Personalizing Interventions with Diversity Aware Bandits

Colton BOTTA [a], Avi SEGAL [b] and Kobi GAL [a,b]
[a] *University of Edinburgh*
[b] *Ben-Gurion University of the Negev*

Abstract. Online systems utilize user data, such as demographics, past performance, preferences and skillset to construct an accurate model of users and maximize personalization. Some of these user features are "shallow" traits which seldom change (e.g. age, race, gender) while others are "deep" traits that are more volatile (e.g. performance, goals, interests). In this work, we explore how reasoning about this diversity of user features can enhance performance of personalized systems. By modeling the personalization process as a Reinforcement Learning (RL) problem, we introduce Diversity Aware Bandits for Intervention Personaliztion (DABIP), a novel contextual multi-armed bandit algorithm that leverages the dynamics within user features to cluster users while maximizing outcomes. We demonstrate the efficacy of this approach using two real world datasets from different domains.

Keywords. Contextual Multi-Armed Bandit, Interventions, Incentives

1. Introduction

Advancements in Artificial Intelligence (AI) have resulted in vastly improved models of users' behaviour [3,14]. Algorithms that use these models rely on data that describes users' online interactions, as well as their demographic information, previous performance, success on diagnostic tasks, etc. This data can be collectively referred to as the *context* of the user. How these varying contextual features collectively model the complexities of human beings is of particular interest in this work, an idea we refer to as *human contextual diversity*. Reasoning about the diversity of users when personalizing content and interventions is critical for optimizing such personalization. Specifically, we hypothesize that combining insights from social science about diversity can enrich models of users' behavior and improve the performance of personalization algorithms.

We present a novel reinforcement learning algorithm, Diversity Aware Bandits for Intervention Personalization (DABIP). DABIP is a "diversity aware" [15] Contextual Multi-Armed Bandit (CMAB) algorithm with three main steps: calculating the dynamics of the underlying human contextual diversity in a group, forming clusters of users with similar feature dynamics, and utilizing these clusters and past user performance to personalize content and interventions to users. We compare the performance of DABIP against LOCB [1], a state-of-the-art contextual bandit algorithm, as a baseline in two domains. Our results show that DABIP achieves a higher average reward than LOCB in each domain when predicting intervention outcomes per user.

2. Background

We give an overview of CMAB algorithms and diversity.

2.1. Contextual Multi-Armed Bandits

Contextual Multi-Armed Bandit (CMAB) is an extension of the Multi-Armed Bandit (MAB) problem where, at each timestep, the agent is presented with a list of arms (actions) and a context vector (additional data) about the environment. The agent needs to select and perform a single action. The agent then receives a reward for that arm only. Over time, the agent learns the underlying reward distribution of each arm and how that distribution is influenced by the context, and endeavors to maximize the total reward received over time [16]. One recent work introduced the Local Clustering in Bandits (LOCB) algorithm [1] which implemented a "soft" clustering approach, by which users are clustered together if their preferences are within a certain threshold of each other.

2.2. Diversity

The existence of differences between humans in a group is one notion of diversity [2], with these differences often falling into two distinct categories: surface-level differences and deep-level differences [8]. Surface-level differences include, for example, age, sex, ethnicity, and race and are generally defined by their low-dynamics and ability to be observed immediately [9]. Deep-level differences, on the other hand, may include skills, values, preferences, and desires. These are more volatile and can only be observed through prolonged interaction between people [8]. One example of the importance of this classification is highlighted by the WeNet project, which places human diversity at the center of a new machine mediated paradigm of social interactions [10,2].

3. DABIP

We now describe a Diversity Aware Bandit for Intervention Personalization algorithm.

3.1. Problem Definition

Let $N = \{1,...,n\}$ represent a set of n total users and $T = 1,...,t$ represent a sequence of timesteps. At timestep, t, a user, i_t, is drawn such that $i_t \in N$. Alongside i_t, the agent receives the context, $C_t = \{c_{1,t}, c_{2,t}, ..., c_{k,t}\}$ with one context vector for each of k arms and each context vector having dimension d such that $c_{k,t} \in \mathbb{R}^d$. The agent chooses one arm $x_{k,t}$, to recommend to i_t and receives reward r_t in return. We assume that each user is associated with an unknown bandit parameter $\theta_{i,t}$ that describes how i_t interacts with the environment and can be thought of as a representation of how user i_t behaves [1]. As in previous bandit settings [11,1,5], the goal is to minimize the total regret, R_T given by:

$$R_T = \sum_{t=1}^{T} [\theta_{i,t}^{\intercal}(argmax_{c_{k,t} \in C_t} \theta_{i,t}^{\intercal} c_{k,t}) - \theta_{i,t}^{\intercal} c_t] \tag{1}$$

where, at each round, t, we compute the regret by taking the reward achieved from the best possible arm choice, $x_{k,t}$, and subtracting the reward achieved from the agent's chosen arm, x_t. We also assume that each user, i, has a set of features, F, of length q such that at any time, t, there exists $F_{i,t} = \{f_{i,1,t}, f_{i,2,t} ..., f_{i,q,t}\}$.

3.2. DABIP Algorithm

The algorithm has three main steps: (1) Calculate the underlying feature dynamics of all users over time, (2) Form clusters of users with similar feature dynamics, then (3) Utilize the clusters and past user performance to personalize interventions to users. The full details of the algorithm are given in Appendix A.

4. DABIP Performance in Multiple Domains

We apply the DABIP algorithm to two datasets from two different domains.

4.1. Eedi Dataset

Eedi[1] [17] dataset includes over 17 million interactions of students answering multiple choice questions. It provides interaction logs of the student ID, question ID, student answer (range a-d), and the correct answer (range a-d). Every question has an associated list of features including a question ID, and a list of subject IDs. Every student has an associated list of features including gender, date of birth etc.

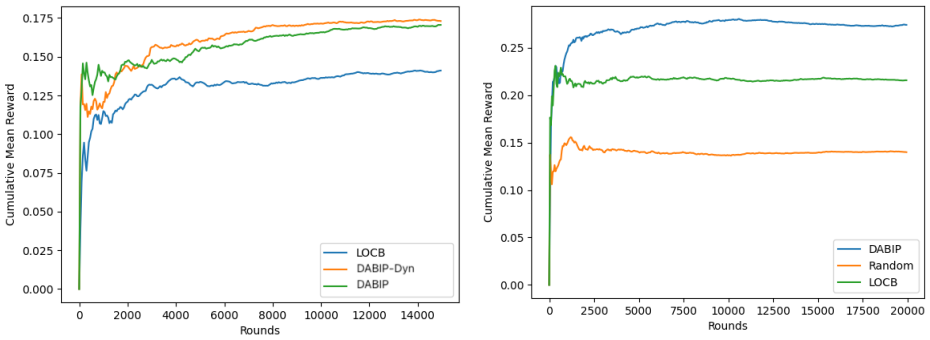
4.2. WeNet Dataset

The WeNet dataset includes 6600 interactions of users participating in WeNet's Ask4Help pilot [10,4,7]. Users participated in asking and answering questions, while receiving one of 4 different interventions messages that encourage their participation. The dataset provides interaction logs of the user ID, intervention messages ID, user activity level following the intervention. Additionally, every user has an associated list of features including location, big-5 characteristics, music and sports preferences, and past activity in the app. Finally, a binary label is computed for each intervention denoting if user activity post intervention surpassed a given threshold (median over post intervention activities).

4.3. Experiments

We apply DABIP to both domains. In the educational domain, the algorithm chooses personalized mathematics questions, based upon past student performance, that are likely to be answered correctly by the student. In the WeNet domain, the algorithm chooses, based upon users' past behaviour, personalized interventions that are likely to increase users' future engagement beyond a median based threshold. We compared DABIP to the LOCB baseline on both datasets. LOCB is available in open source [2] which we extended and adapted to operate on our datasets.

[1] https://eedi.com
[2] https://github.com/banyikun/LOCB



(a) Educational Dataset (b) WeNet Dataset

Figure 1. A comparison of the performance on both datasets based on cumulative average reward.

5. Results and Analysis

We compare the performance of DABIP and LOCB on the two datasets. As shown in Figure 1a, DABIP outperforms the LOCB baseline by about 25% on the education dataset. The DABIP-Dyn approach uses only the deep diversity features and shows comparable results to DABIP for this dataset. For the WeNet dataset, DABIP outperforms LOCB by about 30%. Additionally, DABIP demonstrates an improvement of more than 75% when compared to a random approach which chooses interventions randomly.

Our results show that identifying and extracting feature dynamics can improve RL algorithm performance, harnessing human diversity proxy information. We argue that identifying the highly dynamic features allows DABIP to search the space of context-reward associations more completely and more quickly, thus leading to better reward. This theory requires further testing, but the results of applying DABIP to real data are promising, and further research into augmenting our clustering approach is planned for the future.

6. Conclusion

In this work, we designed, implemented, and tested DABIP, a diversity aware RL algorithm that uses feature dynamics as a proxy for underlying human-contextual diversity. We hypothesized that this technique could improve RL algorithms that operate in environments where user data is highly dynamic, and this proved true when applying DABIP to two different domains. We believe that extensions to DAABIP can make it an ideal tool for building more performant personalized applications.

Acknowledgements

This work was supported in part by the European Union Horizon 2020 WeNet research and innovation program under grant agreement No 823783.

References

[1] Y. Ban and J. He. Local clustering in contextual multi-armed bandits. In *Proceedings of the Web Conference 2021*, pages 2335–2346, 2021.

[2] I. Bison, M. Bidoglia, M. Busso, R. C. Abente, M. Cvajner, M. D. R. Britez, G. Gaskell, G. Sciortino, S. Stares, et al. D1. 3 final model of diversity: Findings from the pre-pilots study. 2021.

[3] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. Das3h: modeling student learning and forgetting for optimally scheduling distributed practice of skills. *arXiv preprint arXiv:1905.06873*, 2019.

[4] A. De Götzen, P. Kun, L. Simeone, and N. Morelli. 21 mediating social interaction through a chatbot to leverage the diversity of a community. *Artistic Cartography and Design Explorations Towards the Pluriverse*, page 234, 2022.

[5] C. Gentile, S. Li, P. Kar, A. Karatzoglou, G. Zappella, and E. Etrue. On context-dependent clustering of bandits. In *International Conference on Machine Learning*, pages 1253–1262. PMLR, 2017.

[6] C. Gentile, S. Li, and G. Zappella. Online clustering of bandits. In *International Conference on Machine Learning*, pages 757–765. PMLR, 2014.

[7] F. Giunchiglia, I. Bison, M. Busso, R. Chenu-Abente, M. Rodas, M. Zeni, C. Gunel, G. Veltri, A. De Götzen, P. Kun, et al. A worldwide diversity pilot on daily routines and social practices (2020). 2021.

[8] D. A. Harrison, K. H. Price, and M. P. Bell. Beyond relational demography: Time and the effects of surface-and deep-level diversity on work group cohesion. *Academy of management journal*, 41(1):96–107, 1998.

[9] S. E. Jackson, V. K. Stone, and E. B. Alvarez. Socialization amidst diversity-the impact of demographics on work team oldtimers and newcomers. *Research in organizational behavior*, 15:45–109, 1992.

[10] P. Kun, A. De Götzen, M. Bidoglia, N. J. Gommesen, and G. Gaskell. Exploring diversity perceptions in a community through a q&a chatbot. In *DRS2022: Bilbao*. Design Research Society, 2022.

[11] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[12] S. Li, W. Chen, and K.-S. Leung. Improved algorithm on online clustering of bandits. *arXiv preprint arXiv:1902.09162*, 2019.

[13] S. Li, A. Karatzoglou, and C. Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548, 2016.

[14] H. Nakagawa, Y. Iwasawa, and Y. Matsuo. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *2019 IEEE/WIC/ACM International Conference On Web Intelligence (WI)*, pages 156–163. IEEE, 2019.

[15] L. Schelenz, I. Bison, M. Busso, A. De Götzen, D. Gatica-Perez, F. Giunchiglia, L. Meegahapola, and S. Ruiz-Correa. The theory, practice, and ethical challenges of designing a diversity-aware platform for social relations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 905–915, 2021.

[16] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[17] Z. Wang, A. Lamb, E. Saveliev, P. Cameron, Y. Zaykov, J. M. Hernández-Lobato, R. E. Turner, R. G. Baraniuk, C. Barton, S. P. Jones, et al. Instructions and guide for diagnostic questions: The neurips 2020 education challenge. *arXiv preprint arXiv:2007.12061*, 2020.

A. The DABIP Algorithm

We now give a detailed description of the algorithm. DABIP (Algorithm 1) is initialized with the number of clusters to maintain (s), the frequency with which to update the clusters ($T_{cluster}$), the frequency with which to update the user feature dynamics ($\mathscr{U}$), and an exploration parameter (α). Then, all users are initialized (Lines 2-4) and the algorithm begins iterating over all timesteps sequentially (Line 5). In each round, t, a user i_t is presented along with the set of context vectors C_t (Line 6). DABIP begins without any user clusters. DABIP first checks if there are any clusters (Line 7), and if there are none (length($\mathscr{G} \leq 0$)), then the arm with the highest upper confidence bound (UCB) is chosen. As is standard practice [11] in bandit algorithms, UCB is computed using the estimation of user i_t's unknown bandit parameter, $\hat{\theta}_{i,t}$ (Lines 14-16) where $A^{-1}_{i,t-1}$ is the covariance matrix and $b_{i,t-1}$ is a normalizing matrix for user i at timestep $t-1$ that are used to compute the ridge regression solution of the coefficients [11]. On the other hand, if a user clustering has been established (length($\mathscr{G} > 0$)), then the cluster holding user i_t is set as $g_{s,t}$ (Line 8) and DABIP calculates $\hat{\theta}_{g_{s,t}}$, which represents the unknown bandit parameter for the entire cluster (Line 9).

Finally, to choose an arm, we compare the UCB using the user's unknown bandit parameter, $\hat{\theta}_{i,t}$ to the UCB using the average unknown bandit parameter of all users in cluster $g_{s,t}$, $\hat{\theta}_{g_{s,t}}$ (Lines 10-12). The maximum of these two UCB values is selected (Line 13). The reasoning behind this is that previous work has established that clustering users by unknown bandit parameter is an effective strategy for identifying users who behave similarly in a task, thus resulting in a collaborative filtering effect [6,5,12,13,1]. In datasets where changes in user features are not available or considered, these past works still represent the state of the art in clustering bandit algorithms. Our approach, by comparison, is to gain an advantage in datasets where user feature dynamics are available and changing. In these cases, we expect the collective bandit parameter of the cluster where user i_t resides, $\hat{\theta}_{g_{s,t}}$, to estimate expected behavior better than $\hat{\theta}_{i,t}$.

With an arm chosen and pulled, we observe the reward, r_t, then update user parameters and cluster parameters for the cluster that user i_t resides in (Lines 17-22). Then, any user features, $F_{i,t}$ are updated (Lines 23-24). This step will be tailored to the specific implementation and dataset, as the number, type, and sophistication of the user features will be entirely dependent on the problem definition and setup. The count for how many times user i_t has been considered is also updated (Line 25). Finally, the most up to date clusters, $\mathscr{G}_t$, are calculated and returned by the CLUSTER function (Line 26 - see Algorithm 2), which ends round t.

The second component of DABIP is clustering users based upon the similarity of their feature dynamics. The CLUSTER algorithm (Algorithm 2) assumes that each user has a set of features, F, of length q such that at any time, t, there exists $F_{i,t} = \{f_{i,1,t}, f_{i,2,t} ..., f_{i,q,t}\}$. The values of each individual user feature, $f_{i,q,t}$ may change over time, which can be tracked to cluster users based upon the similarity of their feature dynamics. To do this, one can observe the value of a feature at some initial timestep, then again at a later timestep, and calculate the absolute value of the difference between them. More formally, at some initial timestep, $T_{initial}$, we store the values of all features for a given user, i_t: $F_{i_t,T_{initial}}$. We also initialize a set Y_t that contains one value for each user such that $Y_t = \{y_{1,t}, y_{2,t} ... y_{i,t}\}$ and $y_{i,t}$ represents the number of times that the agent has made a recommendation to user i_t. Thus, each time user i_t is selected by the algorithm,

Algorithm 1 DABIP

Require: number of clusters to form s, cluster update frequency $T_{cluster}$, user feature dynamics update frequency $\mathscr{U}$, exploration parameter α

$T_{initial} \leftarrow 0$
for each $i \in N$ **do**
  $A_{i,0} \leftarrow I, b_{i,0} \leftarrow 0$
  $y_i \leftarrow 0$
for $t \leftarrow 1,2...T_{final}$ **do**
  receive $i_t \in N$ and obtain $C_t = \{c_{1,t}, c_{2,t}..., c_{k,t}\}$
  if length of $\mathscr{G} \geq 0$ **then**
    $g_{s,t} \leftarrow$ Cluster where i_t resides at round t
    $\hat{\theta}_{g_{s,t}} \leftarrow \frac{1}{|g_{s,t-1}|} \sum_{j \in g_{s,t-1}} A_{j,t-1}^{-1} b_{j,t-1}$
    $x_{cluster} \leftarrow argmax_{c_{a,t} \in C_t} \hat{\theta}_{g_{s,t}}^{\intercal} c_{a,t} + CB_{r,g_{s,t}}$ where $CB_{r,g_{s,t}} \leftarrow \frac{1}{|g_{s,t-1}|} \sum_{j \in g_{s,t-1}} \alpha \sqrt{c_{a,t}^{\intercal} A_{j,t-1}^{-1} c_{a,t}}$
    $\hat{\theta}_{i_t t} \leftarrow A_{i,t-1}^{-1} b_{i,t-1}$
    $x_{user} \leftarrow argmax_{c_{a,t} \in C_t} \hat{\theta}_{i_t t}^{\intercal} c_{a,t} + CB_{r,i}$ where $CB_{r,i} \leftarrow \alpha \sqrt{c_{a,t}^{\intercal} A_{i,t-1}^{-1} c_{a,t}}$
    $x_t \leftarrow max(x_{cluster}, x_{user})$
  else
    $\hat{\theta}_{i_t t} \leftarrow A_{i,t-1}^{-1} b_{i,t-1}$
    $x_t \leftarrow argmax_{c_{a,t} \in C_t} \hat{\theta}_{i_t t}^{\intercal} c_{a,t} + CB_{r,i}$ where $CB_{r,i} \leftarrow \alpha \sqrt{c_{a,t}^{\intercal} A_{i,t-1}^{-1} c_{a,t}}$
  pull x_t and observe reward r_t
  $A_{i,t} \leftarrow A_{i,t-1} + x_t x_t^{-1}$
  $b_{i,t} \leftarrow b_{i,t-1} + r_t x_t$
  if length of $\mathscr{G} \geq 0$ **then**
    $A_{g_{s,t},t} \leftarrow A_{g_{s,t},t-1} + x_t x_t^{-1}$
    $b_{g_{s,t},t} \leftarrow b_{g_{s,t},t-1} + r_t x_t$
  for $f_{i,q,t} \in F_{i,t}$ **do**
    update $f_{i,q,t}$ according to information gathered from problem setup and r_t
  $y_{i,t} \leftarrow y_{i,t} + 1$
  $\mathscr{G}_t \leftarrow CLUSTER(\mathscr{U}, Y, T_{cluster}, i_t)$

we can update $F_{i,t}$ based upon the observed user features at timestep t, and increment $y_{i,t}$ by 1. Once the agent has made a recommendation to a user $\mathscr{U}$ times, say at time T_{final}, such that $y_{i,t} = \mathscr{U}$, the feature dynamics for user i, δ_i, can be computed based upon how the features have changed between $T_{initial}$ and T_{final} (Algorithm 2 Line 2). The differences are summed over time to compute δ_i, and $\mathscr{U}$ is a hyperparameter that controls how often user feature dynamics are updated. After this calculation, $T_{initial}$ is set to T_{final} and $y_{i_t,t}$ is set to 0. The process repeats when $y_{i_t,t} = \mathscr{U}$ until all timesteps are complete.

By performing this operation for every user, we constantly have access to δ_i which represents the current dynamics of user i's features. We use the similarity between user's δ values to cluster them together, rather than $\theta_{i,t}$ as done in previous works [6,5,12,13,1]. To that end, we assume that there exists a set of clusters $\mathscr{G}$ of length s such that $\mathscr{G}_t = \{g_{1,t}, g_{2,t}...g_{s,t}\}$. For simplicity, we assume that each user must appear in exactly one

cluster and all users are split evenly amongst the clusters. This results in each cluster containing $\frac{n}{s}$ users. See Algorithm 2 for the full clustering pseudocode.

DABIP updates clusters after a period of timesteps have passed $T_{cluster}$. This is because calculating the dynamics of the user features requires observing changes in those features over a period of time. To re-cluster after every timestep would not allow sufficient time to observe any true dynamics, so we update δ_i for each user after every $\mathscr{U}$ timesteps in which that user is selected.

Algorithm 2 *CLUSTER*

Require: user feature dynamics update frequency $\mathscr{U}$, user update counts Y, cluster update frequency $T_{cluster}$, user i_t

if $y_i == \mathscr{U}$ **then**
  $\delta_i = \sum_{q=1}^{Q}\{|F_{i,t} - F_{i,T_{initial}}|\}$
  $T_{initial} \leftarrow t$
  $y_i \leftarrow 0$
if t % $T_{cluster} == 0$ **then**
  $\delta_{sorted} \leftarrow$ sort δ in ascending order
  $\mathscr{G}_t \leftarrow$ split(δ_{sorted},s) where split(x,y) splits x into $length(x)\%y$ groups each of size $\frac{length(x)}{y}$ + 1 and the rest of size $\frac{length(x)}{y}$
return $\mathscr{G}_t$