Predicting Privacy Preferences for Smart Devices as Norms

Marc Serramia
King's College London
London, United Kingdom
marc.serramia_amoros@kcl.ac.uk

William Seymour
King's College London
London, United Kingdom
william.1.seymour@kcl.ac.uk

Natalia Criado
Universitat Politècnica de València
València, Spain
ncriado@upv.es

Michael Luck
King's College London
London, United Kingdom
michael.luck@kcl.ac.uk

ABSTRACT

Smart devices, such as smart speakers, are becoming ubiquitous, and users expect these devices to act in accordance with their preferences. In particular, since these devices gather and manage personal data, users expect them to adhere to their privacy preferences. However, the current approach of gathering these preferences consists in asking the users directly, which usually triggers automatic responses failing to capture their true preferences. In response, in this paper we present a collaborative filtering approach to predict user preferences as norms. These preference predictions can be readily adopted or can serve to assist users in determining their own preferences. Using a dataset of privacy preferences of smart assistant users, we test the accuracy of our predictions.

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent systems**; • **Security and privacy**;

KEYWORDS

Norms; Privacy; Preferences; Collaborative filtering; Smart devices

ACM Reference Format:
Marc Serramia, William Seymour, Natalia Criado, and Michael Luck. 2023. Predicting Privacy Preferences for Smart Devices as Norms. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Artificial intelligence (AI) technologies are making their way into our daily lives and into our homes. We have grown accustomed to using our devices to call friends, set reminders, or check the weather.However, for these technologies to be adopted and trusted by users, they must act as users expect, and this problem is especially apparent in the area of privacy preferences. Studies show that users are deeply concerned about how their data is being collected online [10]. Interestingly, while they expect AI to act as they desire, they are unwilling to spend time setting their preferences. For example, despite users' concerns about privacy, studies show that they ignore or blindly accept cookie banners [8] and privacy policies in social networks [16]. Furthermore, in social networks, a large proportion of users do not change default privacy settings [9]. This can be explained as a result of privacy fatigue [4], the sensation of loss of control and futility over protecting one's privacy. This leads to privacy cynicism, when users do not adopt a privacy protecting behaviour even if they are concerned about their privacy [6]. Thus, the current approach of directly asking the user when a preference is unknown but needed fails to capture the user's true preferences. Additionally, continual questioning prevents users from achieving their objectives with the device. In response, this paper advocates for an approach that can understand user preferences with less user involvement, in turn bringing more importance to user interactions whenever such preferences are needed.

A particular platform in which capturing privacy preferences is challenging and yet essential is that of smart speakers and other smart personal assistants. These devices have benefited from widespread early adoption, and it is estimated that 500 million units were installed in the last quarter of 2021 [23]. Nonetheless, the early adoption of these technologies means that they still have several vulnerabilities that pose a threat to the security and privacy of their users [5]. Indeed, there have already been cases reported in which smart assistants have not functioned as expected; for example, a smart speaker recorded and sent a private conversation without the user's consent [27]. These situations hinder user trust in the technology and can ultimately lead users to limit the functionalities of the devices used, or even to adopting coping mechanisms [1].

This paper describes an alternative approach that addresses the issues outlined above. The critical observation underpinning our approach is that smart devices are just one part of a larger ecosystem (e.g. see [5] for a description of the ecosystem of smart speakers), and they interact and share data with agents like services, apps, and other devices. For example, a smart watch might send a voice recording to a smart speaker, or might share the wearer's heart rate with a health app. In this respect, we can understand this ecosystem as a multi-agent system in which the use of norms can help to regulate these interactions, implementing privacy preferences.

Norms can effectively summarise complex privacy preferences into simple sets of regulations, as shown by Abdi et al. [2], who gathered over 800 privacy preferences on data transmissions, yet produced just 17 norms. Moreover, although here we assume no knowledge of the domain, if such knowledge is available there exist techniques to generalise norms (see [13, 14] for an example) or find and resolve inconsistencies among them [26]. Furthermore, norms are also used by people, and are naturally understood by them, representing a good base upon which to construct explanations.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

This can be used not only to generate explanations for a user if something unexpected happens, but also to tailor interactions with a user to validate predicted norms. Norms are regarded as expected patterns of behaviour [28], causing agents (each component in the smart device ecosystem) to coordinate better and function more efficiently. As an example with smart devices, imagine a service knows in advance the privacy norms of a user with regard to each component of the ecosystem. If this service needs to interact with other components, it can use the user's norms to adapt its behaviour to avoid violating norms or to avoid performing unregulated transmissions of information, which might require consent.

As informally outlined in [18], we can exploit the large user bases of smart devices to use knowledge of previously specified privacy preferences to infer new preferences or to assist users in specifying their preferences. In particular, we aim to exploit similarities between users to make privacy preference predictions using collaborative filtering [7]. Effectively, we see the smart device ecosystem as a multi-layered multi-agent system. The lower level represents the multi-agent system associated with a single user (that is, the user's device, and the other devices, skills, and services that can be accessed from it). The higher level is that of the multi-agent system composed of all the users. Our approach is centred on the norm creation stage in the lower level multi-agent systems related to each user. Therefore, each device user has its own associated set of norms, and all agents within its lower level multi-agent system, be they devices, skills, or other services, are informed and affected by the norms whenever they want access to the user's personal data. While many researchers have studied different approaches to constructing norm systems, like norm synthesis [12, 13] or norm emergence [17, 21, 24], we are not aware of any similar approach like the collaborative filtering presented here.

In taking this approach, we make the following contributions.

- Formalisation of the problem of predicting norms to ensure that computational behaviour aligns with user preferences. This is divided into two subproblems, namely preference approximation (predicting unknown user preferences) and norm inference (inferring norms from predicted preferences).
- Formalisation of preference prediction functions. We provide a specific example of this type of function based on the preferences of similar users.
- Inference of norms from the predicted preferences, and specification of different methods to do so based on the confidence of the prediction or other variables.

The paper is structured as follows: Section 2 formalises the core problems we aim to address in the paper. In Section 3 we detail the process of predicting preferences. We then use these predictions to infer norms in Section 4. Section 5 is dedicated to validate our findings. In Section 6 we discuss related work. Finally, in Section 7 we discuss conclusions and future work.

2 PROBLEM DEFINITION

Consider a set of users U and a set of agents Ag, such that $ag_u \in Ag$ is the agent (i.e. the smart device) of $u \in U$[1]. Consider also a finite number of elements $X = \{x_1, \dots, x_{|X|}\}$ over which users have preferences. These elements will commonly be actions an agent can perform, but can also be more complex, for example containing the context in which an action happens (e.g. "share if the user is notified"). For generality purposes, we do not specify the formalisation of these elements since, for the problem definition (and our proposed resolution), it is not necessary. This not only allows our notation to be kept simple, but it also allows us to define the preference domain with as much or as little complexity as needed. Given an element $x \in X$, we assume the user's preference towards x is a number in $[-1, 1]$, where 1 means the user totally approves of x, -1 means the user totally disapproves of x, and 0 means neutrality towards x. Note that we can make this assumption without loss of generality as we can always transform any user preferences into $[-1, 1]$[2]

For each user u and agent ag_u, we consider the following tuples of preferences.

- The user's preference profile p_u represents the real preferences of user $u \in U$. Note that $p_u \in [-1, 1]^{|X|}$, and the i^{th} position in the tuple represents the preference of user u towards element x_i.
- The agent's preference profile p_{ag_u} represents the preferences of user u known by agent ag_u. This tuple has the same structure as p_u, but unlike p_u, this tuple has gaps of knowledge. We represent an unknown preference as $\bigcirc$, therefore $p_{ag_u} \in ([-1, 1] \cup \{\bigcirc\})^{|X|}$.

Having introduced these elements, we now present a running example with smart personal assistants, which we use throughout the paper to illustrate the concepts we introduce.

Example 1. *We consider users u_1, u_2 and u_3, who have smart personal assistants ag_{u_1}, ag_{u_2} and ag_{u_3} respectively. We consider three elements over which users have preferences: sharing data with the AI assistant manufacturer (x_1), with internet provider (x_2), and with developers of third party skills (x_3). When it comes to the user's real preferences, we have: $p_{u_1} = (-1, -1, -1)$, $p_{u_2} = (-1, -1, -1)$, $p_{u_3} = (1, -1, 1)$. As for the agent's known preferences, we have: $p_{ag_{u_1}} = (-1, -1, \bigcirc)$, $p_{ag_{u_2}} = (-1, \bigcirc, -1)$, $p_{ag_{u_3}} = (1, \bigcirc, 1)$.*

Finally, we define a process to complete preferences, noted as *comp*. This process takes p_{ag_u} and completes it, producing $p^*_{ag_u} \in [-1, 1]^{|X|}$. Note that in $[-1, 1]^{|X|}$ we can assess distances between preference tuples (which are points in the space). With this in mind, we can formalise the first problem we address in this paper.

Def. 1 (Preference approximation problem). *Consider the space $[-1, 1]^{|X|}$, and dis a distance function in this space, the preference approximation problem consists of finding the process comp, with the aim of minimising the distance $dis(p_u, p^*_{ag_u})$.*

Example 2. *The process to complete preferences could be, for example, asking the users about their preferences directly. For example, user u_1 interacts with a third party skill that requires unknown preferences and therefore asks the user about them. The user might want to use*

[1] To simplify, and without loss of generality we assume that for each $u \in U$ there is only one $ag_u \in Ag$. Note that if one user had more than one device, we could consider a mock second user.

[2] On the one hand, numerical preferences can be re-scaled into $[-1, 1]$, because the number of users and the number of elements is finite and therefore preferences will always be bounded. On the other hand, ordinal preferences will also be bounded and can be transformed into numerical preferences in $[-1, 1]$.

the skill and responds affirmatively automatically (against their real preferences), thus $p^*_{ag_{u_1}} = (-1, -1, 1)$. *Then, considering the euclidean distance, we would have* $dis(p_{u_1}, p^*_{ag_{u_1}}) = 2$, *which means that, in this case, this process can be improved.*

Our final aim is to align agent behaviour to user preferences. To that end, we resort to norms to regulate how each agent ag_u behaves. Note that there is no standard definition of norm; for example, [29] considers rewards and punishments in norms, whereas [12] ignores these and instead considers the context of application of the norm. In our case, we use a very simple definition of norm in support of generality, as a more complex definition would require domain knowledge which would hinder the applicability of our approach.

Def. 2 (Norm). *Given an element* $x \in X$, *a norm is a structure* $\theta(x)$, *where* $\theta \in \{Prh, Per\}$, *where* $Prh(x)$ *is the norm prohibiting* x, *whereas* $Per(x)$ *is the norm permitting* x.

Having defined our notion of norm, we can now define the second problem we address in this paper, as follows:

Def. 3 (Norm inference problem). *Given an agent* $ag_u \in Ag$ *and its completed preferences* $p^*_{ag_u}$, *the norm inference problem consists of enacting preferences in* $p^*_{ag_u}$ *as norms, such that when following them, the agent will behave as expected by user* u.

Example 3. *Supposing we correctly completed the preferences of user* u_3 ($p^*_{ag_u} = (1, -1, 1)$), *our aim would be to find a way to encode these preferences as norms, those close to 1 into permission norms, and those close to -1 into prohibition norms. In this case,* $Per(x_1)$, $Prh(x_2)$, *and* $Per(x_3)$.

3 PREFERENCE PREDICTION

In this section we consider how to predict a user's preference toward an element x for which we don't know their preference. To do so, we will infer preferences from similar users. As argued above, we can assume that users who share similar views on known preferences will also share similar views on unknown ones. Our aim is to formally define a separation measure between users so that we can build predictions based on a set of users deemed similar enough. This can be an aggregation of the preferences of similar users over the element in question. First, however, we must introduce some preliminary notation and definitions and, to simplify, we reuse the notation introduced above. Given a user $u \in U$ with corresponding agent ag_u and an element $x \in X$, we note the real preference of u towards x as $p_u(x)$, the known preference of u towards x by agent ag_u as $p_{ag_u}(x)$, and the preference of u towards x by agent ag_u after the prediction process as $p*_{ag_u}(x)$. First, we define common known preference elements. Given a pair of users, the common known preference elements are those elements for which we know both users' preferences, and can be used to measure their separation.

Def. 4 (Common known preference elements). *Consider two users* u_1, u_2, *their agents* ag_{u_1}, ag_{u_2}, *and the users' known preferences* $p_{ag_{u_1}}$ *and* $p_{ag_{u_2}}$. *We call the common known preference elements of* ag_{u_1} *and* ag_{u_2}, *the set of elements for which we know both agents' preferences. If* $p_{ag_{u_1}} = (p^1_1, \ldots, p^{|X|}_1)$ *and* $p_{ag_{u_2}} = (p^1_2, \ldots, p^{|X|}_2)$, *this is formalised as:*

$$C(ag_{u_1}, ag_{u_2}) = \{x_i | p_{u_1}(x_i), p_{u_2}(x_i) \neq \bigcirc\}$$

With these preliminary definitions, we now turn to formalising a measure of separation between users. While we can define distance functions in the space of real user preferences $[-1, 1]^{|X|}$, our aim here is to assess distances between users with only partial knowledge of their preferences. Therefore, we want to define a distance in the space of known preference tuples $([-1, 1] \cup \{\bigcirc\})^{|X|}$, though in this case we do not want a strict distance function, but a more relaxed version of a distance function. Consider, for example, the points $(1, \bigcirc, \ldots, \bigcirc)$ and $(\bigcirc, \ldots, \bigcirc, 1)$, where a distance function would have to assign a distance between these two points, but there are no commonly known preferences between them, so we choose not to assign a distance in this case. In other words, the separation between two users should only depend on their commonly known preferences. Hence, instead of defining a formal distance we define a function, called a preference separation function, for which we require similar properties to those of distances, albeit more relaxed.

Def. 5 (Preference separation). *Given a set of pairs of users with common elements* $U_{com} = \{(u, u') \in U \times U | C(u, u') \neq \emptyset\}$, *a user separation measure is a function* $sep : U_{com} \to \mathbb{R}$ *that measures the separation of two users* $(u_1, u_2) \in U_{com}$, *based on their known preferences by the agent (i.e.* $p_{ag_{u_1}}$ *and* $p_{ag_{u_2}}$*). This function must satisfy the following properties:*

- ***Dependence of commonly known preferences:*** $sep(u_1, u_2)$ *only depends of* $p_{ag_{u_1}}(x)$ *and* $p_{ag_{u_2}}(x)$, $\forall x \in C(ag_{u_1}, ag_{u_2})$
- ***No-negativity:*** $sep(u_1, u_2) \geq 0$.
- ***Symmetry:*** $sep(u_1, u_2) = sep(u_2, u_1)$.
- ***Zero separation for equal known common preferences:*** $sep(u_1, u_2) = 0 \Leftrightarrow p_{ag_{u_1}}(x) = p_{ag_{u_2}}(x) \forall x \in C(ag_{u_1}, ag_{u_2})$.
- ***Triangle inequality for known common preferences:*** *Given a third user* u_3, *if we note* $C = C(ag_{u_1}, ag_{u_2})$, *then* $sep(u_1, u_2) \leq sep|_C(u_1, u_3) + sep|_C(u_3, u_2)$. *where* $sep|_C$ *is the separation function restricted to only the common elements of* ag_{u_1} *and* ag_{u_2} *(i.e. applying* sep *as if* $X = C$*).*

As argued earlier, the first property ensures that user separation only depends on the commonly known preferences of the users. The next four properties correspond to the properties of distances adapted to our case, as follows. First, we require that the separation measure is positive, as 0 is the closest possible separation, disallowing negative separations. Second, this function must be symmetric as the separation between two preferences should be the same no matter the order. Third, two users have separation of 0 if and only if their commonly known preferences are the same. This property is more general than $sep(u_1, u_2) = 0 \Leftrightarrow u_1 = u_2$ as we do not want to take into account what happens with not commonly known preferences. The fourth property is a general version of the triangle inequality where we only consider the commonly known preferences. Again, we want to be more general because we want to disregard preferences for which we do not have complete knowledge of both users. To illustrate, we provide an example of such a function, which we call cumulative user separation.

Def. 6 (Cumulative user separation). *The cumulative user separation function is a function* $sep_+ : U_{com} \to \mathbb{R}$ *that, for users*

$(u_1, u_2) \in U_{com}$ and respective agents ag_{u_1}, ag_{u_2}, assesses their separation as follows:

$$sep_+(u_1, u_2) = \sum_{x_i \in C(ag_{u_1}, ag_{u_2})} |p_{ag_{u_1}}(x_i) - p_{ag_{u_2}}(x_i)|$$

Proving that sep_+ satisfies the properties in Definition 5 is straightforward, but omitted due to space constraints.

The following example illustrates the concepts introduced so far, in context of the scenario introduced in Example 1.

Example 4. *We assess the separation between u_1 and the other users. For both u_2 and u_3, we have $C(ag_{u_1}, ag_{u_2}) = C(ag_{u_1}, ag_{u_3}) = \{x_1\}$, as for u_1 we know preferences over x_1 and x_2, but for the other users we know preferences over x_1 and x_3, but not x_2. Thus, in this case the cumulative separation is $sep_+(u_1, u_2) = 0$ and $sep_+(u_1, u_3) = 2$.*

Given a user for which we want to predict a preference over x, we can gather a set of similar users for which we know their preferences over x considering their separation with regard to the original user. With this set of similar users, we can then predict the targeted preference by aggregating the preferences of similar users towards that element. With this aim, we define the set of similar users. Since we build this set to make predictions, we must require that we know the preference of the users in the set with regard to the targeted element. Ideally, similar users should be those with separations less or equal to a maximum ϵ. However, since our aim is to use similar users to build predictions, we require a minimum number of similar users ν (those with the least separation), so that predictions are founded on a reasonable number of users.

Def. 7 ($\epsilon\nu$-similar users). *Given a user u, an element x, and parameters $\nu \in \mathbb{N}$ and $\epsilon \in \mathbb{R}$, we call $\epsilon\nu$-similar users the set $Sim^{\epsilon}_{\nu}(u, x)$ of similar users to u, such that they have preferences over x, and which contains at least the ν most similar users and all users that are closer than ϵ (in terms of separation). Hence, if $K(x) = \{u | p_{ag_u}(x) \neq \bigcirc\}$ is the set of users for whom we know their preference over x, we formalise $Sim^{\epsilon}_{\nu}(u, x) = Sim^{\epsilon}(u, x) \cup Sim_{\nu}(u, x)$, where:*

- *$Sim^{\epsilon}(u, x) = \{u' \in K(x) | sep(u, u') \leq \epsilon\}$ is the set of users with known preference over x who have a separation with u less or equal to ϵ.*
- *$Sim_{\nu}(u, x) = \{u_1, \ldots, u_{\nu} \in K(x) | sep(u, u_1) \leq \cdots \leq sep(u, u_{\nu})$ and $\nexists u' \in K(x) \setminus \{u_1, \ldots, u_{\nu}\} s.t.\ sep(u, u') < sep(u, u_{\nu})\}$ is the set of the ν closer users to u with known preference over x.*

Using the set of similar users, we can predict the targeted preference, for which we use a prediction function, defined as follows.

Def. 8 (Preference prediction function). *A preference prediction function is a function $pre : U \times X \to [-1, 1]$ which, given a pair of a user $u \in U$ and elements $x \in X$, predicts the preferences of u towards x. Given $\nu \in \mathbb{N}$ and $\epsilon \in \mathbb{R}$, this function must depend only on the preferences towards x of similar users to u, $\{p_{ag_{u'}}(x) | u' \in Sim^{\epsilon}_{\nu}(u, x)\}$*

We provide an example of a preference prediction function called average preference prediction function. This function builds a prediction of the preference of u towards x as the average of the preferences over x of similar users to u. Formally:

Def. 9 (Average preference prediction function). *Given a separation measure sep, the average preference prediction function $pre_{avg} : U \times X \to [-1, 1]$, takes a user $u \in U$ (with $p_{ag_u}(x) = \bigcirc$) and an element $x_i \in X$, and predicts the preference of u towards x in $[-1, 1]$, as follows:*

$$pre_{avg}(u, x) = \frac{\sum_{u' \in Sim^{\epsilon}_{\nu}(u,x)} p_{ag_{u'}}(x)}{|Sim^{\epsilon}_{\nu}(u, x)|}$$

We continue with the following illustration of a prediction.

Example 5. *We want to predict the preference of u_1 with regard to x_3. Considering the separations found in Example 4, we aim at selecting the similar users. To do so, we require at least one user (i.e. $\nu = 1$) and we consider them similar if the separation is less than 0.5 (i.e. $\epsilon = 0.5$), then $Sim^{\epsilon=0.5}_{\nu=1}(u_1, x_3) = \{u_2\}$. Then, to predict the preference of u_1 towards x_3, we average the preferences of the similar users towards x_3. In the case of Example 4, the result would be $pre_{avg}(u_1, x_3) = -1$*

Given a prediction function, we can complete the unknown preferences of the user as follows.

Def. 10 (Complete predicted preferences). *Given a preference prediction function pre and user u with known partial preferences p_{ag_u}, the tuple of complete predicted preferences for the user is $p*_{ag_u} = (p*^1_{ag_u}, \ldots, p*^{|X|}_{ag_u})$, composed of the known preferences and predictions of the unknown ones. Formally:*

$$p*^i_{ag_u} = \begin{cases} p^i_{ag_u} & \text{if } p^i_{ag_u} \neq \bigcirc \\ pre(u, x_i) & \text{if } p^i_{ag_u} = \bigcirc \end{cases}$$

Note that in Example 5, the preference predicted along with the known preferences form the complete predicted preferences for u_1.

Using Definition 10, we obtain the complete preferences of the user from the already known preferences and the newly predicted ones. This offers a solution to the preference approximation problem, and we show the validity of our approach in Section 5. First, however, we tackle the norm inference problem in the next section.

4 NORM INFERENCE FROM PREDICTIONS

At this point, we can predict user preferences from similar users. However, our broader aim is to build norms from these preferences so that agents can follow them. This means transforming numerical preferences in [-1, 1] into norms. In this section, we propose several methods to perform this transformation and discuss under which circumstances these methods would be appropriate to be used.

4.1 Hard thresholds

The simplest method we can use to transform numbers in [-1, 1] into norms is through hard thresholds. Thus, we would consider two thresholds ϵ_{prh}, and ϵ_{per} that divide [-1, 1] into three blocks, referring to (in the following order): prohibition, no norm, and permission. Hence, for consistency, we require that the threshold of prohibition must be on the negative side of the preferences interval, and the permission threshold on the positive side, $\epsilon_{prh} \in [-1, 0]$, and $\epsilon_{per} \in [0, 1]$. Then, considering the completed preferences $p*_{ag_u}$, we would build norm $Prh(x)$ if $p*_{ag_u}(x) \leq \epsilon_{prh}$, no norm if $\epsilon_{prh} < p*_{ag_u}(x) < \epsilon_{per}$, or $Per(x)$ if $\epsilon_{per} \leq p*_{ag_u}(x)$.

Example 6. *If we have thresholds $\epsilon_{prh} = -0.25$ and $\epsilon_{per} = 0.25$, then elements x with preferences in $[-1, -0.25]$ would be prohibited*

(Prh(x)), those in $[-0.25, 0.25]$ *would not be regulated, and those in* $[0.25, 1]$ *would be permitted (Per(x)).*

4.2 Thresholds based on prediction confidence

Note that hard thresholds can be problematic when predictions are not particularly accurate (for example, due to p_{ag_u} having many unknown preferences). In this case, for unknown preferences that are close to the threshold our prediction can easily fall on either side. Thus, in these cases we can consider variable thresholds depending on the confidence of our predictions.

Here, we consider thresholds to be a function of prediction confidence. If we consider confidence to be a number in $[0, 1]$, then we formalise thresholds as functions: $\epsilon_{prh} : [0, 1] \rightarrow [-1, 0]$ and $\epsilon_{per} : [0, 1] \rightarrow [0, 1]$.

The remaining task now is to define prediction confidence. Note that we would consider a prediction based on other very similar agents, with very similar preferences, as being accurate, whereas a prediction obtained from agents close in opinion but not entirely similar, and whose preferences span over an array of options, likely not very accurate. If we do not know the real preference we cannot be entirely sure of the quality of predictions but confidence gives us an intuition on the quality of the data they are drawn from. Formally, we define a prediction confidence function as follows.

Def. 11 (Confidence function). *A prediction confidence function* $conf : U \times X \rightarrow [0, 1]$ *is a function that takes a pair of user and element and gives the confidence of prediction* $pre(u, x)$ *in* $[0, 1]$*, where 0 means no confidence and 1 is absolute confidence. Note that in general* $conf(u, x) > conf(u', x')$ *should imply* $|pre(u, x) - p_u(x)| < |pre(u', x') - p_{u'}(x')|$*. In other words, a higher confidence should correlate with a better prediction (one closer to the real preference).*

We provide an example prediction confidence function called $\rho\mu$-Confidence based on the following two measures:

- The separation between u and users in $Sim^{\epsilon}_{\nu}(u, x)$ (for some separation measure sep)
- The distribution of preferences of users in $Sim^{\epsilon}_{\nu}(u, x)$ towards x (i.e. their standard deviation).

We define $\rho\mu$-Confidence as the weighted average of these two measures where ρ and μ are the weights.

Def. 12 ($\rho\mu$-Confidence). *Let sep be a separation measure as in Def. 5 and* $Sim^{\epsilon}_{\nu}(u, x)$ *be the set of similar users to user u (using sep). We can then define the confidence of prediction* $p(u, x)$ *as:*

$$conf_{\rho,\mu}(u, x) = 1 - \rho \cdot \min\left(\frac{\sum_{u' \in Sim^{\epsilon}_{\nu}(u,x)} sep(u, u')}{|Sim^{\epsilon}_{\nu}(u, x)|}, 1\right) - \mu \cdot \min(sd(SP), 1)$$

Where $\rho, \mu \in [0, 1]$*,* $\rho + \mu = 1$*, sd refers to the standard deviation of a set, and* $SP = \{p_{ag_{u'}}(x) | u' \in Sim^{\epsilon}_{\nu}(u, x)\}$.

Note that, in order to have confidence between 0 and 1, we set an upper bound of 1 to each of the two parts. The first part of the $\rho\mu$-Confidence refers to the separation between the users for the prediction, and the higher this separation, the less confidence in the prediction. In this case, we measure the average separation between u and the users in $Sim^{\epsilon}_{\nu}(u, x)$. The second part refers to the distribution of the real preferences of the similar users, hence the more these preferences differ, the lower confidence in our prediction. Here, we use the standard deviation of the preferences.

Once we have a confidence function, we can use it to define variable thresholds to create norms from preferences. One possibility is to favour the creation of norms when we have confident predictions, while limiting their production when we have low confidence. In other words, we can consider variable thresholds that are closer to the middle point (0) when confidence is high, and closer to the extremes (-1 and 1) when confidence is low.

Def. 13 (Confident norm thresholds). *Given a confidence function* $conf(u, x)$*, we define confident norm thresholds as* $\theta_{prh}(u, x) = -1 + \frac{conf(u,x)}{3}$, $\theta_{per}(u, x) = 1 - 2\frac{conf(u,x)}{3}$.

We use confident norm thresholds for our running example.

Example 7. *We want to infer a norm for* u_1 *and element* x_3*, in Example 5 where we predicted a preference of* -1*. Note that, if we consider* $\rho = \frac{1}{2}, \mu = \frac{1}{2}$*, in this case we have* $conf_{\rho,\mu}(u_1, x_3) = 1$ *(as both parts of the function are 0). Hence, we would have* $\theta_{prh}(conf) = -\frac{2}{3}$*,* $\theta_{per}(conf) = \frac{1}{3}$*, and would infer* $Prh(x_3)$ *in this case, because* $-1 < \theta_{prh}$.

4.3 Thresholds based on other variables

Much like with prediction confidence, threshold functions can also depend on other relevant variables like the context of the elements (assuming they have contexts). For generality purposes, we have avoided defining formally any type of these variables, and have considered them implicitly in each $x \in X$. However, in some applications it might be important to consider them when setting norm thresholds. For example, if we want to avoid inappropriate actions in sensitive contexts, we can consider the sensitivity of the context as a variable to set the thresholds. Then, $\epsilon_{prh}(c)$ would be closer to 0 for contexts c that are considered sensitive than for non-sensitive ones. Formally, in this case, we would consider thresholds as functions depending on multiple variables $\epsilon_{prh} : V_1 \times V_n \rightarrow [-1, 0]$ and $\epsilon_{per} : V'_1 \times V'_m \rightarrow [0, 1]$, where we consider n and m variables respectively and $V_1 \times V_n$ and $V'_1 \times V'_m$ are the possible values of these n and m variables. As for hard thresholds, we require that the ϵ_{prh} and ϵ_{per} functions have ranges in $[-1, 0]$ and $[0, 1]$ respectively.

4.4 The suitability of the different approaches

The suitability of each of the previous approaches depends largely on the domain of application. Apart from particular application requirements, when deciding which method to apply we should also consider the accuracy and distribution of predictions. To be concise, we discuss this in relation to two general measures: the average prediction distance from the real preference (denoted as APD), as well as the standard deviation of these predictions (denoted PSD). The average prediction distance tells us the accuracy of our predictions, while the standard deviation gives us an indication of the polarisation of predictions with regard to average distance. These two measures lead to the following four differentiated cases:

- Low APD and low PSD: This is the ideal scenario in which predictions work best, where any method is valid. Hard thresholds are useful for cases that demand an easily explainable method. Function thresholds can also be useful, especially if required by the application (for example, one that explicitly demands consideration of environmental variables like context sensitivity when determining norms).

- Low APD and high PSD: Here, the predictions seem accurate but not reliable enough, so hard thresholds are best avoided. Instead, the other approaches prevent norms being enacted in cases of vague predictions coming from bad quality data.
- High APD and low PSD: Here the predictions are consistently wrong, and consistently deviate from the truth. This case should not usually arise and tells us that there is something wrong with the prediction formula, so predictions should not be used to build norms.
- High APD and high PSD: Here, the predictions are seemingly random. This can be a consequence of insufficient information (e.g., when known preferences are far fewer than unknown ones). At this stage, norms could be built using function thresholds, but if more information is collected this scenario should then settle into one of the other three, and norms would be selected using the relevant advice. Predictions at this stage may not be reliable so the resulting norms should be rebuilt once more information is known.

5 PROOF OF CONCEPT: PRIVACY NORMS FOR SMART PERSONAL ASSISTANTS

In order to validate the preference prediction and norm inference models presented here, we return to the problem of Smart Personal Assistants. We consider this case by virtue of the privacy preferences dataset used by [2], which is available at [3].

5.1 Description of the dataset

The dataset contains the responses of 1737 participants in a survey concerning privacy preferences when using Smart Personal Assistants (SPAs). The questions in the survey[3] ask participants how acceptable it is to share data in a particular context. More specifically, the survey considers 15 data types (e.g. emails, banking data, healthcare data, voice recordings) and presents 8 scenarios for each. These scenarios or contexts consider different recipients of the data (e.g. parents, friends, visitors), the purpose of sharing the data, different conditions on data transmission, etc. Each scenario has a different number of associated questions, amounting to 55 for all 8 scenarios. Overall, the survey consists of 825 different preference questions, for each of which participants answer on a 1 to 5 Likert scale (1 meaning the sharing of that datatype is completely unacceptable in that context, and 5 meaning it is completely acceptable). Participants did not answer all questions, with each participant answering questions related to 4 scenarios for 6 datatypes (both selected randomly). Note that different scenarios have different amounts of associated questions, so participants responded to different questions and different numbers of questions, ranging from 144 to 199, with an average of 170 questions answered.

5.2 Prediction validation

In this section we assess the accuracy of our predictions using the previously described dataset[4]. We show that our predictions are more accurate than random guesses and also more accurate than the preferences found by Abdi et al. in [2]. Finally, we record the confidence measure for each prediction and show that there is a correlation between confidence and prediction quality.

First, we describe the experiments on accuracy. Out of the 1737 participants, we selected 20% of participants (347) randomly to test the accuracy of our predictions. The remaining 80% of participants (1390) represent our base of knowledge to build predictions. For each test participant, we randomly picked 20% of their answers as the test set we sought to predict (this was an average of 35). We did not consider all remaining answers to assess user similarity, but instead used only 40% of answers (an average of 71), and applied this reduction to all participants. To proceed, for each test participant and answer to predict, we filtered the pool of 1390 participants to keep only those relevant; we needed to filter out participants who did not answer the question we aim to predict. In addition, we discarded participants with less than 5 questions in common with the test participant (as we wanted to finding similar users with a certain degree of reliability). Then, we assessed the separation between users using the cumulative user separation measure of Def. 6. Using this separation measure, we selected participants similar to our test participant as in Def. 7, with $\epsilon = 0$, and $\nu = 5$. In other words, we selected all users at distance 0, and then selected those with the least distance until we had 5 participants. Then, we predicted the test participant's answers to the test questions using the average preference prediction function (see Def. 9).

To test our prediction, we calculated the distance between our predicted answer and the real answer, and collected all these distances for all test participants and test questions. Below, we report the mean distance and the standard deviation. Apart from the experiment considering all participants (which we now refer to as the *regular* experiment) we wanted to test accuracy for two further levels of difficulty. First, we hypothesised that participants that responded similarly to all questions would be easiest to predict. Thus, when selecting test participants we avoided those with small standard deviations on their given answers (but retained them in the pool of possible similar participants). Here, we required the standard deviation of any selected test participant to be no less than 1. We repeated the experiments as explained above, and refer to this as the *medium hardness* experiment. Second, to test the extreme case, we selected the 100 participants with highest standard deviation in their answers and repeated the experiments as explained above. These are arguably the most difficult participants to predict. By selecting many of them, we limit the chances of finding very similar participants in the remaining pool (since, to be similar, they also need to have a large standard deviation and therefore might have been selected as test participants). We call this the *hard* experiment.

Table 1 provides the mean distance as well as the standard deviation for each run of tests. We provide a histogram for the regular experiments to better understand the distribution of distances between prediction and reality (the histograms for medium and hard experiments are almost identical, so we omit them to save space).

The regular experiment resulted in 12007 predicted answers with a mean distance from the real answer of 0.5954 and a standard deviation of 0.6757. Figure 1 shows the distribution of these 12007 distances between prediction and reality. We can see that most predictions fall within 0.25 of the real answer, while only a small number of predictions have distances larger than 1 with regard

[3]For our tests, we only use the main block of questions in the survey, so to not consider data from questions on demographics, IUIPC, and security attitudes

[4]The code necessary to run these experiments can be found at: https://github.com/secure-ai-assistants/norm-prediction

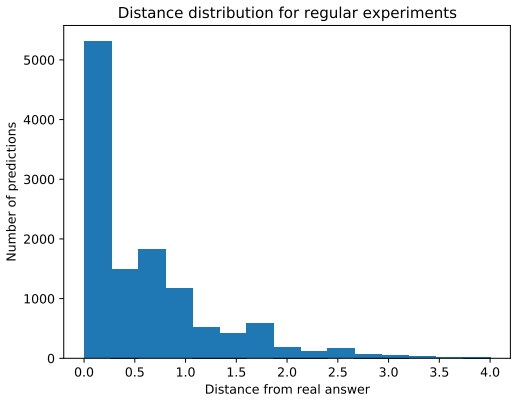

Figure 1: Histogram of distances between our predictions and the real preferences

to the real answer. The medium hardness experiment resulted in 12015 predicted answers[5]; as expected, the mean distance grows a little to 0.6538 and the standard deviation to 0.7207. Overall, while distances grow slightly, we can see that predictions are still of good quality. Finally, the hard experiment resulted in 3461 predicted answers[6] with a mean distance of 0.7480 and a standard deviation of 0.8707. While the mean distance has increased from the distance of the regular experiment it has increased by less than 0.2. The standard deviation has increased by almost 0.2, confirming that the predictions of these participants may lead to more outliers.

Thus, while increasing the difficulty of predictions slightly increases the distance between the prediction and the real answer, we have seen that our predictions are still reliable with the hardest participants to predict. Furthermore, our predictions improve the preferences presented by Abdi et al. in [2], which are the average preferences for all participants in their survey. Most notably, our model can capture preferences outside the majority view; for example, Abdi et al. point out that people are less inclined to share video call data with assistant providers (the average preference for sharing this while being able to delete it is 2.44), yet in many instances we correctly predicted favourable preferences towards sharing. Table 2 shows the results of our experiments considering their preferences, and as an additional point of reference, Table 3 provides results for the experiments with random predictions.

When it comes to confidence, we tested how $\rho\mu$-Confidence correlates with prediction quality (its closeness to the real preference)

[5]Different experiments have different numbers of predictions because we selected as test answers 20% of a participants' answers and different participants answered different numbers of questions

[6]Note that the hard experiment has fewer predicted answers because it consists of 100 test participants instead of 347

	Regular	Medium	Hard
Mean distance from real answer	0.5954	0.6538	0.7480
Standard deviation	0.6757	0.7207	0.8707

Table 1: Results for regular, medium and hard experiments

	Regular	Medium	Hard
Mean distance from real answer	1.0437	1.0611	1.2895
Standard deviation	0.7069	0.7093	0.7765

Table 2: Results with preferences found by Abdi et al. [2]

using Spearman's correlation coefficient[7]. Note that the coefficient must be a negative number as we should have an inverse correlation (the higher the confidence, the lower the distance between the real and predicted preferences). While our confidence formula considers two weights ρ and μ, these depend on each application and could be adjusted at runtime to maximise correlation (i.e. to minimise the correlation coefficient). For regular experiments, we have a correlation coefficient of -0.67 with $\rho = 0$ and $\mu = 1$. With medium hardness experiments, the minimum was -0.63 with $\rho = 0.01$ and $\mu = 0.99$. Finally, for hard experiments, the minimum was -0.74 when $\rho = 0.15$ and $\mu = 0.85$. We can therefore detect an inverse correlation between confidence and prediction quality. We also see that in this case, confidence largely depends on the distribution of preferences from which the prediction is made, whereas the separation between the user and the similar users is not pertinent to assess confidence. Importantly, we see that our confidence function is more reliable for hard predictions. We believe this is because since confidence is always in [0,1] it is easier for it to correlate with prediction quality in cases where the quality has more variability (i.e. in the case of hard experiments).

5.3 Evaluating inferred norms with real users

To test the norm inference process with real users we performed a user study with the scenarios from [2] and the preference data collected in [3], with the aim of validating user perceptions of our predicted norms. Through Prolific,[8] we recruited 50 participants matching the demographics of the original data set who answered 32 preference questions over 5 randomly selected scenarios from [2, 3]. We then selected three unknown preferences at random and made predictions for them, inferring the norms using the hard thresholds function (see Section 4.1); if no norm was generated for a preference we randomly selected another unknown preference[9]. We also interleaved three control norms with the same structure but randomly generated outcomes. Participants rated these norms using 5 point Likert items from completely inappropriate (1) to completely appropriate (5) and could leave a text comment explaining their reasoning. After discarding 3 incorrect responses to the included attention check, the remaining 47 participants had an average age

[7]We cannot ensure that our data follows a normal distribution, hence Spearman's is the appropriate correlation test to use

[8]prolific.co

[9]Note that this does not compromise our results as we only aim to validate the generated norms. If we do not predict a clear preference, our approach does not produce any norm and instead we resort to other approaches (like asking for consent).

	Regular	Medium	Hard
Mean distance from real answer	1.6083	1.578	1.8768
Standard deviation	1.0864	1.080	1.1286

Table 3: Results with random predictions

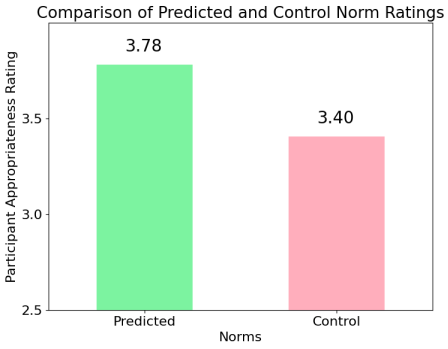

Figure 2: Comparison between mean participant ratings of predicted and control norms

of 37.5 (σ = 13.8), and 49% identified as women. The study was approved by our Institutional Review Board (IRB).

Figure 2 shows the mean ratings of the 141 predicted and 141 control norms, showing a substantial improvement of predicted norms over the control set. Overall we predicted 126 unique preferences covering 15% of the original set. A follow up t-test (t=2.88, p=0.003) confirms that participants' higher agreement with our predicted norms compared to the control norms is statistically significant, with the prediction process eliminating almost a quarter (24%) of the difference between the control ratings and the perfect acceptance score of 5. As in similar user studies (e.g., [25]), we emphasise that it is very unlikely that even ground truth preferences would receive a perfect score of 5 given the variability of self-ratings and the tendency for participants to bring in outside context when evaluating norms (as seen in accompanying text comments).

6 RELATED WORK

Some previous approaches have addressed the problem of privacy norms in AI assistants. For example, Abdi et al. [2] surveyed users on how acceptable some information exchanges are under some context (this survey produced the data we used for our experiments [3]) and then crowd-sourced norms that aligned with their answers. While this work is useful to understand the general preferences of users towards privacy, it is restricted in terms of the scenarios covered. Nonetheless, it could be useful as a default set of preferences when information about the user is sparse (e.g. when the user first uses the assistant). The work of Zhan et al. [30] instead proposes to construct AI assistant privacy norms using an approach based on rule mining and machine learning (exploiting the idea of contextual integrity [15]). Unfortunately, we cannot compare our norm prediction approach with their approach as [30] reports test results as percentages of accuracy which cannot be compared with our acceptability rates. While this approach achieves an accuracy of 70-80% it requires domain knowledge. In contrast our work in this paper can customise norms to each user without the need for specifying or formalising contexts. Note that the elements in our set X can be actions, or tuples of actions and contexts, and in both cases our method is able to predict preferences and infer norms without the need for additional knowledge of the elements in X.

Closely related to this is the area of AI ethics and norms. We can assume that morality influences a user's preferences towards AI so that, for example, a user that highly values privacy will be less inclined to share data. Works like [19, 20] have investigated the selection of norms with regard to their promotion of moral values and preferences over these values. Similarly, [22] proposes to enact those norms that will benefit those state transitions leading to an increase in value alignment with the considered values and preferences. In this direction, Montes et al. [11] describe how these norms can be synthesised. While these approaches could produce privacy norms (provided privacy is a value considered) we argue that in practice this would not be possible since they require knowledge that is hardly attainable by smart devices, like states of the world, contexts, the user's value preferences, or a measurement of value alignment (with regard to privacy and other desirable values).

7 CONCLUSIONS

Collaborative filtering is a useful tool in recommender systems. For example, online stores use it to recommended products by considering purchases of similar users. This paper provides a novel application of collaborative filtering, with the aim of predicting user preferences towards AI. However, our approach offers far more than just recommending preferences to the user. Indeed, while users expect smart devices to act as they desire, constant interaction not only annoys the user but fails to capture their true preferences. Hence, our approach has two purposes: understanding user preferences while minimising interaction, and bringing more value to interactions regarding preferences by considering predictions. Thus, coupling collaborative filtering with norms allows us to both add a component of explainability to user preferences, and to propagate user preferences to other parts of the AI ecosystem. For example, in the case of privacy in smart assistants, norms could govern the management of data not only by the device itself but also for other components of the ecosystem, like skills.

Admittedly, our approach requires large quantities of users and partial preferences for each of these users to function properly, and the more information the more accurate the predictions. Thus, our approach might be better suited to smart devices with a reasonable number of users. Even if the number of users is sufficient, it is also possible that predictions could be unreliable. However, we can detect this using a confidence measure, such as that of Definition 12. Crucially, however, as the number of users grows, and as knowledge of their preferences increases, low confidence preferences can be recalculated, which should increase the confidence in the prediction.

In addition, we have assumed a single user for each device, but it is unclear how this method would apply when multiple users share the same device (for example, a family sharing a smart speaker). This will be the subject of future work. Other interesting aspects we plan to investigate include the addition of rewards or punishments associated with norms (which could be derived from context sensitivity), and how to produce explanations from norms.

ACKNOWLEDGMENTS

Research funded by project SAIS Secure AI AssistantS via Grant EP/T026723/1, funded by the UK Engineering and Physical Sciences Research Council; and by project TED2021-131295B-C32, funded

by MCIN/AEI/ 10.13039/501100011033 and the European Union NextGenerationEU/PRTR.

REFERENCES

[1] Noura Abdi, Kopo M. Ramokapane, and Jose M. Such. 2019. More than Smart Speakers: Security and Privacy Perceptions of Smart Home Personal Assistants. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. USENIX Association, Santa Clara, CA, 451–466. https://www.usenix.org/conference/soups2019/presentation/abdi

[2] Noura Abdi, Xiao Zhan, Kopo M. Ramokapane, and Jose Such. 2021. Privacy Norms for Smart Home Personal Assistants. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 558, 14 pages. https://doi.org/10.1145/3411764.3445122

[3] Noura Abdi, Xiao Zhan, Kopo M. Ramokapane, and Jose Such. 2021. Privacy Norms for Smart Home Personal Assistants Survey Dataset. https://osf.io/63wsm/?view_only=571e8ed94d9c4dd19f09e8046a2d1abf Accessed April 2022.

[4] Hanbyul Choi, Jonghwa Park, and Yoonhyuk Jung. 2018. The role of privacy fatigue in online privacy behavior. *Computers in Human Behavior* 81 (2018), 42–51. https://doi.org/10.1016/j.chb.2017.12.001

[5] Jide S. Edu, Jose M. Such, and Guillermo Suarez-Tangil. 2020. Smart Home Personal Assistants: A Security and Privacy Review. *ACM Comput. Surv.* 53, 6, Article 116 (dec 2020), 36 pages. https://doi.org/10.1145/3412383

[6] Christian Hoffmann, Christoph Lutz, and Giulia Ranzini. 2016. Privacy cynicism: A new approach to the privacy paradox. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace* 10 (12 2016). https://doi.org/10.5817/CP2016-4-7

[7] Jun Hong, Xiaoyuan Su, and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence* 2009 (2009), 421425.

[8] Michael Kretschmer, Jan Pennekamp, and Klaus Wehrle. 2021. Cookie banners and privacy policies: Measuring the impact of the GDPR on the web. *ACM Transactions on the Web (TWEB)* 15, 4 (2021), 1–42.

[9] Balachander Krishnamurthy and Craig E. Wills. 2009. On the Leakage of Personally Identifiable Information via Online Social Networks. In *Proceedings of the 2nd WOSN* (Barcelona, Spain). ACM, NY, USA, 7–12.

[10] Mary Madden. 2014. Public perceptions of privacy and security in the post-Snowden era. https://www.pewresearch.org/internet/2014/11/12/public-privacy-perceptions/ Accessed on April 2022.

[11] Nieves Montes and Carles Sierra. 2022. Synthesis and Properties of Optimally Value-Aligned Normative Systems. *J. Artif. Int. Res.* 74 (sep 2022), 36. https://doi.org/10.1613/jair.1.13487

[12] Javier Morales, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Wamberto Vasconcelos, and Michael Wooldridge. 2015. On-line Automated Synthesis of Compact Normative Systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 10, 1 (March 2015), 2:1–2:33.

[13] Javier Morales, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. 2013. Automated Synthesis of Normative Systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* (St. Paul, MN, USA) *(AAMAS '13)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 483–490.

[14] Javier Morales, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. 2014. Minimality and Simplicity in the On-line Automated Synthesis of Normative Systems. In *AAMAS 2014* (Paris, France). IFAAMAS, Richland, SC, 109–116.

[15] Helen Nissenbaum. 2004. Privacy as contextual integrity. *Washington Law Review* 79 (2004), 119.

[16] Jonathan A. Obar and Anne Oeldorf-Hirsch. 2018. The Biggest Lie on the Internet: Ignoring the Privacy Policies and Terms of Service Policies of Social Networking Services. In *Proceedings of the 44th Research Conference on Communication, Information and Internet Policy*. Information, Communication & Society, Arlington, VA, USA, 1–20. https://doi.org/10.2139/ssrn.2757465

[17] Bastin Tony Roy Savarimuthu, Maryam Purvis, Stephen Cranefield, and Martin Purvis. 2007. Mechanisms for norm emergence in multiagent societies. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems* (Honolulu, Hawaii) *(AAMAS '07)*. ACM, New York, NY, USA, Article 173, 3 pages. http://doi.acm.org/10.1145/1329125.1329335

[18] Marc Serramia, Natalia Criado, and Michael Luck. 2023. Collaborative Filtering to Capture AI User's Preferences as Norms. In *PRIMA 2022: Principles and Practice of Multi-Agent Systems*, Reyhan Aydoğan, Natalia Criado, Jérôme Lang, Victor Sanchez-Anguix, and Marc Serramia (Eds.). Springer International Publishing, Cham, 669–678.

[19] Marc Serramia, Maite Lopez-Sanchez, Stefano Moretti, and Juan A Rodriguez-Aguilar. 2021. On the dominant set selection problem and its application to value alignment. *Autonomous Agents and Multi-Agent Systems (JAAMAS)* 35, 2 (2021), 42. https://doi.org/10.1007/s10458-021-09519-5

[20] Marc Serramia, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Manel Rodriguez, Michael Wooldridge, Javier Morales, and Carlos Ansotegui. 2018. Moral Values in Norm Decision Making. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1294–1302.

[21] Yoav Shoham and Moshe Tennenholtz. 1997. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence* 94, 1-2 (1997), 139–166.

[22] Carles Sierra, Nardine Osman, Pablo Noriega, Jordi Sabater-Mir, and Antoni Perello-Moragues. 2019. Value alignment: a formal approach. In *Responsible Artificial Intelligence Agents Workshop (RAIA) in AAMAS*. IFAAMAS, Montreal, Canada, 15.

[23] Strategy analytics. 2021. Global Smart Speaker and Screen Vendor & OS Shipment and Installed Base Market Share by Region: Q4 2021.

[24] Toshiharu Sugawara. 2011. Emergence and Stability of Social Conventions in Conflict Situations. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One* (Barcelona, Catalonia, Spain) *(IJCAI'11)*. AAAI Press, US, 371–378.

[25] Sharadhi Alape Suryanarayana, David Sarne, and Sarit Kraus. 2022. Justifying Social-Choice Mechanism Outcome for Improving Participant Satisfaction. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems* (Virtual Event, New Zealand) *(AAMAS '22)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1246–1255.

[26] Wamberto Weber Vasconcelos, Martin J. Kollingbaum, and Timothy J. Norman. 2009. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 19, 2 (2009), 124–152.

[27] Sam Wolfson. 2018. Amazon's Alexa recorded private conversation and sent it to random contact. https://www.theguardian.com/technology/2018/may/24/amazon-alexa-recorded-conversation Accessed Feb. 2022.

[28] Michael Wooldridge. 2002. *An Introduction to MultiAgent Systems* (1st ed.). John Wiley & Sons, New Jersey, US.

[29] Fabiola López y López, Michael Luck, and Mark d'Inverno. 2002. Constraining Autonomy through Norms. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2* (Bologna, Italy) *(AAMAS '02)*. Association for Computing Machinery, New York, NY, USA, 674–681. https://doi.org/10.1145/544862.544905

[30] Xiao Zhan, Stefan Sarkadi, Natalia Criado, and Jose Such. 2022. A Model for Governing Information Sharing in Smart Assistants. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society* (Oxford, United Kingdom) *(AIES '22)*. Association for Computing Machinery, New York, NY, USA, 845–855. https://doi.org/10.1145/3514094.3534129