A Bayesian Approach for Preference Alignment for Language Models

Bissmella Bahaduri
MVA, ENS Paris-Saclay
bissmellabahaduri@gmail.com

Adrien Letellier
MVA, ENS Paris-Saclay
adrien.letellier@ensae.fr

Abstract

In this report we take a Bayesian approach to provide a better solution to the problem of noisy labels for LLM alignment based on Wang et al. (2023). We propose two contributions. First, we link the setting of the paper to the more general Bayesian framework of Rolf et al. (2022). Second, we experiment with other losses for the training of the preference model, namely the reverse KL divergence that has more theoretical guarantees, and the Jensen-Shannon entropy. The code for this report is available at `https://github.com/Bissmella/llm_bayesian_preference.git`.

1 Introduction

Alignment with human preferences is a crucial task for LLMs after pre-training. Trained on masked token prediction, the LLMs get only the capability of next token prediction. However, this capability is of limited use for data-to-day usage. That is why there comes the next step of fine-tuning of the pre-trained model to human preferences so that the LLM can learn the capability of instruction following and prompt responding. During the last few years there has been major advancements in this field. A lot of the current techniques and methods are based on a first fine-tuning step followed by reinforcement learning such as reinforcement learning based on human feedback, DPO, and recently introduced GRPO. During this stage of fine-tuning the model is mainly trained to provide good responses by ranking of the generated responses by human annotators to good and bad responses. Based on the feedback provided by the human feedback the model learns to generate acceptable responses and avoid unacceptable responses. More recently, the DPO method changes this problem to a more supervised learning problem by first collecting acceptable and unacceptable responses to a given prompt. The ranking of the responses are being determined again by multiple human annotators and the final label is an aggregation of these multiple rankings. However, as expected the ranking of the good and bad responses can be controversial and not all annotators may have the same idea of the rank of a given response. In a worse scenario the ranking of the annotators might cancel each other out and lead to a trivial neutral ranking. More specifically in the case of emotional analysis and emotional support, the subjectivity of the responses is even higher due to the different human values, priorities, and personal standards. Finding the true label from such a noisy dataset becomes extremely difficult and fine-tuning a model based on supervised learning on such noisy labels becomes problematic.

2 Bayesian methodology

In this section, getting inspiration from Rolf et al. (2022) we first develop a Bayesian approach for dealing with a more general version of the problem - noisy labels.

2.1 Uncertainty in the labels

In a supervised learning setting, a neural network is trained on hard labels to minimize cross-entropy $-\sum_i \sum_l p_i^d(l) \log q_i(l)$ between the model prediction $q_i(l) = q(l|x_i, \theta)$ and the provided hard labels $p_i^d(l) \in \{0, 1\}$. However, in the case where the labels are only soft priors on the true unknown labels, by training on such labels the model will also learn to make uncertain and noisy predictions. Converting such priors to hard targets $\mathbf{1}[l = \arg\max_l p_i^d(l)]$ is also problematic as the model will make highly confident but wrong predictions.

2.2 Solution based on a Bayesian approach

Using a Bayesian approach we consider the provided uncertain labels $p_i(l)$ as a prior on our belief on the label of the data instance i. Then we have the likelihood as $p(x_i|l)$ which shows the likelihood of data point x_i given the label l. The given likelihood unlike the belief does not suffer from uncertainity but it is difficult to compute from the data. In a Bayesian model, this likelihood is assumed to be a well-specified conditional distribution. It does not include uncertainty about which label is correct—instead, it assumes that for a fixed l, the mechanism generating x_i is fixed (even if it is stochastic in nature due to inherent noise in the data). Any uncertainty about l is deferred to the prior. This can also be thought as the generative model generating data x_i conditional on l. However, formulating this as generative model makes it even more difficult as it requires a generative network, sampling complexities, and generative training.

On the other hand the Bayes law allows us to write the posterior based on the provided likelihood and prior belief. As in the variational inference framework, a good approximation of the true posterior $p(l|x_i)$ can be attained by introducing a variational posterior distribution $q(l|x_i)$. Here, we do not have simplifying assumptions on the variational posterior distribution but instead make it conditional to the data x_i. Minimizing a divergence measure between this variational posterior and the true posterior will allow to obtain a good approximation of the true posterior.

2.3 Numerical optimization

Several choices are possible for the divergence measure. In Rolf et al. (2022), explicit computations are derived for the reverse Kullback-Leibler divergence $KL\big(q(l|x_i) \,||\, p(l|x_i)\big)$. It can indeed be shown that minimizing the reverse Kullback-Leibler divergence is equivalent to minimizing the free energy (a negated evidence lower bound (ELBO)):

$$\min_{q(l|x_i), p(x_i|l)} -\sum_i \sum_l q(l|x_i) \log \frac{p(x_i|l) p_i(l)}{q(l|x_i)} \tag{1}$$

Minimizing negative ELBO involves both the forward distribution $p(x_i|l)$ and the variational posterior $q(l|x_i)$. This is achieved in VAE (Kingma et al. (2013)) by defining a parameterized encoder as $q(l|x)$ and a parameterized decoder for $p(x|l)$. However, Rolf et al. (2022) propose parameterizing only the $q(l|x; \theta)$ as a neural network. The generative conditional $p(x_i|l)$ is only defined for individual data point x_i and is computed by minimizing 1 for fixed q subject to constraint that p is a proper probability distribution. Using this trick, they first derive an explicit solution to the minimization problem with respect to $p(x_i|l)$.

We prove it using Lagrange multipliers. Since we are optimizing in the first step only over $p(x_i|l)$, we treat $q(l|x_i)$ as fixed. Excluding the terms that do not depend on $p(x_i|l)$, we are facing the problem:

$$\min_{p(x_i|l)} -\sum_i \sum_l q(l|x_i) \log p(x_i|l) \quad \text{s.t.} \quad \sum_i p(x_i|l) = 1, \forall l$$

We hence define the Lagrangian:

$$\mathcal{L} = -\sum_i \sum_l q(l|x_i) \log p(x_i|l) + \sum_l \lambda_l \left(\sum_i p(x_i|l) - 1 \right).$$

Applying the first order condition, we get:

$$\frac{\partial \mathcal{L}}{\partial p(x_i \mid \ell)} = -\frac{q(\ell \mid x_i)}{p(x_i \mid \ell)} + \lambda_\ell = 0.$$

Solving for $p(x_i \mid \ell)$ and using the constraint $\sum_i p(x_i \mid \ell) = 1$, we end with:

$$\lambda_\ell = \sum_j q(\ell \mid x_j).$$

Substituting back, the optimum is achieved by:

$$p(x_i|l) = a_{i,l} = \frac{q(l|x_i)}{\sum_j q(l|x_j)} \tag{2}$$

This allows us to rewrite the true posterior $p(l|x_i)$:

$$p(l|x_i) = \frac{p_i(l)p(x_i|l)}{p(x_i)} = \frac{p_i(l)q(l|x_i)}{p(x_i)\sum_j q(l|x_j)}$$

This is still not fully tractable because of the normalization constant $p(x_i)$. The authors handle it by estimating it within batches, and conduct further analysis of the effect of the batch size on performance. Hence, it is possible to perform gradient descent on the reverse Kullback-Leibler divergence to optimize the variational distribution:

$$\sum_i KL\big(q(l|x_i;\theta) \mid\mid p(l|x_i)\big) = \sum_i KL\big(q(l|x_i;\theta) \mid\mid \frac{p_i(l)q(l|x_i;\theta)}{p(x_i)\sum_j q(l|x_j;\theta)}\big)$$

2.4 Discussion on the choice of the divergence

The authors also include the forward Kullback-Leibler divergence $KL\big(p(l|x_i) \mid\mid q(l|x_i)\big)$ in their experiments, which is shown to generally perform better than the reverse one, although the previous theoretical justification does not hold. However, the authors underline an interesting property of the forward KL divergence case, namely that it reduces to the minimization of the standard cross entropy loss in the case of provided hard labels. Indeed, assuming $p_i^d(l)$ are provided, we have:

$$\begin{aligned} KL\big(p_i^d(l) \mid\mid q(l|x_i)\big) &= \sum_l p_i^d(l) \log\Big(\frac{p_i^d(l)}{q(l|x_i)}\Big) \\ &= \sum_l p_i^d(l)\log(p_i^d(l)) - \sum_l p_i^d(l)\log(q(l|x_i)) \end{aligned}$$

where the right term is the cross entropy loss.

A limitation of the method is that the authors do not systematize and formalize settings where one variant of the KL would be superior to the other. An hypothesis, following Murphy (2012), would be that the reverse KL works better for unimodal posterior distributions whereas the forward is better for multimodal ones. We try to evaluate the best performing loss in the experiments part of this report.

A more concerning point is that it is surprising that they are able to run their experiment using the forward KL. In general, this is untractable because it requires to integrate with respect to the true posterior distribution. The authors bypass this using their tricks of not parameterizing $p(x_i|l)$. However, this raises concerns about overfitting to the observations x_i.

In the following section we go in more details on the specific case of human preference modeling.

3 Preference modeling

Human preference labels are also noisy labels due to controversies and the potential sensitivity of the concerned topic. Following the section above we can model the preferences using a Bayesian approach and learn it from data. The dataset can be given in a format of (c, s, y) context, response, and the noisy measure of preference given in the data respectively. Hence, we replace the x_i of the general setting by $s_i|c_i$ and we model the true preference distribution that generated the data y by ρ, where the distribution $p_i(\rho)$ will act as the prior $p_i(l)$ of the previous section. The log probability of a response given a context can be written as $\sum_i \log p(s_i|c_i) = \sum_i \log \sum_\rho p(s_i|c_i,\rho)p_i(\rho)$, where

$p_i(\rho)$ is a prior over the preferences given by the noisy data. Now we are interested in the posterior over the preferences. It can be written as $p(\rho|s_i, c_i) = \frac{p(s_i|c_i,\rho)p_i(\rho)}{p(s_i|c_i)}$ and the denominator can be written as:

$$p(s_i|c_i) = \sum_{\rho} p(s_i|c_i, \rho)p_i(\rho)$$

However this is intractable. So, as in the previous section, we approximate the posterior with a variational posterior distribution $q(\rho|s_i, c_i)$ and we optimize it to make it as close as possible to the true posterior by minimizing the KL divergence. The preference model is defined as the variational posterior distribution that we will obtain:

$$q(\rho|s_i, c_i) = \mathcal{R}(s_i, c_i|\theta)$$

We write the objective as:

$$\sum_i KL(q(\rho|s_i, c_i)||r_i(\rho)) \tag{3}$$

where $r_i(\rho) = \alpha_i p_i(\rho)p(s_i|c_i, \rho)$ is another notation for the posterior $p(\rho|s_i, c_i)$ with α_i as normalization factor.

$$r_i(\rho) = \alpha_i p_i(\rho)\frac{q(\rho|s_i, c_i)}{\sum_j q(\rho|s_j, c_j)} \tag{4}$$

Finally, we get the optimal parameter for the preference model $\mathcal{R}$ by minimizing the following sum of reverse KL divergences:

$$\sum_i KL(q(\rho|s_i, c_i)||r_i(\rho)) = \sum_i KL(R(s_i, c_i, \theta)||\alpha_i p_i(\rho)\frac{R(s_i, c_i, \theta)}{\sum_j R(s_j, c_j, \theta)}) \tag{5}$$

It is not made explicit in the article but we notice in the code released by the authors[1] that they do not minimize the reverse KL but the forward KL, that is:

$$\sum_i KL(\alpha_i p_i(\rho)\frac{R(s_i, c_i, \theta)}{\sum_j R(s_j, c_j, \theta)}||R(s_i, c_i, \theta)) \tag{6}$$

Hence, as in the more general framework detailed in the previous section, we cannot rely on the previous calculations. Nevertheless, we have the interesting property that it reduces to the cross entropy in the case of hard labels.

4 LLM alignment

It is obvious that at the beginning the generative model $G(\epsilon_0)$ can not produce a response that can conform to all of the preferences. Now that we have derived the preference model, we want to align the LLM based on the proposed Bayesian preference model.

In order to calibrate the model to generate high preference response, Wang et al. (2023) use a three step process consisting of generating multiple candidates using diverse beam search and then ranking them using the d-PM model. Further, a contrastive learning strategy is used to calibrate the generation likelihood same as the preference model.

Beam search and diverse beam search: is a technique used in next token generation, with the goal of finding the best generated text. Since at each step the model predicts a probability distribution over the entire vocabulary size, then finding the best generated text becomes a tree search problem which is computationally expensive. By limiting the width of considered possible tokens at each step to a prefixed beam, the complexity can be reduced drastically. While beam search may end up generating very similar texts, the diverse beam search adds a similarity penalty to the score by encouraging the selection of diverse and different options.

[1] https://github.com/wangjs9/Aligned-dPM/tree/master

More specifically, for a dataset (X, Y), first a set of K candidates $\{\tilde{y_1}, \tilde{y_2}, \ldots, \tilde{y_K}\}$ are generated via diverse beam search, where $\tilde{y_k} = G(x, \xi_0)$. Next a preference score is calculated for each of the generated responses as $S(\tilde{y_k}, x) = R(\tilde{y_k}, x)$ and ranked based on the predicted preference score as $[y'_1, \ldots, y'_k]$ where $S(y'_i) > S(y'_j)$ for $i < j$. Next, a ranking loss is calculated as:

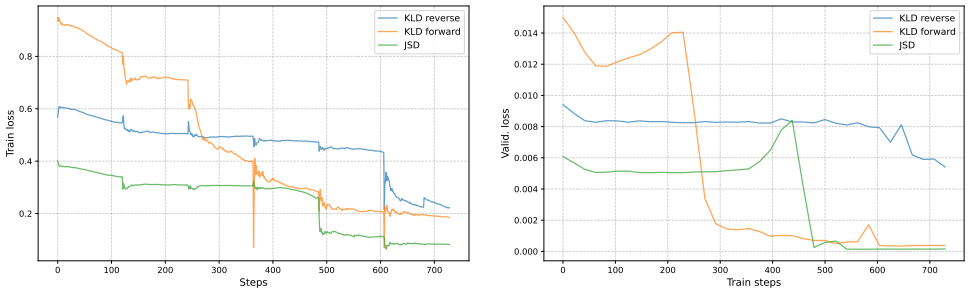
$$L^r = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \max(0, P(y_j; \xi) - \tilde{P(y_i, \xi)} + \lambda_{ij} \tag{7}$$

where P is the sum of log probabilities normalized by the length of the generated text and $\lambda_{ij} = \lambda(j - i)$ is the default ranking loss. In order to avoid the model from forgetting the ground truth an additional term is added as following to get the final loss:

$$L = -\lambda \frac{1}{|y|} \sum_{t=1}^{|y|} \log G(y_t | x, y_{t,}, \xi) + L^r \tag{8}$$

5 Experiments

With the same setting and configuration as Wang et al. (2023) we conducted experiments on training preference models. We focused on comparison of forward KL divergence, reverse KL divergence, and Jenson-Shannon divergence. The training and evaluation comparison of the three loss functions can be seen in figure 1. The Jenson-Shannon divergence loss shows better performance in terms of convergence and also generalization. However, due to a high computational cost, we were not able to compare the final performance of the trained model after calibration.



(a) Training loss curve

(b) Evaluation loss curve

Figure 1: Loss curves for three different loss functions.

References

Kingma, D. P., Welling, M., et al. (2013). Auto-encoding variational bayes.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Rolf, E., Malkin, N., Graikos, A., Jojic, A., Robinson, C., and Jojic, N. (2022). Resolving label uncertainty with implicit posterior models. *arXiv preprint arXiv:2202.14000*.

Wang, J., Wang, H., Sun, S., and Li, W. (2023). Aligning language models with human preferences via a bayesian approach. *Advances in Neural Information Processing Systems*, 36:49113–49132.