

Welfare-Optimized Recommender Systems

Benjamin Heymann¹, Flavian Vasile¹ and David Rohde¹

¹*Criteo AI Lab, 32 Blanche, Paris, France*

Abstract

We present a recommender system based on the Random Utility Model. Online shoppers are modeled as rational decision makers with limited information, and the recommendation task is formulated as the problem of optimally enriching the shopper’s awareness set. Notably, the price information and the shopper’s Willingness-To-Pay play crucial roles. Furthermore, to better account for the commercial nature of the recommendation, we unify the retailer and shoppers’ contradictory objectives into a single welfare metric, which we propose as a new recommendation goal. We test our framework on synthetic data and show its performance in a wide range of scenarios. This new framework, that was absent from the Recommender System literature, opens the door to Welfare-Optimized Recommender Systems, couponing, and price optimization.

Keywords

discrete choice, random utility models, neural networks, recommendation, dynamic pricing, welfare optimization

1. Introduction

1.1. Recommender Systems

Recommender Systems (RS) have proven successful almost everywhere in the digital economy, as they are a crucial component of the engines that run digital advertising, e-commerce websites, digital media, music and video providers and social networks. For the last few years, there has been a growing effort – partially inspired by economic theory– in providing a principled way to design these systems. We contribute to this research effort by providing a proof of concept that recommender systems used by e-tailer and advertisers could be welfare maximizers. To do so, we leverage a user behavioural model summarized in Figure 1, and which will be formalized in Section 4.

1.2. Shortcomings of current approaches

Most reward-optimized recommendation systems measure an abstract form of user utility and not an actual monetary value. This situation likely stems from the preponderance of clicks as immediate reward feedback in real-world systems. But as the field and the industry mature, we need consistent and rigorous approaches for conversion-optimized recommendation systems. The apparent similarity of conversions and clicks from the point of the merchant/advertiser is misleading, and the current click-optimization approaches are conceptually lacking when it comes to conversion modelling. The three main differences are:

- While clicks are free for the users, conversions cost real money, and therefore the item price should play a big role in the utility computation within the Recommender System. In other words, the price of an item is a disincentive to buy, but not to click.
- Due to finite budgets, users have to make choices and items serving the same need are mainly in competition in the user shopping process. This is much better modelled as a Categorical distribution over the choices, instead of the standard Bernoulli used for click modelling.

Presented at the SURE workshop held in conjunction with the 18th ACM Conference on Recommender Systems (RecSys), 2024, in Bari, Italy.

✉ b.heyman@criteo.com (B. Heymann); f.vasile@criteo.com (F. Vasile); d.rohde@criteo.com (D. Rohde)

🌐 <https://benhey.github.io/> (B. Heymann)

🆔 0000-0002-0318-5333 (B. Heymann); 0009-0006-3578-5655 (F. Vasile); 0000-0002-0661-6266 (D. Rohde)



© 2024 Copyright 2024 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

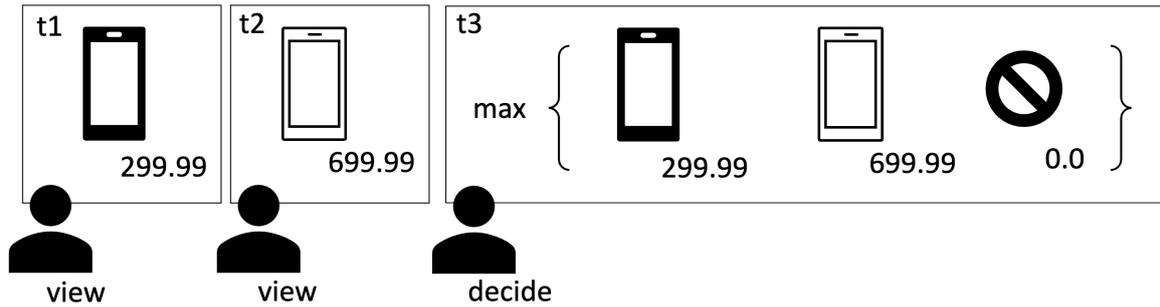


Figure 1: The model of the user as a rational decision-maker with a limited awareness set. At time t_1 and t_2 , the user observes different models of phones, with their prices. At time t_3 , they decide whether to buy one of the two phones, or to leave the website empty-handed. This decision is the output of a rational process of utility maximization that takes into account both the features and the prices of the two phones. The novelty of our framework is that we make explicit the difference between what the user knows (the awareness set) and the entire catalog.

- While clicks are conditioned on existing recommendation systems policies (they are bandit feedback by nature), conversions are in a large part organic and therefore less sensitive to changes in recommendation policy.

Also, because the goal of recommendation is to facilitate two-party transactions, one needs to clearly state on whose behalf the Recommender System is operating (the buyer or the seller's side).

1.3. Our approach

To address all these shortcomings, we rely at learning time on the user's browsing history to infer the set of items over which the user is making the decision to buy or to leave (Figure 1), and then we frame the problem of recommendation as the problem of finding *the optimal enrichment of the user's awareness set* (Figure 2).

Also, we propose a novel recommendation objective: *welfare maximization*, moving the focus from ways of splitting the pie – which is the main focus of multi-stakeholder literature – to how to maximize the pie. This leads us to an objective based on the Willingness-To-Pay (WTP), which measures the maximum price that the user is willing to pay for an item. More precisely, it is the price point where the user is indifferent to buying or not the item [1]. To be able to maximize the WTP objective, we propose a simple algorithm which can be seen as an extension of the classical Matrix Factorization (MF) algorithm [2, 3] that incorporates price and that can serve both as a user decision model and as a WTP estimator.

1.4. Paper contents

In Section 2, we cover the related work on recommendation with prices. In Section 3 we discuss potential applications of the Welfare objective for Recommendation. In Section 4, we introduce the user discrete choice model, and propose RUM-MF (random utility model via matrix factorization), a simple extension of existing MF algorithms. In Section A, we introduce our simulated environment and present our results against multiple performance-optimizing baselines on classical and welfare-based metrics. In Section 6, we conclude and outline future research directions.

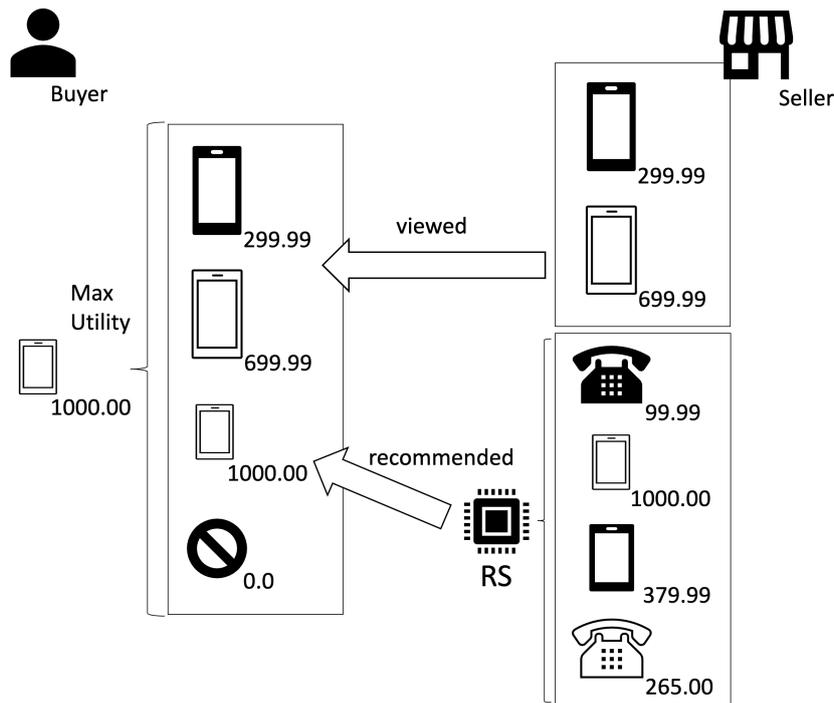


Figure 2: The role of recommendation in a limited information setup. Based on its knowledge of the user, the RS points the user to a new item that is likely to be useful. The value of RS is equal to the difference in utility between the user's decisions with or without the recommendation.

2. Related Work

2.1. Performance-optimized Recommendation

Performance-optimized Recommendation has a long history, with many initial works related to the problems of click-through rate (*CTR*) optimization for online advertising and search ranking. There are two categories of methods. The first relies on the label or reward given by the user, the second directly learns an order on the items.

The first class of methods was motivated by online advertising, and mostly relies on likelihood-based models. They are preferred due to their calibration properties, and range from the use of large-scale logistic regression with sparse features as in [4], Gradient Boosted Trees [5], Factorization and Field-aware Factorization Machines [6, 7], and more recently, Deep Neural Networks [8], Bayesian Methods [9], Variational Autoencoders [10], and Policy-based methods [8], leading to state-of-the-art results.

The second class of methods was motivated by search engine design and mostly relies on learning-to-rank models. This includes classical ranking methods such as SVM-Rank [11], Triplet Loss and Syamese Networks [12] and that have been more recently supplemented by Bayesian Personalized Ranking (BPR) [13, 14].

2.2. Price in Recommendation

Moving to specialized approaches for conversion modelling and the use of price in recommendation, the literature is quite varied and ranges from full user shopping decision modelling via multiple objective matrix factorization of the shopping data as in [15], to ranking approaches that supplements user - item affinity measures with global and local budget preferences, as in [16]. More recently, [17] proposes a Graph Convolution Network-based method to infer category - specific price elasticity, and [18] explores several methods to compute item price bands and explicitly model user price affinity for various types of products.

2.3. Recommendation and Econometric Models

Probably the closest area of research is the existing body of work around the use of economic-inspired recommendation models. In [19], the authors outline the relationship between recommendation and surplus or welfare maximization and propose a way to estimate the personalized WTP based on the concavity of the utility function. In our paper, we extend their ideas to support what we think is the realistic case of limited awareness, and propose a different way to estimate the WTP that is closer to standard Matrix Factorization approaches. Similar limitations of the user decision set were already noted in the economic theory [20] literature.

In [21], the authors propose learning the WTP with auctions on mechanical turk, and using it for personalized promotion. In the follow-up paper [22], the authors propose Multiproduct Utility Maximization (MPUM) based on the complementarity property of some existing choices. In [16] the authors propose learning the in-session WTP using views and purchase decisions. More recently, in a series of papers [23, 24] the authors propose two economic recommendation objectives, Marginal Utility per Dollar and Weighted Expected Utility. Unlike in our approach, the authors are only considering the user-side. On the vendor side, [25, 26] both raise the point that the product prices have an impact on the seller's revenue and profit and propose alternative ways to incorporate this information in the final recommendation ranking function. In [27], the authors propose an extension to dynamic recommendation strategies that take into account demand saturation, and show that the problem is NP-complete and propose tractable heuristics.

2.4. Random Utility Models

Discrete Choice Theory is a classical branch of Economics, and Random Utility Models are one of the main tools of the discipline. Discrete Choice theory has its origins in the 60s with work such as [28]. Its use to answer marketing questions was pioneered by Daniel McFadden in [29]. He won a Nobel Prize for his work on Random Utility Models (RUM), and more precisely on Conditional Logit models [30]. Random Utility models (RUM) have been introduced in the 1970s, a time where the availability of user consumer data was scarce and the observability of the salient user and product features was reduced. However, in modern times, a lot of the user shopping activity moved online, therefore tremendously increasing the volume of available user shopping data and its granularity. Furthermore, it can be argued that in many cases, the modeller has access to a full view of the product information available to the consumer at the moment of the decision. That means that at learning time, we have total access to the full feature set used in the decision-making process, albeit in a raw form. For this reason, recent work has explored the possibility of applying RUM to consumer basket modelling and dynamic pricing (SHOPPER) [31]. However, though the authors drew parallels to existing recommendation models, the model was not used and benchmarked for recommendation.

3. Welfare and Recommendation

There are many ways to frame the problem of designing a Recommender System algorithm as a machine learning problem. For some, it is the question of guessing the most probable next item, or of matrix completion, for some others, it is an exploration-exploitation problem (bandit setting). A more recent line of approaches frames the problem of designing a Recommender System algorithm in terms of economic value maximization [19]. Our claim is that in the presence of sales data, recommendation algorithms can use the price information to directly optimize the welfare of the whole system (which is made of the advertisers and the users), instead of maximizing the probability of an action (click or conversion for instance). We use the simple but powerful idea that users convert when the price is below a stochastic threshold value, the Willingness-To-Pay (WTP) [1], and leverage a mature recommendation algorithm –Matrix Factorization (MF)– to learn the user decision process and the underlying WTP.

We concur with authors in [32] who wrote that “recommender systems serve multiple goals and that the purely user-centered approach found in most academic research does not allow all such goals

to enter into their design and evaluation”. In the case of online shopping, if the Recommender System is a user assistant, one has to raise the concern of contradictory objectives because on the one hand, the sellers might want to maximize their revenues or profit, and, on the other hand, the user is looking to maximize their utility.

The difference between what the user actually pays and the WTP can be seen as a surplus/utility that the user wants to maximize. On the seller side, we suppose the seller to be a revenue maximizer – we could equivalently suppose a profit maximizing criterion by taking into account the seller’s procurement cost. *We propose to maximize the total welfare of the system buyer+seller, defined as the sum of the utility of the buyer and the seller.* If, when a sale occurs, the buyer’s utility is their surplus (WTP - price) and the seller utility is the revenue generated by the sale (the price), then the sum of the two results in the system maximizing the expected WTP.

4. A shopper’s behavioural model

4.1. User Decision Model

In this section, we present a simple model for the user behaviour. Such model can be understood by describing what happens between the moment the user reaches the e-tailer page, to the moment the user leaves or decides to buy something. In this description, we suppose that the user is looking for one specific category of product (for example, a pair of shoes or a mobile phone). By design, our model does not encompass the purchase of several goods, hence while not adapted for grocery shopping, we believe this encompasses a lot of the situations where Recommender Systems play a key role as shopping assistants.

Our main assumption is that while the user browses the e-tailer website – for example, looking for a new mobile phone –, they *learn* about the different alternatives proposed by the website. More precisely, *we suppose that we can infer the awareness set over which the user decides to purchase something, or simply leave.* For instance, in order to build the awareness set, one can take the browsing history and add the most popular items in the category, as well as the previous purchases.

By doing so, and assuming the user is rational and risk-neutral, we can define its decision process as choosing the item j^* that maximizes their utility over the associated awareness set at time t $D_i(t)$, to which we add the **no buy** option (by definition, the no buy option brings 0 utility). More formally, we have that:

$$j^*(t) \in \operatorname{argmax}_{j \in D_i(t)} u_{i,j}(t), \quad (1)$$

where $u_{i,j}(t)$ is the total utility of user i for item j :

$$u_{i,j}(t) = w_{i,j} - p_j(t), \quad (2)$$

computed as the surplus between the Willingness-To-Pay (WTP) $w_{i,j}$ and the price $p_j(t)$ of item j at time t . In our case, time t is the index of the browsing session, to support the case where the user has performed multiple shopping trips.

4.2. Random Utility Maximization/ Matrix Factorization (RUM-MF)

Next, we introduce two additional assumptions that allow us to link the utility maximization to Matrix Factorization.

4.2.1. Gumbel noise

We extend the Equation (2) and connect our model with Random Utility Models: we suppose that there exists noise $\epsilon_{i,j}(t)$ on the latent variable $u_{i,j}$, such that at each decision time, the shopper makes their decision based on $\hat{u}_{i,j}(t) = u_{i,j} + \kappa_i \epsilon_{i,j}(t)$, where κ_i is a positive, scalar model parameter. With this change, the shopper’s decision – knowing i and p –, denoted $x_i(t)$ is now random. Also, we suppose

the noise samples $\epsilon_{i,j}(t)$ to be independent and follow a standard Gumbel distribution, which leads us to the classical softmax decision rule, for $j \in D_i(t)$:

$$\Pr(x_i(t) = j) = \frac{\exp(u_{i,j}(t)/\kappa_i)}{\sum_{j' \in D_i(t)} \exp(u_{i,j'}(t)/\kappa_i)}. \quad (3)$$

4.2.2. Factorizable form

We suppose there exists a dimension d , and a d -dimensional vectorial representation of the users X_i and of the items Y_j such that $w_{i,j} = X_i \cdot Y_j$, where \cdot is the dot product. Such representation is very close to the Matrix Factorization, and can benefit from pre-existing user and/or item embeddings, that can be built for example using clicks, number of co-occurrences, user segmentation or catalog input. With these two assumptions, the decision model becomes

$$\Pr(x_i(t) = j) = \frac{\exp((X_i \cdot Y_j - p_j(t))/\kappa_i)}{\sum_{j' \in D_i(t)} \exp((X_i \cdot Y_{j'} - p_{j'}(t))/\kappa_i)}. \quad (4)$$

It is classical but notable that in our formulation, we get a relation between the utility, and the odds ratio between two alternatives:

$$\log \frac{\Pr(x_i(t) = j | (j, j') \in D_i(t))}{\Pr(x_i(t) = j' | (j, j') \in D_i(t))} = \frac{u_{i,j} - u_{i,j'}}{\kappa_i}. \quad (5)$$

The interpretation of Equation 5 is that the odd ratio between two alternatives is fully explained by the difference in utility the two alternatives provide to the user. So if κ_i is known, we can relate the WTP, the prices, and the odd ratio between two alternatives.

5. Experiments

While many RecSys performance studies rely on datasets with logged partial feedback to back their claim, such methodology is not adapted to the problem at hand, since we cannot have access to the true WTP in real life datasets. For this reason, we build our own simulated environment in which we observe the behavior of our proposed method against existing state-of-the-art. For more discussion on the use of simulators in the analysis of Recommender Systems, see [33]. For space consideration, we only report here the main insights. The experiments detail as well as the tables of results can be found in the long version of this article as well as in the appendix, that contains an extended version of this section.

Our data generator takes as input the number of users and items, the number of sessions per user, the number of items seen in each session and the dimension d of the latent space. It then generates two vectors from a d -dimensional Gaussian distribution. Those two vectors are then used as means for generating the users and items d -dimensional representations (by sampling from a Gaussian distribution). Each item's price is set to be revenue maximizing plus uniform noise. Once the items and the users are generated, we generate user sessions by exposing users to random sub-samples of items from the catalog and at the end of each session we run the utility computation to create the session outcome. We implemented our models in PyTorch and used Adam [34] to optimize over a L_2 -regularized cross-entropy loss.

To compute our metrics, we assume that our recommendations will be shown in slates of various sizes. We use a greedy approach to fill the banner using the top k products ranked by their expected Value per Sale (eVPS): $eVPS_{j|i} = \Pr(\hat{u}_{i,j} > 0) \times Value_j$ where $Value_j$ is the value of the sale, which can be one of the four: volume (value equal to 1), user's utility, seller's revenue and overall welfare. While the greedy slate-filling approach is not optimal, it is by far the most popular approach and it has been shown in [35] to be a good approximation of the global maximum. To note, because of the greedy strategy, metrics that are not aligned with the user's utility, such as the seller's revenue, might not be monotonous in k .

We tested the following methods.

- *Oracle* Represents the best k products for each user, according to the true item features and true user preferences. We run experiments where we rank the products by their user’s utility (*Oracle-utility*) or by the system welfare (*Oracle-welfare*).
- *BestOf* Returns the k most popular products in the training data, sorted by number of sales.
- *RUM-MF* Returns the k most likely to be bought products according to the proposed model.
- *MF-SM(Softmax)* This is the same method as *RUM-MF*, but without using the price in the model training. As a note, we predict the WTP w_i by considering that the innerproduct between the user and item vectors is still approximating the utility u_i for the pair and adding the item price to it: $\Pr(x_i(t) = j)_{MF-SM} = \frac{\exp(X_i \cdot Y_j)}{\sum_{j' \in D_i(t)} \exp(X_i \cdot Y_{j'})}$.
- *MF-PCLICK* Treats every conversion as an independent outcome, similar to Factorization Machine methods used for Click-through Rate Prediction [7]. Since the method is not recovering the willingness-to-pay, only two objectives are available for the slate generation, namely volume of sales and seller’s revenue: $\Pr(x_i(t) = j)_{MF-PCLICK} = \sigma(X_i \cdot Y_j)$.

5.1. Findings

Optimizing for welfare leads to the best welfare output. As expected, optimizing for welfare works and it leads to the best welfare metric, *RUM-MF-welfare*, surpassing by more than 30% the performance matrix factorization methods (see long version for details).

We also observe that RUM-MF outperforms the other methods not only in Welfare metrics but also in classical Precision and Sales metrics, albeit by a smaller margin. The performance is maintained for slates with $k > 1$

Switching to our formulation can lead to improvements even in the absence of prices. We observe that while not as competitive as *RUM-MF*, *MF-SM* outperforms clearly the *MF-PCLICK* approach. Indeed, even in the absence of prices, the categorical formulation of the problem and the presence of the no-buy option in the model leads to a better estimation of the user’s utility vector.

6. Conclusion

We believe that as new algorithms tend to leverage the price information, welfare maximizing Recommender Systems are likely to become a reality in the near future. Because they will explicitly model the user’s behaviour, this new class of RS will allow for a better understanding of the user interaction with the RS. The shift from click maximizer to welfare maximizer will permit a better control of the value sharing between the different stakeholders. Also, we implicitly relied on the classical Luce axiom of independence from irrelevant alternative, which might not be always satisfied. This is, however, a general problem in discrete choice theory that is not specific to our approach, so that one could investigate if the current approach could borrow, for instance, from nested decision models. Our goal was to provide a minimal working algorithm, but in practice such algorithm could be adapted to the available information. For example, if we have access to embedded representations of the user and/or the items (based on clicks or sales or browsing behaviours...), then we can use these representations as inputs in a neural network.

References

- [1] L. J. Savage, Elicitation of personal probabilities and expectations, *Journal of the American Statistical Association* 66 (1971) 783–801.
- [2] H.-J. Xue, X. Dai, J. Zhang, S. Huang, J. Chen, Deep matrix factorization models for recommender systems., in: *IJCAI*, volume 17, Melbourne, Australia, 2017, pp. 3203–3209.
- [3] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, *Computer* 42 (2009) 30–37.

- [4] O. Chapelle, E. Manavoglu, R. Rosales, Simple and scalable response prediction for display advertising, *ACM Transactions on Intelligent Systems and Technology (TIST)* 5 (2014) 1–34.
- [5] K. S. Dave, V. Varma, Learning the click-through rate for rare/new ads from similar ads, in: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010, pp. 897–898.
- [6] S. Rendle, Factorization machines, in: *2010 IEEE International conference on data mining*, IEEE, 2010, pp. 995–1000.
- [7] Y. Juan, Y. Zhuang, W.-S. Chin, C.-J. Lin, Field-aware factorization machines for ctr prediction, in: *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 43–50.
- [8] A. Swaminathan, T. Joachims, Counterfactual risk minimization: Learning from logged bandit feedback, in: *International Conference on Machine Learning*, PMLR, 2015, pp. 814–823.
- [9] O. Sakhi, S. Bonner, D. Rohde, F. Vasile, Blob: A probabilistic model for recommendation that combines organic and bandit signals, in: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 783–793.
- [10] D. Liang, R. G. Krishnan, M. D. Hoffman, T. Jebara, Variational autoencoders for collaborative filtering, in: *Proceedings of the 2018 world wide web conference*, 2018, pp. 689–698.
- [11] T. Joachims, Optimizing search engines using clickthrough data, in: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 133–142.
- [12] X. Dong, J. Shen, Triplet loss in siamese network for object tracking, in: *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 459–474.
- [13] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, *arXiv preprint arXiv:1205.2618* (2012).
- [14] T. Joachims, A. Swaminathan, T. Schnabel, Unbiased learning-to-rank with biased feedback, in: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017, pp. 781–789.
- [15] M. Wan, D. Wang, M. Goldman, M. Taddy, J. Rao, J. Liu, D. Lymberopoulos, J. McAuley, Modeling consumer preferences and price sensitivities from large-scale grocery shopping transaction logs, in: *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1103–1112.
- [16] D. Hu, R. Louca, L. Hong, J. McAuley, Learning within-session budgets from browsing trajectories, in: *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18*, Association for Computing Machinery, New York, NY, USA, 2018, p. 432–436. URL: <https://doi.org/10.1145/3240323.3240401>. doi:10.1145/3240323.3240401.
- [17] Y. Zheng, C. Gao, X. He, Y. Li, D. Jin, Price-aware recommendation with graph convolutional networks, in: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, IEEE, 2020, pp. 133–144.
- [18] S. Wadhwa, A. Ranjan, S. Xu, J. H. Cho, S. Kumar, K. Achan, Personalizing item recommendation via price understanding, in: *ComplexRec-ImpactRS@ RecSys*, 2020.
- [19] Y. Zhang, Q. Zhao, Y. Zhang, D. Friedman, M. Zhang, Y. Liu, S. Ma, Economic recommendation with surplus maximization, in: *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2016, p. 73–83. URL: <https://doi.org/10.1145/2872427.2882973>. doi:10.1145/2872427.2882973.
- [20] J. S. Lleras, Y. Masatlioglu, D. Nakajima, E. Y. Ozbay, When more is less: Limited consideration, *Journal of Economic Theory* 170 (2017) 70–85.
- [21] Q. Zhao, Y. Zhang, D. Friedman, F. Tan, E-commerce recommendation with personalized promotion, in: *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15*, Association for Computing Machinery, New York, NY, USA, 2015, p. 219–226. URL: <https://doi.org/10.1145/2792838.2800178>. doi:10.1145/2792838.2800178.
- [22] Q. Zhao, Y. Zhang, Y. Zhang, D. Friedman, Multi-product utility maximization for economic recommendation, in: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17*, Association for Computing Machinery, New York, NY, USA, 2017, p. 435–443. URL: <https://doi.org/10.1145/3018661.3018674>. doi:10.1145/3018661.3018674.

- [23] Y. Ge, S. Xu, S. Liu, S. Geng, Z. Fu, Y. Zhang, Maximizing marginal utility per dollar for economic recommendation, in: The World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 2757–2763. URL: <https://doi.org/10.1145/3308558.3313725>. doi:10.1145/3308558.3313725.
- [24] Z. Xu, Y. Han, Y. Zhang, Q. Ai, E-commerce recommendation with weighted expected utility, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1695–1704. URL: <https://doi.org/10.1145/3340531.3411993>. doi:10.1145/3340531.3411993.
- [25] A. Das, C. Mathieu, D. Ricketts, Maximizing profit using recommender systems, arXiv preprint arXiv:0908.3633 (2009).
- [26] D. Jannach, G. Adomavicius, Price and Profit Awareness in Recommender Systems., arXiv: Information Retrieval (2017).
- [27] W. Lu, S. Chen, K. Li, L. V. S. Lakshmanan, Show me the money: dynamic recommendations for revenue maximization, Proc. VLDB Endow. 7 (2014) 1785–1796. URL: <https://doi.org/10.14778/2733085.2733086>. doi:10.14778/2733085.2733086.
- [28] K. J. Arrow, H. B. Chenery, B. S. Minhas, R. M. Solow, Capital-labor substitution and economic efficiency, The review of Economics and Statistics 43 (1961) 225–250.
- [29] D. McFadden, Econometric models of probabilistic choice, Structural analysis of discrete data with econometric applications 198272 (1981).
- [30] D. McFadden, et al., Conditional logit analysis of qualitative choice behavior (1973).
- [31] R. Donnelly, F. J. Ruiz, D. Blei, S. Athey, Counterfactual inference for consumer choice across many product categories, Quantitative Marketing and Economics (2021) 1–39.
- [32] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, L. Pizzato, Beyond personalization: Research directions in multistakeholder recommendation, arXiv preprint arXiv:1905.01986 (2019).
- [33] M. D. Ekstrand, A. Chaney, P. Castells, R. Burke, D. Rohde, M. Slokom, Simurec: Workshop on synthetic data and simulation methods for recommender systems research, in: Fifteenth ACM Conference on Recommender Systems, 2021, pp. 803–805.
- [34] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
- [35] M. Derakhshan, N. Golrezaei, V. Manshadi, V. Mirrokni, Product ranking on online platforms, Management Science (2022).

A. Experiments: extended version

While many RecSys performance studies rely on datasets with logged partial feedback to back their claim, such methodology is not adapted to the problem at hand, since we cannot have access to the true WTP in real life datasets. For this reason, we build our own simulated environment in which we observe the behavior of our proposed method against existing state-of-the-art. For more discussion on the use of simulators in the analysis of Recommender Systems, see [33].

A.1. The simulator

Our data generator takes as input the number of users and items, the number of sessions per user, the number of items seen in each session and the dimension d of the latent space. It then generates two vectors from a d -dimensional Gaussian distribution. Those two vectors are then used as means for generating the users and items d -dimensional representations (by sampling from a Gaussian distribution). Each item's price is set to be revenue maximizing plus uniform noise. Once the items and the users are generated, we generate user sessions by exposing users to random sub-samples of items from the catalog and at the end of each session we run the utility computation to create the session outcome.

	Medium1	Medium2	Hard
nb-sessions	3	15	3
nb-items-session	10	2	10
nb-users	1000	1000	1000
nb-prods	100	100	1000
dimension	10	10	10

Table 1

The three environment configurations

Simulation environments For our experiments, we define three types of configurations with various levels of difficulty for our simulator, as shown in Table 1. We classify the level of difficulty of the environment based on the user and item-level sparsity of the training data. In the medium configurations, each user sees 30% of the catalog and each item has associated around 300 events. In the hard case each user sees only 3% of the catalog and each item is visited on average only 30 times. Both the item and user vectors are of dimension 10 and the associated Gaussian distributions have their variance set to 3. The only other simulator parameter is the price noise which we kept fixed to the range 0-5.

A.2. Training

We implemented all our models in PyTorch and used Adam [34] to optimize over a L_2 -regularized cross-entropy loss. In the implementation of the *RUM-MF* model we opted for a slightly different parametrization than the one in Equation.4, which is still equivalent to it, but in which the parameter κ_i parameter can be seen as the price sensitivity of the user specific to the product category:

$$\Pr(x_i(t) = j)_{RUM-MF} = \frac{\exp(X_i \cdot Y_j - \kappa_i p_j(t))}{\sum_{j' \in D_i(t)} \exp(X_i \cdot Y_{j'} - \kappa_i p_{j'}(t))} \quad (6)$$

The Neural Network jointly learns the embedding and κ on sales signals by LLH maximization. As a result, we will estimate the WTP as the ratio of the dot product between the user and item vectors and the user sensitivity to price: $w_{i,j} = \frac{X_i \cdot Y_j}{\kappa_i}$. When estimating WTP, we cap all our learnt sensitivities to minimum 0.1.

A.3. Metrics

In order to compute our metrics, we assume that our recommendations will be shown in slates of various sizes. We use a greedy approach to fill the banner using the top k products ranked by their expected Value per Sale (eVPS), as defined below:

$$eVPS_{j|i} = \Pr(\hat{u}_{i,j} > 0) \times Value_j \quad (7)$$

where $Value_j$ is the value of the sale, which can be one of the four: volume (value equal to 1), user's utility, seller's revenue and overall welfare. While the greedy slate-filling approach is not optimal, it is by far the most popular approach and it has been shown in [35] to be a good approximation of the global maximum. To note, because of the greedy strategy, metrics that are not aligned with the user's utility, such as the seller's revenue, might not be monotonous in k .

To define our metrics, we introduce the following notation:

- j_i^* is the true optimal choice for the user i over the catalog, $j_i^* = \operatorname{argmax}_{j \in C} u_{i,j}$
- S_i^k is the set containing the top k choices sorted by their predicted utility for the user i , to which we add the **no buy** option. S_i^k is *practically an awareness set built by the RS*.
- j_{ik}^* is the optimal choice for the user i out of the set of items present in the slate S_i^k , $j_{ik}^* = \operatorname{argmax}_{j \in S_i^k} u_{i,j}$

We divide our metrics into classic metrics, such as Precision@k and Sales@k – respectively $|j_i^* \in \hat{S}_i^k|/|I|$ and $|u_{i,j_{ik}^*} > 0|/|I|$ – and novel metrics, such as BuyerUtility@k, SellerRevenue@k and Welfare@k ($\sum_{i \in I} u_{i,j_{ik}^*}/|I|$, $\sum_{i \in I} p_{j_{ik}^*}/|I|$ and $\sum_{i \in I} (u_{i,j_{ik}^*} + p_{j_{ik}^*})/|I|$). For the classical metrics, while both of them are well-known, computing probability of sale at various slate sizes is novel, and only possible due to our access to an oracle.

A.4. Methods

Below we present the set of methods that we take into consideration in our experiments:

Oracle Represents the best k products for each user, according to the true item features and true user preferences. We run experiments where we rank the products by their user’s utility (*Oracle-utility*) or by the system welfare (*Oracle-welfare*).

BestOf Returns the k most popular products in the training data, sorted by number of sales.

RUM-MF Returns the k most likely to be bought products according to the proposed model.

MF-SM(Softmax) This is the same method as *RUM-MF*, but without using the price in the model training. As a note, we predict the WTP w_i by considering that the innerproduct between the user and item vectors is still approximating the utility u_i for the pair and adding the item price to it:
$$\Pr(x_i(t) = j)_{MF-SM} = \frac{\exp(X_i \cdot Y_j)}{\sum_{j' \in D_i(t)} \exp(X_i \cdot Y_{j'})}$$

MF-PCLICK Treats every conversion as an independent outcome, similar to Factorization Machine methods used for Click-through Rate Prediction [7]. Since the method is not recovering the willingness-to-pay, only two objectives are available for the slate generation, namely volume of sales and seller’s revenue: $\Pr(x_i(t) = j)_{MF-PCLICK} = \sigma(X_i \cdot Y_j)$

A.5. Results

Below we summarize our main findings:

Optimizing for welfare leads to the best welfare output On the first axis, we wanted to understand the impact of different choices of objective (and the associated value per sale – VPS) on the 5 metrics, as shown in Table 4. The first observation is that, as expected, optimizing for welfare works and it leads to the best welfare metric, *RUM-MF-welfare*, surpassing by more than 30% the performance of the *MF-SM* method and almost doubling the performance of *MF-PCLICK*. This observation is confirmed by Table 5, that shows that the magnitude of the improvement in performance is maintained over all configurations and runs.

RUM-MF outperforms the baselines in all other metrics In the same Table 4, we can observe that RUM-MF outperforms the other methods not only in Welfare metrics but also in classical Precision and Sales metrics, albeit by a smaller margin.

The performance is maintained for slates with k>1 In Table 6. we confirm that the results hold also for banners/slates with k bigger than 1. This is important since in real-world recommendations, the performance is judged not over single item recommendations but over recommendation sets. We observe that, as expected, the gap in performance becomes smaller as k becomes bigger and all methods start retrieving good items in their top k set.

Algo	Objective	Welfare@k	Utility@k	Revenue@k	Sales@k	Precision@k
oracle	welfare	21.51	11.31	10.19	0.89	0.41
oracle	utility	20.58	14.32	6.25	1.00	1.00
bestof	sales	8.39	5.27	3.12	0.57	0.11
rum-mf	welfare	11.74+/-0.26	6.78+/-0.21	4.96+/-0.11	0.78+/-0.01	0.14+/-0.01
rum-mf	utility	10.67+/-0.3	6.59+/-0.21	4.08+/-0.12	0.75+/-0.02	0.12+/-0.01
rum-mf	revenue	11.43+/-0.36	4.37+/-0.18	7.06+/-0.22	0.64+/-0.02	0.08+/-0.01
rum-mf	sales	10.82+/-0.3	6.68+/-0.21	4.14+/-0.11	0.78+/-0.01	0.12+/-0.01
mf-sm	welfare	8.59+/-0.25	5.97+/-0.21	2.62+/-0.06	0.75+/-0.01	0.1+/-0.01
mf-sm	utility	8.16+/-0.21	5.73+/-0.18	2.42+/-0.05	0.73+/-0.01	0.09+/-0.01
mf-sm	revenue	8.93+/-0.24	6.08+/-0.19	2.85+/-0.07	0.75+/-0.01	0.11+/-0.01
mf-sm	sales	7.53+/-0.22	5.51+/-0.18	2.01+/-0.05	0.73+/-0.01	0.08+/-0.01
mf-pclick	revenue	7.03+/-0.21	5.13+/-0.18	1.9+/-0.05	0.69+/-0.01	0.07+/-0.01
mf-pclick	sales	6.42+/-0.2	4.88+/-0.18	1.54+/-0.04	0.69+/-0.02	0.06+/-0.01

Table 2

Performance of various models on top product recommendation (k=1) (Medium2, run2)

	Medium1		Medium2		Hard1	
	run1	run2	run1	run2	run1	run2
oracle	20.10	23.38	19.32	21.51	27.81	28.50
bestof	5.75	9.06	5.30	8.39	5.83	5.80
mf-rum	11.61	12.89	7.88	11.74	7.06	7.43
mf-sm	8.85	8.94	7.31	8.93	4.2	5.06
mf-pclick	7.26	9.06	5.23	7.03	5.17	5.61

Table 3

Welfare@1 metric over multiple environments

	Welfare@1	Welfare@5	Welfare@10
oracle	21.51	20.84	20.62
bestof	8.39	11.48	14.9
rum-mf	11.74	16.92	18.27
mf-sm	8.93	14.51	16.56
mf-pclick	7.03	12.7	14.95

Table 4

Welfare metric over multiple k (Medium2, run2)

Switching to our formulation can lead to improvements even in the absence of prices. We observe both in Table 4. and 5. that while not as competitive as *RUM-MF*, *MF-SM* outperforms clearly the *MF-PCLICK* approach. Indeed, even in the absence of prices, the categorical formulation of the problem and the presence of the no-buy option in the model leads to a better estimation of the user’s utility vector.