Experimental Design: Testing DPL on Real-World Data

Allegra Laro

2025-01-02

1 Identified Failure Mode

Standard preference learning assumes a single latent utility function shared across annotators, with noise that is i.i.d. and homoscedastic. When annotators differ systematically in hidden context, such as risk tolerance or safety thresholds or even their core values, this assumption is violated. Preference noise may become heteroskedastic or multimodal: some options may generate consistent agreement, while others may induce sharp disagreement depending on annotator type. Standard preference learning will collapse this disagreement into a single mean utility estimate, whereas mean-variance distributional preference learning (DPL), because it explicitly models per-option utility variance, allows the learner to represent disagreement rather than average it away. This should improve robustness under hidden context heterogeneity.

Mathematically, we can understand the differences in standard preference learning and DPL by their assumed data generating processes. Mean-variance DPL models each preference's utility as a Gaussian with its own mean and variance parameters, supporting heteroskedasticity. Because drawing from a softmax distribution is equivalent to adding independent Gumbel distributed noise to each choice and choosing the one with the largest noisy value (the Gumbel max trick, see `https://arxiv.org/abs/2110.01515`), standard preference learning can be understood as modeling each alternative's utility as being drawn from a Gumbel distribution with a fixed global scale parameter. The Gaussian and Gumbel distributions are similar aside from the long right tail of the Gumbel distribution, so the main difference comes from the treatment of heteroskedasticity.

Thus, the situations in which we expect DPL to outperform standard preference learning can be thought of as those in which there is significant heteroskedasticity in the data. In other words, if choices tend to have significantly different variances, this presents a problem for standard preference learning and should be an area where DPL will shine. Conversely, if all choices have similar variances, DPL may perform worse than standard preference learning due to the unnecessary additional free parameters.

Concrete failure mode: Theorem 3.4 tells us that no learning algorithm can hope to guarantee maximization of social welfare given only paired preference data in the presence of hidden context. However, there are certainly settings where the estimated mean is more accurate under DPL than standard preference learning, as is convincingly demonstrated in Figure 3 of the paper. This synthetic experiment points to a key failure mode of standard preference learning: *in cases where choices with higher means also have higher variances (due to hidden context), standard preference learning will often choose the lower utility choice because it has lower variance* (since the estimate is based on how often the choice wins, not by how much it wins). The synthetic results shown in Figure 3 have proven this point, but the question remains as to how likely this scenario is in settings of real language data.

2 Research Question

Does DPL significantly improve the mean utility estimates on real-world prompt-response datasets where there may be significant heteroskedasticity due to hidden context?

3 Experimental Design

3.1 Data Construction

Because most real prompt-response datasets lack true utility scores, this is not a straightforward question to study. To circumvent this issue, we propose using the **Stanford Human Preferences (SHP) dataset** [1], which contains paired Reddit responses with associated upvote scores.

Rationale for SHP: Raw upvotes, though not perfect utility estimates, provide a useful proxy for population-level preference strength. While they are a population statistic rather than an individual preference, the dataset should still naturally contain hidden context in the form of different Reddit user preferences and subreddit cultures. Importantly, the upvote scores allow us to use the scores as approximate ground truth utilities that are masked during preference learning but can be compared against after training.

Data filtering: We filter to only examples where the time difference between response postings is less than 60 seconds to avoid cases where responses had drastically different visibilities, which could confound upvote counts.

Hidden context: The hidden context z in this dataset represents the diverse preferences of Reddit users who voted on responses. Different users have different values, risk tolerances, and preferences for content style, creating heteroskedastic noise in the preference data.

3.2 Learning Setup

We reuse the following components from the original DPL paper:

1. **Standard preference learning baseline:** Train using Bradley-Terry MLE loss (Equation 1 from the paper). For a single preference comparison where alternative a is preferred to b, the loss is:

$$-\log \sigma(\hat{u}(a) - \hat{u}(b)) \tag{1}$$

   where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic sigmoid function. The total loss is averaged over all preference comparisons in the training set.

2. **DPL mean-and-variance model:** Train using the distributional loss from Section 4 of the paper. For a single preference comparison where alternative a is preferred to b, the loss is:

$$-\log \Phi\left(\frac{\hat{\mu}(a) - \hat{\mu}(b)}{\sqrt{\hat{\sigma}(a)^2 + \hat{\sigma}(b)^2}}\right) \tag{2}$$

   where Φ is the standard normal CDF, $\hat{\mu}(a)$ is the learned mean utility, and $\hat{\sigma}(a)$ is the learned standard deviation.

3. **Model architecture:** Following the paper's approach (Appendix C.2), we fine-tune a pretrained language model to serve as the reward model. Specifically, we use LLaMA-2-7B with LoRA (low-rank adaptation) for efficient fine-tuning. We replace the standard language model head with a linear layer that outputs:

   - 1 scalar value $\hat{u}(x, y)$ for standard preference learning
   - 2 values for DPL mean-and-variance: $f_1(x, y)$ and $f_2(x, y)$, which are transformed into $\hat{\mu}(x, y) = f_1(x, y)$ and $\hat{\sigma}(x, y) = \log(1 + \exp(f_2(x, y)))$ (using softplus to ensure variance is positive)

   The model takes as input a concatenation of prompt x and response y, and produces the utility estimate(s).

4. **Training regime:** We start by following the paper's setup (Appendix C.2), we use the AdamW optimizer with learning rate 3×10^{-6} decayed via a cosine schedule to 3×10^{-7}, batch size of 2 comparisons (i.e., 4 responses total: 2 pairs of $(x, y_{\text{preferred}})$ and $(x, y_{\text{dispreferred}})$), and weight decay of 0.0001. However, because we are using a different dataset, we may need to adjust these training parameters.

3.3 Training Procedure

1. On the train split, train a DPL mean and variance model on SHP preference comparisons to obtain $\hat{\mu}_{\text{DPL}}(x, y)$ and $\hat{\sigma}_{\text{DPL}}(x, y)$
2. On the same train split, train a standard preference learning model on SHP to obtain $\hat{u}_{\text{SPL}}(x, y)$

3.4 Evaluation Metrics

We evaluate on a held-out test set using the following metrics:

1. **Pseudo-regret:** For each prompt-response pair (x, y_A, y_B) in the test set, calculate the pseudo-regret for each method:

$$R_{\text{SPL}} = \max(u_{\text{true}}(x, y_A), u_{\text{true}}(x, y_B)) - u_{\text{true}}(x, y_{\text{selected-SPL}}) \quad (3)$$

$$R_{\text{DPL}} = \max(u_{\text{true}}(x, y_A), u_{\text{true}}(x, y_B)) - u_{\text{true}}(x, y_{\text{selected-DPL}}) \quad (4)$$

   where $y_{\text{selected-SPL}} = \arg\max_{y \in \{y_A, y_B\}} \hat{u}_{\text{SPL}}(x, y)$ and $y_{\text{selected-DPL}} = \arg\max_{y \in \{y_A, y_B\}} \hat{\mu}_{\text{DPL}}(x, y)$.

   This measures the opportunity cost of using each learned reward model to select between responses: how much true utility is lost by following the model's preference compared to the oracle best choice. Report average pseudo-regret across all test pairs.

2. **Selection accuracy:** For each test pair, report the percentage of cases where each method correctly identifies the higher-utility response according to ground truth:

$$\text{Acc}_{\text{SPL}} = \frac{1}{N} \sum_{i=1}^{N} 1[\hat{u}_{\text{SPL}}(x_i, y_A) > \hat{u}_{\text{SPL}}(x_i, y_B) \iff u_{\text{true}}(x_i, y_A) > u_{\text{true}}(x_i, y_B)] \quad (5)$$

   and similarly for DPL using $\hat{\mu}_{\text{DPL}}$.

3.5 Expected Results if DPL Helps

If DPL successfully addresses the identified failure mode, we expect to observe:

1. **Lower pseudo-regret:** The DPL reward model should incur lower average pseudo-regret when used to select between responses: $E[R_{\text{DPL}}] < E[R_{\text{SPL}}]$. This demonstrates that DPL's better mean estimates translate to better decision-making when choosing between alternatives.
2. **Higher selection accuracy:** $\text{Acc}_{\text{DPL}} > \text{Acc}_{\text{SPL}}$, meaning DPL correctly ranks response pairs more often than standard preference learning.
3. **Systematic bias correction in reward models:** Examining reward model predictions, standard preference learning should systematically underestimate utilities for high-variance, high-mean responses (consistent with Borda count behavior described in Theorem 3.1). A scatter plot of $\hat{u}_{\text{SPL}}$ vs. u_{true} should show predictions compressed toward the mean, while $\hat{\mu}_{\text{DPL}}$ vs. u_{true} should show less bias and better calibration across the full utility range.

4 Limitations and Considerations

While upvotes are imperfect proxies for true utilities (primarily because they are population statistics), they provide a reasonable approximation of aggregated human preferences. The key assumption is that if DPL better captures heteroskedasticity in upvote patterns, it should also better handle heteroskedasticity in true preference data. The 60-second time filter helps control for visibility confounds but inevitably newer responses may still have somewhat biased scores.

References

[1] Ethayarajh, K., Choi, Y., and Swayamdipta, S. (2022). Understanding dataset difficulty with V-usable information. In *International Conference on Machine Learning*, PMLR.