Electronic Commerce Research and Applications 40 (2020) 100914

Contents lists available at ScienceDirect

Electronic Commerce Research and Applications

journal homepage: www.elsevier.com/locate/elerap

Machine learning and operation research based method for promotion optimization of products with no price elasticity history

Asnat Greenstein-Messica*, Lior Rokach

Software and Information Systems Engineering Department, Ben-Gurion University of the Negev, Beer-Sheva, Israel

ARTICLE INFO

Keywords:
Online stores
Promotion optimization
Decision making

ABSTRACT

Many leading e-commerce retailers adopt a consistent pricing strategy to build customer trust and promote just a small portion of their catalog each week. Promotion optimization for consistent pricing retailers is a challenging problem, as they need to decide which products to promote, with no historical price elasticity information for the candidate products. In this paper, we introduce a novel approach for predicting product price elasticity impact for e-commerce retailers who use a consistent pricing strategy. We combine the commonly used operation research-based log–log demand model with the nonlinear gradient boosting machines algorithm to predict the price elasticity impact of products with no historical price elasticity information. A pessimistic prediction interval measure is used to accelerate the learning period and reduce the probability of selecting low impact promotions due to high model prediction uncertainty. We demonstrate the effectiveness of our approach on a real-world dataset collected from an online European department store.

1. Introduction

Within the retail marketing mix, sales promotions have a strong impact on short-term consumption behavior (Santini et al., 2015). During a sales promotion, the retail price of an item is temporarily reduced, often leading to a dramatic increase in sales volume. The price elasticity impact, which is defined as the responsiveness of a product's purchase quantity (or probability) to changes in its price, varies widely among products and product categories. To optimize the effectiveness of promotions, the retailer must decide which products to promote and set the promotion depth in dynamic demand conditions. In cases where the retailer changes the price frequently, as is done when employing the Hi-Lo pricing strategy (Fassnacht and El Husseini, 2013), previous research (Van Heerde et al., 2000; Cohen et al., 2017) has shown the effectiveness of the log–log demand model for estimating the price elasticity impact, taking into account demand seasonality and trend. This model assumes linear relations between the price elasticity impact log and the price change ratio log, and uses a linear regression technique to estimate the regression coefficients. However, for retailers who use a consistent pricing strategy such as the Every Day Low Price (EDLP) to strengthen their customers' trust (Santini et al., 2015), this model is not suitable, since in most cases, only one historical price point per product is available.

Fig. 1 demonstrates the challenge of predicting the price elasticity impact. This figure presents the price elasticity impact distribution for 6882 products promoted with different discount ratios. The data was collected from a European online department store's transaction log for a period of six months. This store employs a consistent pricing strategy in which only 5% of the products are offered at a reduced price each week, and the discount price is valid for the entire week. As can be seen, for a given discount ratio level, the price elasticity impact varies widely among products.

In this research, we introduce Price Elasticity Impact Learning (PEIL), a novel, nonlinear approach for modeling price elasticity impact, in order to optimize promotion effectiveness for e-commerce retailers who use a consistent pricing strategy. PEIL combines the commonly used log–log demand model and the gradient boosting machines (GBMs) (Chen and Guestrin, 2016) regression algorithm to predict the price elasticity impact of unpromoted products based on their semantic similarity to other, previously promoted products. A GBM is a nonlinear model, which is very effective for handling high-dimensional problems (Friedman, 2001; Chen and Guestrin, 2016; Ma et al., 2018; Bogina et al., 2019) with relatively small datasets. To detect the similarity between products, both catalog-based features (e.g., product category and relative product price) and consumer behavioral-based features relying on browsing clickstream analysis (e.g., click to buy ratio) are used. Previous research demonstrated the potential of leveraging consumers' clickstreams for different e-commerce prediction tasks such as

* Corresponding author.
E-mail address: asnatm@post.bgu.ac.il (A. Greenstein-Messica).

https://doi.org/10.1016/j.elerap.2019.100914
Received 23 March 2019; Received in revised form 19 October 2019; Accepted 28 October 2019
Available online 17 December 2019
1567-4223/

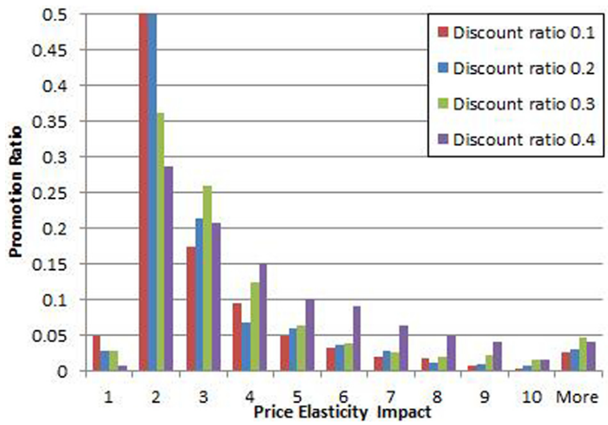

Fig. 1. Price elasticity impact distribution.

anonymous user purchasing intent (Bogina et al., 2019) and product recommendations (Greenstein-Messica et al., 2017; Pourgholamali et al., 2017). In addition, to capture the fine-grained behavioral similarity among products, a consumer's browsing clickstream is leveraged to generate an item embedding representation of the products, which is added to the GBM algorithm. Finally, PEIL employs a pessimistic prediction interval measure to accelerate the learning and incorporate the prediction uncertainty when ranking candidate products in order to reduce the probability of recommending low impact promotions due to model prediction uncertainty.

We evaluated our approach using six months of data collected from a European online department store during two independent periods. Our evaluation includes both price elasticity prediction and candidate product ranking for promotion revenue optimization when only one historical price point per product is available. The evaluation results show that the proposed approach outperforms the commonly used log–log demand model for this use case for both price elasticity impact prediction and promotion revenue optimization. In addition, employing the pessimistic prediction interval during the training period reduces the number of promotions required to predict the price elasticity of a product with no price elasticity history. To summarize, our research makes the following contributions:

- We introduce PEIL, a novel approach for price elasticity impact learning when no historical price elasticity data is available. PEIL combines the commonly used operation research-based log–log demand model and the nonlinear GBM regression algorithm which incorporates semantic product similarity related features. This use case is important for e-commerce retailers that use a consistent pricing strategy.
- We introduce a set of features based on consumers' browsing clickstreams on an e-commerce website, in order to capture the fine granularity of product similarity, and incorporate it into PEIL to further improve price elasticity impact learning.
- We propose a pessimistic active learning (PAL) approach to reduce the length of the training period.

2. Related work

2.1. Price elasticity learning

Price elasticity estimation is a classical application of revenue management theory. The problem, which is closely related to the field of dynamic pricing, is widely covered in the economics, marketing, and operation research literature (den Boer, 2015). Operation research models assume that the structure of the demand is known and consider several historical prices per product to optimize the model parameters using linear regression techniques (Cohen et al., 2017; Felgate and Fearne, 2015; Syntetos et al., 2005; Van Heerde et al., 2000). Recent studies have used linear and dynamic programming techniques (Cohen et al., 2017; Fisher et al., 2017) for promotion optimization.

Recent research used a machine learning approach to address the challenge of dynamic pricing and promotion optimization. These nonlinear models do not assume a functional form for the demand curve, and they have demonstrated some success in dynamic pricing optimization under competition and incomplete demand information (Fisher et al., 2017; Bauer and Jannach, 2018; Misra et al., 2018; Ye et al., 2018). Bajari et al. (2015) demonstrated the effectiveness of using machine learning models for demand prediction compared to traditional econometric-based models. They derived novel asymptotic properties for the machine learning models and used linear regression to combine multiple models in order to improve the prediction accuracy. Ye et al. (2018) used the GBM algorithm with a customized loss function for dynamic pricing optimization in the Airbnb marketplace. Bauer and Jannach (2018) suggested a novel algorithm based on Bayesian inference combined with bootstrap-based confidence estimation and kernel regression for dynamic pricing optimization.

Our approach addresses the challenge of promotion optimization in a scenario where only one historical price of the candidate product is available, by combining operation research and machine learning. This scenario is common among consistent pricing retailers.

2.2. Item embedding

Word embedding methods such as GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013) have attracted significant attention in recent years. The vector representation of words learned by these methods has been shown to carry semantic meaning and is useful in various natural language processing (NLP) tasks. The GloVe method is based on a global log-bilinear regression model and combines the advantages of the global matrix factorization and local context window methods.

Recent studies have introduced embedding-based models (Cheng et al., 2016; Shan et al., 2016; Greenstein-Messica et al., 2017; Pourgholamali et al., 2017; Wang et al., 2018) to improve the performance of e-commerce prediction related tasks. Wide & Deep (Cheng et al., 2016) and Deep Crossing (Shan et al., 2016) learn feature interactions by placing a multilayer perceptron (MLP) above the concatenation of the embeddings of nonzero features; the MLP is claimed to be capable of learning any cross feature order. Pourgholamali et al. (2017) proposed learning both user and product embedding representations from unstructured textual content available in external information sources using recurrent neural networks (RNNs) and then demonstrated the effectiveness of incorporating these representations into a recommender system to improve its effectiveness. Wang et al. (2018) combined embeddings with a tree-based model to further optimize the prediction performance and model explainability. Greenstein-Messica et al (2017) incorporated item embedding based on GloVe into an RNN for e-commerce anonymous user recommendation. An item embedding enables us to capture the hidden patterns of browsing sessions' clickstreams to learn the user's purchasing intent and preferences.

3. Price elasticity impact learning (PEIL)

3.1. Approach

Our goal is to optimize weekly promotion profits by selecting 5% of the products in the catalog to promote each week that have not yet been promoted and setting the discount depth for each product. These products are then sold at a reduced price, based on the recommended discount depth, the next week.

The proposed PEIL approach combines the log–log demand model, which provides good prediction results for cases where multiple prices are available per product (Cohen et al., 2017), and the nonlinear GBM

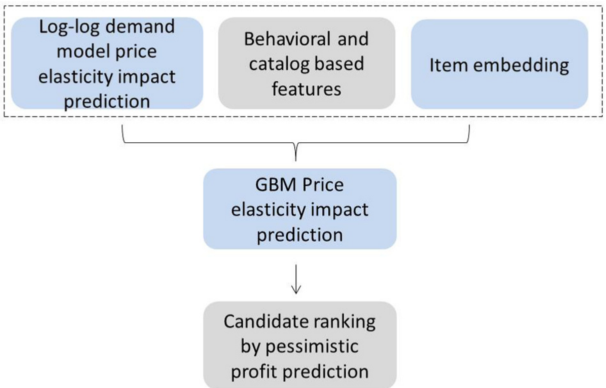

Fig. 2. Overview of the PEIL approach.

model. Fig. 2 presents the main steps of the PEIL approach. First, we generate the features used to train a GBM regression model, in order to predict the price elasticity impact of candidate promotions. For this, we use the transaction log to train a log–log demand model using ordinary least squares (OLS) (Bajari et al., 2015), assuming that the price elasticity and seasonality are common to all of the products within a specific category. We represent the products by a set of features, based on catalog attributes, transaction logs, and the clickstream of consumers. We leverage the training clickstream data to train a GloVe (Pennington et al., 2014) model and generate an item embedding representation which captures the fine granularity of semantic relations among products based on consumers' clickstreams (Greenstein-Messica et al., 2017). Second, we train the GBM model, leveraging the features calculated in the first step, to predict the price elasticity impact of candidate promotions. Finally, to avoid low impact promotions due to model uncertainty, we estimate the GBM model prediction uncertainty to rank the candidate promotions by their pessimistic price elasticity impact prediction.

To better understand the PEIL approach and the material presented in this section, Table 1 summarizes the notations used to describe the PEIL approach.

3.2. Problem formulation

In our promotion optimization context, a product $i \in I$ is represented by a set of attributes, such as the relative product price and average number of transactions per day. The regular product price is p, and the discount level $d \in D$ equals the relative change in product price during the promotion week. The average number of transactions per day for product i during week t is q_{it}. We assume that the product cost equals a ratio c of the product's regular price p and is equal for all products. Assuming that the seasonality's contribution to the promotion's profit is much lower than the discount's contribution, the price elasticity impact $e_{i,d}$equals the increase in sales as a result of the change in price, and the promotion profit $pp_{i,d}$ (Cohen et al., 2017) is equal to the relative change in profits during the promotion week:

$$e_{i,d} = \frac{q_{i,t}}{q_{i,t-1}} \tag{1}$$

$$pp_{i,d} = \frac{q_{i,t}(1-d-c)p}{q_{i,t-1}(1-c)p} \tag{2}$$

Our goal is to select the discount level $d_i \in D$ and the subset S of M tuples of products $i \in I$, in order to maximize the retailer's weekly promotion profits as follows:

$$argmax_S \sum\nolimits_{(i,d_i)\in S} pp_{i,d_i} \tag{3}$$

3.3. Log-log demand model

Eq. (4) describes the log–log demand model estimation for the number of units of product i sold during week t as a function of product price p_t, while taking into account the seasonality and product sales trend (Van Heerde et al., 2000; Cohen et al., 2017). Here, β_i^0, β_i^1denotes the product's intercept and its trend coefficient respectively.$\beta_{it}^2$is a vector with seasonality coefficients for each week of the year, and $week_t$ is a weekly indicator variable which equals one if the observation is from week t and equals zero otherwise. We assume that the weekly seasonality is common to all products belonging to the same product category (Cohen et al., 2017). The model is trained using OLS (Van Heerde et al. 2000). We neglect the interactions between products and promotion fatigue, since we are focusing on a consistent pricing scenario.

$$\ln q_{i,t} = \beta_i^0 + \beta_i^1 t + \beta_{i,t}^2 week_t - \beta_i^3 lnp_{i,t} \tag{4}$$

In our use case of a consistent pricing retailer, there is only one price associated with each candidate product i, and a large portion of the products in the catalog are new (i.e., not available during the previous year). Therefore, we assume that the price elasticity β_i^3 coefficient is common for all products that belong to the same product category. The estimated price elasticity impact, which is the change in the sold quantity for product i which belongs to product category c, is expressed

Table 1
Notation summary.

Notation	Notation Group	Description
q_{it}	General	The average number of transactions per day for product i during week t
$e_{i,d}$	General	Price elasticity impact for product i having promotion discount d; this metric describes the increase in sales as a result of the change in price
$\widehat{y}_i$	General	Price elasticity impact prediction value for product i
y_i	General	Pessimistic price elasticity prediction value for product i
$pp_{i,d}$	General	Promotion profit for product i having a promotion discount d
$pp_{i,d}$	General	Pessimistic promotion profit for product i having a promotion discount d
$G_{i,d}$	General	Pessimistic profit gain *(PEG)* of product i having a promotion discount d
T	GBM model	Number of regression trees in GBM ensemble
L	GBM model	Number of leaves in regression tree
$f^{(k)}$	GBM model	Regression tree k in GBM ensemble; a regression tree k is defined by a vector of scores in leaf $\vec{w}^{(k)}$and a leaf index mapping function q_k that maps an instance to a leaf
q_k	GBM model	Mapping function that maps an instance to a leaf for tree k
$w_l^{(k)}$	GBM model	Score of leaf l in tree k
$\hat{y}_i^{(k)}$	GBM model	Price elasticity prediction value for item i after k iterations (trees)
$O^{(k)}$	GBM model	Objective function for tree k
$\widehat{var}(x_i)$	GBM model	Prediction variance of product i having feature vectorx_i

by:

$$\frac{q_{i,t}}{q_{i,t-1}} = e^{\beta_i^1+\beta_{i,t}^2 week_t-\beta_{i,t-1}^2 week_{t-1}}\left(\frac{p_{i,t}}{p_{i,t-1}}\right)^{-\beta_c^3} \quad (5)$$

3.4. GBM model

An alternative approach to using the average category price elasticity as an estimation when no history is available is to estimate the price elasticity based on an item's similarity to previously promoted products. In recommender systems (Ricci et al., 2015), products are effectively recommended to users based on very sparse historical data. Two of the most common recommendation algorithms are *content-based* where recommended products are selected based on feature similarity to the products the user liked, and *collaborative filtering* where the recommended products are selected based on semantic behavioral similarity to other users who liked the same items the user liked. Influenced by this approach, we propose using the GBM algorithm to predict candidate products' price elasticity impact based on catalog features and consumers' behavioral-based features.

Following recent studies (Greenstein-Messica et al., 2017, Bogina et al., 2019) showing the effectiveness of leveraging consumers' clickstreams and item embedding in capturing the fine granularity of products' semantic similarity, both transaction-based features and clickstream-based features are used. We define a session S as a sequence of click and buy events in an e-commerce site performed by the same user during a time window of 24 h. A "buy session" for a product i, is defined as a session which includes at least one buy event for product i; otherwise, if the session includes at least one click event of product i it is defined as a "click session" for product i.

To generate the item embedding representation, we bagged all of the daily click events per user, sorted by their timestamp, into a sequence of items. We mapped each item into a word and each sequence into a sentence. Then, we used a corpus generated from all of these sessions as input to the GloVe method. Eq. (6) provides the local cost function of the GloVe method which was minimized using gradient descent.

$$\mathrm{J} = \sum_{\mathrm{i,j=1}}^{\mathrm{V}} \mathrm{f(X_{ij})(w_i^T w_j + b_i + b_j - \log(X_{ij}))^2} \quad (6)$$

Here, V is the number of items in the catalog, X_{ij} denotes the number of times item j occurs in the context of item $i$taking into account the distance between the items within the context window, w_i is the vector representation of item i(i.e., the item embedding), w_j is the context item vector, and b_i, b_j are bias terms. The function f described by Equation (7) is a weighting function that avoids overweighting high co-occurrences. The performance of the model depends slightly on the cutoff, which we set at$x_{max} = 100$ for all of our experiments. In this case, w_i, w_j, b_i, b_j are the parameters to be learned during training.

$$f(x) = \begin{cases} x/x_{max}, & x < x_{max} \\ 1, & otherwise \end{cases} \quad (7)$$

Since the change in the number of sold items is also influenced by the seasonality and product trend, we added features that model these attributes to the GBM algorithm as well. An exponent of the weekly seasonality estimation derived from the log–log demand model $\beta_{i,t}^2 week_t - \beta_{i,t-1}^2 week_{t-1}$is used to calculate the weekly category's seasonality change estimation feature. A set of features which estimates the specific product's demand volatility due to product trend and seasonality is derived by calculating the changes in sold products during the weeks prior to the promotion week. Table 2 describes the features we used for each feature category. All of the features refer to the candidate product i, its category c, and the promotion week t.

To generate a hybrid model which combines the log–log demand model predictions with the nonlinear GBM model, we provide the score of the log–log demand model and the regression model's R^2 value as additional features to the GBM model. Our hypothesis is that since there is no price elasticity history for each candidate product, and the log–log demand model's price elasticity coefficient β_i^3 is estimated by the price elasticity impact of all previously promoted products belonging to the same category, it will be less accurate than the nonlinear GBM model which predicts the candidate product's price elasticity impact based on semantic similarity to previously promoted products. Incorporating the clickstream-based features into the GBM model will enable detection of fine-grained similarity aspects into the model. A hybrid approach that brings together the log–log demand model with the GBM algorithm allows us to combine the semantic similarity among previously promoted products to the candidate products with the model-based approach for estimating the product trend and seasonality when predicting the price elasticity.

Table 2
GBM algorithm features.

Feature category: clickstream	Feature category: category, seasonality
#click to buy sessions ratio	Estimated weekly category seasonality change
#click to buy sessions ratio vs category	Average category elasticity impact
Category #click to buy sessions ratio Item embedding representation Category #click to buy sessions ratio	Category's price elasticity impact standard deviation
Feature category: transaction	**Feature category: catalog**
Average daily transactions#	Price percentile within category
Average daily transactions# vs category	Price percentile vs total
Category daily transactions#	Category's price percentile
Average daily transactions ratio 1 to 2 weeks before	Discount price vs regular price ratio
Average daily transactions ratio 2 to 3 weeks before	
Max. average weekly transactions ratio during last 4 weeks	
Average zero transaction days per week	

3.5. Pessimistic product ranking

The GBM algorithm provides a prediction for the price elasticity impact of each candidate product and discount depth combination. Based on this prediction, an estimation of the promotion revenues and profits is calculated, and the retailer can rank the candidate products. Since our scenario involves a high uncertainty level, we want to incorporate the prediction uncertainty into the candidate product ranking.

The confidence of a prediction can be understood as how likely the prediction is to be correct (Rokach et al., 2008; Zhang et al., 2016) and to what extent the prediction can be trusted. Previous research (Li et al., 2016; Zhang et al., 2016) used the upper confidence limit when ranking candidates in order to expose new candidates. We propose a new measure termed the pessimistic prediction value to incorporate prediction uncertainty into the prediction value when ranking the candidate products for the weekly promotion. Such an approach is suitable in cases where the cost of selecting unsuitable candidates is higher than the potential loss associated with leaving out good candidates. We define the pessimistic prediction *value* as the lower prediction interval for a certain probability $100(1 - \alpha)\%$. To calculate the pessimistic prediction value for each prediction, we first calculate the prediction confidence interval of the GBM regression machine and then use it to estimate a pessimistic prediction interval for each prediction. For a GBM regression machine that is composed of a set of T regression trees, the prediction $\widehat{y_i}$ for an instance i is described by (Chen and Guestrin, 2016):

$$\hat{y}_i = \sum_{k=1}^{T} f^{(k)}(x_i) f^{(k)} \in \mathscr{F} \quad (8)$$

Here, $\mathscr{F}$ is the space of functions containing all regression trees. Defining a regression tree k by a vector of scores in leaf $\overrightarrow{w}^{(k)}$ and a leaf index mapping function q_k that maps an instance to a leaf, the regression tree $f^{(k)}$ for an input vector x_i is equal to:

$$f^{(k)}(x_i) = w^{(k)}_{q_k(x_i)} \quad (9)$$

The model is trained in an additive manner. The objective function for regression tree k is described by:

$$O^{(k)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(k-1)} + f^{(k)}(x_i)) + \frac{1}{2}\lambda\|w\|^2 \quad (10)$$

Here, $y_i$is the actual target value for instance i, $\hat{y}_i^{(k-1)}$ is the model prediction of the ith instance in the k-1 iteration (k-1 trees), l is the loss function, and λ is a regularization parameter. Assuming a Gaussian distribution for each leaf prediction, the scores and mappings are assigned in such a way that the objective function $O^{(k)}$ is minimized using the maximum likelihood estimation principle. Defining the instance set I in leaf j as $I_j = \{i|q_k(x_i) = j\}$ and the first (Gradient) and second (Hessian) order gradient statistics on the loss function as $g_i^{(k)}$ and $h_i^{(k)}$, respectively, and having a regularization parameter λ, the objective function for regression tree k with L leaves equals:

$$O^{(k)} = \sum_{j=1}^{L} \left(\sum_{i\in I_j} g_i^{(k)}\right) w_j^{(k)} + 0.5\left(\sum_{i\in I_j} h_i^{(k)} + \lambda\right) w_j^{(k)2} \quad (11)$$

Using a square loss to minimize the regression error, the Gradient $g_i^{(k)}$and Hessian $h_i^{(k)}$ for an instance i in tree k are described by:

$$g_i^{(k)} = 2(\hat{y}_i^{(k-1)} - y_i);\ h_i^{(k)} = 2 \quad (12)$$

Assuming the tree structure $q(x)$is fixed, the optimal weight (leaf prediction) $w_j^{(k)*}$ for leaf j in tree k is equal to:

$$w_j^{(k)*} = -\frac{\sum_{i\in I_j}(\hat{y}_i^{(k-1)} - y_i)}{\sum_{i\in I_j} 1 + 0.5\lambda} \quad (13)$$

and the leaf variance equals:

$$\widehat{var}_j^{(k)*} = \frac{\sum_{i\in I_j}(\hat{y}_i^{(k-1)} - y_i)^2 - \left(\sum_{i\in I_j}(\hat{y}_i^{(k-1)} - y_i)\right)^2}{\left(\sum_{i\in I_j} 1 + 0.5\lambda\right)^2\left(\sum_{i\in I_j} 1 - 1\right)} \quad (14)$$

Based on the assumption that the predictions in each regression tree leaf are sampled from a Gaussian distribution, we can conclude that the overall GBM prediction (expressed by Eq. (5)), which equals the sum of each tree prediction, is sampled from a Gaussian distribution as well. Similarly, the GBM prediction distribution variance for an instance x_i can be calculated using the distribution variance of each leaf the instance is assigned to:

$$\widehat{var}(x_i) = \sum_{k=1}^{T} \widehat{var}(q^k(x_i)) + 2\sum_{1\le k<k'\le T} Cov\left(q^k(x_i), q^{k'}(x_i)\right) \quad (15)$$

Since there is no consistency with the assignment of instances in each tree of the GBM, we omitted the covariance item from the calculation.

Finally, to incorporate the prediction uncertainty in candidate product ranking, we used the lower bound prediction value for the ranking rather than the point prediction value. By using a pessimistic prediction value based on the lower prediction interval, candidate products with a high point prediction value and relatively high variance will be ranked lower than candidate products with relatively low variance. The pessimistic prediction value $y_i$for an instance $x_i$is described by:

$$y_i = \hat{y}_i - T_\alpha\sqrt{\sum_{k=1}^{T} \widehat{var}(q^{(k)}(x_i))(1 + 1/n_{q^{(k)}(x_i)})} \quad (16)$$

Here, $T_\alpha$is the $100(1-\alpha)th$ percentile of Student's t-distribution with $n-1$ degrees of freedom, and $n_{q^{(k)}(x_i)}$ is the number of instances assigned to the leaf x_i in the regression tree k using the mapping function $q^{(k)}(x_i)$.

In our case, the predicted value is the price elasticity impact $y_{i,d}$ for candidate product i having promotion discount depth d. The *pessimistic promotion profit* value $pp_{i,d}$, defined as the ratio between the profits during the promotion week and the previous week, can be estimated using the pessimistic price elasticity impact prediction value y_i:

$$pp_{i,d} = \frac{y_i(1-d-c)}{(1-c)} \quad (17)$$

3.6. Pessimistic active learning

Active learning (Cohn et al., 1994) refers to data mining policies which actively select unlabeled instances for labeling. During an exploration phase, promotions are offered, and the price elasticity impact is measured. Based on the results, the firm uses the active learning algorithm to select the next promotions to offer. Since the learner chooses the promotions to label, the number of promotions needed to train a supervised machine learning model can often be much lower than the number of promotions required in normal supervised learning. In the exploitation phase, the firm simply applies the trained supervised machine learning model to select the upcoming promotions, with no intention of improving the model. Thus, the model evolves during the exploration phase and is fixed during the exploitation phase.

Our objective is not only to maximize the net profit obtained during the exploitation phase but also to minimize the number of promotions needed to train the model and thereby reduce the length of the exploration phase (Rokach et al., 2008; Bauer and Jannach, 2018). To support this, we should select the candidate products that increase knowledge about promotions' effectiveness during the exploration phase. Agarwal et al. (2009) presented the ε–greedy approach with random exploration on a fraction of the traffic and greedy exploitation of the rest. Recent research (Li et al., 2010; Li et al., 2016; Gentile et al. 2017) suggested a contextual bandit model for exploration–exploitation optimization and leveraged the upper confidence bound of the prediction to collect users' feedback for new candidates.

We propose a new measure called pessimistic profit gain (PEG) which is used for ranking the candidate products as part of the exploration task. The proposed approach aims to improve decision-making by measuring the change in the profit when candidate products with high uncertainty are selected (Auer, 2002), while considering the potential promotion profit. This approach is suitable in cases where the cost of selecting unsuitable candidates is higher than the potential loss involved with missing potential good candidates. We define the PEG $G_{i,d}$for a candidate product i having discount depth d as the change in the pessimistic promotion profit prediction before and after promoting candidate product i with discount depth d. Since the actual price elasticity impact of the product is not known prior to its promotion, the pessimistic promotion profit is recalculated based on the model's estimation. As can be seen from Eqs. (16) and (17), the price elasticity impact point predictions before and after the promotion decision are equal, and the difference arises from the prediction interval component. Candidates assigned to leaves with relatively high variance and a small number of labeled products will have higher pessimistic profit gain values than candidates assigned to leaves with relatively low variance and a large number of labeled products. The recalculated leaf variance value $\widehat{var}'(q^{(k)}(x_i))$will decrease when a new product with a value equal to its score is added. The $G_{i,d}$ for product i and discount d for a given mapping function q is described by:

$$G_{i,d} = \frac{1-d-c}{1-c}\left(T_\alpha\sqrt{\sum_{k=1}^{T}\widehat{var}(q^{(k)}(x_i))(1+1/n_{q^{(k)}(x_i)})} - T_\alpha \sqrt{\sum_{k=1}^{T}\widehat{var}'(q^{(k)}(x_i))(1+1/2+n_{q^{(k)}(x_i)})}\right) \tag{18}$$

4. Experiment

In order to evaluate the PEIL approach, we leveraged 18 months of data from a European online department store that employs a consistent pricing strategy. Our study addresses the following:

1) PEIL's effectiveness at predicting price elasticity impact and promotion profit in a consistent pricing scenario.
2) The contribution of incorporating clickstream data and item embedding to the prediction accuracy.
3) The contribution of incorporating prediction uncertainty using the pessimistic prediction value and pessimistic profit gain measures to the promotion profit ranking and the required learning period.

4.1. Dataset description

The dataset contains the online department store's product catalog and event log for an 18 month period (January 2017–June 2018). Details about the product catalog and consumer log data used are presented in Table 3.

4.2. Experimental settings

The first year of the dataset, defined as the seasonality dataset, is used to calculate the weekly seasonality for each product category, and the last six months, defined as the promotion dataset, is used to evaluate the proposed approach. To calculate the category's weekly seasonality, we relied on Eq. (4) and included only products that were sold during 2017 but were not promoted during this period. We split the promotion dataset into two sub-datasets which cover two periods of three months each, which are referred to as P1 (the first three months) and P2 (the second three months), to evaluate the consistency of our results over various time periods. Each of the sub-datasets is split into training (80%), validation (10%), and test (10%) sets. We maintained the chronological order of dataset samples in the train-validation-test split in order to adhere to real-world conditions. The promotion dataset consists of the product, discount depth, calendar week, weekly sales during the promotion week, and weekly sales during the four weeks before the promotion week. We filtered out products that had more than three days of zero sales during at least one week in the month before the promotion, in order to avoid intermittent demand complexity (Syntetos et al., 2005). In addition, since our goal is to model price elasticity impact, we filtered out categories with a high seasonality contribution to changes in sales, based on 2017 seasonality and special calendar events. After filtering, we had 1414 promotions for P1 and 1405 promotions for P2. The promoted products belong to 21 categories, and the discount depth varied between 10 and 40 percent for both periods. Table 4 summarizes the dataset's descriptive information for the two periods. It can be seen that when the discount ratio is higher, the price elasticity impact and its standard deviation are also higher, on average. The distribution of the discount ratio is similar across the two periods.

To calculate the transaction and clickstream-based features, we used the event log information for the relevant training period. We bagged all of the daily events per user (sorted by the timestamp) into a sequence of items. For a logged-in user, the user ID is used to assign the relevant events to the user, whereas for an anonymous user, either the cookie mechanism or session ID provided by the Web server is used to assign the relevant clicks to the anonymized user. To obtain the item embedding representation (Greenstein-Messica et al., 2017) for each product, we mapped each item into a word and each session into a sentence and then applied the GloVe (Pennington et al., 2014) method on this corpus. After filtering out products that appear in less than five sessions during the training period and one click sessions, we had 11,696,659 sessions and 30,562 products for P1 and 13,123,187 sessions and 32,362 products for P2. We used the validation set for hyperparameter optimization in all of the experiments.

5. Evaluation

5.1. Price elasticity impact learning

To evaluate the performance of the different models, we predicted the price elasticity impact on the test set using each model. We then calculated the average mean absolute error (MAE) and the prediction correlation coefficient between the predicted price elasticity and the actual values. We evaluated the following models:

- *Average:* This is our baseline method. Here, the price elasticity impact prediction equals the average price elasticity impact for the selected discount.
- *Category:* In this case, the price elasticity impact prediction equals the category's average price elasticity impact for the selected discount ratio.
- *Regression:* This is a log–log demand regression model assuming the same price elasticity coefficient for all products belonging to the same category.
- *GBM:* This is a GBM model in which we evaluated two flavors: basic, which includes the features described in Table 1, and embedding, which also includes the item embedding features.
- *Hybrid:* This is a GBM model that includes both the item embedding features and the log–log demand model prediction.

Table 5 presents the results of the price elasticity prediction for the two periods: P1 and P2. The average test set price elasticity impact is 3.28, and the standard deviation is 2.38. As can be seen, although the GBM model flavors perform better than the regression model, the hybrid model, which combines the regression model predictions with the GBM model, outperforms all of the models. Furthermore, incorporating the item embedding-based features, which detect the fine granularity of semantic similarity among items, into the GBM model improves the model's accuracy. The regression model outperforms the category model, meaning that taking into account the specific product trend and category seasonality, as described by Eq. (4), improves the prediction accuracy. The category method outperforms the average method,

Table 3
Dataset description.

Source	Information
Product catalog	Date, product ID, category ID, price, promotion indication
Event log: click event	Timestamp, anonymous session ID/user ID, product ID
Event log: buy event	Timestamp, anonymous session ID/user ID, product ID, quantity, price

Table 4
Dataset's descriptive statistics.

	Promotion number P1;P2	Promotion ratio P1;P2	Price elasticity impact average P1;P2	Price elasticity impact standard devication P1;P2
10% discount promotions	382;337	0.27;0.26	2.11;2.05	1.54;1.65
20% discount promotions	395;422	0.28;0.3	3.12;2.8	1.83;1.8
30% discount promotions	382;351	0.27;0.25	3.8;3.5	2.7;2.41
40% discount promotions	255;295	0.18;0.21	5.1;4.7	4.16;3.86
Total promotions	1414;1405	1;1	3.38;3.2	2.4;2.35

Table 5
Price elasticity impact prediction results.

Model	MAE P1;P2	Correlation Coefficient P1;P2
Average	1.73;1.68	0.37;0.28
Category	1.51;1.5	0.59;0.49
Regression	1.42;1.38	0.61;0.53
GBM basic	1.27;1.31	0.65;0.59
GBM emb	1.22;1.23	0.7;0.64
Hybrid	1.16;1.20	0.73;0.67

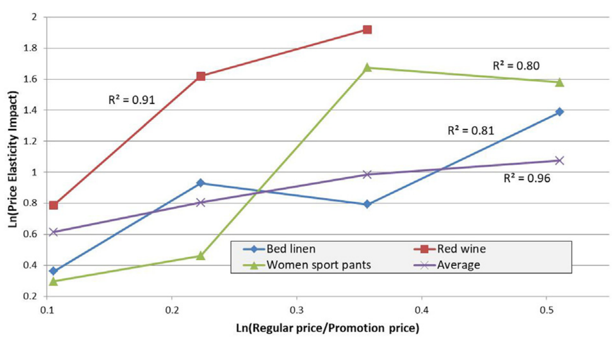

Fig. 3. Log-log graph of the price elasticity impact vs. regular to promotion price ratio for different categories and the average across all products in the training set (the regression R^2 values for each category are presented).

meaning that the variance of the promotion impact within each category is lower than the variance across all of the categories. Fig. 3 presents the log–log graph of the average price elasticity impact vs the regular to promotion price ratio for different categories after seasonality normalization, as well as an average across all products in the training set. In addition, we present the training data in sample regression R^2 values for the different categories. Although on average the regression model fit is good, there is a difference between the different categories and the various products within each category.

This supports our hypothesis that when no price elasticity impact history is available, the log–log demand model's price elasticity coefficient β_i^3 (described by Equation (4)) estimation is not accurate, and hence the GBM model approach will be more accurate.

To determine the confidence level of the results, we performed the Friedman test on the null hypothesis that each of the methods will provide the same results. The MAE of the predicted price elasticity impact was used as the observation value, and the ranking was set in descending order of the absolute error value. We used the Friedman test, since all of the methods were tested on the same population (281 samples). We obtained a Friedman chi-square value of 92.18 with a p-value lower than 2.2e-16. Hence, we reject the null hypothesis that each of the methods provides the same results. Once the null hypothesis is rejected, we apply post hoc tests which are based on the mean rank differences of the different methods (Bortz et al., 2008) for pairwise comparisons between the methods, and specifically between the hybrid method and the other methods. Table 6 presents the post hoc analysis results. As can be seen in the table, the hybrid model is superior to the regression model with 95% confidence. We cannot conclude that the

Table 6
Post hoc pairwise comparison test - price elasticity impact prediction.

Models	Post Hoc Test p-value
Average – Hybrid	< e-13
Category – Hybrid	0.002
Regression – Hybrid	0.049
GBM basic – Hybrid	0.098
GBM emb – Hybrid	0. 10
GBM basic – Regression	0.081
GBM emb – Regression	0.063

hybrid approach significantly outperforms the other GBM model flavors that do not include the regression model prediction. However the post hoc comparison test results between the other GBM flavors and the regression model indicate that the GBM model flavors excluding the regression model prediction do not significantly outperform the regression model. The addition of the regression model prediction and score to the GBM model, as done in the hybrid approach, is required in order to obtain statistically significant superiority to the regression model.

As shown in previous studies, price elasticity impact varies among categories (Fassnacht and El Husseini, 2013). As a result, once the superiority of the hybrid approach for predicting the price elasticity impact was demonstrated, we wanted to gain more insight regarding its prediction accuracy for different product categories. To have a meaningful number of promotions per category in the test set, we aggregated the 21 categories into four major categories: food, home, fashion, and sport, and combined the two test set periods into one period. We managed to aggregate about 90% of the promotions into the four major categories, because some of the promotion categories, such as self-care and beauty, didn't have a meaningful number of promotions during the test period, and thus were not aggregated. We then analyzed the hybrid approach's price elasticity impact prediction accuracy across these four categories. To evaluate the variability of the price elasticity impact across the different categories, we calculated the average and standard deviation of the price elasticity impact on the training dataset. The dataset promotion distribution for each major category and the ratio of promotions for each discount promotion value are presented in Table 7. Here, we calculated the price elasticity impact average and standard deviation per major category and discount ratio, and then calculated a weighted average price elasticity impact and standard deviation across the different discount ratios for each major category. The home category has a relatively higher ratio of low (10%) discount promotions and

Table 7
Dataset's categorical distribution.

	Fashion	Sport	Home	Food
Ratio instances train;test	0.27;0.24	0.15;0.18	0.30;0.31	0.28;0.27
Instances number train;test	548;61	304;46	608;78	568;75
10% discount promotion ratio	0.1	0.05	0.14	0.05
20% discount promotion ratio	0.37	0.4	0.45	0.40
30% discount promotion ratio	0.33	0.33	0.35	0.37
40% discount promotion ratio	0.2	0.21	0.08	0.18
Price elasticity impact average	2.15	3.1	2.82	5.74
Price elasticity standard deviation	1.8	2.19	2.2	2.91

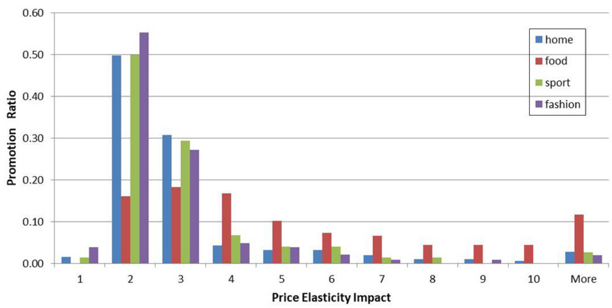

Fig. 4. Price elasticity impact distribution for major categories.

a lower ratio of high (40%) discount promotions. The ratio of promotions from each major category is stable between the training and test set. The relatively low standard deviation of the fashion category can be explained by the fact that we filtered out fashion related categories with high seasonality (e.g., accessories) in the preprocessing phase, as described earlier. The standard deviation of the food category is relatively higher than the other categories, and it has a relatively high price elasticity impact. Fig. 4 presents the distribution of price elasticity impact for each of the four major categories.

To evaluate whether there is a difference in the prediction accuracy across categories, we analyzed the prediction error MAE for each major category in the test set for the different models evaluated. Table 8 contains the prediction error MAE for each category and evaluated model. As can be seen, the hybrid approach's prediction error is smaller than all of the other evaluated models for all categories. The prediction error is higher for the food category where the price elasticity impact standard deviation is relatively high.

To further evaluate the contribution of the overall clickstream-based features to the hybrid model's performance, we analyzed the top 10 features by leveraging the GBM feature importance score (Chen and Guestrin, 2016). The results are presented in Table 9. The clickstream related features are ranked relatively high, starting at fourth place. Other important features relate to the demand volatility during the weeks prior the promotion week and the category's price elasticity impact. The regression model prediction feature received a rank of 10 out of 21 features.

5.2. Promotion profit

In this section, we compare the promotion profit for each of the models. We present the optimal promotion profit, which is equal to the promotion profit achieved by selecting the promotions with the highest actual profit from the test set. In this aspect of our evaluation, we added random selection of the candidate promotions and another model, the pessimistic model, which uses the pessimistic prediction value to incorporate prediction uncertainty into the hybrid model's prediction value when ranking the candidate products for the weekly promotion. For the pessimistic model, we used a prediction interval of 0.7 ($\alpha = 0.3$), which provided the best results on the validation set. Table 10 presents the evaluation results for different selection ratio and cost ratio values. The values for the second evaluation period are presented in parentheses. As can be seen, the pessimistic approach

Table 8
Price elasticity impact prediction error (MAE) per category.

Category/Model MAE	Hybrid MAE	GBM MAE	Regression MAE	Category MAE
Fashion	1.17	1.27	1.39	1.49
Sport	1.09	1.2	1.40	1.49
Home	1.14	1.25	1.37	1.48
Food	1.33	1.45	1.54	1.6

Table 9
Hybrid model feature importance.

Feature	Importance (Rank)
Average daily transactions ratio (ratio of weeks 1 and 2 before)	0.0304 (1)
Max. average weekly transaction ratio during last 4 weeks	0.0256 (2)
Category's price elasticity impact standard deviation	0.0219 (3)
#click to buy sessions ratio vs category	0.0202 (4)
Price percentile within category	0.0191 (5)
#click to buy sessions ratio vs category	0.0191 (6)
Price percentile vs total	0.0191 (7)
#click to buy sessions ratio	0.0190 (8)
Item embedding (neuron 24)	0.0174 (9)
Regression model prediction	0.0167 (10)

Table 10
Promotion profit results.

Selection Ratio	Model	Promotion Profit. Cost = 0 (P2); Cost = 0.1 (P2)
0.05	Optimal	7.27 (7.24); 7.25 (7.05)
0.05	Random	2.85 (1.97); 2.36 (1.15)
0.05	Average	2.97 (2.8); 2.96 (1.49)
0.05	Category	4.8 (4.15); 4.8 (3.54)
0.05	Regression	5.63 (5.56); 5.48 (5.48)
0.05	Hybrid	6.39 (5.96); 6.07 (5.85)
0.05	Pessimistic	6.48 (6.47); 6.22 (6.31)
0.1	Optimal	6.56 (6.03); 6.50 (5.8)
0.1	Random	2.47 (2.23); 2.37 (1.5)
0.1	Average	2.98 (2.6); 2.96 (2.07)
0.1	Category	4.52 (4.13); 4.34 (4.0)
0.1	Regression	4.99 (4.40); 4.91 (4.31)
0.1	Hybrid	5.40 (4.71); 5.21 (4.58)
0.1	Pessimisitc	5.54 (4.75); 5.28 (4.65)

outperforms all other models for the different selection and cost ratios. The relative improvement is higher for the smaller selection ratio, since the average promotion profit decreases when the selection ratio increases.

Fig. 5 presents the promotion revenue (setting the cost at zero) for the various selection ratios and models evaluated. It can be seen that the pessimistic approach outperforms the other models. The optimal promotion revenue decreases as the selection ratio increases.

In order to analyze whether the differences between the reported performance are statistically significant, we generated 100 datasets by bootstrap resampling (Bortz et al., 2008; Koehn, 2004) the training and testing datasets for the two evaluated periods. The same data partitions were used by all methods. To determine the confidence level of the results, we performed the Friedman test on the null hypothesis that each of the methods will provide the same results. The promotion profit was used as the observation value, and the ranking was set in ascending order. Since consistent pricing retailers usually promote a small portion of products, we performed the significance tests for a selection ratio of

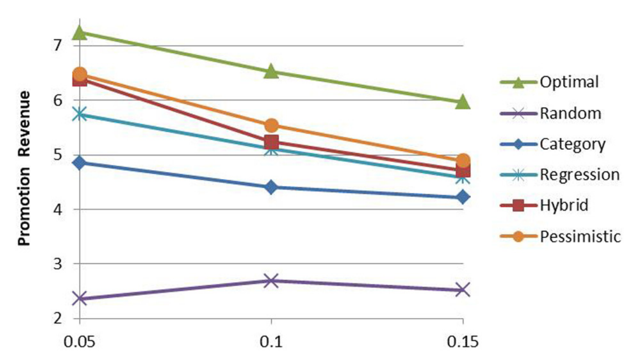

Fig. 5. Promotion revenue for different selection ratios.

Table 11
Post hoc pairwise comparison test results – promotion profit.

Models Pair	Post Hoc Test p-value	Cost Ratio
Hybrid – Pessimistic	0.088	0
Hybrid – Regression	0.047	0
Pessimistic – Regression	0.014	0
Hybrid – Pessimistic	0.076	0.1
Hybrid – Regression	0.043	0.1
Pessimistic – Regression	0.009	0.1

0.05 and cost ratios of zero and 0.1. We obtained a Friedman chi-square value of 230.1 for a cost ratio of zero and 229.7 for a cost ratio of 0.1. For both cost ratios the p-value is lower than 2.2e-16. Hence, we reject the null hypothesis that each of the methods provides the same results. Once the null hypothesis is rejected, we apply post hoc tests (Carpenter and Bithell, 2000; Koehn, 2004) for pairwise comparisons between the models, and specifically between the three leading models: hybrid, pessimistic, and regression. Table 11 presents the post hoc pairwise comparison test results. It can be seen that for a confidence level of 95%, both the pessimistic and hybrid model significantly outperform the regression model for both cost ratios, however we can't conclude that the pessimistic model is superior to the hybrid model with a confidence level of 95%.

5.3. Pessimistic active learning

In this section, we evaluate the effectiveness of using the PEG measure to reduce the length of the GBM algorithm's training period. We selected the first week of the promotions training dataset as a seed. A fixed number of instances *M* (the batch size) was chosen in each iteration by each method evaluated. We set *M* to be 10% of the training dataset. A GBM model for each method was then trained using all of the instances selected so far by the method. Once again, to compare the methods, the price elasticity impact prediction MAE and the average promotion revenue for the test set were calculated. In order to provide reliable estimates of the algorithms' performance and analyze whether the differences between the models' performance are statistically significant, we generated 100 datasets by bootstrap resampling (Carpenter and Bithell, 2000; Koehn, 2004) the training and testing datasets. To reduce the experimental variance, the same data partitions were used by all methods. We evaluated the following methods of selecting the candidate promotions during the training period:

- *Random:* This is our baseline method. Here, in each iteration the promotions are randomly selected from the group of promotions that hadn't yet been selected from the training set.
- *Epsilon Greedy:* This well-known decision-making method combines best predicted action selection and knowledge expansion. Epsilon, the portion of random instances selected in each batch, is set to 0.25.
- *Pessimistic Profit Gain (PEG):* This is our proposed method, in which the PEG is used for ranking the candidate products while considering the potential promotion profit.

Fig. 6 presents the predicted price elasticity impact MAE on the test set for each method as a function of the training set proportion. The results presented are averaged across the bootstrap sampled datasets. Naturally, all of the methods converge to the same values when using the entire training set. It can be seen that in contrast to the other methods, learning is accelerated with the PEG method. The gap between the prediction MAE becomes smaller after 30% of the batches and converges after 60% of the batches.

We performed the Friedman test on the null hypothesis that each of the methods will provide the same results. The area under the curve (AUC) of the predicted price elasticity impact MAE as a function of the proportion of the training data set was used as the observation value

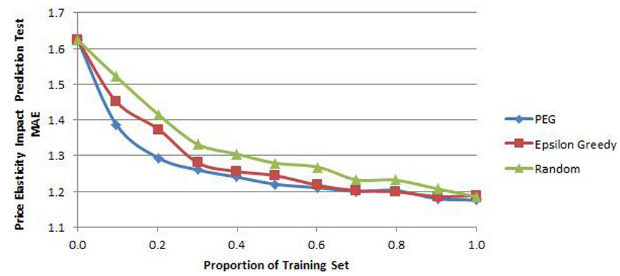

Fig. 6. Active learning test set price elasticity prediction MAE.

Table 12
. Active learning test set's promotion revenue evaluation.

Model	AUC	PEG Comparison Test p-value
PEG	14.19	–
Epsilon Greedy	14.31	0.085
Random	14.40 0.01	

(Rokach et al. 2008). A lower AUC means that the MAE converged earlier, and hence the learning duration is shorter. We obtained a Friedman chi-square value of 9.05 and p-value of 0.01. Hence, we reject the null hypothesis that each of the methods provides the same results. We apply post hoc tests (Bortz et al., 2008) for pairwise comparisons between the methods, and specifically between the PEG method and the baseline methods. Table 12 presents the evaluation results. Although the average AUC of the PEG method is lower than the baseline methods, we can only conclude that it is superior to the Random method with a confidence level of 95%.

.Fig. 7 presents the test set promotion revenue for the different methods as a function of the training set proportion. For simplicity, we assumed a zero cost ratio. The results presented are averaged across the bootstrap sampled datasets. Here as well, the PEG method accelerates the learning, and after only 30% of the batches the test set revenues are almost equal to the revenues achieved after training the GBM model with the complete training set.

We performed the Friedman test on the null hypothesis that each of the methods will provide the same results. The area under the curve (AUC) of the test set's promotion revenue was used as the observation value (Rokach et al., 2008). We obtained a Friedman chi-square value of 52.92 and a p-value of 3.2e-12. Hence, we reject the null hypothesis that each of the methods provides the same results. Table 13 presents the results of the pairwise post hoc comparison tests and the average test set promotion revenue AUC for each method. Here, we can conclude that the PEG method is superior to both the Epsilon Greedy and Random methods with a confidence level of 95%.

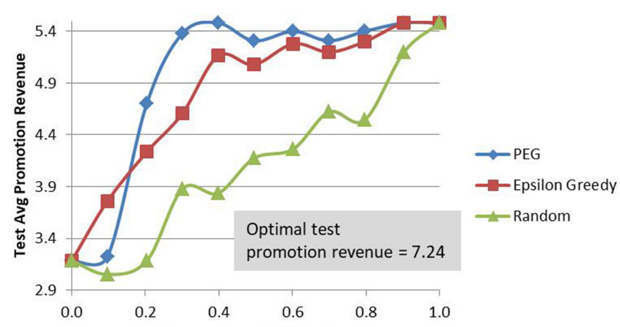

Fig. 7. Active learning test set promotion revenue.

Table 13
Active learning test set's promotion revenue evaluation.

Model	AUC	PEG Comparison test p-value
PEG	50.72	–
Epsilon Greedy	48.23	0.018
Random	43.12	9e-13

6. Conclusion and future work

In this research, we introduce PEIL, a novel approach for predicting the price elasticity impact of promotions for retailers who adopt a consistent pricing strategy. PEIL combines the commonly used log–log demand model, which assumes linear relations between the price elasticity impact and the price change ratio, with the nonlinear GBM model. The GBM model is based on a rich set of features, which capture the similarity between products based on catalog, transaction log, and clickstream data. Furthermore, we introduce a pessimistic prediction value measure to incorporate prediction uncertainty when ranking the candidate promotions to further improve the promotion profit.

We used 18 months of data collected from an online European department store that employs a consistent pricing strategy in order to evaluate the proposed approach. Our results show that PEIL achieved a significant improvement in both prediction accuracy and promotion profit compared to the log–log demand model.

In addition, we showed that using the pessimistic profit gain (PEG) measure for active learning during the training period significantly shortened the required learning period and enabled efficient promotion selection after only three promotion batches.

For future work, we plan to collect competitors' pricing data in order to derive historical price elasticity impact for more products, and incorporate it into the overall optimization method. In addition, we would like to extend the pessimistic active learning approach for a reinforcement-learning scenario where exploration and exploitation (Misra et al., 2018) are optimized simultaneously.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported by the Ministry of Science and Technology, Israel [grant number 8762421]. The authors would like to thank Michael Friedmann, CTO of YOOCHOOSE GmbH for his support and involvement at various stages of this research.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.elerap.2019.100914.

References

Agarwal, D., Chen, B.C., Elango, P., Motgi, N., Park, S.T., Ramakrishnan, R., Zachariah, J., 2009. Online models for content optimization. In: Advances in Neural Information Processing Systems, pp. 17–24.
Auer, P., 2002. Using confidence bounds for exploitation-exploration trade-offs. J. Mach. Learn. Res. 3 (Nov), 397–422.
Bajari, P., Nekipelov, D., Ryan, S.P., Yang, M., 2015. Demand estimation with machine learning and model combination. National Bureau of Economic Research (No. w20955).
Bauer, J., Jannach, D., 2018. Optimal pricing in e-commerce based on sparse and noisy data. Decis. Support Syst. 106, 53–63.
Bogina, V., Mokryn, O., Kuflik, T., 2019. Will this session end with a purchase? Inferring current purchase intent of anonymous visitors. Electron. Commer. Res. Appl., 100836.
Bortz, J., Lienert, G.A., Boehnke, K., 2008. Verteilungsfreie methoden in der biostatistik. Springer-Verlag, pp. 275.
Carpenter, J., Bithell, J., 2000. Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. Stat. Med. 19 (9), 1141–1164.
Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, pp. 785–794.
Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anil, R., 2016. Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, pp. 7–10.
Cohen, M.C., Leung, N.H.Z., Panchamgam, K., Perakis, G., Smith, A., 2017. The impact of linear optimization on promotion planning. Oper. Res. 65 (2), 446–468.
Cohn, D., Atlas, L., Ladner, R., 1994. Improving generalization with active learning. Mach. Learn. 15 (2), 201–221.
den Boer, A.V. (2015). Dynamic pricing and learning: historical origins, current research, and new directions. Surveys in operations research and management science, 2.
Fassnacht, M., El Husseini, S., 2013. EDLP versus Hi–Lo pricing strategies in retailing—a state of the art article. J. Bus. Econ. 83 (3), 259–289.
Felgate, M., Fearne, A., 2015. Analyzing the Impact of Supermarket Promotions: A Case Study Using Tesco Clubcard Data in the UK. In: The Sustainable Global Marketplace. Springer, Cham, pp. 471–475.
Fisher, M., Gallino, S., Li, J., 2017. Competition-based dynamic pricing in online retailing: A methodology validated with field experiments. Manage. Sci. 64 (6), 2496–2514.
Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 1189–1232.
Gentile, C., Li, S., Kar, P., Karatzoglou, A., Zappella, G., Etrue, E., 2017. On context-dependent clustering of bandits. In: Proceedings of the 34th International Conference on Machine Learning. JMLR.org, pp. 1253–1262 Vol. 70.
Greenstein-Messica, A., Rokach, L., Friedman, M., 2017. Session-based recommendations using item embedding. In: Proceedings of the 22nd International Conference on Intelligent User Interfaces. ACM, pp. 629–633.
Koehn, P., 2004. Statistical significance tests for machine translation evaluation. Proceedings of the 2004 conference on empirical methods in natural language processing.
Li, L., Chu, W., Langford, J., Schapire, R.E., 2010. A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web. ACM, pp. 661–670.
Li, S., Karatzoglou, A., Gentile, C., 2016. July). Collaborative filtering bandits. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. (pp. 539–548). ACM.
Ma, X., Sha, J., Wang, D., Yu, Y., Yang, Q., Niu, X., 2018. Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning. Electron. Commer. Res. Appl. 31, 24–39.
Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
Misra, K., Schwartz, E.M., Abernethy, J., 2018. Dynamic online pricing with incomplete information using multi-armed bandit experiments.
Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation. Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014).
Pourgholamali, F., Kahani, M., Bagheri, E., Noorian, Z., 2017. Embedding unstructured side information in product recommendation. Electron. Commer. Res. Appl. 25, 70–85.
Ricci, F., Rokach, L., Shapira, B., 2015. Recommender systems: introduction and challenges. In: Recommender Systems Handbook. Springer, Boston, MA, pp. 1–34.
Rokach, L., Naamani, L., Shmilovici, A., 2008. Pessimistic cost-sensitive active learning of decision trees for profit maximizing targeting campaigns. Data Min. Knowl. Disc. 17 (2), 283–316.
Santini, F.D.O., Sampaio, C.H., Perin, M.G., Vieira, V.A., 2015. An analysis of the influence of discount sales promotion in consumer buying intent and the moderating effects of attractiveness. Revista de Administração (São Paulo) 50 (4), 416–431.
Shan, Y., Hoens, T.R., Jiao, J., Wang, H., Yu, D., Mao, J.C., 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 255–262.
Syntetos, A.A., Boylan, J.E., Croston, J.D., 2005. On the categorization of demand patterns. J. Operat. Res. Soc. 56 (5), 495–503.
Van Heerde, H.J., Leeflang, P.S., Wittink, D.R., 2000. The estimation of pre-and post promotion dips with store-level scanner data. J. Mark. Res. 37 (3), 383–395.
Wang, X., He, X., Feng, F., Nie, L., Chua, T.S., 2018. Tem: Tree-enhanced embedding model for explainable recommendation. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 1543–1552.
Ye, P., Qian, J., Chen, J., Wu, C.H., Zhou, Y., De Mars, S., Zhang, L., 2018. Customized Regression Model for Airbnb Dynamic Pricing. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, pp. 932–940.
Zhang, M., Guo, X., Chen, G., 2016. Prediction uncertainty in collaborative filtering: Enhancing personalized online product ranking. Decis. Support Syst. 83, 10–21.