PASTO: STRATEGIC PARAMETER OPTIMIZATION IN RECOMMENDATION SYSTEMS – PROBABILISTIC IS BETTER THAN DETERMINISTIC

Weicong Ding[*1], Hanlin Tang[†2], JingShuo Feng[‡3], Lei Yuan[1], Sen Yang[1], Guangxu Yang[1], Jie Zheng[1], Jing Wang[1], Qiang Su[1], Dong Zheng[1], Xuezhong Qiu[1], Yongqi Liu[1], Yuxuan Chen[1], Yang Liu[1], Chao Song[1], Dongying Kong[1], Kai Ren[1], Peng Jiang[1], Qiao Lian[1], and Ji Liu[§]

[1]Kuaishou Technology
[2]University of Rochester
[3]University of Washington

August 23, 2021

ABSTRACT

Real-world recommendation systems often consist of two phases. In the first phase, multiple predictive models produce the probability of different immediate user actions. In the second phase, these predictions are aggregated according to a set of '*strategic parameters*' to meet a diverse set of business goals, such as longer user engagement, higher revenue potential, or more community/network interactions. In addition to building accurate predictive models, it is also crucial to optimize this set of '*strategic parameters*' so that primary goals are optimized while secondary guardrails are not hurt. In this setting with multiple and constrained goals, this paper discovers that *a probabilistic strategic parameter regime can achieve better value compared to the standard regime of finding a single deterministic parameter*. The new probabilistic regime is to learn the best distribution over strategic parameter choices and sample one strategic parameter from the distribution when each user visits the platform. To pursue the optimal probabilistic solution, we formulate the problem into a stochastic compositional optimization problem, in which the unbiased stochastic gradient is unavailable. Our approach is applied in a popular social network platform with hundreds of millions of daily users and achieves $+0.22\%$ lift of user engagement in a recommendation task and $+1.7\%$ lift in revenue in an advertising optimization scenario comparing to using the best deterministic parameter strategy.

1 Introduction

Consider the pipeline to generate a list of personalized contents (e.g., videos, ads) to users in a real (probably oversimplified) recommendation system. Such pipelines typically include two phases: 1) predicting the probability of user's immediate actions and 2) calculating a rank score for final recommendation, as illustrated in Figure 1. In the first phase, the goal is to accurately predict various immediate user actions such as clicking on an ad or liking a recommended content. The prediction functions to these actions can be directly learned from the user's behavior data. However, it is not optimal to use any single one of these predicted scores to rank and display the final contents because the ultimate goal of recommender is to improve *multiple* downstream business goals. For example, the goals could be to maximize the average user engagement time while the average content like rate does not decrease. These business purposes are usually the global and overall metric, which can hardly be modeled by users' immediate actions and are not directly learnable. Therefore, in the second phase, one needs an aggregation function to compute an overall score

[*]weicongding@kuaishou.com
[†]tanghl1994@gmail.com
[‡]jingsf@uw.edu
[§]ji.liu.uwisc@gmail.com

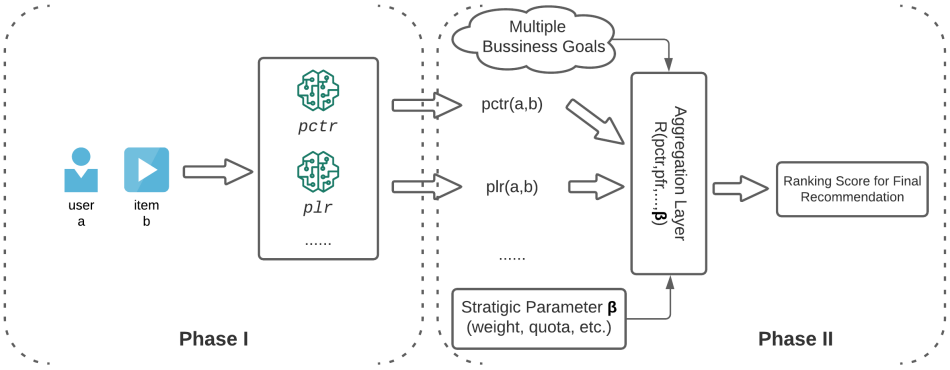

Figure 1: A illustrating example of strategic parameters in a simplified recommendation system with two phases. The 1st phase has multiple predictors for click-through-rate (pctr), like-rate (plr), etc. The 2nd phase is an aggregation layer with strategic parameter $\boldsymbol{\beta}$ to generate the final recommendation score. Note that the strategic parameters can be in many other components of a recommendation system.

for each user-item pair (a, b) based on the predictions from the first phase,

$$R_{\boldsymbol{\beta}}(a, b) = R(\text{pctr}(a, b), \text{plr}(a, b), \cdots; \boldsymbol{\beta}),$$

where $\text{pctr}(a, b)$, $\text{plr}(a, b)$ are the predicted content click through rate, like rate, etc. $\boldsymbol{\beta}$ is the hyper parameter of the function $R()$, which is referred as the **strategic parameter** in recommendation. A simple version of $R_{\boldsymbol{\beta}}(\cdot)$ can be a weighted sum [22, 30, 36] where $\boldsymbol{\beta}$ are the mixing weights. Our paper focuses on the strategic parameters.

Examples like Figure 1 is ubiquitous in industrial recommendation systems. The strategic parameter $\boldsymbol{\beta}$ can be in many components of a practical recommendation system and are often in the form like weights, thresholds, or quotas. While the problem of building predictive models (e.g, models in the 1st phase in Figure 1) has been extensively studied[10, 35], deciding the strategic parameter has not yet received equal attention. It is mainly because the problem is generally classified into a *standard* black-box optimization problem – tuning the strategic parameters to optimize the overall business metrics. Manual tuning via controlled experiments are popular in practice [28, 29, 36] and more sophisticated methods like bandit optimization [4, 11], evolution algorithms [5], Pareto-efficiency [21, 23, 26], and Bayesian optimization [3, 16, 19] have been explored. All of these existing methods pursue a *deterministic* (or single) strategic parameter; that is, the same strategic parameter $\boldsymbol{\beta}$ is applied to all the requests/users to the recommendation system.

In this paper, we argue that the deterministic (single) strategic parameter is *not* the optimal solution. We discover that a *probabilistic* choice for the strategic parameter can be superior to any deterministic one, especially when there are multiple business goals to pursue simultaneously. In the probabilistic solution, we learn an optimal distribution over all the candidate strategic parameter choices. The system then works as follows: when there is a request to the recommender (e.g., a user visit), we first sample one of the multiple strategic parameters using the learned distribution and then apply the randomly selected strategic parameter in the recommendation pipeline. A common recommender is visited by users hundreds of millions of times daily. Therefore, the proposed solution can be viewed as a probabilistic mixture over multiple strategic parameters. The probabilistic solution achieves supreme performance over deterministic solution since the latter can therefore be viewed as restricting the distribution vector in the probabilistic solution to a one-hot vector. We demonstrate the supreme performance using extensive simulation studies and online AB testing results later in this paper.

The challenges of finding the optimal probabilistic parameters are two folded. On the one hand, the distribution of the unknown metrics can only be learned by interacting with the online customer-facing systems and then observing sparse and noisy samples via multiple iterations. On the other hand, the optimization problem to find the best probabilistic distribution falls into the family of stochastic compositional optimization, in which the unbiased stochastic gradient is unavailable. We propose to use an average of the unbiased estimator over the history in approximating the unbiased stochastic gradient. This trick also helped to address the sparse observation and reduced the noise. We also incorporated proximal projection to remove the simplex constraint. The proposed approach, Probabilistic pArameter optimization with unbiased STOchastic gradient approximation (`PASTO`), converges at a rate $\mathcal{O}\left(1/\sqrt{T}\right)$ and admits the regret in an order of $\mathcal{O}\left(\sqrt{T}\right)$.

In sum, the key contributions of this paper are,

- We discover a *probabilistic* strategic parameter solution that outperforms the classic deterministic strategic parameter when we are pursuing multiple goals in recommendation systems.
- We formulate the problem of finding the optimal probabilistic parameter solution as a compositional stochastic optimization task, and developed an efficient stochastic gradient algorithm. We proved that the proposed algorithm converges to the optimal probabilistic distribution at a rate of $\mathcal{O}\left(1/\sqrt{T}\right)$ and the regret admits an order of $\mathcal{O}\left(\sqrt{T}\right)$ where T is the number of iterations.
- We implement the proposed probabilistic strategic parameter solution at a leading social network platform with hundreds of millions of daily active users and tens of billions annual revenue. Note that in a platform of this scale, a slight percentage gain provides enormous business value. The proposed approach achieved $+0.22\%$ lift of user engagement in a recommendation scenario and $+1.7\%$ lift in revenue in an advertising optimization scenario compared with the optimal deterministic parameter choice.

The rest of this paper is organized as follows. We first discuss related literature in Section 2. In Section 3 we formulate the strategic parameter searching as an optimization problem and argue why the probabilistic regime is better. We discussed our solution in Section 4 and provide the theoretical analysis in Section 5. We include a series of simulation studies in Section 6, and present online AB testing results in Section 7.

2 Related Work

Black-box strategic parameter tuning has been extensively studied using both heuristic and Bayesian approaches to find the maximizer of an unknown and noisy objective function. Popular heuristic approaches such as Genetic Algorithm [24], Cross-Entropy-Methods [28], and Particle Swarm Optimization [17] have been widely adopted empirically. [16, 19] proposed to use the Bayesian approach to sequentially explore the strategic parameters. [1, 9, 14] proposed to use the Bayesian approach with contextual information in industrial recommendation systems. Multi-armed bandits (MAB) and bandit optimization problems are also widely used in searching the best deterministic strategic parameter [4, 31]. The contextual bandit setting considers environmental conditions and can select a deterministic parameter for each different context [13, 18]. We note that this is different than the proposed probabilistic regime (see Section 3 for detailed discussion).

Multi-task learning has raised attention recently in recommendation systems [10, 35]. However, the focus of existing literature is in the phase of predicting immediate user actions [8, 12, 15, 30] and not the strategic parameters. Different architecture for model-parameter sharing, loss sharing, etc., have been explored [8, 22, 30, 36]. Other ways of addressing multiple losses have also been studied, such as weighted sum [21, 27] and adaptive optimization optimization[8, 30]. Regarding learning deterministic strategic parameters with multiple objectives, [6] uses an evolutionary algorithm to include a diversity indicator on top of item rating evaluation; [26] maximizes a so-called economic value based on reinforcement learning. Recently, much attention has been paid to finding the Pareto frontier of multiple goals [23], where a set of Pareto optimal items is selected, and no alternative can improve every objective simultaneously. And Pareto efficient algorithms can help coordinates multiple objectives [21, 27] and the problem can be solve by entropy search [3, 29], expected hyper-volume improvement [7], etc.

Recently, [32] proposed to personalize the strategic parameter assignment based on the user's demographic group or device type. However, in their proposed settings, deterministic single-best parameters are still the solution for each user group or context. We also note that [32] uses post AB-testing data to estimate the stochastic effect of treatment/parameters. Our paper considered a dynamic and iterative approach that is more effective in industrial applications.

3 Why Probabilistic Solution is Better Than Singleton Solution

Now we formally define the probabilistic strategic parameter solution and show its advantage over the deterministic one. We start by formulating the strategic parameter selection problem into an optimization task. Then we show that the proposed probabilistic solution defines a larger feasible domain than the deterministic solution, implying its superior performance.

3.1 Optimization View of the Strategic Parameter Tuning Problem

Recall the motivating example in Section 1 where we tune the strategic parameter $\boldsymbol{\beta}$ to optimize one or multiple business goals, e.g.,

- maximize the expected averaged engagement time (μ^X),
- maximize the expected averaged engagement time (μ^X) while the like rate (μ^Y) does not drop,
- maximize a utility function of both the expected averaged engagement time (μ^X) and the like rate (μ^Y).

In this example, to be more precise, we denote by $\mu^X(\boldsymbol{\beta})$ the expected daily averaged engagement time when applying the strategic parameter $\boldsymbol{\beta}$ in the system, and $\mu^Y(\boldsymbol{\beta})$ follow a similar definition. Note that the ground truth value of $\mu^X(\boldsymbol{\beta})$ or $\mu^Y(\boldsymbol{\beta})$ is difficult to obtain. We can usually obtain an unbiased sample (denote as $\hat{u}^X(\boldsymbol{\beta})$) by applying $\boldsymbol{\beta}$ to a group of randomly selected users, namely,

$$\hat{u}^X(\boldsymbol{\beta}) := \frac{1}{\#\text{users}} \sum_j \text{user } j\text{'s } X \text{ metric (i.e., daily engagement time) applying strategic parameter } \boldsymbol{\beta}.$$

Here $\mathbb{E}(\hat{u}^X(\boldsymbol{\beta})) = \mu^X(\boldsymbol{\beta})$ is a unbiased observation of $\mu^X(\boldsymbol{\beta})$ whose variance depends on the number of users/requests in our observation. In general, we use X to denote the metrics we primarily want to improve and Y to denote the guardrail metrics we do not want to drop. Next we can formulate the **conventional** strategic parameter optimization in the form of a numerical optimization problem

$$\max_{\beta \in \mathcal{B}} \quad f(\mu^X(\boldsymbol{\beta}), \mu^Y(\boldsymbol{\beta})) \tag{1}$$

where $\mathcal{B}$ is the set of all possible strategic parameters. For different goals, f can be designed as,

- $f(x, y) = x$ if we only have one metric to optimize. This is less common in practice.
- $f(x, y) = x - h(y; c)$ where $h(y) = \infty$ if $y \leq c$ and $h(y) = 0$ otherwise. This f maximizes the metric μ^X while strictly requiring metric μ^Y to be higher than a lower-bound c.
- $f(x, y) = x - \lambda \min(0, y - c)^2$ and $\lambda > 0$ is some constant. This combined utility function aims to improve metric X while imposing a square penalty if the metric Y drops below a pre-define threshold c.

3.2 Probabilistic Strategic Parameter Solution

Now to illustrate our proposed probabilistic solution, we first reformulate (1) into an mathematically equivalent form. Here we assume that the number of possible strategic parameter options $\mathcal{B} = \{\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^K\}$ is finite and is of size K for simplicity. The infinite scenario follows the same spirit. [5] Let's first denote the rewards of different strategic parameters compactly in vector form

$$\boldsymbol{\mu}^X(\mathcal{B}) = \left[\mu^X(\boldsymbol{\beta}^1), \ldots, \mu^X(\boldsymbol{\beta}^K)\right] \quad \boldsymbol{\mu}^Y(\mathcal{B}) = \left[\mu^Y(\boldsymbol{\beta}^1), \ldots, \mu^Y(\boldsymbol{\beta}^K)\right]$$

And re-write the optimization Eq (1) in an equivalent form,

$$\begin{aligned} \max_{\boldsymbol{p}} \quad & f(\boldsymbol{\mu}^X(\mathcal{B})\mathbf{p}, \boldsymbol{\mu}^Y(\mathcal{B})\mathbf{p}) \\ \text{s.t.} \quad & \mathbf{p} \in S := \left\{\mathbf{p} \in \{0,1\}^K | \Sigma_{k=1}^K p_k = 1\right\}. \end{aligned} \tag{2}$$

It is worth noting that the optimal solution to Eq. (2) selects the best option from $\mathcal{B}$ to optimize the target goal. We refer to this as the **deterministic** solution since the same strategy parameter is applied to all the recommendation requests.

Next we are ready to propose the probabilistic solution. Revisiting (2), one can mathematically change feasible set selection variable $\mathbf{p}$ to a larger set, $\bar{S} = \{\mathbf{p} \in [0,1]^K | \Sigma_{k=1}^K p_k = 1\}$, which is the convex hull formed by S. $\bar{S}$ is the K-dimensional probability simplex and $\mathbf{p} \in \bar{S}$ can be viewed as a pmf over K strategic parameters in $\mathcal{B}$.

The pmf view of $\mathbf{p}$ suggests a new **probabilistic** solution of applying strategic parameters in recommendation systems. Say for example $K = 3$ and $\mathbf{p} = [0.8, 0.2, 0]$. When a user visits the platform and a recommendation request is created, we randomly select either $\boldsymbol{\beta}^1$ with probability 0.8 or $\boldsymbol{\beta}^2$ with 0.2, and then apply the selected $\boldsymbol{\beta}^1$ or $\boldsymbol{\beta}^2$ as the strategic parameter to generate final recommendation list for the user. If we consider the average effect over hundreds of millions of daily requests in a industrial recommender, the inner product $\boldsymbol{\mu}^X(\mathcal{B})\mathbf{p} = \sum_k p_k \mu^X(\boldsymbol{\beta}^k)$ still represents the expected metric X (here the expectation is over both the randomness of sampling from $\mathbf{p}$ and the noise in $\hat{u}^X$). Formally, we propose to pursue the optimal probabilistic solution by solving,

$$\begin{aligned} \max_{\boldsymbol{p}} \quad & f(\boldsymbol{\mu}^X(\mathcal{B})\mathbf{p}, \boldsymbol{\mu}^Y(\mathcal{B})\mathbf{p}) \\ \text{s.t.} \quad & \mathbf{p} \in \bar{S} = \{\mathbf{p} \in [0,1]^K | \Sigma_{k=1}^K p_k = 1\}. \end{aligned} \tag{3}$$

[5] See appendix for discussion on continuous parameter space.

3.3 Probabilistic Solution Achieves better Objective Value than Singleton Solution

Since f is in general a non-linear in Eq (3), the optimal $\mathbf{p}$ to Eq (3) may not be a one-hot solution. For the same f and $\boldsymbol{\mu}$'s in (2) and (3), let $\mathbf{p}_S^\star$ be the optimal (deterministic parameter) solution to Eq. (2) and $\mathbf{p}_{\bar{S}}^\star$ be the optimal (probabilistic parameter) solution to Eq. (3). Since the feasible set of (2) (deterministic solution) is a subset of (3). It is straightforward to verify that,

Observation 1 The optimal objective value $f(\boldsymbol{\mu}^X \mathbf{p}_S^*, \boldsymbol{\mu}^Y \mathbf{p}_S^*) \leq f(\boldsymbol{\mu}^X \mathbf{p}_{\bar{S}}^*, \boldsymbol{\mu}^Y \mathbf{p}_{\bar{S}}^*))$. Namely, *the probabilistic parameter solution achieves better reward compared to the deterministic parameter.*

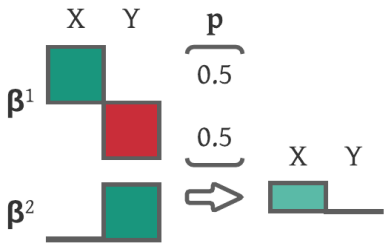

Figure 2: An Illustrating Example on why probabilistic parameter can achieve better overall rewards than single deterministic parameter. X, Y refer to two different business metrics, $\boldsymbol{\beta}^1, \boldsymbol{\beta}^2$ are two possible strategic parameter choices. $\mathbf{p} = [0.5, 0.5]$ is a probabilistic parameter mixing weight that samples $\boldsymbol{\beta}^1$ or $\boldsymbol{\beta}^2$ with equal probability for each request. The right-hand side indicated the overall rewards achieved with $\mathbf{p}$.

An Illustrating Example: To illustrate Observation 1, consider $K = 2$ strategic parameters and $\mu^X(\mathcal{B}) = [2, 0]$,i.e., the first choice $\boldsymbol{\beta}^1$ has an expected engagement time of 2 and the engagement time is 0 for $\boldsymbol{\beta}^2$). In practice we focus more on the relative gain over some baseline. So setting $\mu^X(\boldsymbol{\beta}^2) = 0$ means there is no relative gain on engagement time over baseline. We also set $\mu^Y(\mathcal{B}) = [-2, 2]$ for metric Y. Let the objecive function be $f(x, y) = x - h(y, c = 0)$ where h imposes infinity penalty when $y < 0$. For deterministic solutions, selecting first parameter $\mathbf{p}_{(1)} = [1, 0]^\top$ results in $f = -\infty$ since the metric Y is below the threshold $c = 0$ for first strategic parameter $\boldsymbol{\beta}^1$. On the other hand, selecting the second parameter $\mathbf{p}_{(2)} = [0, 1]^\top$ result in $f = 0$ since $\boldsymbol{\beta}^2$ has a lower metric Y. Now for the probabilistic case, we can choose $\mathbf{p}_{(\text{mix})} = [0.5, 0.5]^\top$ and the average metric $\boldsymbol{\mu}^Y \mathbf{p} = 0$ is above the threshold so the reward is $f = 1$. Therefore, the $\mathbf{p}_{(\text{mix})}$ achieves better objective value than the best deterministic solution $\mathbf{p}_{(2)}$.

The illustrating example we've discussed represents a common real-world scenario, that is, one strategic parameter choice ($\boldsymbol{\beta}^1$ in this example) improves metric X but at the same time results in a lower average value of metric Y, and vice versa for another strategic parameter choice ($\boldsymbol{\beta}^2$). This scenario is common as multiple business metrics in real applications often compete with each other (given limited user attention on the platform). We will demonstrate this in Section 6 and 7.

Before concluding this section, note that we used two metrics X, Y so far for simplicity. In real application, there are typically multiple metrics X's to improve and various metrics Y's to protect. The formulation and conclusions in this section directly extends to this general setting.

4 How to Solve the Probabilistic Parameter Optimization

We now discuss how to solve the probabilistic parameter optimization problem in Eq (3), which falls into the family of constrained stochastic compositional optimization [34] tasks. The key difficulty in solving Eq (3) is that the ground-truth rewards $\boldsymbol{\mu}^X(\boldsymbol{\beta})$ and $\boldsymbol{\mu}^Y(\boldsymbol{\beta})$ are unknown. To obtain an empirical sample $\hat{u}$ in practice, one needs to choose one of the strategic parameters (say $\boldsymbol{\beta}^k$), apply it to the actual recommender system (typically to a small group of users/requests), and observe the empirical average metrics $\hat{u}^X(\boldsymbol{\beta}^k), \hat{u}^Y(\boldsymbol{\beta}^k)$ (in our motivating example, they are the user's daily engagement time and his/her like rate), and repeat this for multiple rounds/iterations. The observation at each iteration is a sparse sample of $\boldsymbol{\mu}^X(\boldsymbol{\beta})$ and $\boldsymbol{\mu}^Y(\boldsymbol{\beta})$. Furthermore, the observation of one iteration can only be collected after a certain time. Say, if hourly engagement time is our goal, then, we need to apply $\boldsymbol{\beta}^k$ in the system for at least one hour.

We choose to solve our problem using stochastic gradient approach. The sparse and noisy observation in our problem raises two technical challenges: 1) the unbiased stochastic gradient is generally hard to obtain; 2) the constraint is quite tricky to enforce. We next discuss our technical solutions to address them.

4.1 Unbiased Stochastic Gradient Approximation

In solving Eq (3), an unbiased stochastic gradient for is hard to obtain since the our observation in each round to approximate the true expectations $\boldsymbol{\mu}^X(\boldsymbol{\beta})$ and $\boldsymbol{\mu}^Y(\boldsymbol{\beta})$ are noisy and sparse. To be concrete, let's define in the matrix form

$$\boldsymbol{\mu} = \boldsymbol{\mu}(\mathcal{B}) = \begin{bmatrix} \mu^X(\boldsymbol{\beta}^1), \ldots, \mu^X(\boldsymbol{\beta}^K) \\ \mu^Y(\boldsymbol{\beta}^1), \ldots, \mu^Y(\boldsymbol{\beta}^K) \end{bmatrix} \in \mathbb{R}^{2\times K} \tag{4}$$

and compactly write $f(\boldsymbol{\mu}\mathbf{p})$ where $\boldsymbol{\mu}\mathbf{p} = \left[\boldsymbol{\mu}^X(\mathcal{B})\mathbf{p}, \boldsymbol{\mu}^Y(\mathcal{B})\mathbf{p}\right]^\top$ for the objective function in Eq (3). Note that there could be more than 2 rows in $\boldsymbol{\mu}$ if we have more than 2 metrics of interests. For simplicity, we will keep the dimension as 2 throughout this section.

Since vector $\boldsymbol{\mu}$ are the expectation of the noisy metric observations, the optimization objective $f(\boldsymbol{\mu}\mathbf{p})$ in (3) is different from the common stochastic optimization admitting the form of $\mathbb{E}\left[f(\cdot)\right]$ (with no expectation inside f). This falls into the family of stochastic compositional objectives and the unbiased stochastic gradient is generally not achievable [33]. More specifically, the true gradient of (3) at learning step t admits the form

$$\frac{\mathbf{d}f(\boldsymbol{\mu}\mathbf{p})}{\mathbf{d}\mathbf{p}} = \boldsymbol{\mu}^\top \nabla f(\boldsymbol{\mu}\mathbf{p}),$$

which requires us to obtain unbiased estimator of the term $\boldsymbol{\mu}^\top \nabla f(\boldsymbol{\mu}\mathbf{p})$ at each iteration (learning step) t. Recall that at each round t, we can sample one strategic parameter $\boldsymbol{\beta}^{i_t}$ from the current distribution $\mathbf{p}_t$, and observed empirical metrics $\hat{u}^X(\beta^{i_t}), \hat{u}^Y(\beta^{i_t})$, etc. We can then construct

$$\hat{U}_t = \begin{bmatrix} 0, & \ldots & , 0, \hat{u}^X(\beta^{i_t})/p_{t,i_t}, 0, \ldots, 0 \\ 0, & \ldots & , 0, \hat{u}^Y(\beta^{i_t})/p_{t,i_t}, 0, \ldots, 0 \end{bmatrix} \tag{5}$$

and verify that $\mathbb{E}(\hat{U}_t) = \boldsymbol{\mu}$ is unbiased estimator of $\boldsymbol{\mu}$. However, this cannot ensure that the stochastic gradient we compute is unbiased, because $\mathbb{E}(\hat{U}_t^\top \nabla f(\hat{U}_t\mathbf{p}_t)) \neq \mathbb{E}(\hat{U}_t)^\top \nabla f(\mathbb{E}(\hat{U}_t)\mathbf{p}_t) = \boldsymbol{\mu}^\top \nabla f(\boldsymbol{\mu}\mathbf{p}_t)$ unless f is linear. In this case, our solution is to use the averaged value of all historical (unbiased) estimator

$$\hat{V}_t := \frac{1}{t}\sum_{s=1}^{t} \hat{U}_s \tag{6}$$

which gets more and more accurate as training continues. We then approximate the unbiased gradient using

$$\boldsymbol{g}_t = \hat{V}_t^\top \nabla f(\hat{V}_t\mathbf{p}_t). \tag{7}$$

We note that our gradient estimation is different than the stochastic compositional gradient descent (SCGD) framework in [33, 34]. This difference is due to the linear form $\boldsymbol{\mu}\mathbf{p}$ inside the f function, a special case of the generic SCGD. Since we used the history average $\hat{V}_t$, the estimation of gradient in (7) is more stable and converges faster than the generic SCGD solution [33] (see Section 6 for simulation study).

4.2 KL divergence removes the simplex constraint

The second technical challenge is to handle the simplex constraint $\mathbf{p} \in \bar{S}$. One can certainly apply the projected step after the (approximate) stochastic gradient descent step. We choose KL divergence (other than the Euclidean distance) as the Bregman distance which can naturally ensure the next iterate $\mathbf{p}_{t+1}$ within the simplex constraint $\bar{S}$ even without considering the simplex constraint. More specifically, $\mathbf{p}_t$ is updated by the following constraint-free proximal step with a given step-size γ_t

$$\mathbf{p}_{t+1} = \text{argmin}_{\mathbf{p}} - \langle \boldsymbol{g}_t, \mathbf{p}\rangle + \frac{1}{\gamma_t}\text{KL}(\mathbf{p}\|\mathbf{p}_t)$$

resulting in the following closed form update rule,

$$w_{t,k} = p_{t,k}\exp\left(\gamma_t \boldsymbol{g}_{t,k}\right),\ k = 1, \ldots, K, \qquad p_{t+1,k} = w_{t,k}/\sum_{k'=1}^{K} w_{t,k'}.$$

4.3 Overall Algorithm

We summarize the proposed robabilistic Parameter Optimization with Unbiased Stochastic Gradient Approximation in Algorithm 1 as an iterative exploration and optimization procedure. At each round, we first sample one strategic parameter from the current probabilistic pmf $\mathbf{p}_t$, observed its rewards from online recommender, and then conduct the gradient optimization. Note that instead of using the $\mathbf{p}_T$ from last round as the final return, we used the averages $\bar{\mathbf{p}}_T = \frac{1}{T}\sum_{s=1}^{T}\mathbf{p}_s$ as the estimated best probabilistic solution, which is more stable in practice. We show this average do converges to the optimal rewards.

Algorithm 1 **P**robabilistic p**A**rameter Optimization with unbiased **STO**chastic Gradient Approximation (PASTO)

Require: Initial estimator of rewards $\widehat{U}_0$ ($2 \times K$ matrix), learning rate γ, smoothing parameter $\epsilon_t, t = 1, \ldots, T$, the total number of iteration steps T;

Ensure: $\bar{\mathbf{p}}_T$

1: Initialize $\mathbf{w}_0 = [1, \ldots, 1]^\top =: \mathbf{1} \in \mathbb{R}^K$, $\hat{V}_0 \leftarrow \hat{U}_0$
2: **for** each iteration $t = 1, \ldots, T$ **do**
3: Sample one strategic parameter $\boldsymbol{\beta}^{i_t}$ from $\mathcal{B}$ based on the pmf $\boldsymbol{p}_t = (1-\epsilon_t)\frac{\boldsymbol{w}_{t-1}}{\|\boldsymbol{w}_{t-1}\|_1} + \epsilon_t/K \cdot \mathbf{1}$
4: Apply the sampled strategic parameter $\boldsymbol{\beta}^{i_t}$ to the online system and observe $\hat{u}^X(\boldsymbol{\beta}^{i_t}), \hat{u}^Y(\boldsymbol{\beta}^{i_t}), \ldots$. Construct $\hat{U}_t$ based on Eq (5)
5: Update $\hat{V}_t = \frac{t}{t+1}\hat{V}_{t-1} + \frac{1}{t+1}\hat{U}_t$ (This is the same as Eq (6) but in a recursive form.)
6: Compute the stochastic gradient using Eq (7) $\boldsymbol{g}_t := \left(\hat{V}_t\right)^\top \nabla f(\hat{V}_t \boldsymbol{p}_t)$
7: Update $w_{t,i} = w_{t-1,i}\exp\left(\gamma g_{t,i}\right)$
8: **end for**
9: Return $\bar{\mathbf{p}}_T := \frac{1}{T}\sum_{t=1}^{T}\mathbf{p}_t$.

Note that following [2], we added a smoothing parameter ϵ_t at each iteration to ensure all candidate parameters have a non-zero chance of being selected for exploration and to cap the unbiased estimation (5) . Intuitively, higher ϵ_t also introduces more exploration of the strategic parameters. Similarly, the step-size γ also controls the degree of exploration.

5 Theoretical Analysis

We now present the theoretical analysis for Algorithm 1. Without loss of generality, we assume that f in the optimization problem (3) is a concave objective function. [6] For the results to hold, we first introduce a few common technical conditions on our objective function f and the underlying ground-truth rewards $\boldsymbol{\mu}$'s.

Assumption 1. *We make the following commonly used assumptions for $f(\cdot)$ and $\boldsymbol{\mu}(\mathcal{B})$:*

- *The gradient of objective $f(\cdot)$ defined in Eq* (3) *is bounded, and $f(\cdot)$ is L_f-smooth, i.e.,*

$$\|\nabla f(\boldsymbol{\theta}_1) - \nabla f(\boldsymbol{\theta}_2)\| \le L_f\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\| \quad \text{and } \|\nabla f(\boldsymbol{\theta}_1)\| \le G_f, \quad \forall \boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2$$

- *The Frobenius norm of $\boldsymbol{\mu}(\mathcal{B})$ defined in Eq* (4) *is bounded, i.e., $\|\boldsymbol{\mu}(\mathcal{B})\|_F \le G_U$*
- *The magnitude of the gradient estimation $\boldsymbol{g}_t$ is bounded for all iterations, i.e., $\|\boldsymbol{g}_t\| \le G, \forall t$.*

5.1 Convergence Analysis

Since $\bar{\mathbf{p}}_T$ is what we deploy in the actual production system after the learning process, our **primary** focus here is the convergence of Algorithm 1. Namely, we want to show the estimate $\bar{\mathbf{p}}_T$ do converge to the optimal probabilistic mixing vector as T goes to infinity. Equivalently, we need to show that the overall reward that can be achieved by $\bar{\mathbf{p}}_T$, i.e., $f\left(\boldsymbol{\mu}\bar{\boldsymbol{p}}_T\right)$, can converge to the optimal reward of Eq (3) (the convergence is in the sense of expectation). Formally,

[6] Note that f represents the rewards to be maximized in application. Therefore, we assume f being concave in our analysis.

Theorem 1. ***(Convergence of Alg. 1)*** *Under Assumption 1, set $\epsilon_t = \frac{G}{\sqrt{t}}$ and $\gamma = \frac{1}{\sqrt{T}}$, further, assuming that the noisy observations $\hat{U}_t$ and $\hat{U}_s$ from two different iterations t, s are independent. Then,*

$$\max_{\mathbf{p}\in\bar{S}} f(\boldsymbol{\mu}\mathbf{p}) - \mathbb{E}\left[f(\boldsymbol{\mu}\bar{\mathbf{p}}_T)\right] \leq \frac{1}{\sqrt{T}}\left(\ln K + 4eG^2 + \frac{32(G_f^2 + G_U^2 L_f^2)G_U^2}{G}\right).$$

Here the expectation is over all the randomness in the gradient history $\{\mathbf{p}_t\}_{t=1}^T$.

In short, the estimation value $f(\boldsymbol{\mu}\bar{\mathbf{p}}_T)$ converges to the optimal reward in expectation and the convergence rate admits the order of $\mathcal{O}\left(\frac{\ln K}{\sqrt{T}}\right)$. Theorem 1 implies that using the $\bar{\mathbf{p}}_T$ in the online production system can yield the optimal rewards of the probabilistic solution.

5.2 Regret Analysis

Next, we analyze the regrets bound of the iterative learning process. In this analysis, we assume the ground-truth rewards $\boldsymbol{\mu}$ could be different over time. This is often the case in real-world applications since customer preferences and overall trends are constantly changing. Concretely, we denote by $\boldsymbol{\mu}_t = \boldsymbol{\mu}_t(\mathcal{B})$ the ground-truth metric at t-th iteration (the same format as in Eq (4)).

From a practice viewpoint, till t-th round, the average historical metrics would be $\frac{1}{t}\sum_{s=1}^t \boldsymbol{\mu}_s$ in expectation and $\hat{V}_t = \frac{1}{t}\sum_{s=1}^t \hat{U}_s$ the empirical observation. Given this, $f(\frac{1}{t}\sum_{s=1}^t \boldsymbol{\mu}_s\mathbf{p})$ represents the (ideal) objective value if one can re-select a solution $\mathbf{p}$ at t-th interation given the full history of observations, and the regret can be defined as the gap between this reward objective with $\mathbf{p}_t$ and the global optimal $\mathbf{p}^*$, $f(\frac{1}{t}\sum_{s=1}^t \boldsymbol{\mu}_s\mathbf{p}^*) - f(\frac{1}{t}\sum_{s=1}^t \boldsymbol{\mu}_s\mathbf{p}_t)$, where $\mathbf{p}_t$ is from Algorithm 1. Similarly, the regret can be also defined as the gap $f(\frac{1}{t}\sum_{s=1}^t \boldsymbol{\mu}_s\mathbf{p}^*) - f(\frac{1}{t}\sum_{s=1}^t \hat{V}_s\mathbf{p}_t)$ between the global optimal and the empirical loss, which is more close to the empirical regrets in the procedure of Algorithm 1.

With all these notations, we show the following bounds on the total regret,

Theorem 2. ***(Regret of Alg. 1)*** *For Algorithm 1, under Assumption 1, $\epsilon_t = \frac{G}{\sqrt{t}}$ and $\gamma = \frac{1}{\sqrt{T}}$, the regrets are:*

$$\max_{\mathbf{p}\in\bar{S}} \sum_{t=1}^T f\left(\frac{1}{t}\sum_{s=1}^t \boldsymbol{\mu}_s\mathbf{p}\right) - \sum_{t=1}^T \mathbb{E}\left[f\left(\frac{1}{t}\sum_{s=1}^t \boldsymbol{\mu}_s\mathbf{p}_t\right)\right] \leq \sqrt{T}\left(\ln K + 4eG^2 + \frac{32(G_f^2 + G_U^2 L_f^2)G_U^2}{G}\right).$$

and

$$\max_{\mathbf{p}\in\bar{S}} \sum_{t=1}^T f\left(\frac{1}{t}\sum_{s=1}^t \boldsymbol{\mu}_s\mathbf{p}\right) - \sum_{t=1}^T \mathbb{E}f\left(\frac{1}{t}\sum_{s=1}^t \hat{U}_s\mathbf{p}_t\right) \leq \sqrt{T}\left(\ln K + 4eG^2 + \frac{32(G_f^2 + G_U^2 L_f^2)G_U^2}{G} + 2G_U^2\right).$$

As Theorem 2 suggests, our Algorithm 1 can achieve a regret at the order of $\mathcal{O}(\sqrt{T})$. We defer all the proof details in supplementary.

6 Simulation Study

We conduct simulation to demonstrate the **gain** of probabilistic solution over single deterministic ones. For all experiments, we choose objective of the form $f(x, y_1, y_2, \ldots) = x - 5.0 * \min(0, y_1 - c_1)^2 - 5.0 * \min(0, y_2 - c_2)^2 - \ldots$ if there are multiple guardrail metrics ys. For Algorithm 1, we set $\epsilon_t = 0.1/\sqrt{t+10}$ and $\gamma = 0.1/K$. Since our goal is to compare the performance of probabilistic solution against the deterministic one. Therefore, when reporting the performance of the deterministic(single) solution, we always assume having access to the noise-less rewards and knowing which single strategic parameter yields the highest expected objective value, which will be the upper bound for *any* deterministic parameter algorithm.

Parallel Querying: in each round of Algorithm 1, we could choose $Q \geq 1$ choices of strategic parameters by sampling from $\mathbf{p}_t$. Practically, we can randomly split the users or requests into Q subgroups and apply one different strategic parameter for each group accordingly.

6.1 Simulation on Synthetic Data

We first verify that the proposed Algorithm 1 does converge empirically using the following setup:

(A) the illustrating example in Section 3 where $K = 2$, $\boldsymbol{\mu}^X = [2, 0]$, $\boldsymbol{\mu}^Y = [-2, 2]$ and $c = 0$. If the k-th paramter is queried in one round, we add a Gaussian noise of $\mathcal{N}(0, 5.0)$ to μ_k^X (and μ_k^Y resp.) to simulate the noisy observations $\hat{u}_k^X$ and $\hat{u}_k^Y$. We set parallel querying $Q = 1$.

(B) $K = 100$ and μ_k^X, $\mu_k^{Y_1}$, $\mu_k^{Y_2}$ for $k = 1, \ldots, K$ are sampled independently and uniformly within $[-1, 1]$. We set $c^{Y_1} = c^{Y_2} = 0.5$. An additive Gaussian noise $\mathcal{N}(0, \sigma^2)$ is added to each round's observation if a strategic parameter is selected. We set parallel querying $Q = 10$.

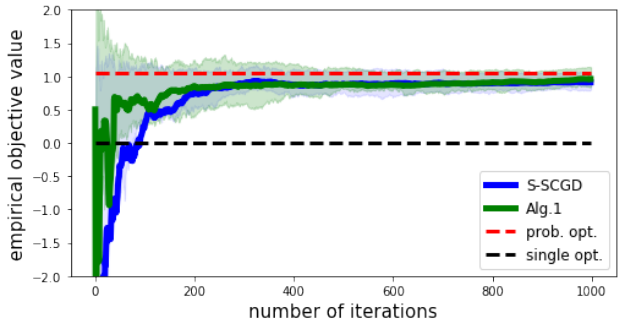

Figure 3: Empirical convergence of our proposed Algorithm 1 and S-SCGD on the example A. Shaded area indicate the 75th and 25th percentile of the objective in each round in 1000 Monte Carlo runs.

For the first toy example (A), we conduct 1000 Monte Carlo runs and report the average empirical objective value. To evaluate the efficiency of the proposed approach, we also compared against the stochastic compositional gradient descent approach [33] (S-SCGD). As shown in Figure 3, our proposed algorithm does converge to the correct optimal probabilistic solution. As one can easily verify from the simulation setting, this probabilistic reward is indeed better than the single parameter solution. We also note that the convergence of Algorithm 1 is faster than the simple S-SCGD due to its reduction of noise. We will only use proposed Algorithm 1 in later sections due to its efficiency and robustness to sparse observations.

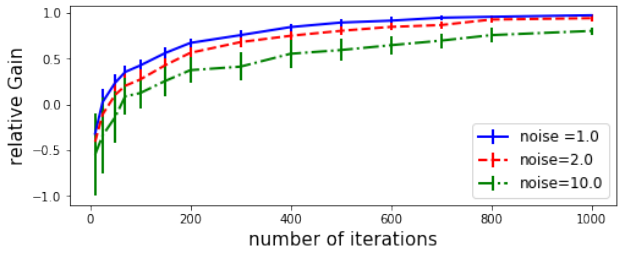

Figure 4: Relative gain of Algorithm 1 on synthetic dataset with different noise level σ. Average and standard deviation of 1000 random runs are reported.

For the second simulation setting (B), we also conduct 1000 Monte Carlo runs to generate ground-truth $\boldsymbol{\mu}$ and simulate our algorithm. To make it comparable across different ground-truth setting $\boldsymbol{\mu}$, we measure the relative gain of the probabilistic parameter over the single best parameter, defined as $r := \frac{|f(\bar{\mathbf{p}}_t) - f(\mathbf{p}_S^*)|}{|f(\mathbf{p}_{\bar{S}}^*) - f(\mathbf{p}_S^*)|}$, i.e., the gain over single best parameter normalized by the ideal objective gap between probabilistic parameter and single parameter. $r > 0$ indicates achieving a better objective than the single-best parameter. Figure 4 summarizes the average relative gain as the number of iteration for different noise levels. The error bars indicate the standard deviation of the 1000 Monte Carlo runs. We note that `TASCO` does converge and approaches the best possible gain. The noise of observation does impact the convergence speed of the approach.

6.2 Simulation on Real Dataset

Motivated by [22, 23], we simulate on two publicly available datasets, Amazon Books [25] and MovieLens 20M[7]. Each of them has customer-item rating interactions as well as price, genre information. We binaries the 5-star rating and set

[7] `https://grouplens.org/datasets/movielens/20m/`

ratings ≥ 3 as positive. Users and items with at least 5 ratings are kept in the processed data. Data were then split into training and test with $70\%, 30\%$.

We considered three targeted metrics: 1) the accuracy of relevance and measure it with the standard Recall@K (R@K) metrics, 2) the revenue achieved by ranking measured by the recall metrics weighted by the actual price of item (denoted as Revenue@K or REV@K) (for Amazon dataset), and 3) the accuracy of predicting a specific category video of 'Documentary' genre (for Movielens 20M dataset) (R-D@K). Higher metrics indicate better performance. We build Variational Auto-Encoder (VAE) [20] with 2 hidden layers using the training set as the underlying predictive model for each of the 3 targets. All the setups are summarized in Table 1.

Table 1: Setup of real world data set

Data	obj.1	obj.2	#.users	#.items	#.events
Amazon	R@20	REV@20	$93,976$	$25,896$	$964,363$
ML20M	R@20	R-D@20	$132,580$	$8,936$	$6,316,389$

The simulation runs as following. For each dataset, objective 1 in Table 1 is viewed as x and objective 2 as y. The underlying VAE models we have built predict the scores $p_1(\text{item, user}), p_2(\text{item, user})$, these scores are then aggregated using a power-based function $s = p_1(\text{item, user})^{\alpha} p_2(\text{item, user})^{1-\alpha}$ to rank the items. Here the parameter α as our strategic parameter.

We discretize the parameter space of α into $K = 100$ choices and set c being the metric of objective 2 when $\alpha = 0.5$. To simulated the noise in the online reward collection regime, for each round t, we randomly split the testing data into 10 folds, apply $Q = 10$ parallel strategic parameters, and then compute the empirical average objective metrics on these random subsets of testing data.

Table 2: Results on real world data simulation. Threshold is 4.28 for REV@20 an 0.080 for R-D@20.

	Amazon Book		ML20M	
	R@20	Rev@20	R@20	R-D@20
Probabilistic.	$\mathbf{0.284}$	4.31	$\mathbf{0.424}$	0.082
Single best	0.279	4.31	0.405	0.084
Single goal	0.286	–	0.430	–

We report the final rewards and constraints metrics in Table 2. The single best parameter, as discussed before, is identified by iterating through all possible choices using the entire testing set. We also report the reward by having non-constrained single obj.1 to understand the upper limit of our prediction models. Table 2 shows our probabilistic solution can achieve overall better results over the deterministic best parameter choices.

7 Real-World Application

We present two industrial applications on a leading social networking platform with hundreds of millions of daily active users and the AB-testing results.

Table 3: AB Test Result of Online Content Recommendation Task, Watch time is primary reward we would like to optimize. Like and Sharing are two guardrail constraints. A soft threshold of -2.0% was imposed on each of the constraint metrics. All results are reported as lift w.r.t. the baseline model at the time of experiment.

	time	like	sharing
Single Best	$+0.42\%$	-1.67%	-1.31%
Probabilistic	$+\mathbf{0.64}\%$	$+\mathbf{0.33}\%$	$+\mathbf{0.54}\%$

Ensemble Sort in Content Recommendation We consider the content ranking in one of the recommendation scenarios. The strategic parameters are ensemble weights of multiple recommendation queues, each optimized for a particular customer target events. This is similar to the illustrating example in Figure 1. The goal is to increase the average user engagement (time) while not dropping the 'like' action rate and the 'share' action rate. The existing

baseline parameter is a deterministic parameter and is extensively optimized. The AB testing results are outlined in Table 3. Note that the probabilistic setting improves not only the primary metrics but also the constraints metrics. This is due to the fact that some of the parameters in our probabilistic mix do yield higher gain in the 'like' and 'sharing' metrics.

Quota-Based Ads Retrieval Systems We also present real-world testing results in an ads-retrieval system of the platform. In this system, there are multiple modules, each attempting to retrieve a set of relevant ad contents with different types. One needs to combine these candidate sets into a single and smaller group to feed the downstream ad ranking and pricing models. Due to the limitation on computation time and power, quotas need to be set on the maximum number of ad contents generated by each module. Our goal here is to improve the overall advertising revenue and not to hurt a number of specific categories such as cold-start ad content.

We implement our probabilistic parameter solution in this ads system and compare it against an existing single-parameter baseline that has been extensively optimized. In our online AB test, the algorithm achieves a revenue improvement of $+1.7\%$ without significantly hurting the imposed constraints.

8 Conclusion

This paper argues that the probabilistic strategic parameter achieves better rewards than the deterministic parameter solution. We present an algorithm (`PASTO`) based on stochastic gradient descent to solve the probabilistic solution with theoretical guarantees. Both simulation and online applications have shown improvement over the deterministic best arm choice.

References

[1] Deepak Agarwal, Shaunak Chatterjee, Yang Yang, and Liang Zhang. Constrained optimization for homepage relevance. In *Proceedings of the 24th International Conference on World Wide Web*, pages 375–384, 2015.

[2] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[3] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 7825–7835, 2019.

[4] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

[5] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159. PMLR, 2013.

[6] Laizhong Cui, Peng Ou, Xianghua Fu, Zhenkun Wen, and Nan Lu. A novel multi-objective evolutionary algorithm for recommendation systems. *Journal of Parallel and Distributed Computing*, 103:53–63, 2017.

[7] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *arXiv preprint arXiv:2006.05078*, 2020.

[8] Tommaso Di Noia, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. Adaptive multi-attribute diversity for recommender systems. *Information Sciences*, 382:234–253, 2017.

[9] Weicong Ding, Dinesh Govindaraj, and SVN Vishwanathan. Whole page optimization with global constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3153–3161, 2019.

[10] Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):1–19, 2015.

[11] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *ICML*, 2010.

[12] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2493–2500, 2020.

[13] Xu He, Bo An, Yanghua Li, Haikai Chen, Qingyu Guo, Xin Li, and Zhirong Wang. Contextual user browsing bandits for large-scale online mobile recommendation. In *Fourteenth ACM Conference on Recommender Systems*, pages 63–72, 2020.

[14] Daniel N Hill, Houssam Nassif, Yi Liu, Anand Iyer, and SVN Vishwanathan. An efficient bandit algorithm for realtime multivariate optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1813–1821, 2017.

[15] Michael Jugovac, Dietmar Jannach, and Lukas Lerche. Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems with Applications*, 81:321–331, 2017.

[16] Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised bayesian optimisation via thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 133–142. PMLR, 2018.

[17] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[18] Andreas Krause and Cheng Soon Ong. Contextual gaussian process bandit optimization. In *Nips*, pages 2447–2455, 2011.

[19] Benjamin Letham, Brian Karrer, Guilherme Ottoni, Eytan Bakshy, et al. Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

[20] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.

[21] Xiao Lin, Hongjie Chen, Changhua Pei, Fei Sun, Xuanji Xiao, Hanxiao Sun, Yongfeng Zhang, Wenwu Ou, and Peng Jiang. A pareto-efficient algorithm for multiple objective optimization in e-commerce recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 20–28, 2019.

[22] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1137–1140, 2018.

[23] Nikola Milojkovic, Diego Antognini, Giancarlo Bergamin, Boi Faltings, and Claudiu Musat. Multi-gradient descent for multi-objective recommender systems. *arXiv preprint arXiv:2001.00846*, 2019.

[24] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[25] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

[26] Changhua Pei, Xinru Yang, Qing Cui, Xiao Lin, Fei Sun, Peng Jiang, Wenwu Ou, and Yongfeng Zhang. Value-aware recommendation based on reinforced profit maximization in e-commerce systems. *arXiv preprint arXiv:1902.00851*, 2019.

[27] Marco Tulio Ribeiro, Nivio Ziviani, Edleno Silva De Moura, Itamar Hata, Anisio Lacerda, and Adriano Veloso. Multiobjective pareto-efficient approaches for recommender systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(4):1–20, 2014.

[28] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[29] Shinya Suzuki, Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Multi-objective bayesian optimization using pareto-frontier entropy. In *International Conference on Machine Learning*, pages 9279–9288. PMLR, 2020.

[30] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Fourteenth ACM Conference on Recommender Systems*, pages 269–278, 2020.

[31] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[32] Ye Tu, Kinjal Basu, Cyrus DiCiccio, Romil Bansal, Preetam Nandy, Padmini Jaikumar, and Shaunak Chatterjee. Personalized treatment selection using causal heterogeneity, 2020.

[33] Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2):419–449, 2017.

[34] Mengdi Wang, Ji Liu, and Ethan X Fang. Accelerating stochastic composition optimization. *arXiv preprint arXiv:1607.07329*, 2016.

[35] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

[36] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 43–51, 2019.

A Extension to Continuous Parameter Space

We briefly discuss the extension of our formulation to the case when the parameter spaces are continuous. We first revisit the optimization problems in discrete space in Eq (2) and Eq (3). We rewrite the equations in the generic inner-product form as:

(for the deterministic parameter in discrete space)

$$\begin{aligned}\max_{\mathbf{p}} \quad & f\left(\langle \boldsymbol{\mu}^X, \mathbf{p}\rangle, \langle \boldsymbol{\mu}^Y, \mathbf{p}\rangle\right) = f\left(\sum_{\beta\in\mathcal{B}} \mu^X(\beta)p(\beta), \sum_{\beta\in\mathcal{B}} \mu^Y(\beta)p(\beta)\right) \\ \text{s.t.} \quad & \mathbf{p} \in S := \left\{ \mathbf{p} \mid p(\beta) \in \{0,1\}, \forall \beta \in \mathcal{B}, \sum_{\beta\in\mathcal{B}} p(\beta) = 1 \right\}.\end{aligned}$$

(for the probabilistic parameter in discrete space)

$$\begin{aligned}\max_{\mathbf{p}} \quad & f\left(\langle \boldsymbol{\mu}^X, \mathbf{p}\rangle, \langle \boldsymbol{\mu}^Y, \mathbf{p}\rangle\right) = f\left(\sum_{\beta\in\mathcal{B}} \mu^X(\beta)p(\beta), \sum_{\beta\in\mathcal{B}} \mu^Y(\beta)p(\beta)\right) \\ \text{s.t.} \quad & \mathbf{p} \in \bar{S} := \left\{ \mathbf{p} \mid p(\beta) \in [0,1], \forall \beta \in \mathcal{B}, \sum_{\beta\in\mathcal{B}} p(\beta) = 1 \right\}\end{aligned}$$

In the above formulation for the discrete case, the inner product $\langle \boldsymbol{\mu}^X, \mathbf{p}\rangle$ is between vectors $\boldsymbol{\mu}$ and $\mathbf{p}$ and can be expressed as a summation over a finite number of items. When $\mathcal{B}$ is a continuous space, we can similarly define $\boldsymbol{\mu} = \mu(\beta)$ as a real-valued function over $\beta \in \mathcal{B}$ and $\mathbf{p} = p(\beta)$ as a PDF over $\mathcal{B}$. The above form of inner product still hold but should be interpreted as the inner product between two real-valued functions, that is the summation of an infinite number of items or integration, specifically,

(for the deterministic parameter in continuous space)

$$\begin{aligned}\max_{\mathbf{p}} \quad & f\left(\langle \boldsymbol{\mu}^X, \mathbf{p}\rangle, \langle \boldsymbol{\mu}^Y, \mathbf{p}\rangle\right) = f\left(\int_{\beta\in\mathcal{B}} \mu^X(\beta)p(\beta)d\beta, \int_{\beta\in\mathcal{B}} \mu^Y(\beta)p(\beta)d\beta\right) \\ \text{s.t.} \quad & \mathbf{p} \in S := \{p(\beta) \geq 0, \forall \beta \in \mathcal{B} \mid p(\beta) = \delta_{\beta_0}(\beta), \forall \beta_0 \in \mathcal{B}\}\end{aligned}$$

where $\delta_{\beta_0}(\cdot)$ is the standard delta function defining at $\beta_0 \in \mathcal{B}$ satisfying

$$\delta_{\beta_0}(\beta) = \begin{cases}\infty, \beta = \beta_0 \\ 0, \text{otherwise}\end{cases} \qquad \text{and} \qquad \int_{\beta\in\mathcal{B}} \delta_{\beta_0}(\beta)d\beta = 1$$

(for the probabilistic parameter in continuous space)

$$\begin{aligned}\max_{p} \quad & f\left(\langle \boldsymbol{\mu}^X, \mathbf{p}\rangle, \langle \boldsymbol{\mu}^Y, \mathbf{p}\rangle\right) = f\left(\int_{\beta\in\mathcal{B}} \mu^X(\beta)p(\beta)d\beta, \int_{\beta\in\mathcal{B}} \mu^Y(\beta)p(\beta)d\beta\right) \\ \text{s.t.} \quad & \mathbf{p} \in \bar{S} := \left\{ p(\beta) \geq 0, \beta \in \mathcal{B}, \mid \int_{\beta\in\mathcal{B}} p(\beta)d\beta = 1 \right\}.\end{aligned}$$

In sum, we would argue that the continuous space do share the same formulation as the discrete space. The gain of probabilistic parameters still holds in the continuous parameter space case.

B Proofs for Theorem 1 and Theorem 2

We first provide proof for Theorem 2 on the regret bound of our proposed `PASTO` in Algorithm 1 since it requires a more general condition. We then turn to the convergence bound in Theorem 1.

We begin by establishing a few technical lemmas.

Lemma 3. *In PASTO where $\boldsymbol{g}_t = \hat{V}_t^\top \nabla f(\hat{V}_t \mathbf{p}_t)$ (as in Eq (7)), we have,*

$$\mathbb{E}\left\|\boldsymbol{g}_t - \frac{\boldsymbol{d}}{\boldsymbol{dp}} f\left(\frac{1}{t}\sum_s \boldsymbol{\mu}_s \mathbf{p}_t\right)\right\|^2 \leq \frac{2\left(G_f^2 + G_U^2 L_f^2\right) G_U^2 \left(\sum_{s=1}^t \frac{1}{\epsilon_s}\right)}{t^2}.$$

Proof. Notice that we have $\mathbb{E}_t \hat{U}_t = \boldsymbol{\mu}_t$, which means

$$\begin{aligned}&\mathbb{E}\left\|\frac{1}{t}\sum_{s=1}^t \left(\boldsymbol{\mu}_s - \hat{U}_s\right)\right\|^2\\ =&\frac{1}{t^2}\sum_{s=1}^t \mathbb{E}\left\|\boldsymbol{\mu}_s - \hat{U}_s\right\|^2.\end{aligned}$$

In order to upper bound $\mathbb{E}\|\boldsymbol{\mu}_s - \hat{\boldsymbol{\mu}}_s\|^2$, we recall the technical conditions in Assumption 1

$$\mathbb{E}\left\|\boldsymbol{\mu}_s - \hat{U}_s\right\|^2 \leq \sum_{i=1}^K p_i(s)\left\|\frac{\hat{U}_s^{(i)} - \boldsymbol{\mu}_s^{(i)}}{p_i(s)}\right\|^2 \leq \sum_{i=1}^K \frac{\left\|\hat{U}_s^{(i)}\right\|^2 + \left\|\boldsymbol{\mu}_s^{(i)}\right\|^2}{p_i(s)} \leq 2\frac{G_U^2}{\epsilon_s},$$

where the last inequality is true since only one of the K terms in the summation is non-zero. Now, recall that $\widehat{V}_t := \frac{1}{t}\sum_{s=1}^t \hat{\boldsymbol{\mu}}_s$ and analogously let $V_t := \frac{1}{t}\sum_{s=1}^t \boldsymbol{\mu}_s$, the inequality above would leads to

$$\mathbb{E}\left\|\widehat{V}_t - V_t\right\|^2 \leq \frac{G_U^2\left(\sum_{s=1}^t \frac{1}{\epsilon_s}\right)}{t^2}.$$

With this , we can decompose and bound the difference of $\boldsymbol{g}_t$ and $\nabla f_t(\mathbf{p}_t)$ as

$$\begin{aligned}\mathbb{E}\left\|\boldsymbol{g}_t - \frac{\boldsymbol{d}}{\boldsymbol{dp}} f\left(\frac{1}{t}\sum_s \boldsymbol{\mu}_s \mathbf{p}_t\right)\right\|^2 =&\mathbb{E}\left\|\widehat{V}_t \nabla f\left(\widehat{V}_t \mathbf{p}_t\right) - V_t \nabla f(V_t \mathbf{p}_t)\right\|^2\\ =&\mathbb{E}\left\|\left(\widehat{V}_t - V_t\right)\nabla f\left(\widehat{V}_t \mathbf{p}_t\right) - V_t\left(\nabla f(V_t\mathbf{p}_t) - \nabla f\left(\widehat{V}_t\mathbf{p}_t\right)\right)\right\|^2\\ \leq&2\mathbb{E}\left\|\left(\widehat{V}_t - V_t\right)\nabla f\left(\widehat{V}_t \mathbf{p}_t\right)\right\|^2 + 2\mathbb{E}\left\|V_t\left(\nabla f(V_t\mathbf{p}_t) - \nabla f\left(\widehat{V}_t\mathbf{p}_t\right)\right)\right\|^2\\ \leq&2G_f^2\mathbb{E}\left\|\widehat{V}_t - V_t\right\|^2 + 2G_U^2\mathbb{E}\left\|\nabla f(V_t\mathbf{p}_t) - \nabla f\left(\widehat{V}_t\mathbf{p}_t\right)\right\|^2\\ \leq&2G_f^2\mathbb{E}\left\|\widehat{V}_t - V_t\right\|^2 + 2G_U^2 L_f^2\mathbb{E}\left\|V_t\mathbf{p}_t - \widehat{V}_t\mathbf{p}_t\right\|^2\\ \leq&2G_f^2\mathbb{E}\left\|\widehat{V}_t - V_t\right\|^2 + 2G_U^2 L_f^2\mathbb{E}\left\|V_t - \widehat{V}_t\right\|^2\\ \leq&\frac{2(G_f^2 + G_U^2 L_f^2)G_U^2\left(\sum_{s=1}^t \frac{1}{\epsilon_s}\right)}{t^2}.\end{aligned}$$

□

Lemma 4. *In PASTO algorithm, if $\boldsymbol{\mu}_t$ are sampled independently from the same distribution, i.e. $\mathbb{E}\boldsymbol{\mu}_t = \boldsymbol{\mu}$, $\forall t$, then we have*

$$\mathbb{E}\left\|\boldsymbol{g}_t - \nabla f(\mathbf{p}_t)\right\|^2 \leq \frac{2\left(G_f^2 + G_U^2 L_f^2\right) G_U^2\left(\sum_{s=1}^t \frac{1}{\epsilon_s}\right)}{t^2}.$$

Proof. (The result can also be obtained as a special case of Lemma 3. Notice that we have $\mathbb{E}_t \hat{\boldsymbol{\mu}}_t = \boldsymbol{\mu}$, which means

$$\begin{aligned}&\mathbb{E}\left\|\frac{1}{t}\sum_{s=1}^t (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_s)\right\|^2\\ =&\frac{1}{t^2}\sum_{s=1}^t \mathbb{E}\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_s\|^2.\end{aligned}$$

In order to upper bound $\mathbb{E}\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_s\|^2$, we have

$$\mathbb{E}\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_s\|^2 \leq \sum_{i=1}^{K} p_i(s)\left\|\frac{\hat{\boldsymbol{\mu}}_s^{(i)}}{p_i(s)}\right\|^2 = \sum_{i=1}^{K}\frac{\left\|\hat{\boldsymbol{\mu}}_s^{(i)}\right\|^2}{p_i(s)} \leq \frac{G_U^2}{\epsilon_s},$$

Denote $\widehat{V}_t := \frac{1}{t}\sum_{s=1}^{t}\hat{\boldsymbol{\mu}}_s$, the inequality above would leads to

$$\mathbb{E}\left\|\widehat{V}_t-\boldsymbol{\mu}\right\|^2 \leq \frac{G_U^2\left(\sum_{s=1}^{t}\frac{1}{\epsilon_s}\right)}{t^2}.$$

The difference between $\boldsymbol{g}_t$ and $\nabla f_t\left(\mathbf{p}_t\right)$ can be decomposed by

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{g}_t-\nabla f_t(\mathbf{p}_t)\|^2 =& \mathbb{E}\left\|\widehat{V}_t\nabla f\left(\widehat{V}_t\mathbf{p}_t\right)-\boldsymbol{\mu}\nabla f(\boldsymbol{\mu}\mathbf{p}_t)\right\|^2\\
=& \mathbb{E}\left\|\left(\widehat{V}_t-\boldsymbol{\mu}\right)\nabla f\left(\widehat{V}_t\mathbf{p}_t\right)-\boldsymbol{\mu}\left(\nabla f(\boldsymbol{\mu}\mathbf{p}_t)-\nabla f\left(\widehat{V}_t\mathbf{p}_t\right)\right)\right\|^2\\
\leq& 2\mathbb{E}\left\|(\widehat{V}_t-\boldsymbol{\mu})\nabla f(\widehat{V}_t\mathbf{p}_t)\right\|^2+2\mathbb{E}\left\|\boldsymbol{\mu}\left(\nabla f(\boldsymbol{\mu}\mathbf{p}_t)-\nabla f\left(\widehat{V}_t\mathbf{p}_t\right)\right)\right\|^2\\
\leq& 2G_f^2\mathbb{E}\left\|\widehat{V}_t-\boldsymbol{\mu}\right\|^2+2G_U^2\mathbb{E}\left\|\nabla f(\boldsymbol{\mu}\mathbf{p}_t)-\nabla f\left(\widehat{V}_t\mathbf{p}_t\right)\right\|^2\\
\leq& 2G_f^2\mathbb{E}\left\|\widehat{V}_t-\boldsymbol{\mu}\right\|^2+2G_U^2L_f^2\mathbb{E}\left\|\boldsymbol{\mu}\mathbf{p}_t-\widehat{V}_t\mathbf{p}_t\right\|^2\\
\leq& 2G_f^2\mathbb{E}\left\|\widehat{V}_t-\boldsymbol{\mu}\right\|^2+2G_U^2L_f^2\mathbb{E}\left\|\boldsymbol{\mu}-\widehat{V}_t\right\|^2\\
\leq& \frac{2\left(G_f^2+G_U^2L_f^2\right)G_U^2\left(\sum_{s=1}^{t}\frac{1}{\epsilon_s}\right)}{t^2},
\end{aligned}$$

completing the proof. □

Lemma 5. *For* `PASTO`*, for any probability distribution* $\mathbf{p}^*\in\bar{S}$ *in the* K*-dimensional probabilistic simplex, we have for any* t

$$\sum_{t=1}^{T}\mathbb{E}\langle\boldsymbol{g}_t,\mathbf{p}^*-\mathbf{p}_t\rangle \leq \frac{\ln K}{\gamma}+4\gamma(e-2)\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{g}_t\|^2+2\sum_{t=1}^{T}\epsilon_t\mathbb{E}\langle\boldsymbol{g}_t,\mathbf{p}_t\rangle.$$

Proof. Denote $W(t):=\sum_{k=1}^{K}\omega_k(t)$, we have

$$\begin{aligned}
\frac{W(t+1)}{W(t)} =& \sum_{k=1}^{K}\frac{\omega_k(t+1)}{W(t)}\\
=& \sum_{k=1}^{K}\frac{\omega_k(t)}{W(t)}\exp\left(\gamma g_k(t)\right)\\
=& \sum_{k=1}^{K}\frac{p_k(t)-\epsilon_t/K}{1-\epsilon_t}\exp\left(\gamma g_k(t)\right).
\end{aligned}$$

If $\gamma g_k(t) \leq 1$, then we get

$$\begin{aligned}
\frac{W(t+1)}{W(t)} \leq & \sum_{k=1}^{K} \frac{p_k(t) - \epsilon_t/K}{1-\epsilon_t} \left(1 + \gamma g_k(t) + (e-2)\gamma^2 \left(g_k(t)\right)^2\right) \\
= & 1 + \sum_{k=1}^{K} \frac{p_k(t) - \epsilon_t/K}{1-\epsilon_t} \left(\gamma g_k(t) + (e-2)\gamma^2 \left(g_k(t)\right)^2\right) \\
\leq & 1 + \frac{\gamma}{1-\epsilon_t} \sum_{k=1}^{K} p_k(t) g_k(t) + \frac{\gamma^2(e-2)}{1-\epsilon_t} \sum_{k=1}^{K} p_k(t) \left(g_k(t)\right)^2 \\
\leq & 1 + \frac{\gamma}{1-\epsilon_t} \sum_{k=1}^{K} p_k(t) g_k(t) + \frac{\gamma^2(e-2)}{1-\epsilon_t} \sum_{k=1}^{K} \left(g_k(t)\right)^2 \\
\leq & 1 + \gamma(1+4\epsilon_t) \sum_{k=1}^{K} p_k(t) g_k(t) + \gamma^2(e-2)(1+4\epsilon_t) \sum_{k=1}^{K} \left(g_k(t)\right)^2 \quad \left(\text{due to } \frac{1}{1-\epsilon_t} \leq 1 + 4\epsilon_t\right) \\
= & 1 + \gamma(1+4\epsilon_t) \langle \mathbf{p}_t, \boldsymbol{g}_t \rangle + \gamma^2(e-2)(1+4\epsilon_t) \sum_{k=1}^{K} \left\|g_k(t)\right\|^2 .
\end{aligned}$$

Therefore we have

$$\begin{aligned}
\ln\left(\frac{W(T+1)}{W(1)}\right) = & \sum_{t=1}^{T} \ln\left(\frac{W(t+1)}{W(t)}\right) \\
\leq & \sum_{t=1}^{T} \ln\left(1 + \gamma(1+4\epsilon_t) \langle \mathbf{p}_t, \boldsymbol{g}_t \rangle + \gamma^2(e-2)(1+4\epsilon_t) \sum_{k=1}^{K} \left\|g_k(t)\right\|^2\right) \\
\leq & \sum_{t=1}^{T} \left(\gamma(1+4\epsilon_t) \langle \mathbf{p}_t, \boldsymbol{g}_t \rangle + \gamma^2(e-2)(1+4\epsilon_t) \sum_{k=1}^{K} \left\|g_k(t)\right\|^2\right),
\end{aligned}$$

which gives us

$$\mathbb{E} \ln(W(T+1)) - \mathbb{E} \ln(W(1)) \leq \gamma \sum_{t=1}^{T} (1+4\epsilon_t) \mathbb{E} \langle \boldsymbol{g}_t, \mathbf{p}_t \rangle + \gamma^2(e-2) \sum_{t=1}^{T} (1+4\epsilon_t) \mathbb{E} \|\boldsymbol{g}_t\|^2. \tag{8}$$

For $W(T+1)$, with any probability distribution $\mathbf{p}^* = (p_1^*, \cdots, p_K^*)$, we have

$$\begin{aligned}
\ln(W(T+1)) = & \ln\left(\sum_k \omega_k(T+1)\right) \\
= & \sum_{j=1}^{K} p_j^* \ln\left(\sum_k \omega_k(T+1)\right) \\
\geq & \sum_{j=1}^{K} p_j^* \ln\left(\omega_j(t)\right) \\
= & \sum_{j=1}^{K} p_j^* \ln\left(\exp\left(\gamma \sum_{t=1}^{T} g_j(t)\right)\right) \\
= & \sum_{j=1}^{K} \sum_{t=1}^{T} \gamma p_j^* g_j(t),
\end{aligned}$$

then after taking expectation, we have

$$\mathbb{E} \ln(W(T+1)) \geq \mathbb{E}\left[\sum_{i=1}^{K} \sum_{t=1}^{T} \gamma p_i^* \mathbb{E}_t g_i(t)\right] = \gamma \sum_{t=1}^{T} \mathbb{E} \langle \boldsymbol{p}^*, \boldsymbol{g}_t \rangle. \tag{9}$$

Combing (8) and (9), we get

$$\gamma \sum_{t=1}^{T}(1+4\epsilon_t)\mathbb{E}\langle \boldsymbol{g}_t, \mathbf{p}_t\rangle \geq \gamma \sum_{t=1}^{T}\mathbb{E}\langle \boldsymbol{p}^*, \boldsymbol{g}_t\rangle - \ln K - \gamma^2(e-2)\sum_{t=1}^{T}(1+4\epsilon_t)\mathbb{E}\|\boldsymbol{g}_t\|^2.$$

After rearrangement, the inequality above leads to

$$\sum_{t=1}^{T}\mathbb{E}\langle \boldsymbol{g}_t, \mathbf{p}_t\rangle \geq \sum_{t=1}^{T}\mathbb{E}\langle \boldsymbol{p}^*, \boldsymbol{g}_t\rangle - \frac{\ln K}{\gamma} - 4\gamma(e-2)\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{g}_t\|^2 - 2\sum_{t=1}^{T}\epsilon_t\mathbb{E}\langle \boldsymbol{g}_t, \mathbf{p}_t\rangle.$$

Therefore, we get

$$\sum_{t=1}^{T}\mathbb{E}\langle \boldsymbol{g}_t, \mathbf{p}^* - \mathbf{p}_t\rangle \leq \frac{\ln K}{\gamma} + 4\gamma(e-2)\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{g}_t\|^2 + 2\sum_{t=1}^{T}\epsilon_t\mathbb{E}\langle \boldsymbol{g}_t, \mathbf{p}_t\rangle.$$

□

C Proof to Theorem 2

Proof. In this proof, we use $f_t(\mathbf{p})$ as a short notation for $f(\frac{1}{t}\sum_{s=1}^{t}\boldsymbol{\mu}_s\mathbf{p})$. Notice that $f_t(\mathbf{p}_t) - f_t(\mathbf{p}^*)$ can be bounded by $\langle \nabla f_t(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t\rangle$ by the convexity assumption. So we seek to upper bound $\sum_{t=1}^{T}\mathbb{E}\langle \nabla f_t(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t\rangle$ to proof Theorem 2. With Lemma 5, we get

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\langle \nabla f_t(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t\rangle \\
\leq &\frac{\ln K}{\gamma} + 4\gamma(e-2)\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{g}_t\|^2 + 2\sum_{t=1}^{T}\epsilon_t\mathbb{E}\langle \boldsymbol{g}_t, \mathbf{p}_t\rangle + \sum_{t=1}^{T}\mathbb{E}\langle \nabla f_t(\mathbf{p}_t) - \boldsymbol{g}_t, \mathbf{p}^* - \mathbf{p}_t\rangle \\
\leq &\frac{\ln K}{\gamma} + 4\gamma(e-2)\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{g}_t\|^2 + 2\sum_{t=1}^{T}\epsilon_t\mathbb{E}\langle \boldsymbol{g}_t, \mathbf{p}_t\rangle + \sum_{t=1}^{T}4\mathbb{E}\|\nabla f_t(\mathbf{p}_t) - \boldsymbol{g}_t\|^2.
\end{aligned}$$

The last inequality is true since $\langle a, b\rangle \leq \|a\|^2\|b\|^2$ and $\|\mathbf{p}^* - \mathbf{p}\| \leq 2$ for any probabilistic vectors $\mathbf{p} \in \bar{S}$. Now by using Lemma 3, we get

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\langle \nabla f_t(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t\rangle \\
\leq &\frac{\ln K}{\gamma} + 4\gamma(e-2)\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{g}_t\|^2 + 2\sum_{t=1}^{T}\epsilon_t\mathbb{E}\langle \boldsymbol{g}_t, \mathbf{p}_t\rangle + \sum_{t=1}^{T}\frac{8\left(G_f^2 + G_U^2L_f^2\right)G_U^2\left(\sum_{s=1}^{t}\frac{1}{\epsilon_s}\right)}{t^2} \\
\leq &\frac{\ln K}{\gamma} + 4\gamma(e-2)G^2T + 2G\sum_{t=1}^{T}\epsilon_t + \sum_{t=1}^{T}\frac{8\left(G_f^2 + G_U^2L_f^2\right)G_U^2\left(\sum_{s=1}^{t}\frac{1}{\epsilon_s}\right)}{t^2}.
\end{aligned}$$

Setting $\gamma = \frac{1}{\sqrt{T}}$ and $\epsilon_t = \frac{G}{\sqrt{t+1}}$, we can ensure that $\gamma g_i(t) = \frac{\gamma g_i(t)}{p_i(t)} \leq \frac{\gamma G}{\epsilon_t} \leq 1$, then the inequality above becomes

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\left(f_t(\mathbf{p}^*) - f_t(\mathbf{p}_t)\right) \\
\leq &\sum_{t=1}^{T}\mathbb{E}\langle \nabla f_t(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t\rangle \\
\leq &\sqrt{T}\left(\ln K + 4(e-1)G^2 + 4G^2 + \frac{32\left(G_f^2 + G_U^2L_f^2\right)G_U^2}{G}\right).
\end{aligned}$$

For the second part of Theorem 2, we first decompose the loss as

$$\begin{aligned}
&\sum_{t=1}^{T} f\left(\frac{1}{t}\sum_{s=1}^{t}\boldsymbol{\mu}_s\mathbf{p}^*\right) - \sum_{t=1}^{T}\mathbb{E}\left[f\left(\frac{1}{t}\sum_{s=1}^{t}\hat{U}_s\mathbf{p}_t\right)\right] \\
\leq &\sum_{t} f\left(\frac{1}{t}\sum_{s}\boldsymbol{\mu}_s\mathbf{p}^*\right) - \sum_{t}\mathbb{E}f\left(\frac{1}{t}\sum_{s}\boldsymbol{\mu}_s\mathbf{p}_t\right) + \sum_{t}\mathbb{E}f\left(\frac{1}{t}\sum_{s}\boldsymbol{\mu}_s\mathbf{p}_t\right) - \sum_{t=1}^{T}\mathbb{E}\left[f\left(\frac{1}{t}\sum_{s=1}^{t}\hat{U}_s\mathbf{p}_t\right)\right] \\
\leq &\sqrt{T}C_1 + \sum_{t}\mathbb{E}\left[f\left(\frac{1}{t}\sum_{s}\boldsymbol{\mu}_s\mathbf{p}_t\right) - f\left(\frac{1}{t}\sum_{s}\hat{U}_s\mathbf{p}_t\right)\right],
\end{aligned}$$

where the first term is upper bounded based on Theorem 2 and C_1 is the constant in the Theorem. To bound the second term, note that

$$\mathbb{E}f(V_t\mathbf{p}_t) - f(\hat{V}_t\mathbf{p}_t) \leq \mathbb{E}G\|\mathbf{p}_t\|^2\|V_t - \hat{V}_t\|^2 \leq GG_u^2\left(\sum_s \frac{1}{\epsilon_s}\right)/t^2$$

and recall that $\epsilon_t = \frac{G}{\sqrt{t+1}}$, we have

$$\begin{aligned}
&\sum_{t=1}^{T} f\left(\frac{1}{t}\sum_{s=1}^{t}\boldsymbol{\mu}_s\mathbf{p}^*\right) - \sum_{t=1}^{T}\mathbb{E}\left[f\left(\frac{1}{t}\sum_{s=1}^{t}\hat{U}_s\mathbf{p}_t\right)\right] \\
\leq &\sqrt{T}C_1 + \sum_{t=1}^{T} G_U^2\sqrt{t}/t^2 \\
\leq &\sqrt{T}\left(C_1 + 2G_U^2\right).
\end{aligned}$$

It completes the proof. □

D Proof to Theorem 1

Proof. Notice that $f(\mathbf{p}_t) - f_t(\mathbf{p}^*)$ can be bounded by $\langle\nabla f(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t\rangle$, so in order to upper bound $\sum_{t=1}^{T}\mathbb{E}\langle\nabla f(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t\rangle$, by using Lemma 5, we get

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\langle\nabla f(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t\rangle \\
\leq &\frac{\ln K}{\gamma} + 4\gamma(e-2)\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{g}_t\|^2 + 2\sum_{t=1}^{T}\epsilon_t\mathbb{E}\langle\boldsymbol{g}_t, \mathbf{p}_t\rangle + \sum_{t=1}^{T}\mathbb{E}\langle\nabla f(\mathbf{p}_t) - \boldsymbol{g}_t, \mathbf{p}^* - \mathbf{p}_t\rangle \\
\leq &\frac{\ln K}{\gamma} + 4\gamma(e-2)\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{g}_t\|^2 + 2\sum_{t=1}^{T}\epsilon_t\mathbb{E}\langle\boldsymbol{g}_t, \mathbf{p}_t\rangle + \sum_{t=1}^{T}4\mathbb{E}\|\nabla f(\mathbf{p}_t) - \boldsymbol{g}_t\|^2.
\end{aligned}$$

Now by using Lemma 4, we get

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\langle\nabla f(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t\rangle \\
\leq &\frac{\ln K}{\gamma} + 4\gamma(e-2)\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{g}_t\|^2 + 2\sum_{t=1}^{T}\epsilon_t\mathbb{E}\langle\boldsymbol{g}_t, \mathbf{p}_t\rangle + \sum_{t=1}^{T}\frac{8\left(G_f^2 + G_U^2L_f^2\right)G_U^2\left(\sum_{s=1}^{t}\frac{1}{\epsilon_s}\right)}{t^2} \\
\leq &\frac{\ln K}{\gamma} + 4\gamma(e-2)G^2T + 2G\sum_{t=1}^{T}\epsilon_t + \sum_{t=1}^{T}\frac{8\left(G_f^2 + G_U^2L_f^2\right)G_U^2\left(\sum_{s=1}^{t}\frac{1}{\epsilon_s}\right)}{t^2}.
\end{aligned}$$

Setting $\gamma = \frac{1}{\sqrt{T}}$ and $\epsilon_t = \frac{G}{\sqrt{t+1}}$, we can ensure that $\gamma g_i(t) = \frac{\gamma g_i(t)}{p_i(t)} \leq \frac{\gamma G}{\epsilon_t} \leq 1$, then the inequality above leads to

$$\begin{aligned}
& f(\mathbf{p}^*) - \mathbb{E} f\left(\frac{1}{T}\sum_{t=1}^{T} \mathbf{p}_t\right) \\
\leq & \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\langle \nabla f(\mathbf{p}_t), \mathbf{p}^* - \mathbf{p}_t \rangle \\
\leq & \frac{1}{\sqrt{T}}\left(\ln K + 4(e-1)G^2 + 4G^2 + \frac{32\left(G_f^2 + G_U^2 L_f^2\right) G_U^2}{G}\right).
\end{aligned}$$

It completes the proof. □