NEURAL INFORMATION PROCESSING SYSTEMS

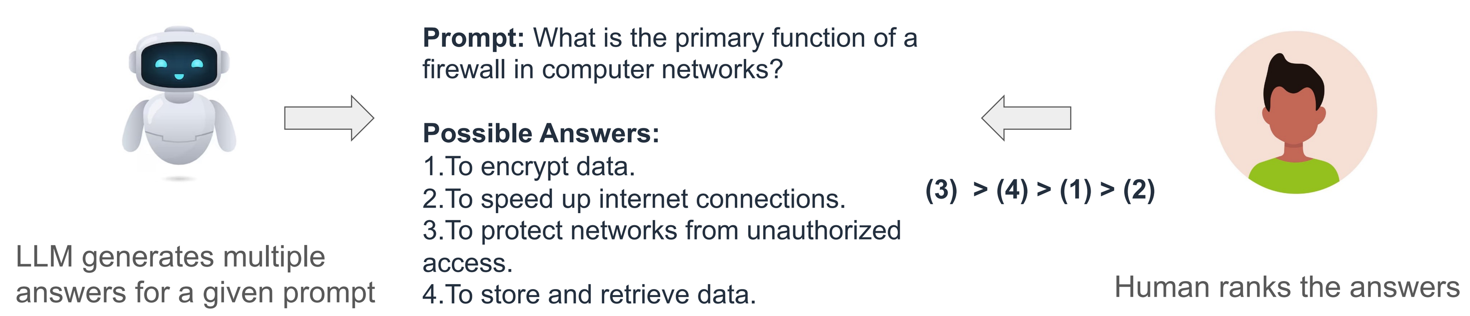
Optimal Design for Human Preference Elicitation

Subhojyoti Mukherjee*, Anusha Lalitha, Kousha Kalantari, Aniket Deshmukh, Ge Liu *, Yifei Ma, Branislav Kveton*
Amazon Web Services *Work Done at Amazon

amazon | science

Learning Preference Models

To effectively learn preference models, we study efficient methods for human preference elicitation



Given a set of L *prompts*, representing *questions*, each with K items representing candidate *answers*. The objective is to learn a preference model that can rank all answers for every prompt by querying humans for feedback.

Problem Setting

We study two models of human feedback, absolute and ranking:

- $\boldsymbol{\theta}_*$ is the unknown human-preference reward model parameter and $x_{i,k}$ is the feature vector for prompt i and candidate answer k
- **Absolute feedback model:** Human provides a reward for each prompt in list I_t chosen by the agent. Agent observes noisy rewards of the form:

$$y_{t,k} = \mathbf{x}_{I_t,k}^\top \boldsymbol{\theta}_* + \eta_{t,k}$$

- **Ranking feedback model:** Human orders all K candidates in prompt I_t selected by the agent. The feedback is a permutation $\sigma_t : [K] \to [K]$, where $\sigma_t(k)$ is the index of the k-th ranked candidate answer. Assume that the human preference follows ***Plackett-Luce Model (PL):***

$$p(\sigma_t) = \prod_{k=1}^{K} \frac{\exp[\mathbf{x}_{I_t,\sigma_t(k)}^\top \boldsymbol{\theta}_*]}{\sum_{j=k}^{K} \exp[\mathbf{x}_{I_t,\sigma_t(j)}^\top \boldsymbol{\theta}_*]}.$$

- Given human preference data, estimate the model parameter by solving a maximum likelihood problem, which is:
  - For absolute feedback, we use the OLS estimator
  - For ranking feedback, we solve the following:

$$\hat{\boldsymbol{\theta}}_n = \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \ell_n(\boldsymbol{\theta}), \quad \ell_n(\boldsymbol{\theta}) = -\frac{1}{n} \sum_{t=1}^{n} \sum_{k=1}^{K} \log \left(\frac{\exp[\mathbf{x}_{I_t,\sigma_t(k)}^\top \boldsymbol{\theta}]}{\sum_{j=k}^{K} \exp[\mathbf{x}_{I_t,\sigma_t(j)}^\top \boldsymbol{\theta}]} \right)$$

- How to collect data so that the solution is close to unknown θ_*?
- **Optimal Design:** Given n samples, how to allocate samples that can efficiently estimate the unknown θ_*?

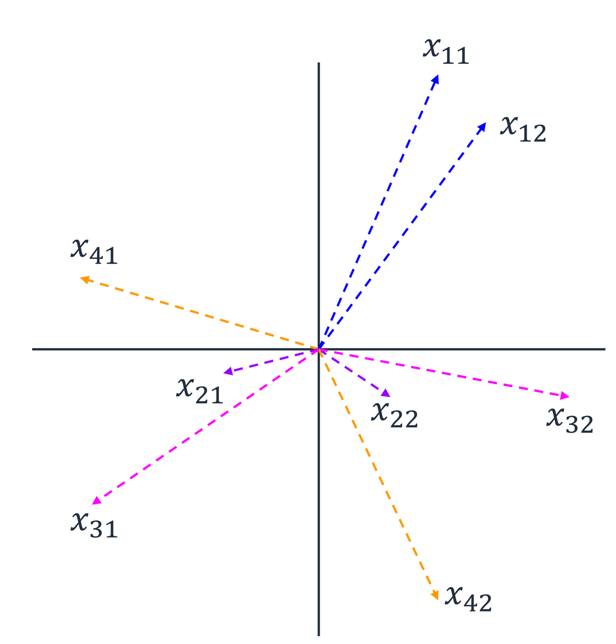
Learning Protocol: D-optimal design (Dope)

1. L questions with K candidate answers, indexed by integers
2. Ask questions $I_t \sim \pi^*$ according to an **optimal K-way ranking design π^***
3. Collect human feedback for n rounds
4. Learn the human-preference reward model

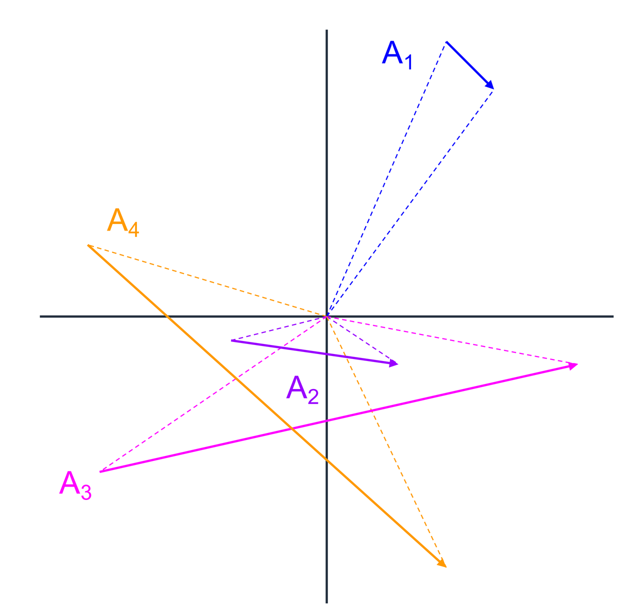
We show that D-optimal design reduces the uncertainty of the estimate $\hat{\boldsymbol{\theta}}_n$ maximally by generalizing the Kiefer-Wolfowitz theorem to matrices

Traditional Vs Matrix D-Optimal Design

- Optimize the data logging distribution π* over prompts
- $\min_{\pi} \max_{x_i \in \mathcal{X}} x_i^\top V_\pi^{-1} x_i \qquad V_\pi = \sum_{i \in \mathcal{X}} \pi_i x_i x_i^\top$
- Sample all prompts that have non-zero supp over x_i
- Traditional D-optimal design samples 1, 3, and 4



- Optimize the data logging distribution π* over prompts $\pi_* = \arg\min_{\pi \in \Delta^L} \max_{i \in [L]} \mathrm{tr}(A_i^\top \Sigma_\pi^{-1} A_i)$
- $\Sigma_\pi = \sum_{i=1}^{L} \pi(i) A_i A_i^\top$
- Absolute feedback: $\mathbf{A}_i = [\mathbf{x}_{i,k}]_{k \in [K]}$
- Ranking feedback: $\mathbf{A}_i = [\mathbf{x}_{i,j} - \mathbf{x}_{i,k}]_{(j,k) \in [K]^2 : j < k}$
- Equivalent to solving $\pi_* = \arg\max_{\pi \in \Delta^L} \log\det(\Sigma_\pi)$
- Optimal distribution is sparse
- Samples 2, 3, and 4



Matrix Kiefer-Wolfowitz

Theorem 1 (Matrix Kiefer-Wolfowitz). *Let $M \geq 1$ be an integer and $\mathbf{A}_1, \ldots, \mathbf{A}_L \in \mathbb{R}^{d \times M}$ be L matrices whose column space spans $\mathbb{R}^d$. Then the following claims are equivalent:*

(a) π_* is a minimizer of $g(\pi) = \max_{i \in [L]} \mathrm{tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i)$, where $\mathbf{V}_\pi = \sum_{i=1}^{L} \pi(i) \mathbf{A}_i \mathbf{A}_i^\top$.

(b) π_* is a maximizer of $f(\pi) = \log\det(\mathbf{V}_\pi)$.

(c) $g(\pi_*) = d$.

Furthermore, there exists a minimizer π_* of $g(\pi)$ such that $|\mathrm{supp}(\pi_*)| \leq d(d+1)/2$.

Prediction Error of Dope

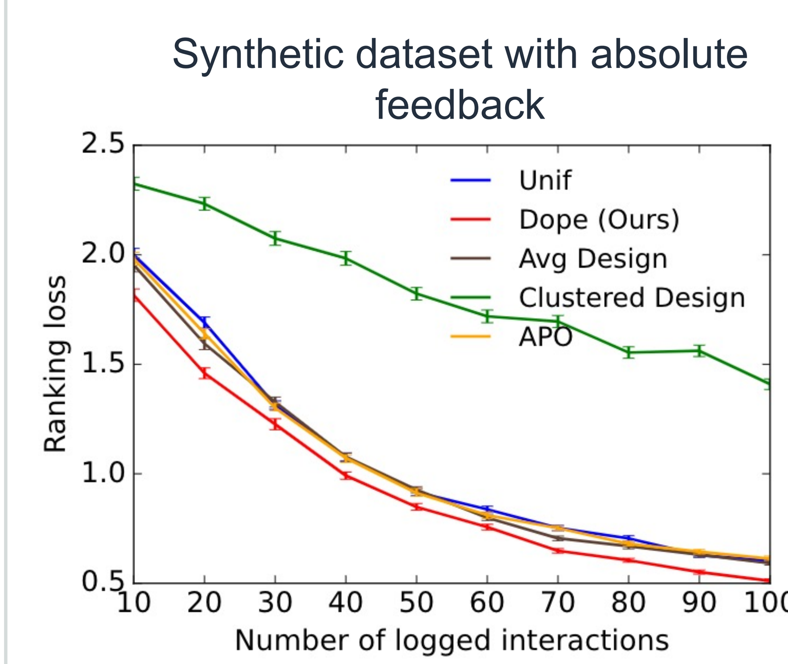
With probability at least 1-δ the prediction error of Dope under ranking feedback is

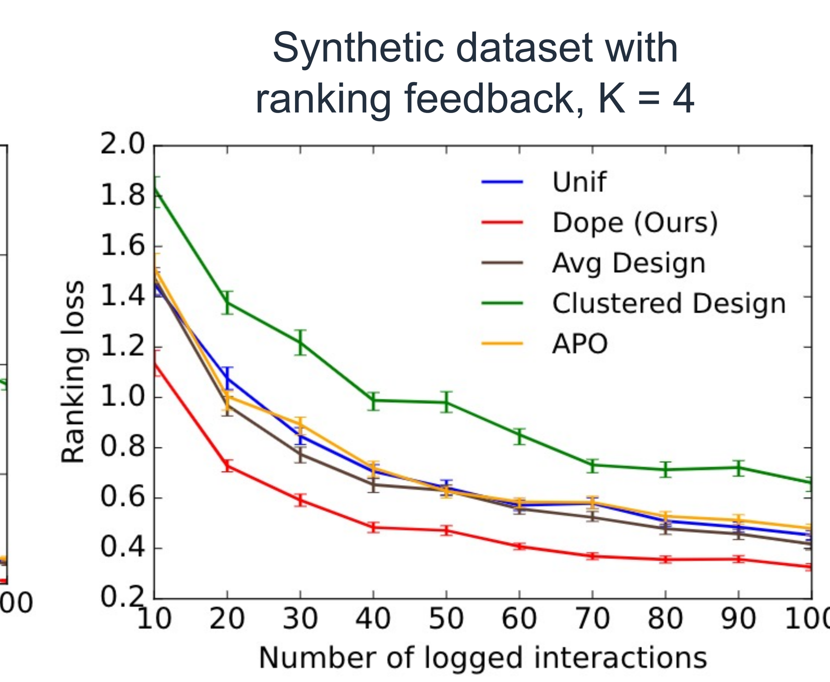
$$\max_{i \in [L]} \mathrm{tr}(\mathbf{A}_i^\top (\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*)^\top \mathbf{A}_i) = O\left(\frac{K^6(d^2 + d\log(1/\delta))}{n}\right)$$

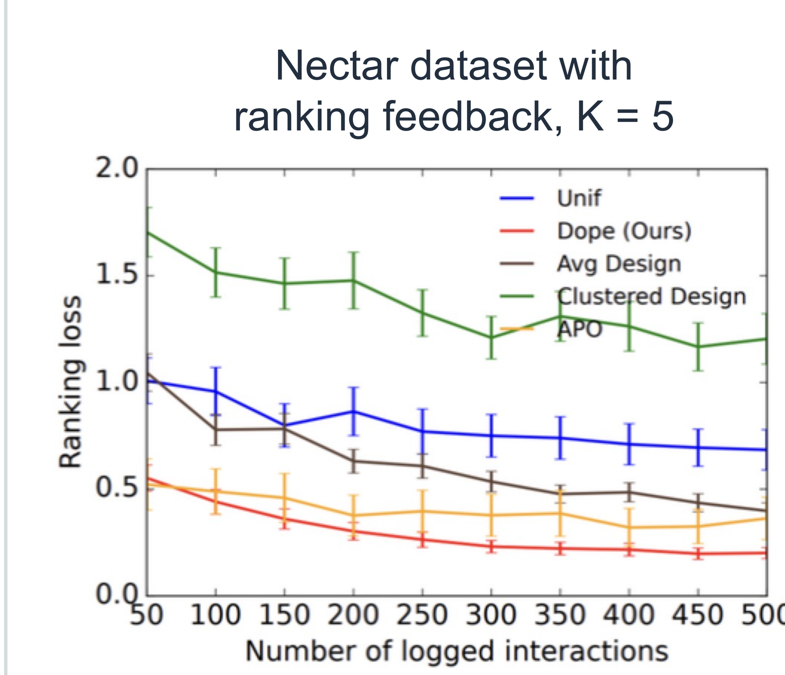
- The LHS is the maximum prediction error and controls the variance of the estimator
- The RHS decreases with the number of samples n
- The dependence on K can be further reduced by more careful analysis
- Prediction error of Dope under absolute feedback has similar form except there is no dependence on K

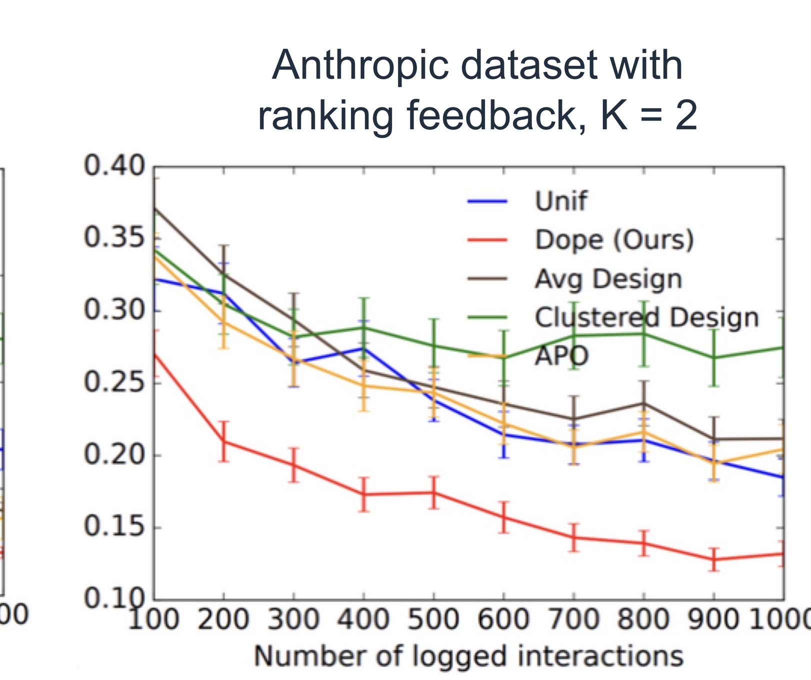
Experiments

- Evaluate the ranking loss defined as $\frac{1}{L} \sum_{i \in [L]} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \mathbb{1}\{\hat{\sigma}_{n,i}(j) > \hat{\sigma}_{n,i}(k)\}$
- We vary logged dataset size n and average over multiple random runs
- Compared methods: (i) Dope: Our proposed approach, (ii) *Unif:* Uniform sampling of lists, (iii) *Avg Design:* Lists are represented by average feature vectors over items, (iv) *Clustered Design:* Avg Design with k-means clustering, (v) *APO:* Dueling design for K = 2 that only focusses on uncertainty reduction
- For both question and answer 768-dim Instructor embedding is projected down and compute the feature vector $\mathbf{x}_{i,k} = \mathrm{vec}(\mathbf{q}_i \mathbf{a}_{i,k}^\top)$









*Corresponding author: anlalith@amazon.com