PROMPT OPTIMIZATION WITH LOGGED BANDIT DATA

Haruka Kiyohara, Yuta Saito, Daniel Yiming Cao, Thorsten Joachims
Computer Science Department, Cornell University
{ hk844, ys552, dyc33, tj36 } @cornell.edu

ABSTRACT

We study how to use naturally available user feedback, such as clicks, to optimize a prompt policy for generating sentences with large language models (LLMs). Naive approaches, including regression-based and importance sampling-based ones, suffer either from bias in the log data or variance caused by the large action space of prompts. To circumvent these challenges, we propose a way to leverage similarity and smoothness in the generated sentence embedding, substantially reducing variance in the policy gradient estimation while maintaining a small bias. Initial experiments on synthetic data demonstrate the effectiveness of our approach. We also plan to publish the benchmark and simulator as open-source software.

1 INTRODUCTION

Prompt tuning is a cost-effective way of optimizing language generation (e.g., personalizing user experiences in search and recommendation settings). As more systems with LLM-generated text are starting to become operational, we are naturally collecting increasing amounts of logged user feedback from their system interactions. These feedback signals provide valuable information on whether the prompt used for generating the sentence was effective for the user. Unlike conventional datasets used for RLHF (Stiennon et al., 2020), this feedback is available for all users at little cost, providing opportunities for personalization of LLMs. There is thus growing interest in using such naturally logged user feedback to optimize a *prompt policy (i.e., which prompt to use for a particular user or situation)* – to enhance the quality and outcome (i.e., reward) of language generation.

However, using logged user feedback for off-policy learning (OPL) of a new prompt policy entails several challenges due to the *partial* nature of the feedback. Specifically, the logged data is bandit feedback, containing the reward for only the action chosen by the *logging* policy (i.e., the one used in past operations) and not for the other actions that a new policy may choose. A naive way for dealing with such counterfactuals is to regress the reward and use imputed rewards instead (Stiennon et al., 2020; Jaques et al., 2017; Snell et al., 2022). However, imputation is often not accurate enough due to covariate shift (Swaminathan & Joachims, 2015). An alternative is the importance-sampling approach, which re-weighs reward observations to enable unbiased estimation of the policy gradient (PG) under support conditions. Nonetheless, this approach suffers from severe variance when the action space is large (Saito & Joachims, 2022) and bias when the logging policy does not fully explore the action space (Sachdeva et al., 2020). These challenges can be particularly problematic in our language generation setting, where the actions are prompts and we aim to support a rich and diverse set of candidate prompts.

The key shortcoming of the standard approaches lies in treating each prompt independently, equivalent to using a one-hot representation of prompts. In contrast, in many NLP tasks, we have witnessed advances from using embeddings and their similarities among words and sentences (Mikolov et al., 2013; Le & Mikolov, 2014). **This paper thus explores and presents a way of leveraging prompt- and sentence-similarities to make large-scale OPL for prompt-guided language generation tractable**. Specifically, to deal with a large action space, we use clustering based on similarity among prompts and generated sentences. Then, we optimize the prompt by the following steps: (1) apply cluster-wise importance sampling to estimate PG to choose a cluster, and (2) determine which prompt within the cluster to choose via a regression-based approach. By doing so, we can relax the conditions for (sufficient) support and reduce the variance in PG estimation, while keeping the bias small either when similar prompts are clustered together or when the within-cluster distribution shift

is small. Finally, we provide initial experiment results in a synthetic setting. We also plan to extend the benchmark with a full LLM setting and publicize it as open-source software for future work.

2 Off-Policy Learning for Prompt Optimization

We start by formulating prompt optimization as a new type of OPL problem, which we call *contextual bandits with auxiliary outputs*. Let $u \in \mathcal{U} \subseteq \mathbb{R}^{d_u}$ be d_u-dimensional user features, sampled from an unknown distribution $p(u)$. Let $q \in \mathcal{Q} \subseteq \mathbb{R}^{d_q}$ be a *query* (or input sentence of a frozen LLM), sampled from a conditional distribution $p(q|u)$. Let $a \in \mathcal{A}$ be a (discrete) *prompt*, where each prompt is associated with some vectorial embedding, i.e., soft prompts ($e_a \in \mathbb{R}^{d_e}$, where d_e is the dimension of the embeddings). The prompt is used to generate a sentence via a frozen LLM. This process can be formulated as a procedure of sampling sentence $o \in \mathcal{O}$ from the output distribution of the LLM: $p_{\text{LLM}}(o|q,a)$. A user will respond to the output sentence and provide some reward $r \in \mathbb{R}$ (e.g., click, stream, purchase), where r follows $p(r|u,q,o)$. Let $\pi \in \Pi$ be a *prompt policy* where $\pi(a|u,q)$ is the probability of choosing *prompt* a for *context* $x := (u,q)$. Our goal is to optimize the prompt policy to maximize the expected reward, defined as

$$V(\pi) := \mathbb{E}_{\underbrace{p(u)p(q|u)}_{=p(u,q)}\pi(a|u,q)\underbrace{p_{\text{LLM}}(o|q,a)p(r|u,q,o)}_{=p(r,o|u,q,a)}}[r] = \mathbb{E}_{p(x)\pi(a|x)p(r,o|x,a)}[r].$$

By formulating the interactions and the objective in this way, we frame prompt optimization as a contextual bandit with auxiliary outputs (o), where the context is x, the action (i.e., prompt) is a, and the reward is r, as illustrated in Figure 1. Note that we parameterize the policy as π_θ using some parameters $\theta \in \Theta$ (e.g., a neural network).

Running a prompt policy π_0 ($\neq \pi_\theta$) as part of an operational system, the *logging* policy π_0 generates logged feedback of the following form:

$$\mathcal{D} := \{x_i, a_i, o_i, r_i\}_{i=1}^n \sim \prod_{i=1}^n p(x)\pi_0(a|x)p_{\text{LLM}}(o|x,a)p(r|x,o),$$

where n is the data size and i is its index. The logged data informs us whether the prompt (a_i) results in a high reward or not (r_i). However, a difficult aspect of using the logged data is that the reward observation is *partial*, i.e., the reward is observed only for the prompt chosen by the logging policy (π_0) and it is not observed for all the other actions. This can be particularly challenging when training a new policy π_θ on the logged data, as π_θ may choose actions that are not chosen by π_0.

2.1 Naive Approaches to Off-Policy Learning of Prompt Policies

Regression-based. A typical way of using (logged) data is to train a reward predictor $\hat{R}$ (Stiennon et al., 2020; Jaques et al., 2017; Snell et al., 2022), and then to use the predicted rewards to estimate the policy gradient (PG) as follows[1].

$$\nabla_\theta V(\pi_\theta) \approx \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{a\sim\pi_\theta(a|x_i)}\left[\nabla_\theta \log \pi_\theta(a|x_i)\hat{R}(x_i,a)\right]$$

Minimizing the MSE loss $\mathcal{L}(\hat{R}) \approx \frac{1}{n}\sum_{i=1}^n(\hat{R}(x_i,a_i) - r_i)^2$ is a common choice for training the regression model $\hat{R}$. Unlike standard supervised learning, regression for OPL entails challenges arising from the partial rewards and from the covariate shift between logging policy π_0 and the target policy π_θ. If the learned regression model $\hat{R}$ is inaccurate, the estimated PG can be heavily biased.

Importance sampling (IS)-based. Instead of relying on potentially inaccurate regression, the IS-based approach corrects the distribution shift between π_0 and π_θ by reweighing the observations:

$$\nabla_\theta V(\pi_\theta) \approx \frac{1}{n}\sum_{i=1}^n \frac{\pi_\theta(a_i|x_i)}{\pi_0(a_i|x_i)}\nabla_\theta \log \pi_\theta(a_i|x_i) r_i$$

The IS-based PG is unbiased under the *support* condition, i.e., $\forall(x,a) \in \mathcal{X}\times\mathcal{A}$, $\pi_0(a|x) > 0 \rightarrow \pi_\theta(a|x) > 0$. However, because the number of candidate prompts ($|\mathcal{A}|$) is typically very large, the support condition may not be satisfied (Sachdeva et al., 2020), and the variance of PG estimation can be extremely high when the importance weight is very large (Saito et al., 2023).

[1]The estimation target is the *true* policy gradient: $\nabla_\theta V(\pi_\theta) = \mathbb{E}_{p(x)\pi_\theta(a|x)p(r,o|x,a)}[\nabla_\theta \log \pi_\theta(a|x) r]$.

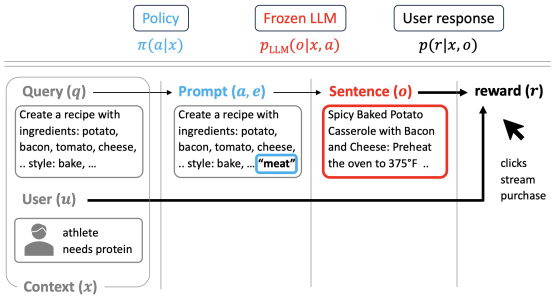

Figure 1: Data generation process.

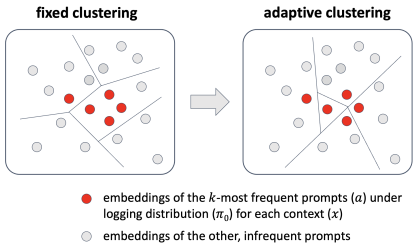

Figure 2: (Left) Fixed action clustering. (Right) Adaptive output clustering.

Key challenge. The key problem causing the issues identified above (especially those related to the IS-based method) lies in treating each prompt independently. This overlooks the potential of using word and sentence similarity, which has been proven effective in many NLP tasks (Mikolov et al., 2013; Le & Mikolov, 2014). Below, we thus discuss how to leverage the similarity among prompts and generated sentences to reduce the variance of PG estimation while preserving a small bias.

3 Proposal: Leveraging Sentence-similarity via Cluster-wise IS

Our solution for dealing with the large action spaces of LLMs is to introduce clustered action spaces in which we can apply importance sampling reliably. For this purpose, we decompose the policy as

$$\pi_\theta(a|x) = \mathbb{E}_{\mathcal{C}\sim p(\mathcal{C}|x)}[\pi_\phi^{\text{cluster}}(c|x,\mathcal{C})\pi_\psi^{\text{within-cluster}}(a|x,c,\mathcal{C})],$$

where $\theta = (\phi, \psi)$ are the policy parameters. $\mathcal{C}$ is the clustering structure and c indicates each cluster. Then, given some within-cluster policy $\pi_\psi^{\text{within-cluster}}$, the *true* policy gradient of the cluster policy $\pi_\phi^{\text{cluster}}$ is defined as

$$\nabla_\phi V(\pi_\theta) = \mathbb{E}_{p(x)p(\mathcal{C}|x)\pi_\phi^{\text{cluster}}(c|x,\mathcal{C})}[\nabla_\phi \log \pi_\phi(c|x,\mathcal{C}) R_{\pi_\psi^{\text{within-cluster}},\mathcal{C}}(x,c)],$$

where $R_{\pi_\psi^{\text{within-cluster}},\mathcal{C}}(x,c) := \mathbb{E}_{\pi_\psi^{\text{within-cluster}}(a|x,c,\mathcal{C})p(r,o|x,a)}[r]$ is the expected reward of choosing cluster c under the clustering structure $\mathcal{C}$ when deploying $\pi_\psi^{\text{within-cluster}}$ to choose an action a within the cluster c. To estimate the above PG, we use cluster-wise IS as follows:

$$\nabla_\phi V(\pi_\theta) \approx \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{p(\mathcal{C}|x)}\left[\frac{\pi_\phi(c(a_i)|x_i,\mathcal{C})}{\pi_0(c(a_i)|x_i,\mathcal{C})}\nabla_\phi \log \pi_\phi(c(a_i)|x_i,\mathcal{C}) r_i\right]$$

$\pi_0(c(a)|x,\mathcal{C}) := \sum_{a'\in\mathcal{A},c(a')=c(a)} \pi_0(a|x)$ is the cluster distribution under the logging policy. The cluster-wise IS corrects the distribution shift w.r.t. the *cluster policy*, while not w.r.t. the *within-cluster policy*, aiming to reduce the variance caused by large importance weights. As a result, the proposed cluster-IS has the following statistical properties:

- **Variance reduction**: Cluster-wise IS reduces the variance since there are typically fewer clusters than actions, and we can consider a low-variance regression policy for $\pi_\psi^{\text{within-cluster}}$.
- **Small bias**: Bias of cluster-wise IS can be small (1) when the within-cluster distribution shift is small or (2) when similar prompts that result in similar rewards are clustered together.

3.1 Clustering methods tailored for prompt tuning

The first idea for clustering prompts is via a k-means clustering of their soft-prompt embeddings (e_a). However, this "**fixed action clustering**" (FAC) comes with two potential shortcomings:

1. When π_0 concentrates on limited subset of prompts, cluster-wise IS still faces high variance by potentially producing clusters that are still infrequent under π_0. (Figure 2 (Left))
2. FAC may not capture which prompts produce similar sentences and thus similar rewards.

Figure 3: Comparing OPL methods with varying data sizes (left) and numbers of actions (right).

To mitigate these shortcomings, we propose a clustering method tailored for prompt optimization called "**adaptive output clustering**" (AOC). Specifically, AOC clusters prompts by the following steps: (1) retrieve k most frequent prompts under logging distribution for each context, (2) sample one "pivot sentence" for each of the k prompts from the LLM $p_{\text{LLM}}(o|x, a)$, and (3) form clusters by assigning sentences generated by other prompts to the most similar pivot sentence. By doing so, we can cluster prompts based on the similarity of associated sentence distribution, i.e., $p_{\text{LLM}}(o|x, a)$.

4 Synthetic Experiments

Synthetic setting. We conduct a synthetic experiment by simulating prompts and sentences with feature vectors. To generate logged data, we first sample 5-dimensional context ($d_u = 5$) and query ($d_q = 5$) from a multivariate normal distribution. Then, for any chosen prompt (or action) a, we sample auxiliary output o from the following distribution

$$f_o(q, e_a) = c \cdot \sin(q^\top M_q + e_a^\top M_e), \qquad o \sim \mathcal{N}(f_o(q, e_a), \sigma_o^2),$$

M_q and M_e are coefficient matrices sampled from a uniform distribution. e_a is the original embedding (i.e., soft prompt) of action a. We use $c = 5.0$ and $\sigma_o = 0.3$. By using the sine function, we simulate a situation where different action embeddings (e_a) result in similar auxiliary outputs (o). Then, a user responds to the sentence (o) with the following reward function.

$$f_r(u, o) = u^\top M_u + o^\top M_o, \qquad r \sim \mathcal{N}(f_r(u, o), \sigma_r^2)$$

where M_u and M_o are coefficient metrics sampled from a uniform distribution. $\sigma_r = 0.5$ is the reward noise. We generate logged data with the softmax logging policy $\pi_0(a|x) = \text{softmax}(\beta_0 \cdot \hat{R}_0(x, a))$, where $\beta_0 = \sqrt[3]{|\mathcal{A}|}$ is a temperature parameter. $\hat{R}_0$ is trained using $n_0 = 10000$ datapoints collected by the uniform random policy. In our experiments, we vary the following configurations: (1) **data size**: $n \in \{1000, 2000, 3000, 4000, \underline{\mathbf{5000}}\}$, (2) **number of candidate prompts**: $|\mathcal{A}| \in \{10, 50, \underline{\mathbf{100}}, 500, 1000\}$. The **bold** font represents the default value.

Baselines and metrics. We compare the proposed method to two baselines: regression-based PG and action-wise IS-based PG. These baselines directly parameterize $\hat{R}$ or π_θ without performing clustering. In contrast, our cluster-wise IS-based PG employs adaptive output clustering (AOC) by default and also uses fixed action clustering (FAC) for ablation. The number of clusters ($|\mathcal{C}|$) is 10. We compare the methods by their *optimality*, defined as $(V(\pi) - V(\pi_{\text{unif}}))/(V(\pi_{\text{opt}}) - V(\pi_{\text{unif}}))$, where π_{opt} is the optimal policy and π_{unif} is the uniform random policy.

Results. Figure 3 shows the optimality of each OPL method averaged over 20 random seeds. The results demonstrate that cluster-wise IS with AOC performs well across various configurations, while the baselines can fall short. Specifically, we observe the baseline methods lose performance as the number of candidate prompts ($|\mathcal{A}|$) increases. This is because the regression-based one fails to accurately estimate the reward of every prompt in $\mathcal{A}$ and the action-wise IS suffers from high variance in such situations. In contrast, the proposed cluster-wise IS with AOC mitigates these negative effects by effectively addressing the cluster-wise distribution shift via cluster-wise IS, while reducing the variance of action-wise IS. We hope our findings will stimulate the discussion of how to use naturally collected bandit data for efficient prompt optimization.

LIMITATION, SOCIAL IMPACT, AND FUTURE WORK

This is ongoing work, and the current limitation of this paper is that we have not yet included the experiment results with actual LLMs, while showing the proof of concept in the synthetic experiment. However, we believe that the idea of using logged data and formulating the problem as OPL of contextual bandits with auxiliary outputs provide fundamentally new opportunities for data-driven language generation to the LLM/NLP community. In particular, OPL for prompt tuning enables even users or third-party companies that do not own LLMs to optimize language generation using prompts and logged feedback. Compared to RLHF (Stiennon et al., 2020), which requires expensive human annotation and huge computational resources to fine-tune LLMs, OPL of prompt policies enables much more cost-effective and effortless optimization by using only naturally collected feedback and prompting.

For future work, we plan to conduct a benchmark experiment with Mistral-7B (Jiang et al., 2023) on a personalized movie description generation task, as well as extend experiments with various configurations on the synthetic setting. We also plan to publicize the benchmark as an open-source software, and we hope it will facilitate future research and practical applications.

REFERENCES

Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*, pp. 1645–1654. PMLR, 2017.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning*, pp. 1188–1196, 2014.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 2013.

Noveen Sachdeva, Yi Su, and Thorsten Joachims. Off-policy bandits with deficient support. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 965–975, 2020.

Yuta Saito and Thorsten Joachims. Off-policy evaluation for large action spaces via embeddings. In *Proceedings of the 39th International; Conference of Machine Learning*, volume 162, pp. 19089–19122, 2022.

Yuta Saito, Qingyang Ren, and Thorsten Joachims. Off-policy evaluation for large action spaces via conjunct effect modeling. In *Proceedings of the 40th International; Conference of Machine Learning*, volume 202, pp. 29734–29759, 2023.

Charlie Victor Snell, Ilya Kostrikov, Yi Su, Sherry Yang, and Sergey Levine. Offline rl for natural language generation with implicit language q learning. In *The 11th International Conference on Learning Representations*, 2022.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1):1731–1755, 2015.