Learning Human Expertise in RLHF using Extended-BTL Model

Kaustubh & Kush
Under the guidance of
Prof. Arpit Agarwal
Department of Computer Science
IIT Bombay
Mumbai, India
{kaustubhshejole, kushm, aarpit}@cse.iitb.ac.in

Abstract

Human-AI alignment is done through Reinforcement Learning through Human Feedback (RLHF) where an AI Model e.g. an LLM is given human-annotated pairs containing a winning response and a losing response for an input. As the size of annotated data is usually large, it is not possible to get an expert opinion on every instance. Hence, crowdsourcing is used where given a pair, workers give their judgment over the preferred response. Human-AI alignment is very important considering fairness in LLMs and constitutes an important part of Responsible AI. However, it is observed that some workers doing annotation may not be well competent and thus may give wrong judgments affecting the quality of annotated data which further affects the RLHF process. Thus, it is essential to consider competencies of workers involved in crowd-sourcing for curating data for Human-LLM Alignment in RLHF. While models like Berry-Terry-Luce (BTL) deals with the calculation of rewards in pairwise comparisons, it misses the point of variable competencies of workers. In this work, we take into consideration the competencies of crowd-sourcing workers and propose an Extended-BTL model. We formulate an EM Algorithm for it having closed form updates for the Maximization Step using Pólya-Gamma Optimization. We also simulate the results on synthetic dataset showing the effectiveness of our model in judging the pairwise-comparisons using the learned rewards and competencies. The results using various settings demonstrate that using the Extended-BTL Model makes the decision on pairwise-comparisons much better than the original true pairwise comparisons.

1 Introduction

The rapid advancement of large language models (LLMs) has underscored the critical need for *human alignment*—ensuring model outputs adhere to human values, ethical standards, and preferences. Despite extensive pre-training and supervised fine-tuning, LLMs may still generate biased, toxic, or misaligned content. To address this, alignment techniques leverage human feedback to refine model behavior, with pairwise comparisons serving as a cornerstone for preference elicitation.

Pairwise comparison frameworks collect human judgments by presenting annotators with two model-generated responses for a given prompt and asking them to select the preferred (winning) response. This structured feedback provides a scalable method to distill subjective human preferences into actionable data. The *Bradley-Terry-Luce (BTL) model* Bradley and Terry (1952) formalizes this process by assigning latent rewards to responses and estimating the probability of one response being

Preprint. Under review.

preferred over another. Through maximum likelihood estimation (MLE), the BTL model infers these rewards, enabling the translation of human preferences into a differentiable optimization signal.

Modern alignment techniques, such as *Reinforcement Learning from Human Feedback (RLHF)* Christiano et al. (2017); Ziegler et al. (2019) and *Direct Preference Optimization (DPO)* Rafailov et al. (2023), build on this foundation. RLHF employs the BTL-derived reward function in a two-step process: first estimating rewards via MLE, then optimizing a policy using reinforcement learning with KL divergence regularization to prevent overfitting. DPO optimizes this by directly optimizing policy parameters to match human preferences, bypassing explicit reward modeling. These methods highlight the interplay between pairwise comparisons, statistical preference models like BTL, and alignment objectives—balancing reward maximization with distributional constraints to maintain coherence and diversity. These approaches ensure LLMs generate outputs that are not only high-quality but also socially aligned and fair.

Motivation

Given the extensive volume of data required, it is impractical to involve domain experts for every annotation instance. Consequently, crowdsourcing has become a prevalent approach, wherein annotators assess pairs of responses to determine the preferred one. While crowdsourcing offers scalability, it introduces variability in annotation quality due to differences in annotator expertise and reliability. Some annotators may lack the necessary competence, leading to erroneous judgments that can adversely affect the quality of the dataset and, subsequently, the RLHF process. This issue underscores the importance of considering annotator competencies in the data curation process for human-LLM alignment.

Traditional models like the Bradley-Terry-Luce (BTL) model are commonly employed to aggregate pairwise comparison data and estimate underlying preference (reward) scores. However, these models typically assume perfect annotator competence and do not account for variations in annotator competence. To address this limitation, extensions to the BTL model have been proposed.

Our work introduces an Extended-BTL model that explicitly models annotator competencies within the pairwise comparison framework. By integrating annotator reliability into the preference estimation process, the Extended-BTL model aims to enhance the quality of aggregated annotations, thereby improving the alignment of LLMs with human preferences.

The structure of the work is as follows: Section 2 discusses Related Work, Section 3 and 4 discusses the formulation of our work and methodology used, Section 5 details the experimentation done on synthetic dataset showing the efficacy of the proposed method, we conclude in Section 6 with some limitations and future works.

2 Related Work

Consider a dataset of pairwise comparisons $\{(w_i, l_i, x_i)\}_{i=1}^{M}$ where item w_i is known to have higher reward l_i. x_i may denote a prompt for which w_i and l_i are two responses. The standard BTL-Model Bradley and Terry (1952) formulates the probability as follows:

$$P(w_i > l_i) = \sigma((r_{w_i} - r_{l_i})) = \frac{1}{1 + e^{(r_{l_i} - r_{w_i})}} \tag{1}$$

with the constraint $\sum_i r_i = 0$. This work deals with the extension of this framework as discussed in the following sections.

Crowdsourcing has become a powerful paradigm for collecting large-scale labeled data from non-expert annotators. However, due to variability in annotator reliability and task difficulty, raw labels are often noisy and require aggregation to infer the true labels. The most basic and widely used approach is *majority voting*, where the label agreed upon by the majority of annotators is taken as the ground truth. Despite its simplicity, majority voting assumes all workers are equally reliable, which often fails in practice.

To address this limitation, probabilistic models such as the *Dawid-Skene* model Dawid and Skene (1979) were proposed. This model introduces latent variables to capture each annotator's confusion matrix, thereby modeling worker-specific reliability. In this work, we try to estimate rewards and competence for RLHF settings where data contains pairwise comparisons.This will help in better alignment of AI models.

3 Problem Formulation

Let us consider a dataset of pairwise comparisons $\{(w_i, l_i, x_i, k_i)\}_{i=1}^{M}$ where worker k_i prefers item w_i over l_i. x_i may denote a prompt for which w_i and l_i are two responses.

The BTL model as discussed in Section 2 lacks the inclusion of competencies of workers, so we propose the formulation for the probability that a worker k_i favours w_i over l_i as follows:

$$P(w_i > l_i \mid k_i) = \sigma(\beta_{k_i}(r_{w_i} - r_{l_i})) = \frac{1}{1 + e^{\beta_{k_i}(r_{l_i} - r_{w_i})}} \tag{2}$$

with the constraints $\sum_i r_i = 0$ and $\beta \in [0, 1]$, where $\sigma(\cdot)$ denotes the sigmoid function, $\mathbf{r} = \{r_j\}_{j=1}^{N}$ represent item reward parameters, and $\boldsymbol{\beta} = \{\beta_k\}_{k=1}^{K}$ model worker competencies. The learning objective is to estimate both $\mathbf{r}$ and $\boldsymbol{\beta}$ from the observed comparison data. The following section discusses the methodology used for estimating $\mathbf{r}$ and $\boldsymbol{\beta}$ from the observed comparison data.

4 Methodology

4.1 Maximum Likelihood Estimation

The likelihood function for the observed comparisons $\mathcal{D} = \{(w_i, l_i, k_i)\}_{i=1}^{M}$ is:

$$L(\mathbf{r}, \boldsymbol{\beta}) = \prod_{i=1}^{M} [\sigma(\beta_{k_i}(r_{w_i} - r_{l_i}))]^{z_i} [1 - \sigma(\beta_{k_i}(r_{w_i} - r_{l_i}))]^{1-z_i} \tag{3}$$

where $z_i = \mathbb{I}(w_i > l_i)$. The log-likelihood becomes:

$$\mathcal{L}(\mathbf{r}, \boldsymbol{\beta}) = \sum_{i=1}^{M} [z_i \log \sigma(\beta_{k_i}\Delta_i) + (1 - z_i) \log \sigma(-\beta_{k_i}\Delta_i)] \tag{4}$$

with $\Delta_i = r_{w_i} - r_{l_i}$. The MLE estimates $(\hat{\mathbf{r}}, \hat{\boldsymbol{\beta}})$ solve:

$$(\hat{\mathbf{r}}, \hat{\boldsymbol{\beta}}) = \arg\max_{\mathbf{r}, \boldsymbol{\beta}} \mathcal{L}(\mathbf{r}, \boldsymbol{\beta}) \quad \text{subject to} \quad \sum_{j=1}^{N} r_j = 0 \tag{5}$$

As we have to estimate both $\mathbf{r}$ and $\boldsymbol{\beta}$ from the observed comparison data, MLE is infeasible due to the computational complexity, hence we use Expectation-Maximization Dempster et al. (1977) for estimation.

4.2 Expectation-Maximization Algorithm

EM alternates between two steps:

1. E-step (Expectation): Compute the expected log-likelihood given current estimates.
2. M-step (Maximization): Update parameters to maximize this expectation.

4.2.1 Gradient-based EM

In gradient-based EM Lange (1995), we use gradients of log-likelihood to maximize the expectation and change the variables accordingly. The E-step is Gradient-EM involves directly computing the gradient of the log-likelihood:

$$\nabla_r \mathcal{L} \quad \text{and} \quad \nabla_\beta \mathcal{L}.$$

Using these gradients, we update the parameters as follows:

$$r \leftarrow r + \eta \nabla_r \mathcal{L}, \quad \beta \leftarrow \beta + \eta \nabla_\beta \mathcal{L}.$$

maintaining the contraints on $\mathbf{r}$ and $\boldsymbol{\beta}$. The major disadvantage of this method is that we do not have any closed-form step for M-update.

4.2.2 Pólya-Gamma Optimized EM

To get closed-form steps for M-update and guarantees for EM convergence, we used an optimization based upon Pólya-Gamma distribution. This method was given in Polson et al. (2013) for Logistic Models Using Pólya–Gamma Latent Variables. Here, $\omega_i \sim \mathrm{PG}(1, 0)$ is introduced and the augmented likelihood becomes:

$$P(w_i > l_i \mid \omega_i, \beta_{k_i}, r_{w_i}, r_{l_i}) \propto \exp\left(-\frac{\omega_i}{2}\left(\beta_{k_i}(r_{w_i} - r_{l_i}) - c_i\right)^2\right)$$

where $c_i = \frac{1}{\beta_{k_i}}\left(z_i - \frac{1}{2}\right)$, $z_i = \mathbb{I}(w_i > l_i)$.

The expectation-step is then given by:

$$\mathbb{E}[\omega_i^{(t)}] = \frac{\tanh\left(\frac{\beta_{k_i}^{(t)}(r_{w_i}^{(t)} - r_{l_i}^{(t)})}{2}\right)}{2\beta_{k_i}^{(t)}(r_{w_i}^{(t)} - r_{l_i}^{(t)})}$$

For maximization step, Pólya-Gamma offers closed-form updates. The updates for Item Rewards Update are given by:

$$r_j^{(t+1)} = \frac{\sum_{i:j\in\{w_i,l_i\}} \mathbb{E}[\omega_i^{(t)}]\beta_{k_i}^{(t)}\Delta_i^{(t)}}{\sum_{i:j\in\{w_i,l_i\}} \mathbb{E}[\omega_i^{(t)}](\beta_{k_i}^{(t)})^2}$$

Worker Competencies Update are given by:

$$\beta_k^{(t+1)} = \frac{\sum_{i\in W(k)} \mathbb{E}[\omega_i^{(t)}]\Delta_i^{(t)}}{\sum_{i\in W(k)} \mathbb{E}[\omega_i^{(t)}](\Delta_i^{(t)})^2}$$

As there exist closed-form updates, we used this model for estimating rewards and competencies from the observed data. The details of Pólya-Gamma Optimization used is given in Appendix 6.

5 Experimental Results

In our synthetic data generation process, we simulate pairwise comparisons under a controlled setting to evaluate learning algorithms. We first generate true latent rewards $\mathbf{r} \in [0, 1]^N$, worker reliabilities $\boldsymbol{\beta} \in [0, 1]^K$, and using them generate M pairwise comparison samples among N items by K annotators. For each sample, a worker k is randomly selected, and two distinct items a and b are drawn and the probability that worker k prefers item a over item b is modeled as given in Equation 2. The comparison outcome is then sampled based on this probability, reflecting both the inherent difficulty of distinguishing items and the worker's competence. This method allows us to emulate realistic noisy preference data, capturing both item rewards and worker's competence.

After the generation of the synthetic data, we run an EM algorithm given in Section 4.2.2. We compare the results by varying N i.e. the number of items, K i.e. the number of workers and m i.e. the number of measurements. After this, we use the estimated rewards to get the winning response from the pair and we then calculate the accuracy considering the true rewards, we compare this method with the original comparisons done by the workers. The quantitative results are shown in Table 1 and Figure 1 shows the comparison using line graphs.

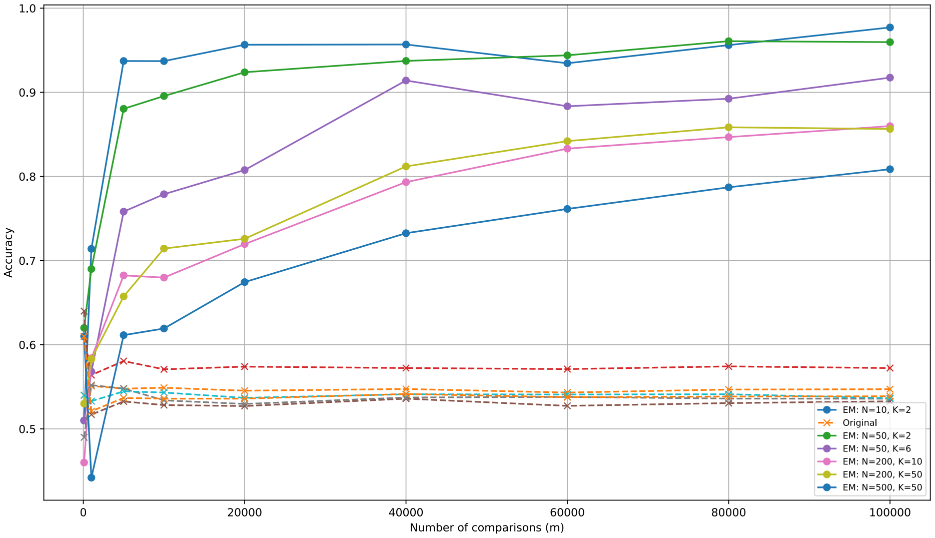

Figure 1: Performance of Extended-BTL Model in various (N, K, m) settings where N denotes the number of items, K denotes the number of workers and m denotes the number of comparisons. Solid lines denotes the performance by extended-BTL Model whereas dashed lines denotes the performance using the original pairwise comparisons.

We can see that the proposed method is working much better than blindly trusting on the worker's opinion. Table 1 and Figure 1 clearly shows the efficacy of our model where the accuracy is seen to be increasing as the number of comparisons increases, and for the estimation of larger number of items and workers, higher number of comparisons are required to get better accuracy, this matches with the normal intuition of larger data for larger number of parameter estimation.

N	**K**	**M**																	
		100		**1000**		**5000**		**10000**		**20000**		**40000**		**60000**		**80000**		**100000**	
		E-BTL	**base**	**E-BTL**	**base**	**E-BTL**	**base**	**E-BTL**	**base**	**E-BTL**	**base**	**E-BTL**	**base**	**E-BTL**	**base**	**E-BTL**	**base**	**E-BTL**	**base**
10	2	0.51	**0.54**	**0.71**	0.55	**0.94**	0.55	**0.94**	0.55	**0.96**	0.55	**0.96**	0.55	**0.93**	0.54	**0.96**	0.55	**0.98**	0.55
50	2	**0.62**	0.61	**0.69**	0.56	**0.88**	0.58	**0.90**	0.57	**0.92**	0.57	**0.94**	0.57	**0.94**	0.57	**0.96**	0.57	**0.96**	0.57
50	6	0.51	**0.64**	**0.57**	0.52	**0.76**	0.53	**0.78**	0.53	**0.81**	0.53	**0.91**	0.54	**0.88**	0.53	**0.89**	0.53	**0.92**	0.53
200	10	0.46	**0.49**	**0.58**	0.55	**0.68**	0.55	**0.68**	0.53	**0.72**	0.53	**0.79**	0.54	**0.83**	0.54	**0.85**	0.54	**0.86**	0.54
200	50	0.53	**0.54**	**0.58**	0.53	**0.66**	0.54	**0.71**	0.54	**0.73**	0.54	**0.81**	0.54	**0.84**	0.54	**0.86**	0.54	**0.86**	0.54
500	50	**0.61**	**0.61**	0.44	**0.52**	**0.61**	0.54	**0.62**	0.54	**0.67**	0.54	**0.73**	0.54	**0.76**	0.54	**0.79**	0.54	**0.81**	0.54

Table 1: Accuracy (E-BTL (ours) vs base (original)) across different numbers of comparisons (M)

Figure 2 shows the Mean Squared Error for estimated rewards, we can see that it is about 0.08 and Figure 4 shows the absolute error in prediction of rewards, the error is 0.25 which suggests that error of predicting rewards is less.

Figure 3 shows the Mean Squared Error for estimated competencies, we can see that it is about 0.29 and Figure 5 shows the absolute error in prediction of competencies, the error is 0.4 which suggests that error of predicting competencies is a little higher, we think that it may be because of the clipping step we do while updating beta values. We will conduct further investigation in this regard in the future.

6 Conclusion

In this work, we propose Pólya-Gamma based optimized EM Algorithm for learning human-expertise in pairwise comparison tasks commnonly found in RLHF settings. The use of Pólya-Gamma

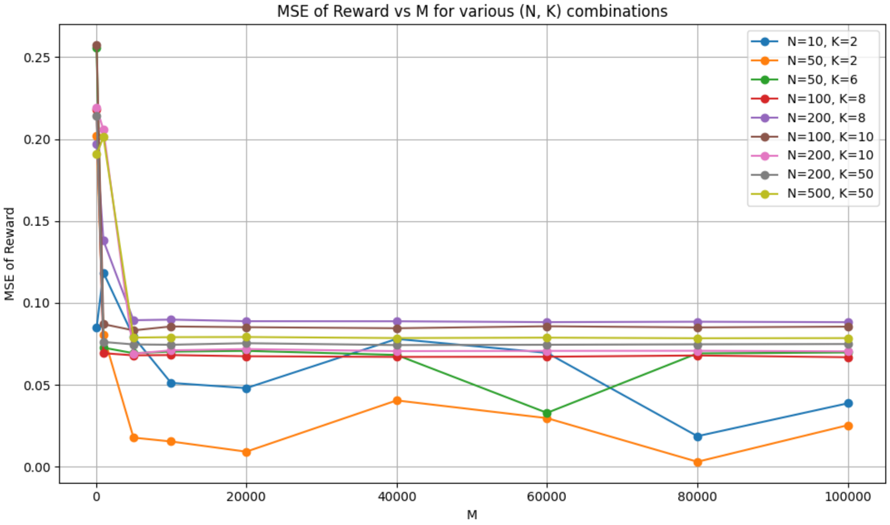

Figure 2: Analysis of MSE difference for rewards in various (N,K) and M settings

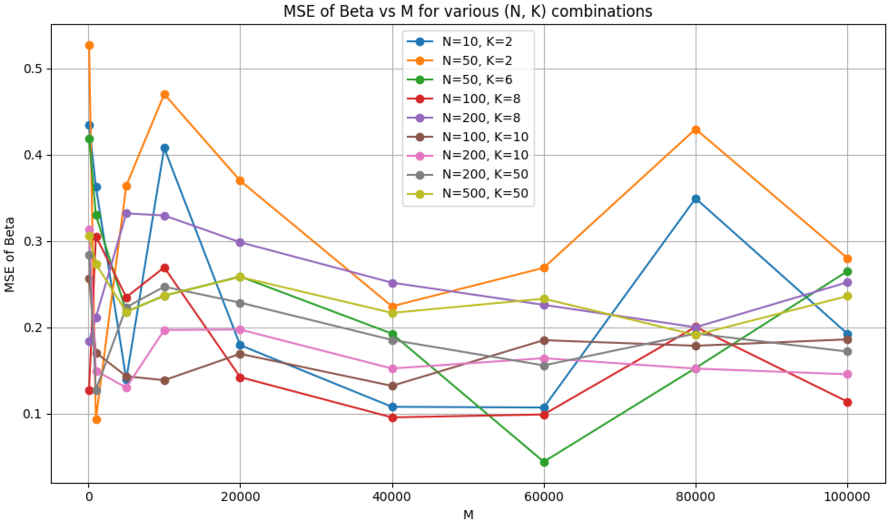

Figure 3: Analysis of MSE difference for competencies in various (N,K) and M settings

based optimization helps in closed-form M-step updates. The results demonstrate that the proposed methodology performs much better (about 20% to 40% depending on the values of N, K and m) than blindly trusting on the worker's opinion. We found that for small number of comparisons (i.e. 100-200) the original judgements are typically more reliable than the proposed method, but for larger M values (commonly found in RLHF values), the proposed methodology works much better.

Limitations and Future Work

Currently, we have done experiments on synthetic dataset, but in the future we will try to use some publicly available real crow-sourced data of pairwise comparisons for RLHF. This will help us to validate our claims on real datasets.

We will investigate the effect of worker's competence by removing the pairwise comparisons of workers having competence less than τ, where τ represent some threshold value. A comparison will be need between original aligned model and a model aligned on those comparisons which were done by workers having competence value greater than τ. This will validate the claim that the proposed

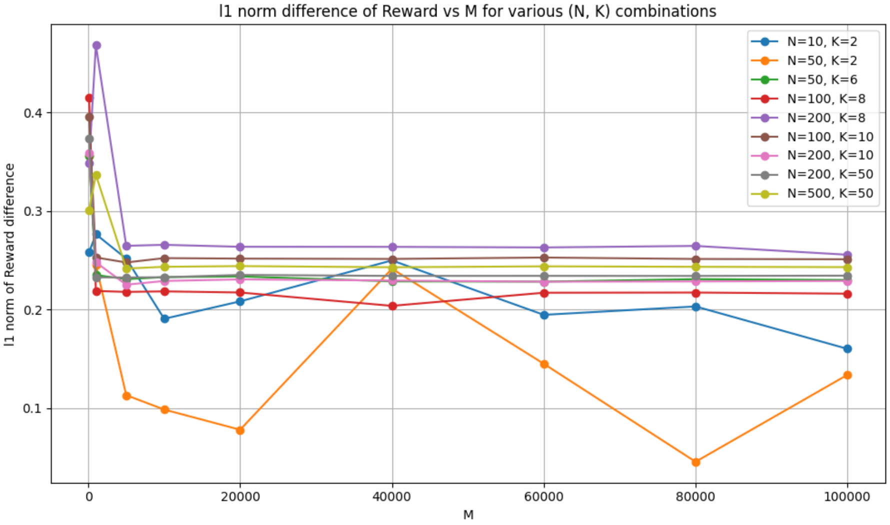

Figure 4: Analysis of absolute error (i.e. l_1 norm) for rewards in various (N,K) and M settings

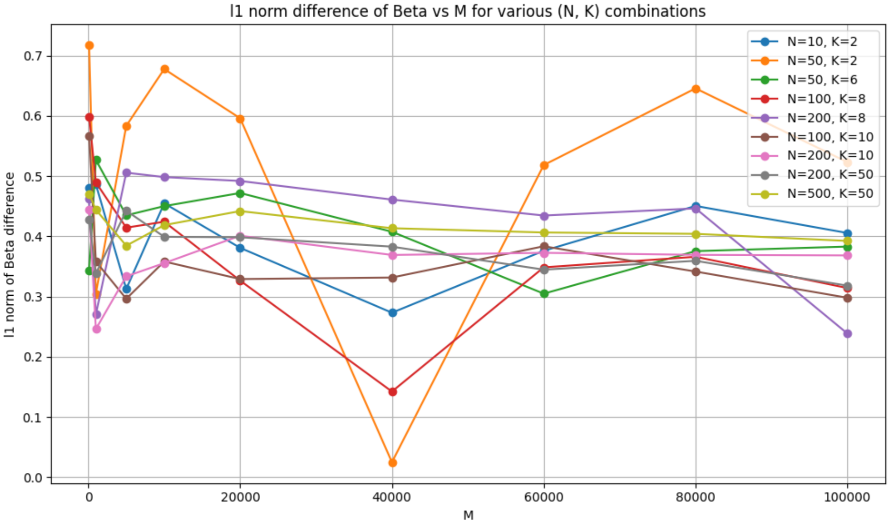

Figure 5: Analysis of absolute error (i.e. l_1 norm) for competencies in various (N,K) and M settings

method finds correct competence values. Inclusion of a real dataset can further help in testing the hypothesis.

References

Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22.

Lange, K. (1995). A gradient algorithm locally equivalent to the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(2):425–437.

Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

Appendix: EM Derivation with Polya-Gamma Augmentation

Problem Formulation

We observe M pairwise comparisons $\{(w_i, l_i, k_i)\}_{i=1}^{M}$, where worker k_i prefers item w_i over l_i. Our goal is to estimate:

- Item rewards $\mathbf{r} = (r_1, ..., r_N)^\top \in \mathbb{R}^N$
- Worker competencies $\boldsymbol{\beta} = (\beta_1, ..., \beta_K)^\top \in [0, 1)^K$

The likelihood follows the Bradley-Terry model with worker reliability:

$$P(w_i > l_i \mid k_i) = \sigma\left(\beta_{k_i}(r_{w_i} - r_{l_i})\right) = \frac{1}{1 + e^{-\beta_{k_i}\Delta_i}} \tag{6}$$

where $\Delta_i = r_{w_i} - r_{l_i}$ and $\sigma(\cdot)$ is the logistic function.

Polya-Gamma Augmentation

The logistic likelihood $P(w_i > l_i \mid k_i)$ is non-linear in β_k and r_j, making direct optimization difficult.

Solution: Introduce a latent variable $\omega_i \sim \text{PG}(1, 0)$ (Polya-Gamma distribution) to linearize the likelihood.

Polya-Gamma Identity Polson et al. (2013)

For any $x \in \mathbb{R}$,

$$\frac{1}{1 + e^{-x}} = \frac{1}{2}\int_0^\infty e^{-\frac{\omega x^2}{2}} p(\omega) d\omega,$$

where $p(\omega)$ is the density of $\text{PG}(1, 0)$.

Augmented Likelihood

Using $x = \beta_{k_i}(r_{w_i} - r_{l_i})$, rewrite the likelihood with ω_i:

$$P(w_i > l_i \mid \beta_{k_i}, r_{w_i}, r_{l_i}, \omega_i) \propto e^{\frac{1}{2}\beta_{k_i}(r_{w_i} - r_{l_i})} \cdot e^{-\frac{\omega_i}{2}\left(\beta_{k_i}(r_{w_i} - r_{l_i})\right)^2}$$

This transforms the logistic function into a Gaussian kernel:

$$P(w_i > l_i \mid \cdots) \propto \mathcal{N}\left(\beta_{k_i}(r_{w_i} - r_{l_i}) \mid \frac{1}{2\omega_i}, \frac{1}{\omega_i}\right)$$

E-Step: Compute $\mathbb{E}[\omega_i]$

Given current parameters $\beta_{k_i}^{(t)}, r_j^{(t)}$, compute the expectation of ω_i:

$$\mathbb{E}[\omega_i^{(t)}] = \frac{\tanh\left(\frac{\beta_{k_i}^{(t)}(r_{w_i}^{(t)} - r_{l_i}^{(t)})}{2}\right)}{2\beta_{k_i}^{(t)}(r_{w_i}^{(t)} - r_{l_i}^{(t)})}$$

Derivation

From the Polya-Gamma expectation formula:

$$\mathbb{E}[\omega \mid x] = \frac{\tanh(x/2)}{2x}.$$

Here, $x = \beta_{k_i}(r_{w_i} - r_{l_i})$.

M-Step: Update Item Rewards r_j

Maximize the expected log-likelihood Q:

$$Q = \sum_{i=1}^{M} \left[\frac{1}{2}\beta_{k_i}(r_{w_i} - r_{l_i}) - \frac{\mathbb{E}[\omega_i]}{2} \left(\beta_{k_i}(r_{w_i} - r_{l_i})\right)^2 \right]$$

Step 1: Collect Terms Involving r_j

For item j, r_j appears in comparisons where j is the winner ($w_i = j$) or loser ($l_i = j$):

$$Q_j = \sum_{i:w_i=j} \left[\frac{1}{2}\beta_{k_i}(r_j - r_{l_i}) - \frac{\mathbb{E}[\omega_i]}{2}\beta_{k_i}^2(r_j - r_{l_i})^2 \right] + \sum_{i:l_i=j} \left[\frac{1}{2}\beta_{k_i}(r_{w_i} - r_j) - \frac{\mathbb{E}[\omega_i]}{2}\beta_{k_i}^2(r_{w_i} - r_j)^2 \right]$$

Step 2: Take Derivative w.r.t r_j

$$\frac{\partial Q_j}{\partial r_j} = \sum_{i:w_i=j} \left[\frac{1}{2}\beta_{k_i} - \mathbb{E}[\omega_i]\beta_{k_i}^2(r_j - r_{l_i}) \right] - \sum_{i:l_i=j} \left[\frac{1}{2}\beta_{k_i} - \mathbb{E}[\omega_i]\beta_{k_i}^2(r_{w_i} - r_j) \right]$$

Step 3: Set Derivative to Zero

Solve $\frac{\partial Q_j}{\partial r_j} = 0$:

$$\sum_{i:w_i=j} \frac{1}{2}\beta_{k_i} - \sum_{i:l_i=j} \frac{1}{2}\beta_{k_i} = \sum_{i:w_i=j} \mathbb{E}[\omega_i]\beta_{k_i}^2(r_j - r_{l_i}) + \sum_{i:l_i=j} \mathbb{E}[\omega_i]\beta_{k_i}^2(r_{w_i} - r_j)$$

Step 4: Rearrange for r_j

$$r_j = \frac{\text{Term1} + \text{Sum1}}{\text{Sum2}},$$

where:

$$\text{Term1} : \frac{1}{2}\left(\sum_{i:w_i=j} \beta_{k_i} - \sum_{i:l_i=j} \beta_{k_i} \right)$$

$$\text{Sum1} : \sum_{i:l_i=j} \mathbb{E}[\omega_i]\beta_{k_i}^2 r_{w_i} + \sum_{i:w_i=j} \mathbb{E}[\omega_i]\beta_{k_i}^2 r_{l_i}$$

$$\text{Sum2} : \sum_{i:l_i=j} \mathbb{E}[\omega_i]\beta_{k_i}^2 + \sum_{i:w_i=j} \mathbb{E}[\omega_i]\beta_{k_i}^2$$

M-Step: Update Worker Competencies β_k

Step 1: Collect Terms Involving β_k

For worker k, β_k appears in all comparisons i where $k_i = k$:

$$Q_k = \sum_{i \in W(k)} \left[\frac{1}{2} \beta_k (r_{w_i} - r_{l_i}) - \frac{\mathbb{E}[\omega_i]}{2} \beta_k^2 (r_{w_i} - r_{l_i})^2 \right]$$

Step 2: Take Derivative w.r.t β_k

$$\frac{\partial Q_k}{\partial \beta_k} = \sum_{i \in W(k)} \left[\frac{1}{2} (r_{w_i} - r_{l_i}) - \mathbb{E}[\omega_i] \beta_k (r_{w_i} - r_{l_i})^2 \right]$$

Step 3: Set Derivative to Zero

$$\beta_k = \frac{\sum_{i \in W(k)} \frac{1}{2} (r_{w_i} - r_{l_i})}{\sum_{i \in W(k)} \mathbb{E}[\omega_i] (r_{w_i} - r_{l_i})^2}$$

In this way, the closed-form updates for M-Step in Polya-Gamma Optimized EM are derived.