Two-Stage Clustering of Human Preferences for Action Prediction in Assembly Tasks

***Abstract*— To effectively assist human workers in assembly tasks a robot must proactively offer support by inferring their preferences in sequencing the task actions. Previous work has focused on learning the dominant preferences of human workers for simple tasks largely based on their intended goal. However, people may have preferences at different resolutions: they may share the same high-level preference for the order of the sub-tasks but differ in the sequence of individual actions. We propose a two-stage approach for learning and inferring the preferences of human operators based on the sequence of sub-tasks and actions. We conduct an IKEA assembly study and demonstrate how our approach is able to learn the dominant preferences in a complex task. We show that our approach improves the prediction of human actions through cross-validation. Lastly we show that our two-stage approach improves the efficiency of task execution in an online experiment and demonstrate its applicability in a real-world robot-assisted IKEA assembly.**

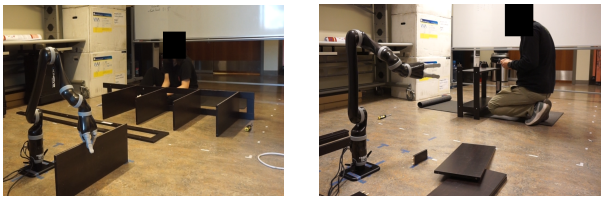
I. Introduction

There are many assembly, service, repair, installation, and construction applications where many different workers may need to perform the same task. For example, consider the task of replacing a bearing on a machine tool located at the shop floor. Even though these tasks have fixed guidelines, there is some variability in the way each worker performs a task because of individual preferences. Robots can help in improving the efficiency of such tasks by adapting to the individualized preferences of human workers and proactively supporting them in their task.

While the space of possible preferences can be very large, previous work has shown that people can be grouped to a few "dominant" preferences: In human-agent teams, users cluster to a set of "reasonable" behaviors, where people in the same cluster have similar beliefs [1]. Similar groupings exist in game playing [2]–[5] and education [6]–[9].

What makes the problem particularly challenging in complex tasks, is that these dominant preferences exist in *different resolutions*. In an IKEA assembly study that we use as a proof-of-concept throughout the paper, we observed that some participants preferred to assemble all shelves in a row, while others alternated between assembling the shelves and assembling the boards (Fig. 1). Moreover, within the first group, participants also differed in how they connected the shelves to the boards: some connected all shelves to boards on just one side first (as in Fig. 1(a)), while others preferred to connect each shelf to boards on both sides.

Therefore to effectively assist a new worker the robot must *learn the dominant preferences of workers at different levels of abstraction*. Using insights from clickstream analysis [10], we propose abstracting action sequences to sequences of *events*, where events are sub-sequences that are shared across different tasks and workers. A high-level preference is captured by a sequence of events, while a lower-level preference captures how each event is executed. We propose clustering users on two levels, over events and over actions within each event, as opposed to clustering based on the sequences of individual actions as in previous work [11].



(a) User preference 1 (b) User preference 2

Fig. 1: Robot-assisted IKEA assembly. (a) Some users preferred to connect all the shelves in a row. (b) Other users connected just two shelves to the small boards first. For effectively assisting the users, a robot must predict their preference and supply the parts accordingly.

We show the applicability of our method in a user study where 20 users assemble an IKEA bookcase. We learn the high and low-level preferences of the users, which enables accurate prediction for a new user performing the same task. Through an online assembly experiment we show that assisting users this way, improves task efficiency. We finally show the applicability of the system in a real-world robot-assisted IKEA assembly demo.

II. Related Work

Clustering dominant preferences. Similar to prior work in task planning, we consider preference as the subset of action sequences from the set of multiple sequences that solve a given task [12]. Related work includes learning user preferences during an assembly task from demonstrations [13], [14], via interactive reinforcement learning [12], [15] or active reward learning [16]–[18], where previous demonstrations can be used as priors [19], [20]. While each user can have a different preference, our goal is to cluster the users to a small set of *dominant preferences*, like human motion prototypes [21] for robot navigation or human preference stereotypes for human-robot interaction [22].

Most relevant to ours is prior work in identifying dominant user preferences from sequences of user actions in a surface refinishing task [11]. Users with similar action transition matrices were clustered using a hard Expectation Maximization (EM) algorithm to obtain the dominant clusters. However the task was simple as each user preference corresponded to a different final robot configuration, and thus one-stage clustering was sufficient for capturing the dominant preferences.

Clustering click-event sequences. The problem of grouping users based on their preferred sequence of doing tasks is similar to the problem of *clickstream analysis* [10], [23]–[25]. A clickstream is a sequence of timestamped events generated by user actions on a webpage and hence is comparable to a sequence of actions. Prior work clusters clickstreams of multiple users based on their longest common sub-sequence [23] or frequency of sub-sequences [25].

To cluster users at different resolutions, prior work uses Levenshtein distance to form macro and micro preference clusters [10]. Recent work [24] uses hierarchical clustering to users first into high-level clusters and then into subsequent low-level clusters. We bring these insights from clickstream clustering to the robotics problem of clustering the action sequences of users in assembly tasks.

III. Methodology

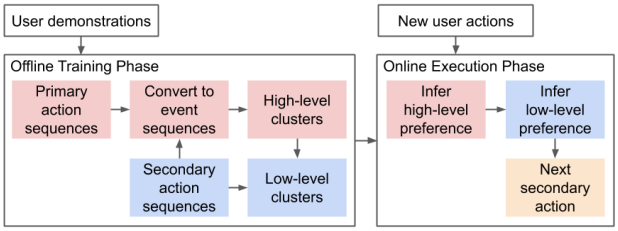


Fig. 2: Flowchart of our proposed two-stage clustering and inference method

The proposed method consists of two phases (see Fig. 2): (1) an offline training phase which takes as input a set of user demonstrations of the entire assembly task and learns the dominant preference clusters at different resolutions, and (2) an online execution phase where we estimate the probability of a new user belonging to one of the clusters based on their observed actions, and predict the next robot action.

Based on our observation that users prefer to perform actions that require the same parts in a row; we first convert each user demonstration into a sequence of such $events$. Thus each event in a demonstration requires a specific set of parts to be supplied by the robot i.e. a specific set of *secondary actions* (non-critical actions like supplying parts). The *high-level preference* of each user is thus the order in which they perform the events. Further, for each event in a high-level preference, users may have a different *low-level preference* for the order in which to supply the set of parts i.e. order in which the secondary actions must be performed.

We learn the high and low-level preferences in the offline phase by clustering users based on their sequence of events and sequence of secondary actions respectively. Accordingly in the online execution phase we first infer the high-level preference of a new user and then infer the low-level preference to determine the next secondary action to execute.

IV. Offline Training Phase

We assume a set of demonstrated action sequences X. Similar to prior work [26], we distinguish the actions A in the demonstrated sequences into two types: **primary actions** ($a^p \in A^P$) which are the task actions that *must be performed by the user*, and **secondary actions** ($a^s \in A^S$) which are the supporting actions that *can be delegated to the robot*.

In the training phase, each user demonstration $x \in X$ is some sequence of primary and secondary actions e.g., $x = [a^s_1, a^s_2, a^p_1, a^p_2, \ldots, a^s_M, a^p_N]$ which has M secondary and N primary actions. We wish to model the online execution as a turn-taking model where at each time step t, the robot performs a set of secondary actions s_t, followed by the user performing a primary action a^p_t. Thus we can re-write the user demonstrations as a sequence of alternate secondary and primary actions e.g., $x = [s_1, a^p_1, s_2, a^p_2, \ldots, s_N, a^p_N]$. Where, in this example, $s_1 = [a^s_1, a^s_2]$ is the set of secondary actions that must be executed before the primary action a^p_1, while $s_2 = [NOOP]$ means that no other secondary action is required to be executed before a^p_2, and so on.

A. Converting User Demonstrations to Event Sequences

We first convert each user demonstration to a sequence of *events*. An event e is defined as - consecutive primary actions that require the same set of secondary actions. Thus an event e from time step t_a to t_b is the sequence of primary actions $p_{t_a:t_b} = [a^p_{t_a}, \ldots, a^p_{t_b}]$ with preceding secondary actions $s_{t_a:t_b} = [s_{t_a}, \ldots, s_{t_b}]$ where,

$$s_i \subseteq \{s_{t_a:i-1}\} \quad \forall i \in [t_a + 1, \ldots, t_b] \tag{1}$$

Example: Consider the demonstration $x = [[a^s_1, a^s_2], a^p_1, [NOOP], a^p_2, [a^s_3], a^p_3]$. Here the secondary actions $[a^s_1, a^s_2]$ precede the first primary action a^p_1, while $NOOP$ precedes the next primary action a^p_2. As $s_2 = [NOOP] = \emptyset$ (null set) is a subset of the set of previous secondary actions $\{a^s_1, a^s_2\}$, we consider the primary actions $p_{1:2} = [a^p_1, a^p_2]$ to belong to the same event. Now if, the next secondary action is also a subset of its previous secondary actions i.e. $s_3 = [a^s_3] \subseteq \{s_{1:2}\}$, all primary actions belong to the same event $e_{1:3}$, with $p_{1:3} = [a^p_1, a^p_2, a^p_3]$ and $s_{1:3} = [[a^s_1, a^s_2], [NOOP], [a^s_3]]$. In this case, the event sequence is $x^e = [e_{1:3}]$. However if $[a^s_3] \not\subseteq \{a^s_1, a^s_2\}$, then a^p_3 will belong to a new event $e_{3:3}$, with $p_{3:3} = [a^p_3]$ and $s_{3:3} = [a^s_3]$. Thus, in this case, the event sequence will be $x^e = [e_{1:2}, e_{3:3}]$, where $e_{1:2} = (p_{1:2}, s_{1:2})$.

Two events are equal if they share the same *set of secondary actions* - $\{s_{t_a:t_b}\}$.

B. Two-Stage Clustering

*1) **Clustering Event Sequences**:* We cluster the converted event sequences $x^e \in X^e$ of each user to determine the *dominant high-level clusters* $z_h \in Z_H$. Details of the method for clustering the sequences are provided in Sec. IV-C. In the online execution phase, we will infer the high-level preference z^*_h of a new user to determine the current event, and thus the *set of secondary actions* for that event.

*2) **Clustering Secondary Action Sequences**:* To learn the low-level preferences for each event e in the dominant clusters Z_h, we cluster the participants based on the sequence of secondary actions in each instance of the event e, to determine the *dominant low-level clusters* $z_l \in Z^e_L$. In the online execution phase, we will infer the low-level preference of the new user for the current event and thus determine the *sequence in which the secondary actions must be executed*.

C. Clustering Method

For each stage, we apply *hierarchical clustering* [27], [28] considering a modification (d_{mod}) of the *Levenshtein distance* [29] (d_{lev}) used in clickstream analysis [10], [30] as the distance metric. The number of clusters formed depends on a distance threshold. We generate clusters for increasing distance thresholds, and select the optimal distance based on the variance ratio criterion (VRC) [31] (also called *calinski-harabasz score*) which is a common metric for distance-based clustering [32].

V. Online Execution Phase

In the online execution phase we infer the high and low level preferences of new users as they are executing the task. At each time step t, as the user performs a primary action, the robot predicts the next secondary action.

1) Inferring high-level preference: At each time step t, we observe the primary action of a new user and append it to the actions observed so far. We then convert the current sequence of actions $x_{1:t}$ of the new user to a sequence of events $x^e_{1:t}$ in the same way as in the offline phase. We use Bayesian inference to predict the high-level preference by computing the probability of observing the event sequence $x^e_{1:t}$ for each high-level cluster $z_h \in Z_H$.

$$p(z_h|x^e_{1:t}) \propto p(x^e_{1:t}|z_h)p(z_h)$$

Here, $p(z_h)$ is simply the ratio of the number of users in the cluster z_h to the total number of users in all the clusters Z_H. However for calculating $p(x^e_{1:t}|z_h)$ we re-compute the event sequences of users in z_h considering their action sequences only up to the time step t. Therefore:

$$p(x^e_{1:t}|z_h) = \frac{\text{No. of users in } z_h \text{ with same } x^e_{1:t}}{\text{Total no. users in } z_h}$$

We then determine the high-level preference as $z^*_h = \arg\max_{z_h \in Z_H} p(z_h|x^e_{1:t})$. If there are two high-level clusters with the same maximum probability, we select one randomly.

2) Inferring low-level preference: Once we infer the high-level preference z^*_h of the new user, we identify the most likely event sequence x^{e*} in that cluster. We assume that the new user follows that sequence x^{e*} to index the event ongoing at time step $t+1$ i.e. e_{t+1} (in a slight abuse of notation). Given the sequence of secondary actions $s^{e_{t+1}}$ performed so far within the event e_{t+1}, we use Bayesian inference to infer the low-level preference $z_l \in Z_L^{e_{t+1}}$, where $Z_L^{e_{t+1}}$ is the set of low-level preferences for the event e_{t+1}:

$$p(z_l|s^{e_{t+1}}) \propto p(s^{e_{t+1}}|z_l)p(z_l)$$

We select the most likely low-level preference z^*_l, identically to the high-level preference case. The robot can then perform the most likely secondary action s_{t+1} in z^*_l to proactively assist the user. If the user accepts s_{t+1} we append that to $x_{1:t}$. If the user rejects s_{t+1} and performs a different secondary action s'_{t+1} instead, we append s'_{t+1} to $x_{1:t}$.

VI. User Study

We wish to show that the proposed method can effectively identify the dominant preferences of users in an IKEA bookcase assembly task, and use the found preferences to accurately predict the next secondary action of a new user.

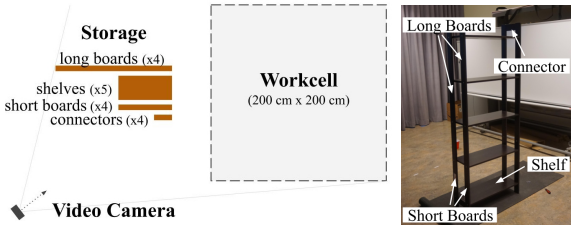


Fig. 3: Top view of study setup (left) and assembled bookcase (right)

A. Study Setup

We conducted an IKEA bookcase assembly study with 20 subjects, out of which 18 ($M = 11$, $F = 7$) completed the assembly. We provided each subject with a labelled image of the bookshelf (Fig. 3) and demonstrated how the connections are made. Users then practiced the connections for five minutes. We asked the subjects to plan their preferred sequence with the goal of assembling the bookcase as fast as they can, but *did not provide any instructions regarding the order of the assembly*. We recorded a video of each user demonstration, and annotated their actions using ELAN [33].

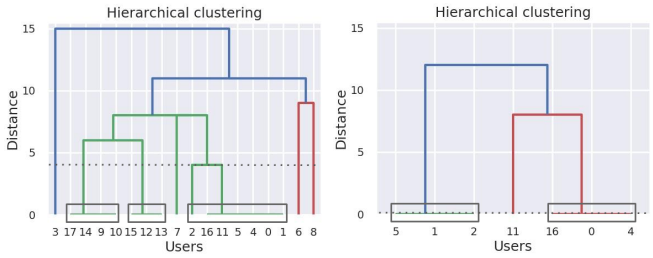
B. Analysis of User Preferences

We consider bringing a part from the storage to the workcell as a *secondary action* and all connections in the assembly as *primary actions*. The bookcase has 4 types of parts: long boards, short boards, connectors and shelves (total 17 parts), and 32 different connections. Thus each user demonstration is a sequence of $N = 32$ time steps.

Fig. 4: Event sequences. 'boards' refers to an event of connecting long and short boards, 'con' refers to an event of connecting assembled boards using connectors, 'shelves' refers to an event of connecting shelves.

The assembly task is fairly complex; the 32 primary actions can be ordered in more than 24! ways! However most users preferred to perform similar actions in a row: 14 users performed all long and short board connections in a row, and 7 users connected all connectors and all shelves in a row.

Event Sequences. We first visualize the sequence of events for each user (shown in Fig. 4). Users



(a) High-level clusters (b) Low-level clusters

Fig. 5: IKEA assembly user study: (a) Dominant high-level clusters (grey rectangles) formed by clustering the event sequences. (b) Dominant low-level clusters (grey rectangles) formed by clustering the secondary actions sequences for the event of all shelf connections (shown in yellow in Fig. 4).

$[0, 1, 4, 5, 11, 16]$ had the same event sequence: short and long board connections (shown in grey), connector and board connections (green), and shelf and board connections (yellow). Similarly, other groups of users - $[12, 13, 15]$, and $[3, 9, 10, 14, 17]$ also had same event sequences.

Dominant clusters. To find the high-level preferences, we cluster the event sequences using the modified Levenshtein distance metric (Sec. IV-C) which results in the hierarchy shown in Fig. 5(a). We partition the users at a distance threshold of $d_{mod} = 4$ (shown as dotted line) based on the VRC [31] to obtain three dominant high-level clusters (shown in grey rectangles). Therefore we see that users cluster to a small set of dominant preferences despite not being provided any instructions regarding the order of assembly.

Users had different preferences for the same event as well. For example, within the event of performing all shelf connections in a row, shown in yellow in Fig. 4, clustering the sequences of secondary actions of all users results in the hierarchy shown in Fig. 5(b). Partitioning at the optimal distance threshold $d_{mod} = 0$ gives us two dominant low-level clusters (shown in grey rectangles). Users 0, 4 and 16 prefer to connect each shelf to boards on one side before connecting on the other side, whereas users 5, 1 and 2 prefer to connect each shelf to boards on both sides at a time.

C. Evaluation & Results

We want to show that our two-stage clustering approach is useful for predicting the next secondary action of a new user. Therefore we make the hypotheses: **H1** - Clustering users based on their sequence of *events* improves the accuracy of predicting the next secondary action, compared to clustering based on the sequences of individual primary actions (baseline). **H2** - The two-stage clustering of users improves the accuracy of predicting the next secondary action, compared to clustering based on just the sequences of events.

We perform leave-one-out cross-validation, and compare the accuracy of predicting the next secondary action at each time step, averaged over 100 trials[1].

1) Importance of High-level Clusters: We compare the prediction results from clustering based on sequence of events (only low-level) to clustering based on the sequence of primary actions. While clustering the abstract event sequences leads to three well-defined dominant clusters (Fig. 5(a)), clustering primary action sequences generates only two clusters for a distance threshold of 63 (based on VRC) with a high variability within each cluster. A two-tailed paired t-test showed a statistically significant difference ($t(17) = -3.232$, $p = 0.004$) in prediction accuracy averaged over all timesteps and trials, between the event-based method ($M = 0.796$, $SE = 0.041$) and the baseline ($M = 0.693$, $SE = 0.041$). This supports hypothesis **H1**.

(a) Importance of high-level (b) Importance of low-level

Fig. 6: Cross-validation accuracy for 18 users averaged over 100 trials. (a) Clustering on event sequences compared to clustering primary action sequences. (b) Two-stage clustering compared to clustering event sequences.

2) Importance of Low-level Clusters: We also compare the prediction accuracy with and without the second stage clustering, averaged over all timesteps and trials. A paired t-test showed a statistically significant difference ($t(17) = -2.34$, $p = 0.03$) between predicting with event sequences only ($M = 0.796$, $SE = 0.041$) and predicting with the two-stage framework ($M = 0.820$, $SE = 0.043$). This supports hypothesis **H2**.

VII. Robot-Assisted Assembly

We also wish to show that predicting the next secondary action using our proposed two-stage approach improves the efficiency of the task. Therefore, we conducted an online experiment on Amazon Mechanical Turk where 80 users played a shelf assembly game three times each, with and without robot assistance. However only 52 (out of 80) users completed all game trials and survey questions. For these users, a paired t-test showed a statistically significant difference ($t(51) = 2.155$, $p = 0.036$) in the time required to assemble with ($M = 55.794$, $SE = 3.439$) without assistance ($M = 66.948$, $SE = 6.05$). This informs us that performing the secondary actions as per our proposed method can indeed be helpful for the users.

Lastly we also perform a real-world IKEA assembly task with two participants[2]. For both the participants, we observed that robot assistance based on our two-stage approach, allowed the user to stay inside the workcell and perform just the primary actions. We hypothesize that this can help to not only improve the efficiency of the task but also to reduce human effort, and we plan to explore this in future work.

[1]We average the accuracy over 100 trials as if the probabilities for two (or more) preferences were same while selecting the most likely dominant preference, we pick one at random with uniform probability.

[2]We were unable to perform a complete human subjects experiment because of COVID-19 restrictions.

VIII. Conclusion

We proposed a two-stage clustering approach, inspired by clickstream analysis techniques, to identify the dominant preferences of users at different resolutions in a complex IKEA assembly task. In future work, we would evaluate our two-stage approach on different assembly tasks, and explore the problem of transferring the preferences for an event in one assembly task to a similar event in a slightly different assembly task (e.g., different number of parts or actions).

References

[1] D. V. Pynadath, N. Wang, E. Rovira, and M. J. Barnes, "Clustering behavior to recognize subjective beliefs in human-agent teams," in *AAMAS*, 2018, pp. 1495–1503.

[2] D. Ramirez-Cano, S. Colton, and R. Baumgarten, "Player classification using a meta-clustering approach," in *Proceedings of the 3rd Annual International Conference Computer Games, Multimedia & Allied Technology*, 2010, pp. 297–304.

[3] G. F. Tondello, A. Mora, and L. E. Nacke, "Elements of gameful design emerging from user preferences," in *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. ACM, 2017, pp. 129–142.

[4] R. Thawonmas, M. Kurashige, K.-T. Chen, *et al.*, "Detection of landmarks for clustering of online-game players." *IJVR*, vol. 6, no. 3, pp. 11–16, 2007.

[5] C. Holmgård, J. Togelius, and G. N. Yannakakis, "Decision making styles as deviation from rational action: A super mario case study," in *AIIDE*, 2013.

[6] T. Peckham and G. McCalla, "Mining student behavior patterns in reading comprehension tasks." *International Educational Data Mining Society*, 2012.

[7] A. Merceron and K. Yacef, "Clustering students to help evaluate learning," in *IFIP World Computer Congress, TC 3*. Springer, 2004, pp. 31–42.

[8] E. Thuillier, L. Moalic, S. Lamrous, and A. Caminada, "Clustering weekly patterns of human mobility through mobile phone data," *IEEE Transactions on Mobile Computing*, 2017.

[9] B. Furletti, L. Gabrielli, C. Renso, and S. Rinzivillo, "Identifying users profiles from mobile calls habits," in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*. ACM, 2012, pp. 17–24.

[10] P. Antonellis, C. Makris, and N. Tsirakis, "Algorithms for clustering clickstream data," *Information Processing Letters*, vol. 109, no. 8, pp. 381–385, 2009.

[11] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in *2015 HRI*. IEEE, 2015.

[12] T. Munzer, M. Toussaint, and M. Lopes, "Preference learning on the execution of collaborative human-robot tasks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 879–885.

[13] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[14] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, 2020.

[15] S. C. Akkaladevi, M. Plasch, C. Eitzinger, S. C. Maddukuri, and B. Rinner, "Towards learning to handle deviations using user preferences in a human robot collaboration scenario," in *International Conference on Intelligent Human Computer Interaction*. Springer, 2016, pp. 3–14.

[16] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions." in *Robotics: Science and Systems*, 2017.

[17] E. Biyik and D. Sadigh, "Batch active preference-based learning of reward functions," in *Conference on Robot Learning*, 2018, pp. 519–528.

[18] E. Bıyık, N. Huynh, M. J. Kochenderfer, and D. Sadigh, "Active preference-based gaussian process regression for reward learning," *arXiv preprint arXiv:2005.02575*, 2020.

[19] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions by integrating human demonstrations and preferences," 2019.

[20] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences," *arXiv preprint arXiv:2006.14091*, 2020.

[21] M. Luber, L. Spinello, J. Silva, and K. O. Arras, "Socially-aware robot navigation: A learning approach," in *IROS*. IEEE, 2012, pp. 902–907.

[22] A. R. Wagner, "Using cluster-based stereotyping to foster human-robot cooperation," in *IROS*. IEEE, 2012, pp. 1615–1622.

[23] A. Banerjee and J. Ghosh, "Clickstream clustering using weighted longest common subsequences," in *Proceedings of the web mining workshop at the 1st SIAM conference on data mining*, vol. 143, 2001, p. 144.

[24] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao, "Unsupervised clickstream clustering for user behavior analysis," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 225–236.

[25] G. Wang, X. Zhang, S. Tang, C. Wilson, H. Zheng, and B. Y. Zhao, "Clickstream user behavior models," *ACM Transactions on the Web (TWEB)*, vol. 11, no. 4, pp. 1–37, 2017.

[26] E. C. Grigore, A. Roncone, O. Mangin, and B. Scassellati, "Preference-based assistance prediction for human-robot collaboration tasks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4441–4448.

[27] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola, "Fast optimal leaf ordering for hierarchical clustering," *Bioinformatics*, vol. 17, no. suppl_1, pp. S22–S29, 2001.

[28] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.

[29] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[30] L. Yujian and L. Bo, "A normalized levenshtein distance metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.

[31] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics - Theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[32] A. EL MEZOUARY, B. HMEDNA, and B. Omar, "An evaluation of learner clustering based on learning styles in mooc course," in *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*. IEEE, 2019, pp. 1–5.

[33] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes, "ELAN: a professional framework for multimodality research," in *5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2006, pp. 1556–1559.