Warming Up to Cold Start Personalization

NIKOLA BANOVIC, Carnegie Mellon University
JOHN KRUMM, Microsoft

Smart agents face abandonment if they are unable to provide value to the users from the very first interaction. Existing smart agents take time to learn about new users before they can offer them personalized services. We present a method for learning personalization information about users quickly and without placing unnecessary hardship on them. Our method enables smart agents to pick which questions to ask the user when they first interact to maximize the agent's overall knowledge about the user. We demonstrate our method on two publically available US census datasets containing 172 user variables from 1,799,394 training and 1,618,489 testing users. The questions selected using our method improve the agent's accuracy when inferring information about future users, including information that they did not ask about. Our work enables smart agents that assist the user with personalized services soon after they start interacting.

CCS Concepts: • **Information Systems → Personalization**

KEYWORDS

Cold start; Personalization; Submodularity.

ACM Reference format:

Nikola Banovic and John Krumm. 2017. Warming Up to Cold Start Personalization. *PACM Interact. Mob. Wearable Ubiquitous Technol.*, 1, 4, Article 124 (December 2017), 13 pages.
DOI: 10.1145/3161175

1 INTRODUCTION

Smart agents that run on personal and mobile devices (e.g., Cortana, Google Now, Siri) provide people with relevant information and assist them in everyday tasks. We focus on agents that provide personalized, content-based services that are relevant to the users' preferences and needs. They are unlike purely context-based agents that provide services using current context, or information from the environment relevant to the current interaction [3]. For example, an agent that recommends the best route for the user to drive home uses context to determine the user's current location, but also requires personal information, such as where the user lives and if she drives, to offer a truly personal experience.

Personalized services for specific users make interactions with mobile devices more efficient [23] and enjoyable [2]. Different online learning methods for cold start personalization (e.g., those using Active Learning [11]) enable the agent to dynamically learn about the users over time. However, sensing and learning user information takes time even with those methods. This leads to significant downtime between when the user starts using the agent and when the agent has learned enough about the user to offer personalized services. For example, the smart agent that erroneously notifies a user who does not own a car about driving conditions could diminish the user's trust in its abilities [4]. Thus, the existing agents that cannot provide relevant quality services early on could lead to user frustration or even abandonment of the technology [27].

Existing smart agents face the cold start personalization problem: how to provide relevant personalized services to the new user about whom the agent knows little. The designers of existing agents often make them default to assumptions

Author's addresses: N. Banovic, Human-Computer Interaction Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA; J. Krumm, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA.

about preferences of the majority of users (as in the vehicle example above). Alternatively, agents could gather knowledge about the users by asking them questions about their preferences. For example, the agent could ask the user whether they own a vehicle and drive to work right after the user unboxes the device. However, this may encumber the users by having to answer questions about each service before their general-purpose agent can provide those specific services to them.

The existing cold start personalization methods for specialized agents that focus on predicting a particular variable in a particular domain do not apply to general-purpose smart agents that assist the users in variety of domains. We would have to run the existing methods impractically many times for each domain and each variable covered by the general-purpose agent. Thus, we require novel approaches that consider all the different variables across different domains that are relevant to the users in a single run.

In this work, we focus on a specific aspect of the cold start personalization problem for general-purpose smart agents. We explore what small number of questions the agents should ask the users the first time they interact to learn as much about them as possible and infer the rest of their properties [22]. We learn the best questions to ask offline on a population of exiting mobile users, which makes the questions readily available before the agent encounters a new user. For example, the agent could learn about the user's home location and try to infer the user's age and income from this information (Fig. 1). This is similar to marketing strategies that target specific consumers based on their demographics or stated preferences [21]. Unlike the traditional marketing strategies, which require expert knowledge about the users that marketers develop over time, we use an automated approach to select variables that the agent should learn about from existing user data.

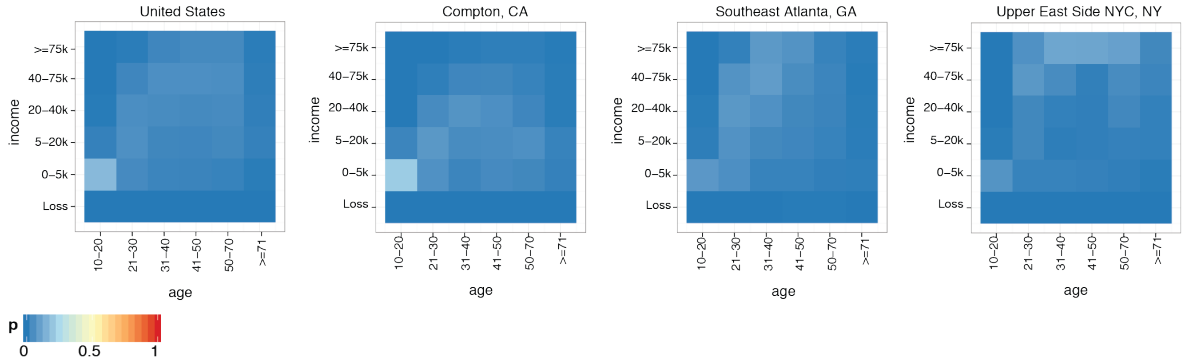

Fig. 1. Learning users' home location improves the agent's ability to infer users' age and income. Figure shows estimated joint probability distribution of age and income of smartphone users (from left to right): in the United States, Compton, CA, Southeast Atlanta, GA, and Upper East Side New York City, NY.

We build our variable selection approach on the insight that the best subset of variables to ask the user about is the one that gives the most information about the rest of the variables that were not selected; i.e., the subset of variables that maximizes the mutual information between the two subsets. We propose an optimization algorithm that avoids selecting variables that the agent can determine from device sensors and settings (e.g., preferred language, home location), but considers the information that those variables provide about other variables. The algorithm also avoids selecting variables that may be impolite or inappropriate to ask about when the user first interacts with the agent (e.g., age, income, education) [24], but that the agent needs to infer to provide the user with services.

We solve this **NP**-hard optimization problem by leveraging the submodularity of mutual information between subsets of user variables. We extend the work by Krause and Guestrin [12], who proved that mutual information between subsets of random variables is submodular. We prove that submodularity of mutual information holds when the algorithm is prevented from selecting certain variables, but needs to include those variables in calculations of mutual information. We illustrate how our approach can be used in practice on the publically available 2013 American Community Survey (ACS) dataset from United States Census Bureau [27] containing 172 user properties (e.g., age, gender, home location, income) from 1,799,394 smartphone users. We evaluate the accuracy of our method to infer user variables on another ACS from 2015 [27], which contained the same 172 properties from another 1,618,489 smartphone users. We show that answers about a selected subset of variables improve the agent's accuracy to infer the rest of the user properties by up to 8.62% compared to a non-informative baseline.

Our main contribution is the insight that submodularity-based optimization is a principled, computationally efficient way to find a subset of variables that best describe any user. We contribute a set of considerations that will guide smart agents when learning about users using our approach. Variables selected using our method represent fundamental properties of users that inform future user modeling efforts and can bootstrap the existing cold start personalization methods that infer user properties from small subsets of data [22] or that match users based on their demographics [16].

2 CHALLENGES IN COLD START PERSONALIZATION

Existing agents face challenges when trying to provide personalized services to users about whom they have limited or no information. Such agents often need to learn explicit information about the user based on a history of interactions with the user [20] (e.g., an agent that personalizes music playlists will play new songs based on previous songs that the user has listened to). Such systems take time to collect user information and cannot provide personalized services to new users.

The existing cold start personalization approaches focus on providing the best personalized experience for specific service they provide (e.g., recommending a movie or a book to a new user). They use the smallest subset of user properties that map to variables that are relevant to the service they provide (e.g., [6]). For example, agents may personalize services by grouping or clustering users based on limited knowledge about their demographics (e.g., [15,16]). The agents then find which group a new user belongs to and provide the new user with services that are relevant to that particular group. They may also use implicit data about some user properties to infer the other properties required to personalize their services (e.g., [18,19,20]). Those existing approaches assume that some knowledge about the user is available beforehand. To learn this information without placing a burden on the user, the agents may elicit only the most relevant preferences (e.g., [7,26]). However, it is not clear what that relevant prior knowledge is for general-purpose agents that provide multiple personalized services (e.g., Cortana, Google Now, Siri) and how to acquire this knowledge quickly for each of the services they offer.

Finding a small subset of questions that contributes to all possible services is challenging. Traditional variable and feature selection methods [9] can be used to rank user properties based on how predictive they are about a specific target variable. Existing cold start personalization methods (e.g., [11]) use properties of information (e.g., maximum entropy) to express uncertainty about new features that the system has not seen and the impact of those features on a specific prediction the algorithm is trying to make. As such, they may be useful to select variables for specific services, but not across services. To select subsets of variables using such approaches would require algorithms that are NP-hard [9].

Recent approaches based on greedy maximization of submodular functions [17] allow approximate solutions to this NP-hard problem in polynomial time [10]. However, existing work explores such optimization in the domain of sensor selection, and there is no evidence that it applies to the domain of selecting variables that best describe users. Krause et al. [14] have developed a specific variable set selection algorithm that penalizes certain variables and thus limits the chances that such variables will be selected. Their approach allows penalized variables to be selected given that the value of the variable information is high enough. That may be acceptable in the domain of privacy where users tradeoff their privacy for more accurate search results. However, that may not be applicable to the cold start personalization setting where the agents should never ask about certain user properties no matter how desirable that property is.

3 PICKING THE BEST QUESTIONS TO ASK

One of the design goals of smart agents is to solve the cold start personalization problem: to start providing relevant information to the user as fast and with as little effort as possible [16]. Users expect that the agent will be able to provide them with relevant services right out of the box. However, when the user interacts with the agent for the very first time, the agent is forced to make inferences about the user without any specific knowledge about that particular user. If the agent makes too many mistakes in these early interactions, the users may stop using it [4].

One way the agent can start learning about the user right away is to ask questions that will help the agent make more accurate inferences. However, there is a limit to the number of questions that the agent can ask before the user grows tired of answering too many questions. The agent should not waste time asking questions that it can compute from the sensors and settings on the users' devices (e.g., using GPS location traces to infer the user's home and work locations). Also, the users may not appreciate being asked about questions that are impolite or inappropriate [24] no matter how much information they contain about the user. Incidentally, some of those inappropriate questions (e.g., about age, income) are

most sought after by marketers because they believe they contain the most information about the user [25]. Therefore, the agent has to choose carefully which questions to ask.

When selecting a limited number of questions to ask, the agent has to consider three important factors: 1) how much information does the question's answer provide about the user? 2) is there a way to determine this information without asking the user? and 3) is this question appropriate when the user first starts interacting with the agent? Our goal is to select a subset of user variables that most effectively reduces the uncertainty about all other user variables, while taking advantage of known variables and avoiding selecting overly sensitive variables. A natural measure of the relationship between subsets of selected variables and the rest of the variables is mutual information.

3.1 Maximizing Mutual Information

Mutual information between subsets of random variables expresses the amount of information one subset gives about the other. If there is high mutual information between what the agent knows and it what it does not know, it can use the variables it knows to estimate values of the variables it does not yet know. It may be tempting for the agent to ask about variables with the highest entropy. However, the advantage of using mutual information to select variables instead of simply picking the variables with the highest entropy is that maximizing entropy aims to reduce the uncertainty of selected variables without considering the predictive power of the selected variables [12]. Krause and Guestrin [12] formalize this selection problem as nonmyopic selection of the most informative subset of variables for graphical models.

Formally, in Krause and Gustrin's conception [12], let V be a finite set of variables that describe the user, and A be a subset of these user variables. A is the subset of variables we want to select for directly questioning the user. Then the mutual information between subsets of selected variables and non-selected variables is given by $I(A; V\backslash A) = H(V\backslash A) - H(V\backslash A|A)$, where $H(V\backslash A)$ is the entropy of the set of unselected variables and $H(V\backslash A|A)$ is the conditional entropy of $V\backslash A$ given the variables in subset A. Intuitively, this means that the subset A that maximizes $I(A; V\backslash A)$ is the subset that best reduces the uncertainty of unselected variables (or the entropy of the set of unselected variables).

In our specific case we have two additional subsets of variables. For simplicity, and without loss of generality, we simply add the subset of variables that the agent can detect on its own to the subset A prior to our optimization. These are variables that could be sensed and inferred automatically by the user's own devices or by external sensors. For variables that we cannot select (e.g. those that are too sensitive to ask about), we define a subset $Z \subseteq V$, where for any subset A we pick, $Z \cap A = \emptyset$; i.e., A cannot contain any elements of Z.

Selecting the most informative set of variables is $\mathbf{NP}^{\mathbf{PP}}$-complete [10] (i.e., brute force approaches that check each combination of variables are infeasible). Instead, we use a method based on theory of submodular functions [17].

3.2 Submodularity of Mutual Information

Submodularity of functions allows for greedy algorithms that approximate optimal solutions in polynomial time with constant function approximation constraints [12]. Using the definition by Nemhauser et al. [17], a set function F is submodular if for all $A \subseteq A' \subseteq V$ and $y \in V\backslash A'$:

$$F(A \cup \{y\}) - F(A) \geq F(A' \cup \{y\}) - F(A') \tag{1}$$

Here we can think of y as a variable to ask about and add to set A. In our algorithm, we are trying to maximize the mutual information function $I(A; V\backslash A)$, using a special greedy optimization function for the subset of variables $Z \subseteq V$, where for any selected subset A, $Z \cap A = \phi$. Thus, we have:

$$F(A \cup \{y\}) = \begin{cases} I(A \cup \{y\}; V\backslash A \cup \{y\}), if\ y \notin Z \\ I(A; V\backslash A), if\ y \in Z \end{cases} \tag{2}$$

In other words, when the function comes across a variable $y \in Z$, it does not add it to the subset A, which in turn does not change mutual information $I(A; V\backslash A)$.

Note that Krause and Guestrin [12] proved that mutual information between sets of variables is submodular for the case when $y \notin Z$. Here we prove the other case ($y \in Z$):

$$F(A \cup \{y\}) - F(A) = F(A) - F(A) = 0 \tag{3}$$

$$0 = F(A') - F(A') = F(A' \cup \{y\}) - F(A') \tag{4}$$

Therefore, our special greedy optimization function, which does not select variables that we cannot ask questions about, is also submodular.

3.3 The Algorithm

We now describe our greedy algorithm that estimates the optimal subset of variables by picking a variable that maximizes the increase in entropy at each step. The number of greedy steps corresponds to the number of variables (L) that the algorithm is intended to select. To simplify our notation, let $MI(A) = I(X_A; X_{V\backslash A})$, and let $\tilde{A}$ mean $V\backslash A \cup \{y\}$. Then, at each of L step, our algorithm picks the single variable y that maximizes [13]:

$$MI(A \cup \{y\}) - MI(A) = H(y|A) - H(y|\tilde{A}) \tag{5}$$

Note that this update step works for the disallowed variables in Z as well because $\forall y \in Z, MI(A \cup \{y\}) - MI(A) = 0$, which follows from Equation 2. This allows us to use the greedy algorithm proposed by Krause and Guestrin [12]. Their algorithm offers a guarantee that the solution is an approximation that is within $(1 - 1/e)$ from the optimum, assuming the function is approximate monotone on the selection interval [13]. Although mutual information is not monotonic (e.g., mutual information will start to decrease after subset A becomes larger than $V\backslash A$), it is approximately monotone for the small number of variables we want to select in our case, where the size of A remains smaller than size of $V\backslash A$.

The pseudo code for our final algorithm is presented in Algorithm 1. Our main departure from the algorithm by Krause and Guestrin [12] is that we prepopulate set A with known (or easily attainable) variables W at the start, and that we prevent the algorithm from picking variables from the restricted set Z.

ALGORITHM 1: Modified Greedy Algorithm

```
input:
L > 0; graphical model G for V;
set of restricted variables Z ⊆ V;
set of known variables W ⊆ V\Z.
output: A ⊆ V\Z.
begin
        A ← W
        δ ← 0
        for i in 1: L do
                for each y in V\A do
                        if y ∉ Z then
                                sample H(y | A) from G
                                sample H(y | V\A ∪ {y}) from G
                                δ_y ← H(y | A) − H(y |V\A ∪ {y})
                        else
                                δ_y ← 0
                        end if
                end for
                x ← argmax_{δ_y} y
                A ← A ∪ {x}
        end for
end
```

4 METHOD FOR SELECTING QUESTIONS

Algorithm 1 offers an efficient way to estimate the subset of variables that maximize the information about all other user variables. However, the efficiency of the algorithm depends on the ability to quickly compute the mutual information between the subsets of variables or the conditional entropies used to compute the mutual information increase at each greedy step. A naïve approach would be to compute the conditional entropies in Equation 4 using a plug-in estimator $\tilde{p}$

that computes joint and conditional probabilities directly from the data. The algorithm would then use the estimator $\tilde{p}$ in entropy calculations:

$$H(y|A) = -\sum_{y} \tilde{p}(y) \cdot \log \tilde{p}(y|A) \tag{6}$$

However, such estimators may not be representative of the true distributions, because the data may not contain examples of all different combinations of variables. Also, the estimation error grows as the number of variables that describe different properties of users increases. Instead, we use a graphical model to represent the relationship between user variables. We then use this probabilistic model to estimate the conditional entropies.

4.1 Modeling User Properties

In this section, we describe our approach to build a probabilistic model of relationships between variables that describe the users (e.g., age, gender, home location, income). Although it is possible to build graphical models manually and using expert knowledge, this process becomes tedious when there are many user variables in the data. Instead, we use an automated approach to infer the structure of the model and the conditional probabilities between the variables in it.

To build graphical models that can be used with Algorithm 1, the training data for our automatically extracted model should include the variables that fully describe the user. Also, we assume that data from each variable is present in the dataset. To make it feasible to build the model and properly estimate the conditional probabilities of variables from the data, we perform unsupervised discretization on continuous and ordinal variables into equal sized bins. To further reduce the dimensionality of the data that can affect the accuracy of the model, it is possible to manually inspect categorical data and further group values together.

We then use the pre-processed data to build the model using the Chow-Liu tree algorithm [1]. The algorithm builds the graphical model by first calculating the mutual information between each pair of the variables in the model. Then the algorithm greedily picks the pair with highest mutual information at each step until all variables are included in the tree. Although this particular algorithm approximates the true distribution using a second-order dependency tree, the main advantage of this algorithm is that it is a fast and efficient way to automatically train the graphical model from large data sets with many variables.

4.2 Sampling Conditional Entropy

The graphical model allows us to sample conditional entropies required to select the next variable in each greedy step of Alg. 1. We use the conditional probabilities between variables in the graphical model to compute the joint and conditional probabilities used in Eq. 5 to estimate the conditional entropy. However, computing the joint and conditional probabilities directly from the graphical model takes time when there are many variables with many values.

In our approach, we use Krause and Guestrin's [12] efficient sampling algorithm for conditional entropy using a graphical model. The algorithm generates samples using the graphical model and then uses the probabilistic inference to compute the joint and conditional probabilities on those samples. In our approach, we use Gibbs sampling which approximates observations based on the probability distributions specified in the graphical model.

4.3 Selecting the Variables

It is worth noting that repeatedly running Algorithm 1 to select a subset of variables may still return different subsets. Although the algorithm always selects the variable with the highest increase in mutual information at each greedy step, the procedure for sampling conditional entropy above may return different results between runs. Picking a large enough sample size to estimate the conditional entropy reduces this error bound significantly [12]. Thus, Algorithm 1 still finds an approximate solution that is within $(1 - 1/e)$ from the optimum. In our approach, we are comfortable performing variable selection only once, knowing that both estimation errors are bounded and low.

5 LEARNING ABOUT USERS FROM CENSUS DATA

We now illustrate our approach on the American Community Survey (ACS) Public Use Microdata Sample [27] dataset collected by United States Census Bureau in 2013. This dataset contains detailed information from a sample of

approximately 3,000,000 people living in the United States. The 172 variables collected in the survey include information about people's demographics (e.g., their race, ethnicity, age, gender); where they live (e.g., their home location, the type of building, if they rent or own), where they work, if employed, or where they go to school, if students; their preferred transportation; their income; their marriage status; etc. Also, this particular dataset for the first time contains data about people's computer and phone use. We use this dataset to illustrate our approach because it is one of the most complete, systematically collected representative samples of potential personalized agent users in the United States.

5.1 Data Pre-processing and Variable Labeling

We begin by training a graphical model on the user data to build a probabilistic model of the potential users. Since most modern personalized agents run primarily on smart phones, we removed all the records for people that reported that they do not own a mobile phone. We also removed children who were 10 and younger at the time of the census. This left 1,799,394 records in the dataset. We performed unsupervised binning of continuous variables (e.g., age, income) into approximately equal sized bins. We then ran the Chow-Liu algorithm [1]. to approximate the structure of the graph and calculated the conditional probability tables for each pair of parent and child variables.

We then manually labeled variables that the algorithm should not select. Those variables included information about the user's gender, age, income, cost of living, marriage status, education level, and any specific questions related to their employment status. Although questions about some of these variables can be asked in a non-threatening way (e.g., "What do you do for a living?"), we decided to err on the side of caution and simply not ask them. We also removed any variables that could lead to awkward questions (e.g., whether the user's home has a toilet). Disallowed variables could be customized by the agent's creators, and our choices illustrate a possible example set.

We also labeled variables that the agent can detect through use of phone sensors and settings soon after the user interacts with the agent for the first time. These variables included home location, work or school location, and when they leave home to go to work or school, all of which can be detected using the location services on the phone in the few days following the first interaction. We also labeled variables about users' access to mobile and wireless data, and other computers the users own because the phone can detect this information based on the users' data connection and computers it connects to. We assume that the agent can detect all of these variables on its own, so they are excluded from the set of variables the agent might ask about. As with the disallowed variables, the set of detectable variables can be adjusted to suit the particular agent.

5.2 Variable Selection Run

To evaluate our approach, we performed three different variable selections: 1) *baseline*, a random selection run where we randomly selected variables without any restrictions; 2) *full*, where we used our selection approach, but did not withhold any variables and did not assume that any variables are already known to the agent; and 3) *restricted*, where we used our selection approach, and withheld variables that the agent should not ask about and assumed knowledge of variables that the agent can detect on its own. To estimate the average mutual information we would get by running the random selection, we repeated the *baseline* run 100 times. Scenario (3) is the most realistic, because it illustrates using our method with restrictions on what can be asked and inferred. We then compare the estimated mutual information for each condition across different numbers of selected variables. We hypothesize that *full* and *restricted* conditions will produce variable subsets with which mutual information is higher than that of randomly selected variables in the *baseline* condition.

We also explore different numbers of questions that the agent can ask the users on the first interaction. Although, the agent should keep the number of questions it asks low, it is still not clear what that number should be. Also, we hypothesize that the benefit of asking more questions (i.e., adding more variables) over time will decrease, and thus our method may be able to calculate a natural break for the number of variables to select. To compute the final subset of questions, we first pick a small number of variables based on mutual information curves across different number of variables, and then for each condition we selected the subset of variables of that size that had the highest mutual information. However, unlike *full* and *restricted* conditions, the random *baseline* condition cannot offer any guarantees about the solution it produces.

5.3 Results

A large mutual information between the known and unknown variables indicates that the known variables can be used to infer the unknown variables. Thus, we aim for high mutual information. We plot mutual information between selected variables and other variables for different number of selected variables (from 1 to 20) in the three conditions in Fig. 2.

As we expected, the two submodularity-based conditions outperform the baseline. Although the curve in the *baseline* condition continues to increase on the interval, the *baseline* mean mutual information does not come close to either of the other two conditions. Also, mutual information in the *restricted* condition is higher than in the *full* condition across all variable counts.

Fig. 2 shows that curves in both *full* and *restricted* conditions start to tail off after selecting five variables. This number coincides with our informal review of the existing phone agents and how many questions they ask when they interact with users for the first time (e.g., an early mobile version of Cortana asked the user three questions the first time they interacted). This small number of questions reduces the chances that the questions will burden or bore the users the first time they interact with the agent. We thus choose a cutoff point at 5 variables.

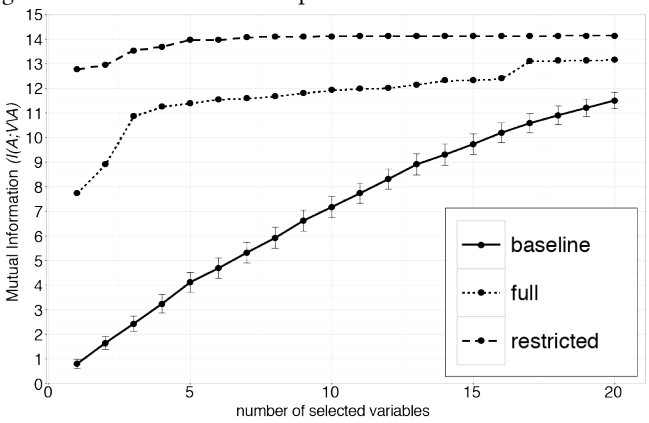

Fig. 2. Mutual information between the selected variables and the rest of the variables across the conditions for different number of selected variables. In the baseline condition, the mutual information represents mean across all 100 runs and the error bars indicate 95% confidence intervals.

The final subsets of five selected variables for each of the conditions (*baseline* includes the best subset in 100 runs) included: 1) gender, rent paid last year, home location, if the user were widowed in the past year, and age ($MI(A)$=11.08) in the *baseline*, 2) home location, work location, place of birth, the user's home location in the past year, and income in the past year ($MI(A)$=11.38) in the *full* condition, and 3) place of birth, the user's home location one year ago, year that the home was built, number of people in household, and birthday season ($MI(A)$=13.96, including known variables) in the *restricted* condition. The *restricted* condition starts at a relatively high point in mutual information, because it already knows about some of the user's variables, i.e. those that can be inferred by other means

The highest *baseline* run was incidentally close to the *full* condition and high compared to the mean mutual information of five selected variables in the *baseline* condition (mean=4.11). However, there is no guarantee that randomly selecting variables would again yield a subset with such high mutual information. Also, without a principled method, such as ours, it would be difficult to know how close the randomly selected subset is to optimal solution.

The variables selected in the *full* condition may better describe other user properties than *restricted* condition variables without the known variables. However, they also contain variables that the agent can detect (e.g., home location) and variables that the agent should not ask about (e.g., income). Asking questions about those variables would unnecessarily waste the user's time or could reduce the rapport between the user and the agent. Thus, we continue our analysis on the variables selected in the *restricted* condition.

5.4 What Selected Variables Tell about Users

Our algorithm selects the variables that best describe the user from a pool of available variables. However, our algorithm does not offer an explanation why certain variables are predictive of the others. The five variables that the algorithm selected in the *restricted* condition may not be obvious choices. We offer some hypotheses about why these variables are important (i.e., what information they carry about other variables). We randomly selected 1,000 people from the training data and manually inspected how truthfully answering the five questions affects knowledge about the rest of the variables. We illustrate this in two scenarios with representative users we identified in our random sample (Fig. 3). We focus on decisions the algorithm made on the training dataset. Later in Section 5.5, we validate our approach on the test dataset and show an increase in inference accuracy on a set of unseen users after they hypothetically answered our questions.

In the first scenario, we pick variables that have been traditionally sought after by advertisers and marketers [21]: age and personal income. Such variables enable smart agents to make decisions about age-sensitive content and target users with appropriate advertisements. Our second scenario explores variables used to automatically suggest transportation options and commute times to a frequent location. Existing agents, such as Google Now, already provide this service to their users (e.g., suggesting routes based on commute time when driving to work). For this scenario, we chose to examine users' main means of transportation and the users' employment status (including if they are students). Knowledge about these variables helps the agent avoid making gross assumptions about the user based on the majority of users in the United States (e.g., assuming that everybody has a car or that most people are employed and go to work).

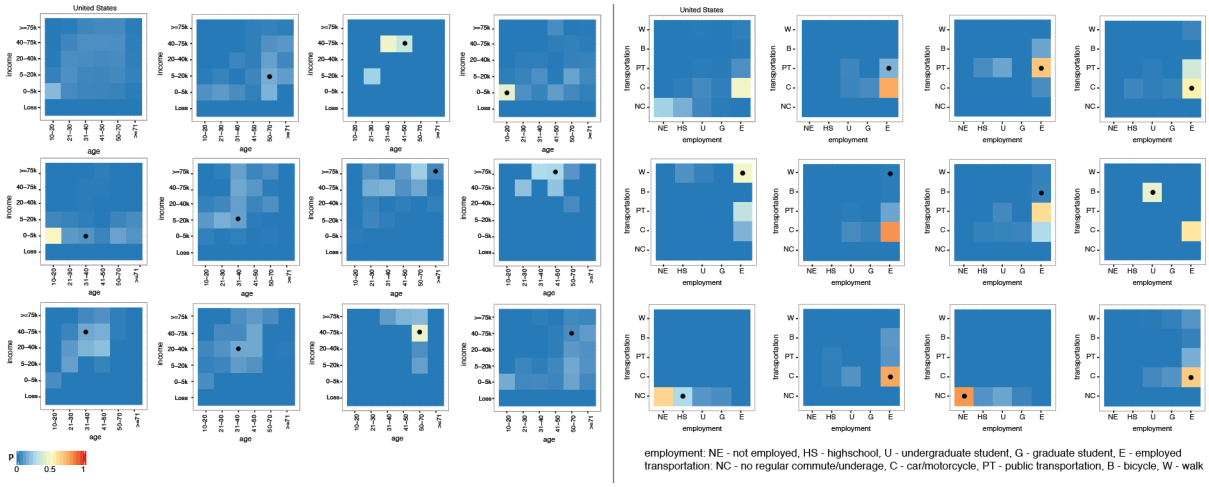

Fig. 3. Joint distribution of age and personal income (left) and users' employment and main transportation (right) in the United States: 1) without any knowledge about the user (top left), and 2) from select 11 randomly sampled people from the US census data assuming the users have answered the selected questions truthfully and the agent has detected the other variables correctly. Black dot indicates actual user properties.

5.4.1 Scenario 1: User Age and Personal Income

It is difficult to infer the user's age and income just from the joint probability distribution of income and age of all potential users in the United States (see top left in Fig. 3). This is because users are close to uniformly distributed across different age groups (i.e., the entropy of the probability distribution of ages is high). Although slightly skewed towards lower values, users can also have a wide range of incomes. However, we see that information from variables we select in the *restricted* condition contained information about both variables (Fig. 3 left).

Fig. 3 (left) shows the joint distributions of age and personal income before and after the algorithm selected the questions for 11 representative people from the US census dataset. We found that in the majority cases the method was able to narrow down the joint probabilities towards the people's actual age and personal income. This is in part due to the variables the agent can detect on its own. For example, home location can tell the difference between affluent and poverty-stricken areas. It can also shift the distribution of ages in case of, say, university towns. User answers to agent questions

further help narrow down the probability distributions. For example, user answers about the year their home was built helped the algorithm differentiate young home owners from older people who have owned their home for a long time.

Knowing these variables would enable a smart agent to provide personalized services to the user as soon as the user completes answering the questions. For example, a match-making service could use the two variables directly to match people based on their age and income. Smart agents can also use age and income indirectly to infer other preferences. A service that provides up-to-date financial news could target users with high income. A service that recommends new restaurants, shopping, or entertainment could also use this information to personalize its suggestions to the user.

5.4.2 Scenario 2: User Work Status and Transportation Preferences

Users' work status and transportation preferences variables differ from age and income variables because, unlike age and income, the joint probability distribution of work status and transportation in the United States is highly skewed towards employed people who own a vehicle or non-employed people who do not commute frequently (see top left corner in Fig. 3 (right)). This means that without any additional information the smart agent will inadvertently suggest a car as the preferred transportation option (as is the case with existing smart agents that offer this service). This could also lead to uncomfortable situations of suggesting "work" as a destination to people who are unemployed.

Fig. 3 (right) shows the joint distribution of employment status (including if the user is a student) and transportation before and after the algorithm selected the variables for 11 representative people from the US census dataset. We found that in most cases the selected variables helped strengthen the algorithm's belief about whether the user owns a vehicle and is employed or not. The algorithm reduced uncertainty about people that take public transportation to work, especially for users living in urban areas. However, selected variables did not provide the algorithm with information about the users that walk or bike to work or school due to a small number of people who have such preference.

The agent can make a suggestion about commutes based on the user's employment status and transportation preferences. For example, instead of suggesting the car option to a user who does not drive, the agent could offer public transportation with appropriate timetables. Inferring that the user is a student could allow the agent to suggest different content (e.g., learning materials and resources). It could also advertise different discounts available only to students.

5.5 Validating Inference Accuracy

Our goal is to show that the agent can make accurate inferences about the user based on the answers to the five questions. Unlike in the previous section where we looked at why the algorithm made certain decisions on the training data, here we use a hold-out set of new users to show that the inference generalizes. Our algorithm already offers theoretical guarantees that the agent will increase knowledge about the users. We hypothesize that inference accuracy will improve as well.

5.5.1 Method and Data Pre-processing

We used another American Community Survey (ACS) Public Use Microdata Sample [27] dataset collected by United States Census Bureau in 2015. The testing dataset contained the same 172 demographics variables from approximately 3,000,000 different people living in the United States. The sampling method used by United States Census Bureau ensures no overlap between the two datasets. We again removed records for people that reported that they do not own a mobile phone and children who were 10 and younger at the time of the census, which left 1,618,489 phone users in the testing dataset.

We then used the graphical model we built on the training data to perform inference in three conditions: 1) *baseline*, in which we did not use answers to any questions in our inference, 2) *full*, in which we assumed that the users in the test dataset truthfully answered questions from the *full* set of questions we obtained in our variable selection run, and 3) *restricted*, in which we assumed that the users in the testing dataset truthfully answered questions from the *restricted* set of questions and that the phone obtained the rest of the variables in that set.

We measured the accuracy for each of the 172 variables in the dataset. We compared the average accuracy over all 172 variables between the three conditions because we wanted to measure the ability of the agent to learn as much about the users rather than about any particular user variable.

5.5.1 Results

The average accuracy for the *default*, *full*, and *restricted* conditions was 67.56%, 72.95%, and 76.18% respectively. The obvious source of increase in average accuracy were variables that the agent learned from the users (e.g., home location accuracy increased from less than 0.1% in the *default* condition to 100% in the *full* and *restricted* conditions). However,

known variables are only a fraction of the 172 total variables, and the increase came from the gain in overall knowledge about each user.

To illustrate this, we again compare the four unknown variables we explored in the previous section: age, income, employment, and transportation. To contrast *full* and *restricted* conditions we also include a variable that describes if a child lives in the same home. The average accuracy for each of these variables across different conditions are in Tab. 1. The results show improvement in accuracy in all but one of these variables (age). The increased accuracy in *restricted* condition comes from nuanced knowledge about additional user variables compared to the *full* condition.

Table 1. Average accuracy for select variables across conditions.

Condition	Age	Income	Employment	Transportation	Child at Home
baseline	20.61%	6.75%	80.31%	15.19%	36.81%
full	19.38%	28.27%	98.45%	80.66%	36.81%
restricted	19.38%	28.27%	98.45%	80.66%	80.32%

6 DISCUSSION

We presented both a theoretical guarantee and empirical evidence that the selected subset of variables will help the agent make, on average, more accurate inferences about the users. The mutual information comparison results in Fig. 2 show that our algorithm will improve on the ability to infer user properties compared to doing the same inference from simple priors or by asking random questions about the user. Our algorithm performs the selection in a principled way that also avoids asking questions that may negatively impact the rapport with the user.

Both our manual analysis of the training data and our empirical evidence show that the algorithm picked meaningful variables. For example, our results indicate that the location variables (e.g., home, work) provide significant amount of information about the user and should be inferred as soon as possible. We also found that the place of birth, which was present in both *full* and *restricted* conditions, could be descriptive of the user because it could indicate whether the user is a citizen, a naturalized citizen, or an immigrant. If the user was born in the same place he or she lives now, it might also mean that the user had time to build social capital among people that live in the same place and could affect variables related to the person's employment. Similarly, the location the user lived in the past year indicates whether the user moved or not. Moving residence to another city or state may indicate changes in jobs, school, or even family relations, such as marriage. The number of people that live with the user could indicate whether the person is married and has a family (has children or is living with parents) or lives alone.

Other variables provided more nuanced description of the users. A high number of people living in the same residence could also indicate the user lives in a university dorm. The age of the building where the user lives may be indicative of their social and economic status: an old home may be indicative of the poverty in neighborhoods where the average income of residents is low, or it may indicate that the user lives in an old mansion, if the general area is affluent.

A notable exception is the birthday season, which illustrates a possible pitfall with our algorithm. This variable may not have a direct impact on other variables, but it may be difficult to guess even when knowing other variables because the entropy of this variable is very high. Thus, the algorithm may be inclined to select variables with high entropy that are not descriptive of other variables. To counter this, we could iterate and add such variables to the list of variables that the algorithm should not select.

Our approach assumes that people will answer all the questions the agent asks and answer them truthfully. In reality, some people many not want to answer any questions or may provide false information if forced to answer them. Additionally, some people may not take the questions seriously and give humorous answers that are not true. Further research is required to determine the likelihood of people answering certain questions and how this impacts our method. Results of such future research may also relax some of the restrictions we have placed on the impolite questions, or may add further questions to the "do not ask" list. Furthermore, since we have access to the underlying probability distributions, we may be able to detect some untruthful answers as those that are highly unlikely.

Our algorithm selects the variables, but does not specify how the agent should ask the questions. This requires additional planning by the agent designers. For example, asking user directly: "How many people do you live with?" may feel awkward. Modifying the question to be playful or conversational changes the tone (e.g., offering some information

about the agent in exchange for information from the user). This is similar to the usual conversations between people that could also build rapport between the agent and the user. Based on the top five variables we selected, we suggest the following questions:

- For place of birth: "I was born in Seattle. Where were you born?"
- For moving home location: "Have you lived in this place long or have you moved recently?"
- For season of birth: "I was born in summer. How about you?"
- For number of people living at the user's home: "This is going to be my first time living with roommates. How many people will I be living with?"
- For the year the home was built: "I am excited to share the home with you. What year was our home built?"

One limitation of our use case dataset is that it did not contain all variables that could be relevant to mobile personalization services (e.g., application preference, music preferences). However, it is unrealistic to expect that every training dataset will always contain all possible user properties. Therefore, it is important to consider the ability to combine data from multiple sources. Because our approach uses a graphical model in the optimization, it is able to use any probabilistic model that has been estimated from multiple data sources.

In our work, we focus on a small subset of questions that the agent should ask the first time it interacts with the user. However, our approach can be used to select a larger subset of variables to ask the users about over time. Also, as the rapport between the user and the agent grows, the agent may be able to start asking some of the restricted questions. For example, after the agent has established rapport with the user, the agent may be able to ask some of the more personal questions, such as their actual age or if they are employed or not. In such cases, it may be beneficial to repeat our approach for different stages of user-agent interactions, where certain questions are removed from the "do not ask" list over time. This would make our approach an essential part of the process of learning about the users throughout their interaction with the agent.

7 CONCLUSION AND FUTURE WORK

We presented a generalizable approach that enables agents to automatically learn fundamental user properties. Our approach does this in a principled way by maximizing the mutual information between sets of variables that describe the users. We extend the existing algorithms that take advantage of the submodularity of the mutual information function to include cases when the algorithm cannot select a subset of variables, but still wants to maximize information about them. As such, our work enables future agents to learn about the users without asking inappropriate questions, which could negatively impact the rapport between the agent and the users.

To offer the best service to the users, future agents will need the ability to consider all variables relevant to the services they offer. However, it is difficult to collect training data that contain all of the user's properties in one dataset. Thus, it is important to find ways to combine data from multiple sources into a single probabilistic model of the users. Although our variable selection algorithm makes no assumptions about the underlying user model, future work should explore what fundamental properties of users the algorithm would be able to detect using those complete user models.

Future work should also explore the cases where the users are unable or choose not to answer certain questions. Future algorithms should consider the probability of people answering certain questions and the probability that the answers are correct. Furthermore, some questions may be easier to answer than the others and future algorithm should consider the difficulty of answering questions when selecting the best subset of variables to ask about.

Our work focuses on immediately improving the first interaction between the agent and the user. However, the agent will progress to learn more about the user after their first interaction. Thus, future research should explore how our work can support continued learning. This includes how algorithms should decide the order of future questions, including how the subsequent answers impact what the agent should ask next.

Our work produces holistic user profiles that help general-purpose agents (e.g., Cortana, Google Now, Siri) when trying to provide value to the user in multiple domains. We envision a future in which such general-purpose agents will try to find out and keep track of as much holistic information about the user as possible to be able to delegate this information to other specialized agents. A consequence of this vision is that future smart agents will have much more high-level knowledge rather than specialized knowledge about the users at first. Our method has implications for future specialized agents, because it will provide them with a starting point for inference about the specific information they need to make their specialized predictions and recommendations.

ACKNOWLEDGMENTS

We would like to thank Chris Meek for his input on this work.

REFERENCES

[1] C. K. Chow and C. N. Liu. 1968. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory, 14(3),* 462-467. DOI: http://dx.doi.org/10.1109/TIT.1968.1054142
[2] Jan O. Blom and Andrew F. Monk. 2003. Theory of personalization of appearance: why users personalize their pcs and mobile phones. *Human-Computer Interact.* 18 (3) (September 2003), 193-228. DOI: http://dx.doi.org/10.1207/S15327051HCI1803_1
[3] Anind K. Dey. 2001. Understanding and Using Context. *Personal Ubiquitous Comput.* 5, 1 (January 2001), 4-7. DOI: http://dx.doi.org/10.1007/s007790170019
[4] Berkeley J. Dietvorst, Joseph P. Simmons, and Massey Cade. 2014. Algorithm Aversion: People Erroneously Avoid Algorithms after Seeing Them Err. *Journal of Experimental Psychology: General* (July 6, 2014). DOI: http://dx.doi.org/10.2139/ssrn.2466040
[5] Linda van der Gaag and Maria Wessels. 1993. Selective evidence gathering for diagnostic belief networks. *AISB Quarterly, 86*, 23–34.
[6] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle and Lars Schmidt-Thieme, 2010. Learning Attribute-to-Feature Mappings for Cold-Start Recommendations. In *IEEE International Conference on Data Mining*, Sydney, NSW, 176-185. DOI: http://dx.doi.org/10.1109/ICDM.2010.129
[7] Nadav Golbandi, Yehuda Koren, and Ronny Lempel. 2011. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the fourth ACM international conference on Web search and data mining* (WSDM '11). ACM, New York, NY, USA, 595-604. DOI: http://dx.doi.org/10.1145/1935826.1935910
[8] Alkis Gotovos, Amin Karbasi, and Andreas Krause. 2015. Non-monotone adaptive submodular maximization. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 1996-2003. AAAI Press.
[9] Isabelle Guyon and André Elisseeff. 2003. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3, 1157-1182.
[10] Andreas Krause and Daniel Goldvin. 2014. Submodular Function Maximization. *Tractability: Practical Approaches to Hard Problems.* Cambridge University Press.
[11] Neil Houlsby, José Miguel Hernández-Lobato, and Zoubin Ghahramani. 2014. Cold-start Active Learning with Robust Ordinal Matrix Factorization. In *International Conference on Machine Learning* (ICML), 766-774.
[12] Andreas Krause and Carlos Guestrin. 2005. Near-optimal Nonmyopic Value of Information in Graphical Models. *In Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence* (UAI).
[13] Andreas Krause, Ajit Singh, and Carlos Guestrin. 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research* 9, 235-284.
[14] Andreas Krause and Eric Horvitz. 2010. A utility-theoretic approach to privacy in online services. *Journal of Artificial Intelligence Research*, 633-662.
[15] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. 2008. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication* (ICUIMC '08). ACM, New York, NY, USA, 208-211. DOI: http://dx.doi.org/10.1145/1352793.1352837
[16] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. Expert Systems with Applications, 41, (4), 2, 2065-2073. DOI: http://dx.doi.org/10.1016/j.eswa.2013.09.005
[17] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming, 14,* 265–294.
[18] Seung-Taek Park, David Pennock, Omid Madani, Nathan Good, and Dennis DeCoste. 2006. Naïve filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '06). ACM, New York, NY, USA, 699-705. DOI: http://dx.doi.org/10.1145/1150402.1150490
[19] Furong Peng, Jianfeng Lu, Yongli Wang, Richard Yi-Da Xu, Chao Ma, and Jingyu Yang. 2016. N-dimensional Markov random field prior for cold-start recommendation. *Neurocomputing* 191 (2016), 187-199.
[20] Francesco Ricci. 2010. Mobile recommender systems. *Information Technology & Tourism* 12(3). 205-231. DOI: http://dx.doi.org/10.3727/109830511X1297870228439
[21] Mark E. Slama and Armen Tashchian. 1985. Selected Socioeconomic and Demographic Characteristics Associated with Purchasing Involvement. *Journal of Marketing, Winter 1985*, 72-82.
[22] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '02). ACM, New York, NY, USA, 253-260. DOI: http://dx.doi.org/10.1145/564376.564421
[23] Choonsung Shin, Jin-Hyuk Hong, and Anind K. Dey. 2012. Understanding and prediction of mobile application usage for smart phones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (UbiComp '12). ACM, New York, NY, USA, 173-182. DOI: http://dx.doi.org/10.1145/2370216.2370243
[24] Helen Spencer-Oatey. 2005.(Im)politeness, face and perceptions of rapport: unpackaging their bases and interrelationships. *Journal of Politeness Research. Language, Behaviour, Culture* 1, no. 1, 95-119.
[25] G. Sridhar. 2007. Consumer involvement in product choice – a demographic analysis. *XIMB Journal of Management, March 2007,* 131-148.
[26] Mingxuan Sun, Fuxin Li, Joonseok Lee, Ke Zhou, Guy Lebanon, and Hongyuan Zha. 2013. Learning multiple-question decision trees for cold-start recommendation. In *Proceedings of the sixth ACM international conference on Web search and data mining* (WSDM '13). ACM, New York, NY, USA, 445-454. DOI: http://dx.doi.org/10.1145/2433396.2433451
[27] United States Census Bureau. *American Community Survey (ACS) Public Use Microdata Sample.* https://www.census.gov/programs-surveys/acs/technical-documentation/pums.html
[28] Viswanath Venkatesh, Michael G. Morris, Gordon B. Davis, and Fred D. Davis. 2003. User acceptance of information technology: toward a unified view. *MISQ.* 27, 3 (September 2003), 425-478.

Received February 2017; revised August 2017; accepted October 2017.