Supplemental Material: A Multi-Armed Bandit Approach for House Ads Recommendations

Nicolás Aramayo[1], Mario Schiappacasse[1], Marcel Goic[2]

[1]I2B Labs, [2]Department of Industrial Engineering, University of Chile

April, 2022

1 Using Dropout to implement Thompson Sampling

In Thompson Sampling (TS), we choose the action $a^* \in A$ according to its probability of maximizing the expected reward. Therefore the probability p_a that an action a is chosen is given by:

$$p_a = \Pr\left\{\mathbb{E}\left[r \mid a^*, \mathbf{x}, \boldsymbol{\theta}\right] = \max_a \mathbb{E}\left[r \mid a, \mathbf{x}, \boldsymbol{\theta}\right]\right\} \tag{1}$$

Here, r is the reward obtained, $\mathbf{x}$ encodes the context and $\boldsymbol{\theta}$ the parameters of the model. In practice, to implement TS, we sample from the posterior predictive distribution the chose an action that maximizes the expected reward. In our case, as the predictions are generated by a neural network, the posterior distribution is usually intractable. The following discussion explains how dropout can be used to characterize uncertainty of optimal decisions providing the basis for a Thompson sampling strategy.

The analysis is based on Gal and Ghahramani (2016) who show that, in the context of neural networks, dropout can be used to characterize the posterior distribution of the forecasts. Interestingly, Gal and Ghahramani do mention that this description of uncertainty can be important for reinforcement learning and that Thompson sampling

can be used through the uncertainty estimates over the actions an agent can take, but they do not fully elaborate on this application as we do in this appendix.

The idea of dropout is to randomly drop internal nodes (along with their connections) from the neural network during training and it was originally thought as a technique to prevent over-fitting (Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov, 2014). Formally speaking, we can consider dropout as a series of binary masks $\mathbf{z}$ that shut-off neurons in the network according to some probability distribution $p(\mathbf{z})$. The basic intuition behind the use of dropout in a Thompson sampling scheme is that the prediction of a neural network after dropout training can be interpreted, from a Bayesian perspective, as the average prediction of the sub-models induced by dropout training and weighted by the posterior distribution of the dropout. Thus, with dropout, the forecast of a context vector $\mathbf{x}^*$ can be obtained by averaging over the distribution of masks $\mathbf{z}$ (Maeda, 2014):

$$\int yp(y \mid x^*, D, \boldsymbol{\theta}) = \sum_z \int yp(y \mid x^*, D, \boldsymbol{\theta}) p(z \mid D, \boldsymbol{\theta}) dy \tag{2}$$

$$\approx \sum_z \int yp(y \mid x^*, D, \boldsymbol{\theta}) q(z) dy \tag{3}$$

The approximation in Equation (3) is justified because the posterior distribution of $\boldsymbol{z}$ is usually intractable and therefore it can be replaced by a tractable trial distribution $q(\boldsymbol{z})$. To complete this discussion, Gal and Ghahramani (2016) show that the use of dropout in neural networks can be interpreted as a Bayesian approximation of a Gaussian process and that the dropout training minimizes the Kullback–Leibler divergence between the approximate distribution $q(\boldsymbol{z})$ and the posterior of the corresponding Gaussian process.

Thus, when obtaining a forecast in a neural network using dropout we are obtaining the probability of maximizing the expected reward after sampling its parameters from the posterior predictive distribution. This is done by simply choosing the arm with the highest estimated reward. Therefore, to implement Thompson sampling for neural networks with dropout, we sample from the posterior distribution of $\boldsymbol{\theta}$, $p(\boldsymbol{\theta} \mid \boldsymbol{z})$, and the chose the action a^* that maximizes the expected reward for that value of $\boldsymbol{\theta}$, as is done in standard Thompson sampling.

2 Alternative Sampling Schemes

As is pointed out in section 3.1, the use of a neural network provides an estimation of the attractiveness of each set of ads. In Algorithm 1, we use those estimates to generate a Thompson sampler were, via dropout we select the display of ads according to their probability of being the action that maximizes the expected reward. However, literature of bandits offer a variety of alternative sampling strategies that can be built based on the outcome of the neural network. In this section, we compare our Thompson sampler against two of those competing strategies:

1. **Deep MNL:** In this sampling scheme, instead of using dropout, we applied a multinomial logit (MNL) model to the last layer of the net and we sample using the probabilities induced by this MNL model.

2. **Deep e-greedy:** In this version, we select the action with the largest reward most of the time, but we display a different set selected at random in a ε fraction of the cases.

We compare these alternative schemes against our proposal in a simulation exercise that follows the same assumptions to the one presented in section 3.3. It is worth noting that the estimation of the neural net does not change between sampling schemes and we only decide different actions conditional on the output of the net. Results of this exercise is presented in Table 1.

Sampling Scheme	Mean	s.d
Deep Thompson	0.293	0.109
Deep MNL	0.286	0.107
Deep ϵ-Greedy	0.242	0.089

Table 1: Mean and standard deviation of click through rates of alternative sampling schemes on simulated data

From this table we infer that the sampling scheme can have a relevant impact in the bandit policy. For instance the mean reward of the ϵ-Greedy algorithm is 17% worse

than the proposed algorithm. Notice however that these results also indicates that the strategy based on MNL probabilities are very competitive and it is only 2% worse than the proposed Thompson sampler based on dropouts.

Considering that the use of dropouts to approximate the posterior distribution is theoretically grounded and that it leads to marginally better results, we believe that our choice for using dropouts is well justified. Nevertheless, the relative good results of the sampler based on a MNL suggest that it can be a reasonable alternative in practice. A more comprehensive comparison of alternative sampling strategies in different settings is beyond the scope of this research, but it could be an interesting avenue for future research.

3 Gini indices for contextual and non-contextual bandits

The main premise of a multi-armed bandit algorithm is to solve efficiently the explore-exploit trade-off resulting from trying to optimize in an uncertain environment. In Figure 1 we show the dispersion of the solutions proposed by the two algorithms we implemented in the first experiment, as measured daily by the Gini index. The ϵ-greedy algorithm starts by fully exploring the action-reward space to then quickly commit to specific superarms, which are the estimated most popular combination of house ads. On the other hand, when there is an opportunity to personalize the display, the contextual bandit plays more diversely even in advanced stages. This is because the neural network is continuously learning a representation of the environment, which encompasses how visitors interact with it and deciding which combination of house ads is the most likely to be clicked by a customer with a particular history, which means not necessarily playing the most overall popular house ads.

4 A product view of ATC

On the second experiment we measured how the display of house ads in the home page can have an impact on subsequent—and arguably more important—decisions like adding

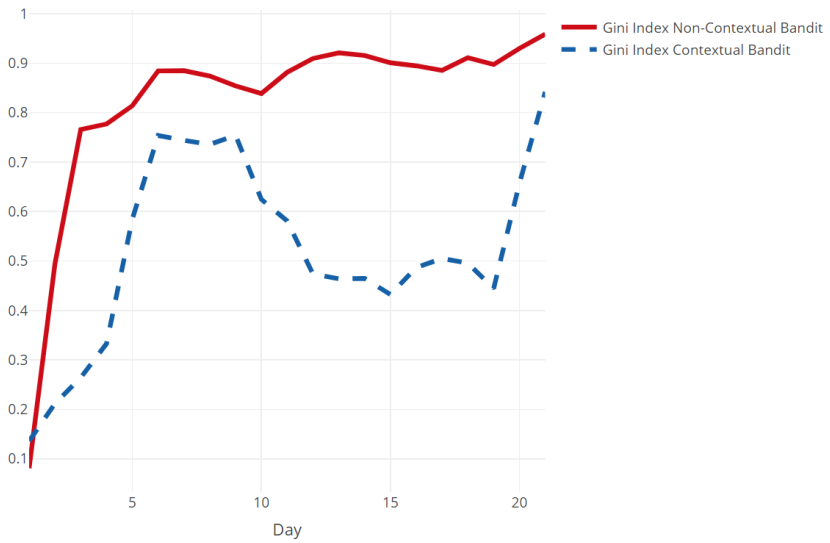

Figure 1: Evolution of daily Gini indexes for contextual and non-contextual bandits during experiment 1

products to the shopping cart. In table 2 we report ATC rates of the most viewed products of the landing pages linked by the expert recommendations, and we compare them against the ATC rates of the same products when visited from the MAB recommendation. The comparison shows that the MAB agent consistently obtains superior results. Given the landing page is the same regardless of who make the house ad recommendation, the difference can only be explained by better personalization as the bandit is effectively showing categories of products with a larger probability of been attractive to each particular visitor

	Expert Policy		MAB Policy	
Product	Display Ranking	ATC Rate	Display Ranking	ATC Rate
P01	1^{st}	13%	28^{th}	33%
P02	2^{nd}	8%	5^{th}	8%
P03	3^{rd}	17%	1^{st}	41%
P04	4^{th}	0%	17^{th}	38%
P05	5^{th}	24%	11^{th}	29%

Table 2: Add-to-cart rates of products after viewing them in their respective product pages for the human dynamic display and the contextual bandit