International Journal of Data Science and Analytics (2023) 16:163–172
https://doi.org/10.1007/s41060-020-00233-1

REGULAR PAPER

Combination of individual and group patterns for time-sensitive purchase recommendation

Anton Lysenko[1] · **Egor Shikov**[1] · **Klavdiya Bochenina**[1]

Received: 20 January 2020 / Accepted: 22 September 2020 / Published online: 8 October 2020

Abstract
Due to the availability of large amounts of data, recommender systems have quickly gained popularity in the banking sphere. However, time-sensitive recommender systems, which take into account the temporal behavior and the recurrent activities of users to predict the expected time and category of next purchase, are still an active field of research. Many researchers tend to use population-level features or their low-rank approximations because the client's purchase history is very sparse with few observations for some time intervals and product categories. But such approaches inevitably lead to a loss of accuracy. In this paper, we present a generative model of client spending based on the temporal point processes framework. The model is built in the way, to bring more individuality for the clients' purchase behavior which takes into account individual purchase histories of clients. We also tackle the problem of poor statistics for people with a low transactional activity using effective intensity function parameterizations, and several other techniques such as smoothing daily intensity levels and taking into account population-level purchase rates for clients with a small number of transactions. The model is highly interpretable, and its training time scales linearly to millions of transactions and cubically to hundreds of thousands of users. Different temporal-process models were tested, and our model with all the incorporated modifications has shown the best results in terms of both error of time prediction and the accuracy of category prediction.

Keywords Point processes · Transactional data · Mixture models · Recommendation · Machine learning

1 Introduction

Banks have been using corporate databases for a long time, which led to the accumulation of a large amount of different data on the purchasing behavior of customers. Thanks to this, as well as the development of machine learning algorithms, banks have moved from using simple models, such as LRFM (length, recency, frequency, and monetary) model to more complex recommendation models. Typically, these models were used to back up bonus programs developed together with trade and service enterprises for a long fixed period, such as a month. However, the use of time-limited offers can be much more profitable. They may sound as follows: "Hurry up and spend 100 dollars at our partner's restaurant and get double cash-back. The offer is valid until 10 pm. April the 5th !!!." The efficiency of limited-time offers is explained by the psychological phenomenon known as loss aversion, which refers to people's tendency to prefer avoiding losses to acquiring equivalent gains. The customer is offered a limited time to make a purchase in a certain category. This offer can be delivered via the bank's mobile application in the form of a coupon. To do this, it is necessary to develop a recommendation system that would predict: (1) the time of the next purchase of the client; (2) the most likely categories of purchase.

The problem of predicting the return time can be solved using classical methods, dividing the time into intervals. First of all, the time can be simply divided into a set of intervals, and static latent feature models can be applied [7,12]. However, such models have several disadvantages: First, it is

A preliminary version of this work was presented at the Most-Rec Workshop at CIKM 2019, but it has not been published in any proceedings or journal before.

✉ Anton Lysenko
blinkop@gmail.com

Egor Shikov
shikovegor86@gmail.com

Klavdiya Bochenina
k.bochenina@gmail.com

[1] ITMO University, 49 Kronverkskiy prospect, Saint Petersburg, Russian Federation

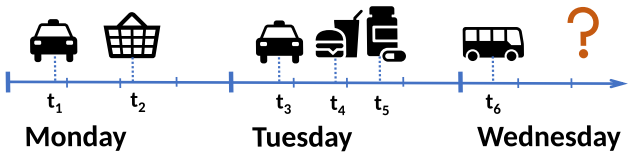

Fig. 1 A fragment of the customer's purchase history. Our model is intended for predicting the category and time of spending based on the client's transaction history

unclear how to choose the interval length parameter; second, different users may have very different time scales; third, the history of last spendings cannot be incorporated into the model.

The point process-based models [1] can overcome these limitations. By nature, they generate continuous timestamps and the length between them can vary depending on the client's activity. Also, the excitation factors can be added to take into account the last client transactions (Fig. 1).

This problem can be formalized in the following way:

Let $[t_0,T]$ be the observation window with some number of transactions of every customer in every category. For each customer u, we have a set of timestamps representing the history of transactions $T_u = \{t_{u,1}, ..., t_{u,n}\}$ and their associated categories $C_u = \{c_{u,1}, \ldots, c_{u,n}\}$ (for example, gas stations, restaurants, transport, etc.).

We need to build a model capable of predicting the time t and category c of the next transaction of the client and the sequence of transactions as well.

In solving this problem, the following features should be taken into account:

- The transaction history of the absolute majority of clients is highly sparse, and many elements (client, category, time) are non-observed.
- The last spendings are quite important and should be taken into account.
- The level of transactional activity of clients differs a lot.
- There are millions of transactions in the dataset, which opens up the question of scalability.

In paper [10], the authors decided not to use any client-specific parameters resulting in a model with only 390 parameters for 10 categories. This approach uses population-generalized consumption levels, and the transactional history of a particular customer is applied only through the introduction of the terms responsible for self-excitation. Therefore, it is obvious that the forecasts will be biased toward the average level of activity for the dataset, which will lead to big errors for customers with a small daily number of transactions.

Approaches based on client–category–time co-occurrence matrix factorization may be viable [16]. However, some authors [8] argue that these methods tend to oversmooth distributions resulting in excessively high probabilities of unseen client–category–time combinations.

In most works, the authors pay great attention to the factor associated with mutual excitement. However, we believe that members associated with the inhomogeneous Poisson process, who are responsible for the "timetable" and their modification can bring the best results, should have a greater impact on our dataset.

Latent representation and deep neural network models are possible to solve the sequential recommendations problem [14]. For example,[13] presents a framework for studying implicit and explicit dependencies of elements, where they pay great attention to the implicit side of the problem. An example of recommendations based on a neural network is presented in [15], where the authors create a sequential network with Purpose-Specific Recurrent Units that captures the membership of items, improving the results of recommendations.

Since the results of this study were planned to be used for the recommendation system in the bank, we set ourselves the task to build an interpretable model and refrained from using neural net approaches [3,5,6,11]. The model presented below is a generative model of client spending and allows us to generate the purchasing activity of the population, which is also of interest to the study.

2 Data

The data that were used during the current work were provided by the partner bank. It includes 67+ millions of transactions of ∼143,000 clients over the period of one year, where each transaction is represented by its client's unique ID, transaction time and date, the amount in rubles, and the category. The category is represented via the merchant category code (MCC)—4-digit code, which is widely used in the banking sphere to mark the transaction category. By using the MCC, we gain a very big number of different categories, while two MCC's can represent pretty same categories, e.g., 3001 code stands for the American airlines and 3009 is Air Canada where both are the airlines' companies. To avoid the issues with much MCC's prediction, we transformed the 4-digit representation to 2-digit, where categories are grouped by their purpose, e.g., every grocery shop becomes one "grocery" category, etc.

By looking at the clients' daily average number of transactions distribution in Fig. 2, we can see that clients are distributed widely—there are clients with very low transaction activity (one transaction), or there are clients with very high activity (4–5 transaction per day). As is intuitively clear, the purchasing activity depends on the current time, which is illustrated in Fig. 3a and b. People spend most on Mondays, and for hours, it is the evening and the middle of the

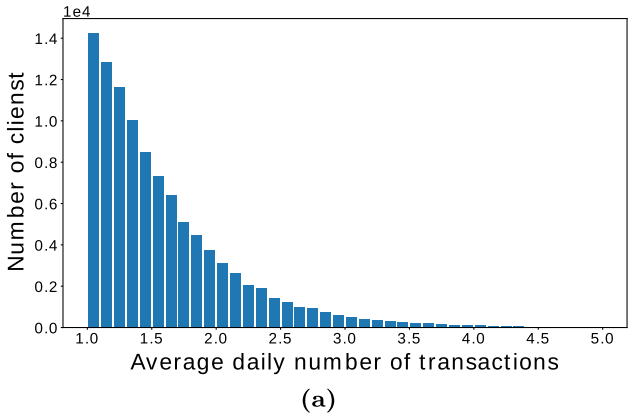

Fig. 2 Properties of the frequency of payments in the dataset: Distribution of the number of transactions in several popular categories

day, which is typically the end of workday and lunchtime, respectively. For our test purpose, we took only the top 10 most frequent categories which are described in Table 1.

Other data filtering includes multiple steps—the objective is to leave only enough active clients and exclude all others, that, for example, hold their bank card only for gaining money on payday and transfer them on the other card or into cash. Another example of unlikely clients is those, whose cards expire just right after the beginning of the observed period, or clients, that got their card right before the end of the year. So the following filtering steps were performed—first of all, we removed all the transactions, that took place at 00:00:00 time due to the bank issue when some of the transactions have delayed operations and are massively performed at night. The second step was to remove categories outside the top 10. Next, to sort out clients, who were not active throughout the whole year we left only those of them, who have at least one transaction before February and at least one after November. At last, we managed to take only clients with at least 20 transactions over a year, which in our opinion are active enough. Going through all the steps described above, we left 115,089 clients with 47,721,556 transactions in total.

3 Model

3.1 Temporal point processes

The temporal point process is usually represented via its conditional intensity function, which can be interpreted as the probability of an event occurring in a small time window. Formally, given the history of previous events at point t as $H_t = \{t_1, t_2, \ldots, t_n\}$, where $t_i < t_{i+1}$ and $t_n < t$, the intensity function looks as follows:

$$\lambda^*(t) = \lim_{h \to +0} \frac{P\big(\text{event } in\ (t, t+h] | H_t\big)}{h}, \tag{3.1}$$

where each point in history H_t can be marked with some event category as a pair (t_i, d_i), which in our scenario is transaction category, and the asterisk means that the intensity function is conditioned by the history of events.

The simplest process is the homogeneous Poisson process, which intensity function is represented only by base rate $\lambda_0 > 0$. It is constant through the whole nonnegative domain, which means that the probability of an upcoming event is independent of any conditions. By itself, the homogeneous Poisson process does not make much sense because in our case, it just evaluates the average frequency of clients purchases and as output gives constant intensity for any client with any history. To capture some time dependencies, we can use the inhomogeneous Poisson process, which is described below.

3.2 Inhomogeneous Poisson process

With inhomogeneous Poisson process, we allow the intensity function to vary according to a deterministic function of t, with bounding $\lambda(t) \geq 0, t \geq 0$. In our case, as the t domain refers to time, we can capture the time dependence with the set of indicator functions F and some weights to each of the time feature, described in Table 2. As a result, we obtain the following intensity function for category d:

$$\lambda_d(t) = \lambda_{d0} + \sum_{f_j \in F} \mu_{dj} f_j, \tag{3.2}$$

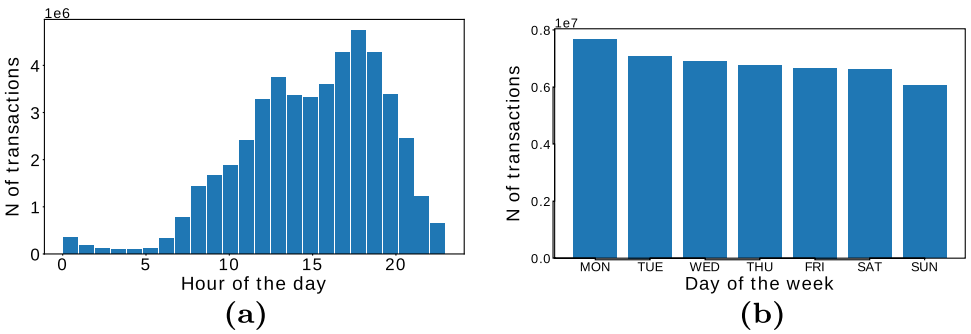

Fig. 3 Total number of purchases in the dataset **(a)** at each hour of the day and **(b)** on each day of the week

Table 1 Purchase categories used

N	Category name	Average monthly number of transactions	Fraction of clients with transaction in category
1	Gas stations (GAS)	1.58	0.43
2	Medical goods (MED)	1.56	0.62
3	Clothing (CLO)	1.07	0.45
4	Personal services (PER)	0.97	0.43
5	Alcohol (ALC)	0.60	0.20
6	Supermarkets (GRO)	16.2	0.95
7	Restaurant (RES)	5.46	0.77
8	Special stores (SPE)	0.84	0.37
9	Transport (TAX)	2.39	0.44
10	Financial services (FIN)	4.05	0.84

Table 2 Time features that are captured by the inhomogeneous Poisson process

Index j	Time feature
0-23	Hour of a day
24	Monday–Thursday
25	Friday
26	Saturday and Sunday

which means that the intensity at some point t_0 is defined by the sum of base rate λ_0 and every μ_{dj}, that is, active at the t_0, e.g., if we want to get the intensity at 1:30 pm on Friday, we sum $\mu_{d,13}$ and $\mu_{d,25}$ with λ_{d_0}. Hour dependency helps the model to capture regular purchases so that we could predict periodic purchases more precisely and make stock with less duration for it, but with longer for aperiodic purchases which patterns can be hardly captured with such short periods as concrete hours. This makes sense because if we will look at the weekday and the hour distributions of our dataset, presented in Fig. 3a and b, we can see the dependence of current time.

By training this model, we are making all the parameters (Λ and M) shared no matter what client we predict for. To do this, we evaluate the log-likelihood, which looks as follows:

$$\mathcal{L}(\{(t_1, d_1), \ldots, (t_n, d_n)\}) = \sum_{i=1}^{n} log(\lambda^*_{d(t_i)}(t_i)) - \sum_{d=1}^{D} \int_0^T \lambda^*_d(\tau)\mathrm{d}\tau - \gamma||\Theta||_2^2, \tag{3.3}$$

where the γ is a L2 regularization parameter and $\Theta = \{\Lambda, M\}$. We do it for each of the clients in the training set and then take the average as a function for maximization. This method brings the possibility to parallelize the learning process well by computing each log-likelihood separately and then just take the average.

But the problem of estimating the parameters in such a way is that it takes much time even if we parallelize it. It takes about ten hours to get the likelihood converged for the data set with the size of 115,000+ clients with 47+ million transactions in total. As it was said earlier, the homogeneous Poisson process intensity function base rate is just average frequency of clients' purchases, and one can see the only difference between the homogeneous and inhomogeneous one in Fig. 4a and b—the latter one is partly constant on some intervals. So, by that we are coming to another approach to learning of the model—estimating average frequencies on that intervals, and as a result, we must obtain the same result, as learning via the maximization of the log-likelihood. To estimate a parameter of the concrete interval, we must calculate the duration of this interval, combining all non-mutual exclusive time features that are active at the interval.

For example, to estimate the intensity at Friday 2 pm, we calculate the duration of the intersection of the time intervals—all Fridays and all 2 pm wall-clock values, and then divide the number of transactions at Friday 2 pm by obtained duration value. Formally, we say that

$$\mu_{d,14} + \mu_{d,25} + \lambda_{d0} = \frac{\sum_{i=1}^{N_{\text{trans}}} I[t_i \in \text{Friday 2 pm}]}{\int_0^T f_{\text{Friday}\cap 2\,\text{pm}}(\tau)\mathrm{d}\tau} \tag{3.4}$$

and if we go in such way with all of non-mutual exclusive combinations, we get the algebraic system of linear equations, which is consistent. Starting from the pair of (Monday–Thursday, 12 pm) and going down to the last pair (Saturday–Sunday, 11 pm), we obtain the following system

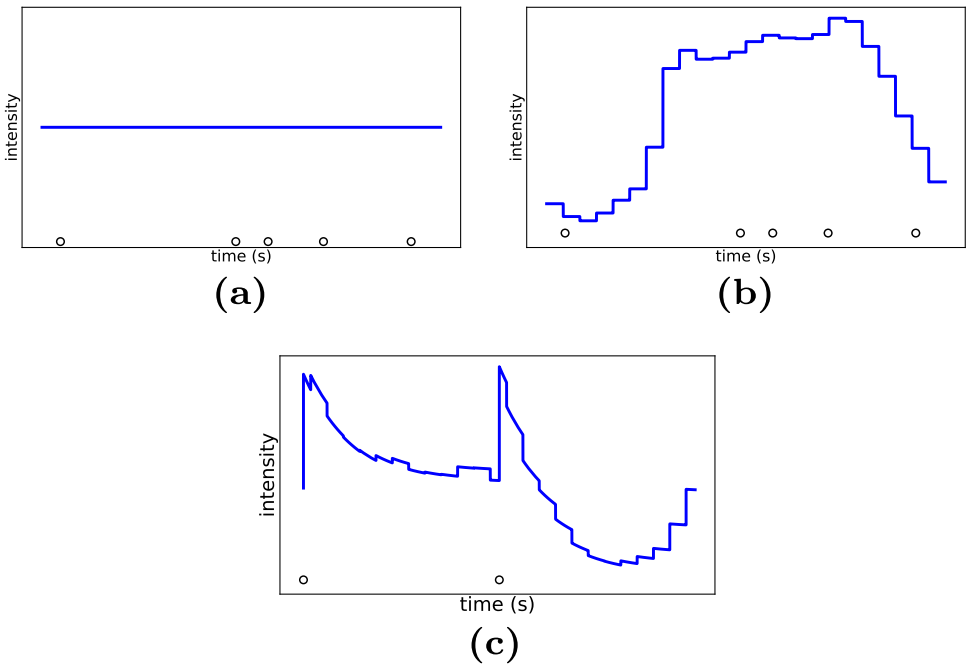

Fig. 4 Intensity functions for homogeneous Poisson **(a)**, inhomogeneous **(b)** and Hawkes **(c)** processes

of equations:

$$\begin{cases} \mu_{d,0} + \mu_{d,24} + \lambda_{d0} = \frac{\sum_{i=1}^{N_{\text{trans}}} I[t_i \in \text{Mon}-\text{Thu } 12\,\text{pm}]}{\int_0^T f_{\text{Mon}-\text{Thu}\cap 12\,\text{pm}}(\tau) d\tau} \\ \mu_{d,1} + \mu_{d,24} + \lambda_{d0} = \frac{\sum_{i=1}^{N_{\text{trans}}} I[t_i \in \text{Mon}-\text{Thu } 1\,\text{am}]}{\int_0^T f_{\text{Mon}-\text{Thu}\cap 1\,\text{am}}(\tau) \text{d}\tau} \\ \dots \\ \mu_{d,23} + \mu_{d,26} + \lambda_{d0} = \frac{\sum_{i=1}^{N_{\text{trans}}} I[t_i \in \text{Sat}-\text{Sun } 11\,\text{pm}]}{\int_0^T f_{\text{Sat}-\text{Sun}\cap 11\,\text{pm}}(\tau) \text{d}\tau} \end{cases}, \tag{3.5}$$

where T is the end of the observation period. By solving it, we get our vector parameter M for category d. This approach speeds up learning time significantly—the boost is about 30 times over likelihood maximization, and this opens new opportunities for creating models, which is described in the next sections.

While having a pretty big dataset with many transactions for each client, we developed a model, where we do not learn the parameters on the whole clients set, but rather we learn them when we want to make a prediction for a particular client by using the approach of solving the linear system equation. This means that we are now conditioned on the client's history with an inhomogeneous Poisson process.

3.3 Intensity factorization

Since the 3-dimensional matrix (client, category, time) describing the process is highly sparse and many elements are unobserved, we also tried to do the following factorization, laying out the client's preference for certain categories and his schedule:

$$\mu(\text{client}, d, t) = \mu(\text{client}, d) \cdot \mu(\text{client}, t), \tag{3.6}$$

where t stands for time and d for category. As input for prediction, we take the history as one of the arguments, with the sequences of timestamps and related categories $H_t = \{(t_1, d_1), \dots, (t_n, d_n)\}$. To not suffer from the case, when the client has not many statistics on some categories, we decided to calculate the parameters that are shared for all categories but are scaled to their frequencies. Formally, by getting the vector of parameters $\theta = \{\lambda_0, \mu_0, \dots, \mu_{26}\}$, calculated without relation to the categories; to bring those vectors for all categories separately, we multiply θ by $\frac{\sum_{i=0}^{N} I[d_i = d]}{N_{\text{trans}}}$ for each category d. By doing that, we obtain the same pattern of purchasing through the time features, which is not the best solution if we got a relatively big history of every category, but it works well if the client has not many statistics on the purchasing. The resulting functions for every category are presented in Fig. 5a.

3.4 Intensity smoothing

Here, we assume that there is some variation in intensity caused by small statistics for some users. And a person can make a purchase a little earlier or a little later. The logic is this: Let's assume that a client has some transactions at 11 am, and no transactions at 12 am. Therefore, his $\mu(12\,\text{am})$ would be zero, which can often be wrong, especially if the customer does not buy a lot. So, we mix the intensities of the adjacent

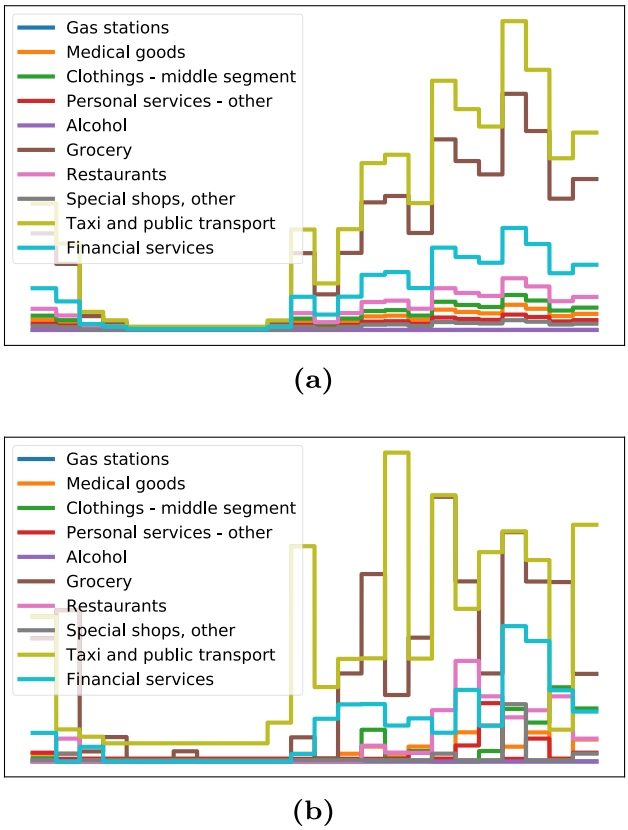

Fig. 5 Intensity functions for one day period with some client's parameters, estimated via linear system for all categories at ones **(a)**, for each category separately **(b)**. Different colors means intensities for different categories

clocks into the intensity of each hour.

$$\widetilde{\mu}_i = (1-\epsilon)\cdot\mu_i + \epsilon\cdot\frac{\mu_{i-1}+\mu_{i+1}}{2} \tag{3.7}$$

$$\widetilde{\widetilde{\mu}}_i = \frac{\sum_{i=1}^{24}\mu_i}{\sum_{i=1}^{24}\widetilde{\mu}_i}\widetilde{\mu}_i, \tag{3.8}$$

where $\epsilon = \frac{\alpha}{N}$. At the same time, if the customer buys often, we believe that the distribution of his purchases by hours deserves more confidence. (And accordingly at the expense of 1/N, we will have almost no mixing.) We also believe that the total intensity per day should remain single before and after smoothing, so we conduct renormalization and get the final normalized $\widetilde{\widetilde{\mu}}_i$ as the result.

3.5 Mixture models

The idea of mixture parameters from the baseline model and parameters, gained from solving the linear system, comes from the case, when we want a particular client to have a chance of purchasing in category or time, that is, not lying on his/her pattern of purchasing, but at the same time, we want to save the individuality with the parameters from the linear system.

Let's say that parameters obtained from learning on the whole dataset are denoted as θ_{avg} and parameters gained from particular client are denoted as θ_{clt}, and then, we define the mixture model as the linear combination of the parameters:

$$\theta = w_{\text{avg}}\cdot\theta_{\text{avg}} + w_{\text{clt}}\cdot\theta_{\text{clt}}, \tag{3.9}$$

where $w_{\text{avg}} + w_{\text{clt}} = 1$. We can interpret the w parameters as how much impact do the baseline and the clients' parameters, respectively.

Now, the problem of having not many statistics on some clients, described in Sect. 3.3, can be dropped out, as we capture some degree from the baseline parameters. And this brings two ideas, how to build the mixture model—the first is to estimate individual parameters as described in Sect. 3.3, and the second is to estimate them separately for each category. The difference of the intensities calculated with both approaches is shown in Fig. 5a and b, where it is seen that when we calculate it separately, we do not repeat the same pattern for all categories with just different magnitudes. We present the w parameters as the hyper-parameter of the model, and to tune it, we can just make the grid search through some set of them.

3.6 Mutual excitation

The model described above does not take into account the impact of recent purchases, which can be very significant. We restricted our consideration of pair interactions purchases in one category purchase in the other and modified the model in the following way:

- Added exponential terms similar to [10]
- Removed the restrictions on the nonnegativity of beta coefficients. In the majority of the works utilizing Hawkes processes, positive β coefficients were used. However, this significantly reduces the expressiveness of the model. And we see evidence of such phenomena in our dataset. We have built distributions of the inter-purchase time in the categories of gas stations versus supermarkets and gas stations versus transport and conducted a seasonal decomposition using moving averages. It can be seen in Fig. 6 that the trend line for these pairs is tilted in different directions, which indicates different signs of the β coefficients. It shows the distribution of inter-purchase time for gas stations versus transport. We observe that with the increase in time after purchase in the gas stations, the number of purchases in the transport category increases.
- In order to fulfill the restriction $\lambda > 0$, we have to modify the intensity, so we take only the positive part:

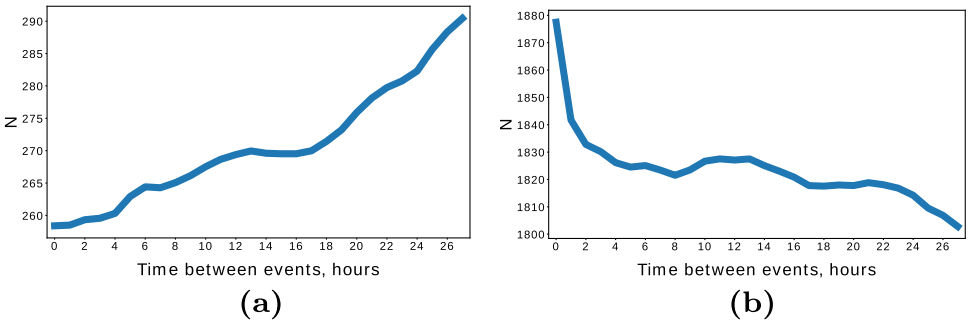

Fig. 6 Indication of different signs of coefficients of mutual excitation. Gas stations versus transport **(a)**; gas stations versus supermarkets **(b)**

$$\lambda_d^*(t) = max\Bigg(+0, \lambda_{d0}^{cl} + \sum_{f_j \in F} \mu_{dj}^{cl} f_j + \sum_{d'=1}^{D} \sum_{\substack{d(t')=d' \\ t' \in H_t}} \beta_{dd'} e^{-\alpha_{dd'}(t-t')} \Bigg) \tag{3.10}$$

The μ coefficients here were not trained but taken from the previous points. We expect the self-generating term is small and therefore will not affect the μ values. β coefficients are not individual, and they depend only on category indices. Since there are only a few beta coefficients (100 for our dataset), there is no reason to perform the training on the whole dataset, and we used a small part of it to reduce the calculation time. The training was conducted by minimizing the likelihood function using the L-BFGS-B [2] method.

4 Experiments

4.1 Prediction

Both models can generate as output next event time and category, and the sequences of the events by predicting the events one right after another, taking previously as a history. To generate a time of the next event, the Ogata's modified thinning algorithm [4] was used in the case of Hawkes process and algorithm for simulation inhomogeneous Poisson process [4] for the other one. In both cases, the following prediction algorithm for time and category was used:

We implemented the evaluation of the median of the 100 runs to take the most probable time prediction (1), and then, we predict category based on that time by the multinomial distribution. For a simulation of an event, we used a simulation algorithm for inhomogeneous Poisson process/Ogata's thinning modified algorithm in case of the process we simulate. To generate a sequence of the events, we can just run Algorithm 1 multiple times, each time starting from the last event time.

Algorithm 1 Prediction time and category of the next event

1: **procedure** PREDICT($t_0, history$)
2: $time[100]$
3: **for** integer i in 100 **do**
4: $time[i] = simulate\ one\ event$
5: **end for**
6: t = $median(time)$
7: c = $multinomial(\{\frac{\lambda_0^*(t)}{\sum_{d=1}^{D}\lambda_d^*(t)}, ..., \frac{\lambda_D^*(t)}{\sum_{d=1}^{D}\lambda_d^*(t)}\})$
8: Return (t, c)
9: **end procedure**

4.2 Evaluation metrics

To measure the quality of the built models, we divided our dataset into the train set and test set as follows—we trained on the time period from the beginning of January till the end of October. We used two types of metrics in this work:

- *Next purchase* Only the first event since the start of the test set for every client was predicted. Then, several metrics were calculated:
  - *Time error* We tried mean / median / 75 percentile error of the timestamp (given in seconds) of the next event and settled with the median relative absolute error.
  - *Accuracy* Accuracy for category prediction averaged among all categories.
- *Sequence of events* A chain of events for every client was generated.
  - *Generation ratio* The ratio of the number of generated events to the number of real events

First, only MAE was calculated. But later, it was realized that the error is too big for clients with high transaction activity—some of them can perform 5–10 transactions per day, but the error can be much higher, than the average time between transactions. The main reason comes from the fact that some clients start their activity only a long period after the test period begins, which is intuitively clear—it can be a vacation or something else. For this reason, we tried the 50 and 75

percentiles as they are more adequate in our case. At last, we settled with MdRAE (median relative absolute error), which looks like the following: $\text{MdRAE}_i = \frac{\text{median}(\text{abs}(t-\hat{t})_i}{\text{mean}(t_{j+1}-t_j)_i}$, where $j = 1 \ldots N_i - 1$, N_i - number of transactions for client i. To estimate the best values for w_{avg} and w_{clt} parameters, we used the grid search within [0; 1] segment. It was discovered that both mixture models showed the best results for w values 0.4 and 0.6, respectively, while the worst case was the (1, 0) pair, which gives 12.7% more time error comparing to the best one, so the resulting parameters look like the following: $\theta = w_{\text{avg}} * \theta_{\text{avg}} + w_{\text{clt}} * \theta_{\text{clt}}$, where $\theta = \{\lambda, \mu\}$.

4.3 Models evaluated

We compared the models with modifications mentioned above with several models from [10] and our recent work [9].

- *Inhomogeneous Poisson process* Simple inhomogeneous Poisson process model with parameters shared among all clients.
- *Hawkes process* Multidimensional Hawkes processes with the time-varying component from [10]. Parameters shared among all clients.
- Scaled Poisson process. Inhomogeneous Poisson process model with intensity scaling proportional to each client's average number of transactions.
- *Individual Poisson process* Inhomogeneous Poisson process with μ coefficients calculated for each client.
- *Smoothing* Smoothing of hourly coefficients.
- *Mixing with group coefficients* Individual coefficients mixed with dataset average coefficients to properly account for the rare categories.
- *Mutual excitation* The self-excitation part added to the previous model similar to the Hawkes process to take into account the impact of recent purchases.

4.4 Prediction performance

All obtained results are shown in Tables 3, 4, 5, where the models' descriptions are related to Table 4.3.

By looking at the time error, we can see that the best performance is obtained with a mixture model of smoothing individual and averaged coefficients including Hawkes process with $\beta > 0$ restriction.

pg

Table 4 illustrates that the modification with mixture, smoothing, and self-excitation achieves the best results in terms of accuracy, while the baseline models perform much worse.

Our experiments reveal that the smoothing methods achieve the best results. (In case of time, we mix smoothing

Table 3 Median relative absolute time error

Model	MdRAE
Poisson baseline	0.53
Hawkes baseline	0.63
Poisson baseline + scaling	0.47
Factorization + Smoothing	0.47
Poisson individual	0.49
Factorization	0.48
Smoothing	0.46
Factorization + Smoothing + Mixture	0.47
Smoothing + Mixture	0.47
Smoothing + Mixture + Hawkes($\beta > 0$)	**0.44**
Smoothing + Mixture + Hawkes	0.58

Table 4 Category accuracy

Model	Accuracy
Poisson baseline	0.233
Hawkes baseline	0.3245
Poisson baseline + scaling	0.3485
Factorization + Smoothing	0.3345
Poisson individual	0.3405
Factorization	0.3345
Smoothing	0.345
Factorization + Smoothing + Mixture	0.287
Smoothing + Mixture	0.3035
Smoothing + Mixture + Hawkes($\beta > 0$)	0.302
Smoothing + Mixture + Hawkes	**0.349**

The bold value row shows the best result for current metric

Table 5 Ratio of the number of events in generated sequences and real sequences

Model	Generation ratio
Poisson baseline	91.55
Hawkes baseline	97.24
Poisson baseline + scaling	94.72
Factorization + Smoothing	94.24
Poisson individual	94.26
Factorization	94.25
Smoothing	93.63
Factorization + Smoothing + Mixture	93.17
Smoothing + Mixture	92.80
Smoothing + Mixture + Hawkes($\beta > 0$)	89.41
Smoothing + Mixture + Hawkes	89.91

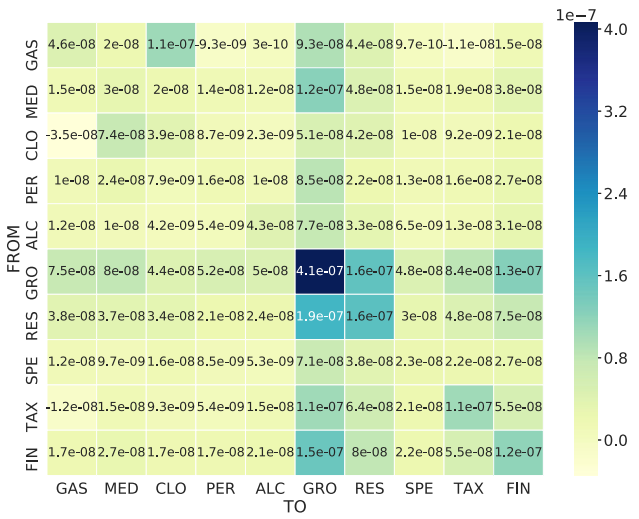

Fig. 7 Analysis of mutual excitation coefficients. Categories are enumerated in Table 1

with averaged parameters, in case of accuracy—smoothing does well by itself.)

These results suggest that most of the purchases are not induced by previous ones but are determined by the average level of consumption by category. This is especially noticeable on the the time error results, with the Hawkes process performing the worst. As for the accuracy, we can conclude that considering the individual purchasing activities of clients improves the results quite significantly compared to the baselines. On the other hand, the experiments indicate that self-excitation improves the results. Also, using this kind of model, we can exclude biases associated with individual consumption levels and analyze the β coefficients to draw conclusions about how purchases in some categories affect the likelihood of purchases in others. Also, it should be noted that all the models without the self-excitation are much more scalable in terms of the training time.

The generation ratio presented in Table 5 can be considered as a kind of sanity test. Although the main task of this work is to predict the category and the timestamp of the next purchase, it was important for us to make sure that all the methods presented generate approximately the same number of events as observed.

4.5 Model interpretation

The structure of our model provides an opportunity for purchase behavior process interpretation. The analysis of the β coefficients can be made, as it is of great interest for marketers. The elements of β matrix are given in Fig. 7, where the values near the 10^{-8} or less show nearly no correlation between the categories and values $> 10^{-7}$ show meaningful correlation. From the matrix, we can distinguish the following patterns:

- Grocery trigger almost all other categories.
- Transport-gas and gas-transport coefficients are both negative.
- Cash withdrawal triggers grocery stores and another cash withdrawal.
- A word of caution is needed about the fact that the greatest β values seem to lie on the diagonal. It does not seem right and may be caused by overfitting.

5 Conclusion

In this work, we proposed a novel set of models that combine the Poisson processes with individual coefficients for each client and mutual/self-excitation behavior and allow predicting the occurrence and time of spending in various categories based on the client's transaction history.

We argue that despite the frequent use of low-rank approximations and group features, individual parameters cannot be ignored. We show that excitation can be both positive and negative and propose a model that allows for this to be taken into account. We also offer several options for modifying individual coefficients to improve the model.

Different variants of the model were tested, and models based on solving the linear system of equations have shown the best results in terms of both error of time prediction and the accuracy of category prediction.

The presented model is interpretable and provides insights on the dynamics of the consumer's purchase behavior.

We show that the model can be used for the modeling of the purchasing activity of the population, which is of fundamental interest.

Acknowledgements This research is financially supported by The Russian Science Foundation, Agreement NO19-71-10078.

References

1. Aalen, O., Borgan, O., Gjessing, H.: Survival and Event History Analysis: A Process Point of View. Springer Science & Business Media, Berlin (2008)
2. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput. **16**(5), 1190–1208 (1995)
3. Dai, H., Wang, Y., Trivedi, R., Song, L.: Recurrent coevolutionary latent feature processes for continuous-time recommendation. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, ACM, pp 29–34 (2016)
4. Daley, D.J., Vere-Jones, D.: An introduction to the theory of point processes: volume II: general theory and structure. Springer Science & Business Media, Berlin (2007)
5. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: Recurrent Marked Temporal Point Processes: Embedding Event History to Vector. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16 pp 1555–1564, https://doi.org/10.1145/

2939672.2939875,http://dl.acm.org/citation.cfm?doid=2939672.2939875 (2016)
6. Grob, G.L., Cardoso, Â., Liu, C.H., Little, D.A., Chamberlain, B.P.: A recurrent neural network survival model: Predicting web user return time. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11053 LNAI:152–168, https://doi.org/10.1007/978-3-030-10997-4_10, arXiv:1807.04098v1 [cs.LG] (2019)
7. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 447–456 (2009)
8. Kotzias, D., Lichman, M., Smyth, P.: Predicting consumption patterns with repeated and novel events. IEEE Trans. Knowl. Data Eng. **31**(2), 371–384 (2018)
9. Lysenko, A., Shikov, E., Bochenina, K.: Temporal point processes for purchase categories forecasting. Proc. Computer Sci. **156**, 255–263 (2019)
10. Manzoor, E., Akoglu, L.: Rush!: Targeted time-limited coupons via purchase forecasts. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 1923–1931 (2017)
11. Mei, H., Eisner, J. M.: The neural hawkes process: A neurally self-modulating multivariate point process. In: Advances in Neural Information Processing Systems, pp 6754–6764 (2017)
12. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web, ACM, pp 811–820 (2010)
13. Wang, S., Cao, L.: Inferring implicit rules by learning explicit and hidden item dependency. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2017)
14. Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., Orgun, M.: Sequential recommender systems: challenges, progress and prospects. arXiv preprint arXiv:2001.04830v1 [cs.IR] (2019a)
15. Wang, S., Hu, L., Wang, Y., Sheng, Q.Z., Orgun, M., Cao, L.: Modeling multi-purpose sessions for nextitem recommendations via mixture-channel purpose routing networks. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, pp 1–7 (2019b)
16. Wang, Y., Du, N., Trivedi, R., Song, L.: Coevolutionary latent feature processes for continuous-time user-item interactions. In: Advances in Neural Information Processing Systems, pp 4547–4555 (2016)