widmer [21] perceived context changes are used to focus the learner specifically on the information relevant to the current context. Klinkenberg [12] 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 context awareness. In contrast with the discussed meta learning approaches [21,12], we incorporate domain expertise and observations in categorization and base predictor selection process.

9 Conclusion

Sales prediction is a complicated task. There are different seasonality patterns across the product assortment. We developed context aware sales prediction approach, via introducing background knowledge into predictor selection process.

In SLIGRO case study we showed that CAPA consistently outperforms baseline methods (based on moving average) in terms of the final prediction accuracy. We also showed different tradeoffs on the accuracy while moving the categorization threshold.

In this study we assumed that the category of a given product is static over time. Next practical step would be to employ the approach in a dynamic setting, where the structural type might change over time as well.

Acknowledgements We thank dr. Carlos Soares for relevant comments and helpful suggestions.

References

1. P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta. *Metalearning: Applications to Data Mining*. Springer Publishing Company, Incorporated, 2008.
2. 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.
3. S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle. A case-based technique for tracking concept drift in spam filtering. In *The 24th SGAI Int. Conf. on Innovative Techniques and Applications of Artificial Intelligence*, pages 3–16. Springer, 2004.
4. W. Fan. Systematic data selection to mine concept-drifting data streams. In *KDD '04: Proc. of the 10th ACM SIGKDD int. conf. on Knowledge discovery and data mining*, pages 128–137. ACM, 2004.
5. J. Gama and G. Castillo. Learning with local drift detection. In *ADMA*, volume 4093 of *LNCS*, pages 42–55. Springer, 2006.
6. M. B. Harries, C. Sammut, and K. Horn. Extracting hidden context. *Machine Learning*, 32(2):101–126, 1998.
7. R. Hyndman and A. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.
8. R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *Int. J. of Forecast.*, 22(4):679–688, 2006.
9. M. P. I. Zliobaite, J. Bakker. Towards context aware sales prediction. In *DDDM-09*, 2009.
10. 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.
11. I. Katakis, G. Tsoumakas, and I. Vlahavas. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *Knowledge and Information Systems*, 2009.

12. 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.
13. J. Z. Kolter and M. A. Maloof. Dynamic weighted majority: An ensemble method for drifting concepts. *J. Mach. Learn. Res.*, 8:2755–2790, 2007.
14. 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.
15. K. Stanley. Learning concept drift with a committee of decision trees. CS Dept., University of Texas-Austin, 2001.
16. N. W. Street and Y. S. Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *KDD '01: Proc. of the 7th ACM SIGKDD int. conf. on knowledge discovery and data mining*, pages 377–382. ACM, 2001.
17. A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen. Dynamic integration of classifiers for handling concept drift. *Information Fusion*, 9(1):56–68, 2008.
18. 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.
19. R. Vilalta and Y. Drissi. A perspective view and survey of meta-learning. *Artificial Intell. Review*, 18:77–95, 2002.
20. H. Wang, W. Fan, P. S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *KDD '03: Proceedings of the ninth ACM SIGKDD int. conf. on Knowledge discovery and data mining*, pages 226–235, New York, NY, USA, 2003. ACM.
21. G. Widmer. Tracking context changes through meta-learning. *Machine Learning*, 27(3):259–286, 1997.
22. G. Widmer and M. Kubat. Effective learning in dynamic environments by explicit context tracking. In *ECML '93: Proc. of the European Conf. on Machine Learning*, pages 227–243. Springer-Verlag, 1993.