Project Proposal: BanditNet

Diwen Lu (dl2309), Zian Jiang (zj444), Lizhong Wang (lw2350)

March 2021

1 Introduction

For the final project, we plan to reproduce the results from the paper [3] and extend the proposed method on a new benchmark dataset. The paper proposes a technique that fits the contextual bandit problem, which is very common in search engine and recommendation system applications, into a deep learning setting.

Given logged contextual bandit feedback from policy π_0, the paper proposes a counterfactual risk minimization approach for training neural networks using the self-normalized inverse propensity scoring (SNIPS) estimator, which is referred to in lecture as the importance weighted (IW) estimator. According to the paper, the SNIPS estimator does not permit stochastic gradient descent (SGD) optimization in its given form, which is an obstacle in efficiently training a neural network. To remedy this problem, the paper performs a reformulation of the SNIPS estimator into a series of constrained optimization problems such that it retains both the desired properties of the SNIPS estimator and the ability to use SGD for training.

2 Main Idea of BanditNet

2.1 Problem Formulation

Consider the contextual bandit setting where a policy π takes an input $x \in \mathcal{X}$ and outputs an action $y \in \mathcal{Y}$. Then we observe risk (or reward) $\delta(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Note that we do not observe $\delta(x, y'), \forall y' \notin Y$.

To reformulate the above in a deep learning setting, we can view the stochastic policy π as a neural network π_w, where w is the weights of the network and $\pi_w(Y|x)$ is a conditional probability distribution (such as a neural network with a softmax output layer) over all possible actions $y \in \mathcal{Y}$. For example, $\pi_w(y|x)$ is the probability of choosing action y given covariate x.

Now given the logging policy π_0, we can incorporate the above setup into our setup from the lecture and formulate the logged contextual bandit feedback D, a collection of n tuples of observed context $x \sim Pr(X)$, action $y_i \sim \pi_0(Y|x_i)$

chosen by the logging policy π_0, propensity score $p_i \equiv \pi_w(y_i|x_i)$, and risk (or reward) $\delta_i(x_i, y_i)$:

$$D = [(x_1, y_1, p_1, \delta_1), ..., (x_n, y_n, p_n, \delta_n)]. \tag{1}$$

Now, we will discuss our batch training objective using our logged contextual bandit dataset D.

2.2 Equivariant Counterfactual Risk Minimization

The paper aims at solving a counterfactual risk minimization (CRM) problem, which can be estimated from our logged contextual bandit feedback D, but this suffers from two problems: 1) we do not know the rewards for actions that were not chosen, 2) D is biased towards π_0. As discussed in lecture, the inverse propensity scoring (IPS) estimator addresses them:

$$\hat{R}_{IPS}(\pi_w) = \frac{1}{n} \sum_{i=1}^{n} \delta(x_i, y_i) \frac{\pi_w(y_i \mid x_i)}{\pi_0(y_i \mid x_i)}. \tag{2}$$

As shown in the lecture, $\hat{R}_{IPS}$ is unbiased and has bounded variance given non-zero conditional probabilities for all actions. But from the lecture we know this estimator is prone to propensity overfitting, which is linked to its lack of equivariance.

Later in the lecture, we introduced the self-normalized IPS (SNIPS) estimator, which is immune to propensity overfitting and has negligible bias and lower variance than the IPS estimator:

$$\hat{R}_{SNIPS}(\pi_w) = \frac{\frac{1}{n} \sum_{i=1}^{n} \delta(x_i, y_i) \frac{\pi_w(y_i|x_i)}{\pi_0(y_i|x_i)}}{\frac{1}{n} \sum_{i=1}^{n} \frac{\pi_w(y_i|x_i)}{\pi_0(y_i|x_i)}} \tag{3}$$

To be consistent with the neural network setting, we will view δ as risk, as opposed to reward, and thus our training objective becomes

$$\hat{w} = \arg\min_{w} \hat{R}_{SNIPS}(\pi_w). \tag{4}$$

2.3 Training Algorithm and an Optimization Obstacle

As we can see in Eq.(3) and Eq.(4), batch training with stochastic gradient descent (SGD) is not permitted in its given form as we are optimizing a ratio. The paper remedies this problem with the following technique.

Intuitively, the denominator of Eq.(3), $S \equiv \frac{1}{n} \sum_{i=1}^{n} \frac{\pi_w(y_i|x_i)}{\pi_0(y_i|x_i)}$, is an one-dimensional quantity. We can perform grid search in sensible values $\{S_1, ..., S_k\}$ for the optimal S^* and instead for each S_j, solve

$$\hat{w}_j = \arg\min_{w} \frac{\frac{1}{n} \sum_{i=1}^{n} \delta(x_i, y_i) \frac{\pi_{w_j}(y_i|x_i)}{\pi_0(y_i|x_i)}}{S_j}. \tag{5}$$

This gives us a list of potential solution $\{\hat{w}_1, ..., \hat{w}_k\}$, and then we simply take the minimum out of them. However, this still leaves the question of how to solve each equality constrained risk minimization problem subject to $S_j = \frac{1}{n}\sum_{i=1}^{n}\frac{\pi_{w_j}(y_i|x_i)}{\pi_0(y_i|x_i)}$ using SGD. Instead, the paper proposes a workaround using its Lagrangian.

Consider the constrained optimization problem

$$\hat{w}_j = \arg\min_w \frac{1}{n}\sum_{i=1}^{n}\delta(x_i, y_i)\frac{\pi_{w_j}(y_i \mid x_i)}{\pi_0(y_i \mid x_i)} \text{ subject to } \frac{1}{n}\sum_{i=1}^{n}\frac{\pi_{w_j}(y_i \mid x_i)}{\pi_0(y_i \mid x_i)} = S_j. \tag{6}$$

Its Lagrangian can be written as

$$L(w, \lambda) = \frac{1}{n}\sum_{i=1}^{n}\frac{(\delta_i - \lambda)\pi_w(y_i \mid x_i)}{\pi_0(y_i \mid x_i)} + \lambda S_j. \tag{7}$$

Since we do not care about the solution to Eq.(6) for any specific S_j, instead merely exploring a sensible range of values for S_j, we can determine the value of S_j in hindsight such that for each $\lambda_j \in \{\lambda_1, ..., \lambda_k\}$, we fix λ_j and solve

$$\hat{w}_j = \arg\min_w L(w, \lambda_j). \tag{8}$$

Using the optimality conditions, we can get the corresponding $\hat{w}_j$ and S_j for a particular λ_j. Then we take the mininum among $\{\hat{w}_1, ..., \hat{w}_j\}$.

2.4 Supervised to Logged Bandit Feedback Conversion

The paper evaluates its results using the ResNet20 architecture [2], with the cross-entropy loss replaced by the counterfactual risk, on the CIFAR-10 [4] dataset so that it is able to compare supervised learning on full-information data with training on logged contextual bandit feedback, and there is a full-information test set for estimating prediction error. Obviously this leaves the question of how to simulate logged contextual bandit feedback using image and label pairs from the CIFAR-10 dataset. The paper performs the standard supervised to bandit conversion proposed in [1].

The main idea of this conversion is that, since the logging policy $\pi_0(y \mid x)$ is a probability distribution over the all possible actions y (predicting 1 out of the 10 labels from the CIFAR-10 images) given an image x, we can generate such a probablity distribution where for each image, we slightly bias toward the correct label such that we still have a reasonably high chance of predicting the correct label (selecting an action with the lowest risk in stochastic-policy terms), compared to random guessing ($\sim 10\%$). In the paper a hand-coded logging policy that achieves about 49% error rate on the training data is used. This biased hand-coded logging policy can be implemented in multiple ways. In one implementation [6] that we have found online, it calculates `np.sqrt(np.arange(1, 11))` for each image, multiplies the entry corresponding to its ground-truth label by 10, and then normalizes this vector as the final conditional probability distribution.

3 Our Planned Approach

In addition to implementing the training algorithm and performing empirical evaluation presented in the paper, we also plan to apply it to other benchmark image datasets such as MNIST [5]. There are two important findings from the paper we are particularly interested in verifying. First, it claims that if given enough bandit feedback, Bandit-ResNet converges to and even outperforms the skyline performance (fully supervised learning using ResNet on CIFAR-10) by a small margin. We would like to verify this on CIFAR-10 and other datasets. Also, it claims that λ in the range of $(0.8, 1.0)$ results in good prediction performance. We would like to verify this and explore with other λ selections as well. Our github repo for the project is at **https://github.com/DiwenLu/ToolsAndTechniquesforML_project**.

References

[1] Alina Beygelzimer and John Langford. The offset tree for learning with partial labels, 2016.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[3] Thorsten Joachims, Adith Swaminathan, and Maarten de Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018.

[4] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[5] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[6] Noveen Sachdeva. Banditnet. `https://github.com/noveens/banditnet`, commit = 7f2c408b7e31943a3a59f1e5e44be1d8c5d365a7, 2019.