Adaptive Preference Aggregation for Recommender Systems

Benjamin Heymann
Criteo AI Lab, Fairplay joint team
Paris, France
b.heymann@criteo.com

ABSTRACT

Social choice theory provides a framework to aggregate preferences, but was not developed for the multidimensional applications typical of recommender systems. Leveraging insights from a recently published urn process, this work introduces a preference aggregation strategy that adapts to the user's context and inherits the good properties of the maximal lottery, a Condorcet-consistent solution concept.

CCS CONCEPTS

• **Information systems** → **Information retrieval**.

KEYWORDS

Maximal Lottery, LLM, RLHF, RecSys, Preference Aggregation

ACM Reference Format:
Benjamin Heymann. 2025. Adaptive Preference Aggregation for Recommender Systems. In *Proceedings of Recsys '25: CONSEQUENCES Workshop.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

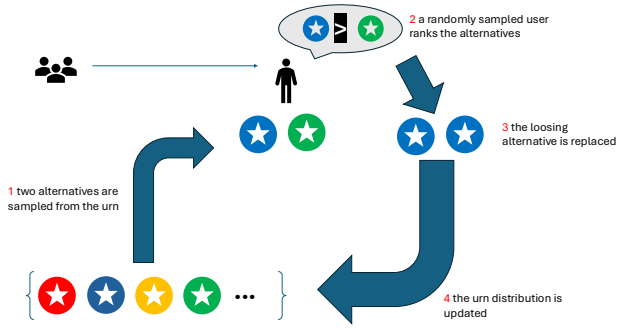

Figure 1: Graphical representation of the urn process introduced by [4], for simplicity we omit the mutation rate in this representation. Iteratively, (1) two alternatives are sampled from the urn, then (2) a randomly sampled user expresses their preference for the two options, (3) the ball of the least preferred option is then replaced in the urn by a ball for the preferred option, which (4) changes the states of the urn.

1 INTRODUCTION

An important solution concept in social choice theory is that of the Condorcet winner [6]. A Condorcet winner is a candidate in an election who wins a majority of votes in a head-to-head comparison against each of the other candidates. A voting method is said to be *Condorcet consistent* if it always selects the Condorcet winner whenever one exists. Similar notions have been rediscovered in different contexts by different communities (because they hide the ubiquituous concept of zero-sum game Nash equilibrium); for example, people interested in dueling bandits might have heard of the von Neumann winner [9], while others might have heard of the randomized Condorcet winner.

In [4], the authors propose a very simple urn process that converges to a *maximal lottery*, a solution concept known as Condorcet consistent. This process involves a single urn containing balls, where each ball represents an alternative — or, equivalently, a political candidate. Initially, the urn is filled with multiple balls for each alternative. The procedure consists of repeatedly selecting two alternatives at random from the urn and having a randomly chosen user —or voter, in the context of [4] — express their preference between them. The ball representing the losing alternative is then replaced in the urn by another ball representing the winning alternative. The general idea is illustrated in Figure 1. Under some technical assumptions, (1) the distribution of winning balls converges to the maximal lottery and (2) the proportions in the urn get closer and closer to the maximal lottery.

We use the main idea of [4] to introduce a novel algorithm for recommender systems. Our innovation was prompted by the parallels between this urn and balls mechanism and the iterative feedback mechanisms used to fine-tune large language models (RLHF). Similar pick-your-preferred-option mechanism can also be observed in industrial recommender systems (e.g., music recommendation) to warm start the system to a given user taste. We show that using some function approximation tricks [11, 8], the urn process idea can be ported to the world of generative AI and recommender systems.

Our goal in this research is to show that processes similar to [4] can be used in an online context to identify maximal lotteries. Because recommender systems [3] and LLMs share many similarities, the general principles of this study can be applied to both domains. In fact, the formalization we propose in Section 2.1 does not distinguish between the two. A longer version of this extended abstract, more tailored to an LLM focused readership, is available online [12]. It contains other experiments and pseudo-code for the algorithms.

Recsys '25: CONSEQUENCES Workshop, September, 2025, Prague

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 PREFERENCE AGGREGATION AT THE USER LEVEL

2.1 Mathematical model

We consider a finite set of alternatives $\mathcal{A}$ over which users from a set $\mathcal{U}$ have preferences. There exists a probability distribution $\mathbb{P}$ over the set of users $\mathcal{U}$, so that $(\mathcal{U}, \mathbb{P})$ is a probability space. The preference of each user $u \in \mathcal{U}$ is encoded with a partial order over $\mathcal{A}$ which we denote by $\preceq_u$: for a_1 and a_2 in $\mathcal{A}$, $a_1 \preceq_u a_2$ simply means that user u prefers alternative a_2 over a_1.

Maintaining a comprehensive representation of each user's preference relation is infeasible (we do not have access to everything, and we need to encode it on a computer), which motivates mapping users into an embedding space $\mathcal{E}$ using the function $\phi : \mathcal{U} \to \mathcal{E}$. When we are given the task of selecting an alternative in $\mathcal{A}$ for a user u_0, who we only know by their embedding representation $\phi(u_0)$, we would ideally select a maximal element of $(\mathcal{A}, \preceq_{u_0})$. However, (1) we do not know u_0, and (2) a maximal element in $(\mathcal{A}, \preceq_{\phi(u_0)})$ does not make any sense (yet). This is where preference aggregation kicks in; see Section 2.2. The representation of u, $\phi(u)$, may contain features known about the user, past interactions, demographic data, information about their preference... we abstract away the specifics here to focus on how the urn process can be ported to this general setting.

We have highlighted an important modeling nuance. In discrete choice theory, an agent's decision is typically modeled stochastically. Randomness is often attributed to exogenous, unobservable random shocks that influence the decision-making process. In contrast, we assume that each agent has a deterministic preference relation. The observed stochasticity arises not from inherent randomness in the agent's decision-making but from the inability to distinguish between agents with distinct preferences using the available data. This perspective shifts the source of uncertainty from exogenous shocks to the limitations of observational data. It also clarifies how preference data can be interpreted as nontransitive. Note that the two sources of randomness are not incompatible, and that we could also account for the agent's stochastic decision-making with our formalism.

2.2 Learning objective

Consider two possible alternative answers to a queries, A and B. It is natural to say that A is better than B if the majority of users prefer A over B. Extending this idea, a "maximal" alternative is one that is preferred by a majority of users over *any* other possible alternatives. This maximal alternative is commonly referred to as the **Condorcet winner**, or when we allow randomization over alternatives, as the **maximal lottery**. Maximal lotteries are known to have several interesting properties that are discussed in several studies [10, 4, 5]. To be more formal: given some users $(u_1, \dots u_n) \in \mathcal{U}$ sampled from $\mathbb{P}$, we consider the task of identifying, using $(u_1, \dots u_n)$, a policy π from $\mathcal{E}$ to $\mathcal{A}$ that maximizes

$$\min_{\pi' : \mathcal{E} \to \mathcal{A}} \mathbb{P}\left(\pi(\phi(u)) \succcurlyeq_u \pi'(\phi(u))\right). \tag{1}$$

This maxmin formulation is reminiscent of the game formulations discussed in [18, 21, 17].

To make this formulation more vivid, in the context of the famous chatbot arena [7] (for LLM), π would be our model, π' would be any competitor's model, and our goal in expression (1) corresponds to maximizing our score against our strongest contender. It is important to note here that the notion of the strongest contender is a priori specific to *our* model, as preference relations can be intransitive, and we might have in practice a Rock-Paper-Scissor like contest; this perspective connects with the literature on evaluating agents [19, 1, 15]. A pure solution to this problem is called the Condorcet winner, but it may not exist. However, its existence is guaranteed when we allow π to be stochastic. In this case, a solution that maximizes (1) is called a randomized Condorcet winner or a maximal lottery.

2.3 Scoring methods for preference aggregation

A popular recommender system technique called Bayesian Personalized Ranking [20] (BPR) shares some similarities with RLHF. We reinterpret the justification of BPR within our setting, where uncertainty arises not from the user's decision making but from the user's embedding representation. This reinterpretation leads to a presentation that differs from the original work. BPR uses implicit rankings derived from dataset interactions to predict the rankings of new alternatives for a specific user. In [20], implicit preferences refer to the relative rankings inferred from user behavior. For example, in a recommender system, a clicked item is assumed to be preferred over an unclicked item. In our setting, feedback is explicitly provided, but the structure of BPR remains relevant owing to its connection with the Elo scoring system.

The central structuring assumption in BPR, relevant to our analysis, is that for any user $u \in \mathcal{U}$ and two alternatives a_1 and a_2, the preference probability is given by

$$\mathbb{P}(a_1 \preceq_u a_2 \mid \phi(u)) = \sigma(x_{\phi(u),a_1} - x_{\phi(u),a_2}), \tag{2}$$

where σ denotes the sigmoid function. The method assumes a hypothesis class Θ for $x_{u,a}$, expressed as $x_{u,a} = f(\theta, \phi(u), a)$ with $\theta \in \Theta$. The solution involves maximizing the likelihood augmented by a Gaussian prior to enable quadratic regularization. **In the context of Recommender Systems, APA can be seen as an alternative to BPR that will handle in a principle manner nontransitivity of the preference data.**

3 ADAPTIVE PREFERENCE AGGREGATION

The Adaptive Preference Aggregation algorithm (APA) learns a mapping from a fixed user embedding $\phi(u)$ to a probability distribution over a finite set of alternatives $\mathcal{A}$, aiming to approximate a maximal lottery for the population of users in the atom $\phi(u)$.

Operating in an online fashion, it iteratively refines a neural network that emulates an urn. At each step, a user $u \in \mathcal{U}$ is sampled according to the probability $\mathbb{P}$, followed by sampling two alternatives (a_1, a_2) with a probability proportional to their presence in the neural urn $f_\theta(\phi(u))$. The user preference between a_1 and a_2 is observed, and the neural network weights θ are updated to minimize the distance between the current output of the urn and the target determined by the urn next state of the urn process. The algorithm maintains the weights of the neural network, which defines the

probability distributions over alternatives for each user embedding, approximating the maximal lottery.

3.1 Function approximation and neural urn

We simply use a multilayer perceptron

$$f_\theta : y \to n \in \mathbb{R}_+^{|\mathcal{A}|}, \tag{3}$$

where n represents the urn state, that is, the number of balls N_a for each alternative $a \in \mathcal{A}$ given a user of embedding $y = \phi(u)$, and θ is the network parameter. We used ReLu for the last layer. [1] The update should be

$$\min_\theta ||f_\theta(\phi(u)) - n_{new}||^2, \tag{4}$$

where $n_{new} = n_{\text{old}} + (e_i - e_j)$, with e_i and e_j being the one hot encoded vector that represent the alternatives i and j that were sampled for user u, and n_{old} is the state of the neural urn when the user is sampled : $f_\theta(\phi(u))$. As explained in [4], we added a small mutation rate. This means that with a small probability, we set

$$n_{new} = n_{\theta_{\text{old}}} + (e_i - e_j), \tag{5}$$

where e_j is taken at random from the urn proportionally to N and e_i is sampled uniformly from $\mathcal{A}$. For the experiments, we used two hidden layers with 32 activation units. To initialize the urn for a given N, we did some learning iterations by assigning random values to N_{new} of amplitude proportional to N

4 EXPERIMENTS

Other experiments can be found in this version of the paper [12]. We took $\mathcal{U} = \mathbb{R}^3$. We also embed the element of $\mathcal{A}$ in $\mathbb{R}^3$ and use the preference rule

$$a_1 \succcurlyeq_u a_2 \iff ||a_1 - u|| \leqslant ||a_2 - u||. \tag{6}$$

The implementation of the experiment will be provided in the supplementary material, and was performed in Julia [2], with the Flux [13, 14] learning library. [2] The setup is described in Figure 2 and the results are displayed in Figure 3. This simple experiment illustrate how the urn process from [4] can be adapted to recommender systems using function approximation. It also provide an example of one of the limit of reward based methods such as BPR.

5 DISCUSSION

In this work, we clarify a key challenge faced when designing a strategy to aggregate preferences while training systems such as LLM or recommender systems: the uncertainty surrounding users due to unrevealed information and their encoding within the system. This uncertainty implies that the data may not be sufficient to determine the optimal response to a query. Moreover, the very notion of what is "optimal" becomes unclear. We propose that one should aim to achieve a sense of Condorcet consistency. We combined [4] with function approximation to develop an APA algorithm that addresses this gap. Using a nontrivial and visual toy example, we demonstrate that our system can learn the maximal lottery. Further research is needed to understand the conditions under which these observations hold. Specifically, because the urn process, which inspired our approach, was not originally designed for this type of application, there may be potential improvements in sample complexity and computational efficiency. From an applied perspective, testing this idea at scale would be an ambitious and valuable follow-up study. Additionally, from a broader viewpoint, the field needs to clarify which properties of social choice theory such systems should possess. Collaborative filtering, a fundamental concept in recommender systems, often relies on the idea that understanding one aspect of a user's preferences can provide insights into other aspects. We did not fully leverage this perspective in our study, which is a limitation of the present work. From this research, there is a clear need to clarify what generalization means in this context and to investigate statistical properties such as convergence of estimators for the lottery.

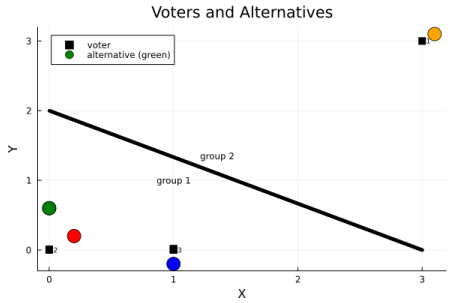

Figure 2: Visualization of the environment. Black squares represent voters, the number on their right their relative mass. The system can only distinguish between voters from group 1 and voters from group 2. The situation in group 1 maps us to a typical majority paradox from social choice theory [17] where reward models will not select an alternative prefered by the majority.

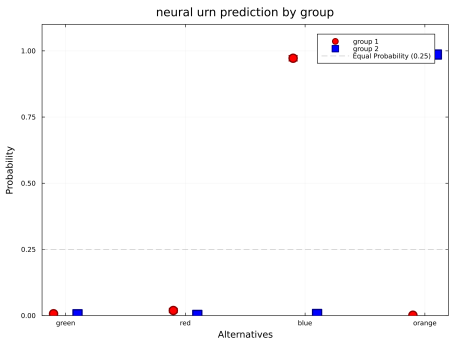

Figure 3: prediction of the neural urn on the two groups after training. One can check that the theoretical maximal lotteries are recovered.

[1] alternatively we could have $f_\theta : y \to g$ where g is a generated alternative to be closer to the LLM setting, or add a softmax layer to the perceptron, to be closer to the standard approaches for modeling probability on a discrete set.

[2] In this version [12], we used JuMP [16] and GLPK to compute the baselines (local or global maximal lotteries). For practical reasons, we use a discretization of $\mathbb{R}^2$ to define user embedding. **Indeed, this allows us to compare the output of APA with the direct computation of the maximal lotteries on the atoms of the grid using an LP solver.**

ACKNOWLEDGMENTS

The author would like to warmly thank Marc Lanctot for the many insightful conversations around this project.

REFERENCES

[1] David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. 2018. Re-evaluating Evaluation. (Oct. 2018).

[2] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. Julia: a fresh approach to numerical computing. *SIAM Review*, 59, 1, 65–98. DOI: 10.1137/141000671.

[3] Craig Boutilier, Martin Mladenov, and Guy Tennenholtz. 2023. Modeling Recommender Ecosystems: Research Challenges at the Intersection of Mechanism Design, Reinforcement Learning and Generative Models. (Sept. 2023). DOI: 10.48550/arXiv.2309.06375.

[4] Florian Brandl and Felix Brandt. 2024. A natural adaptive process for collective decision-making. *Theoretical Economics*, 19, 2, 667–703. DOI: 10.3982/TE5380.

[5] Felix Brandt. 2025. Stochastic choice and dynamics based on pairwise comparisons. In *One Hundred Years of Game Theory: A Nobel Symposium*. Econometric Society Monographs, (Ed.) Cambridge University Press.

[6] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. 2016. *Handbook of computational social choice*. Cambridge University Press.

[7] Wei-Lin Chiang et al. 2024. Chatbot arena: an open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

[8] Ryan D'Orazio, Danilo Vucetic, Zichu Liu, Junhyung Lyle Kim, Ioannis Mitliagkas, and Gauthier Gidel. 2024. Solving hidden monotone variational inequalities with surrogate losses. *arXiv preprint arXiv:2411.05228*.

[9] Miroslav Dudík, Katja Hofmann, Robert E. Schapire, Aleksandrs Slivkins, and Masrour Zoghi. 2015. Contextual dueling bandits. In *Proceedings of The 28th Conference on Learning Theory* (Proceedings of Machine Learning Research). Peter Grünwald, Elad Hazan, and Satyen Kale, (Eds.) Vol. 40. PMLR, Paris, France, (Mar. 2015), 563–587. https://proceedings.mlr.press/v40/Dudik15.html.

[10] Peter C Fishburn. 1984. Probabilistic social choice based on simple voting comparisons. *The Review of Economic Studies*, 51, 4, 683–692.

[11] Daniel Hennes et al. 2020. Neural Replicator Dynamics: Multiagent Learning via Hedging Policy Gradients. *New Zealand*.

[12] Benjamin Heymann. 2025. Adaptive preference aggregation. In *Games, Agents, and Incentives Workshop, AAMAS 2025*.

[13] Michael Innes, Elliot Saba, Keno Fischer, Dhairya Gandhi, Marco Concetto Rudilosso, Neethu Mariya Joy, Tejan Karmali, Avik Pal, and Viral Shah. 2018. Fashionable modelling with flux. *CoRR*, abs/1811.01457. https://arxiv.org/abs/1811.01457.

[14] Mike Innes. 2018. Flux: elegant machine learning with julia. *Journal of Open Source Software*. DOI: 10.21105/joss.00602.

[15] Marc Lanctot, Kate Larson, Yoram Bachrach, Luke Marris, Zun Li, Avishkar Bhoopchand, Thomas Anthony, Brian Tanner, and Anna Koop. 2023. Evaluating Agents using Social Choice Theory. (Dec. 2023).

[16] Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. 2023. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*. DOI: 10.1007/s12532-023-00239-3.

[17] Roberto-Rafael Maura-Rivero, Marc Lanctot, Francesco Visin, and Kate Larson. 2025. Jackpot! alignment as a maximal lottery. *arXiv preprint arXiv:2501.19266*.

[18] Rémi Munos et al. 2024. Nash learning from human feedback. (2024). https://arxiv.org/abs/2312.00886 arXiv: 2312.00886 [stat.ML].

[19] Shayegan Omidshafiei et al. 2019. α-rank: multi-agent evaluation by evolution. *Scientific reports*, 9, 1, 9937.

[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. [n. d.] BPR: Bayesian Personalized Ranking from Implicit Feedback.

[21] Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. 2024. A Minimaximalist Approach to Reinforcement Learning from Human Feedback. (June 2024).