

Making Better Recommendations with Online Profiling Agents

Danny Chin Hock Oh and Chew Lim Tan

■ In recent years, we have witnessed the success of autonomous agents applying machine-learning techniques across a wide range of applications. However, agents applying the same machine-learning techniques in online applications have not been so successful. Even agent-based hybrid recommender systems that combine information filtering techniques with collaborative filtering techniques have been applied with considerable success only to simple consumer goods such as movies, books, clothing, and food. Yet complex, adaptive autonomous agent systems that can handle complex goods such as real estate, vacation plans, insurance, mutual funds, and mortgages have emerged. To a large extent, the reinforcement learning methods developed to aid agents in learning have been more successfully deployed in offline applications. The inherent limitations in these methods have rendered them somewhat ineffective in online applications. In this article, we postulate that a small amount of prior knowledge and human-provided input can dramatically speed up online learning. We demonstrate that our agent HumanE—with its prior knowledge or “experiences” about the real estate domain—can effectively assist users in identifying requirements, especially unstated ones, quickly and unobtrusively.

Electronic profiling agents have been deployed with considerable success in certain agent-based recommender systems, namely, information filtering systems, collaborative filtering systems, and hybrid recommen-

der systems. Unfortunately, the success of the interactive learning method has been restricted to simple consumer goods such as movies, books, clothing, and food. When the information retrieval or collaborative filtering system techniques plus other reinforcement learning methods are applied in online applications for complex consumer products such as real estate, vacation plans, insurance, mutual funds, and mortgages, they fail to enjoy much success, because many reinforcement learning implementations assume that the agent developed knows nothing about the environment to begin with and that the agent must gain all of its information by exploration and subsequent exploitation of learned knowledge.

When dealing with a real, complex online system such as a large-scale real estate listing and brokerage application, however, these approaches are simply not practical. Typically, the state space is too large to explore satisfactorily within the lifetime of the agent (much less within the attention time-span of typical online users). Worse still, making random exploratory recommendations can frustrate and disappoint users, potentially causing them to abandon the system totally.

Accumulated knowledge, in the form of memories and experiences, allows humans to go about performing daily tasks. In the real world, we often go to a human real estate agent for assistance in selling or acquiring real estate

properties. Naturally, we expect the agent to be an expert in the real estate domain, and hence able to offer suitable advice and recommendations. Certainly, we do not expect the real estate agent to have no knowledge about the real estate domain. Hence, in order to take our prior knowledge (which is often implicit) and incorporate it into a reinforcement learning framework, we have examined in this work the idea of supplying the agent with an initial policy about the real estate domain in the HumanE agent (hereafter shortened to HumanE).

Better Recommendations with HumanE

There are ten main problems encountered when developing online profiling agents for complex multidimensional domains. First, the assumption that the agent knows nothing and must acquire its knowledge through exploration and subsequent exploitation of learned knowledge results in slow agent learning for complex domains and makes online implementation difficult. Second, it is difficult to give an agent a large amount of application-specific and domain-specific knowledge. Third, it is difficult to encode this application-specific and domain-specific knowledge in an abstract language. Fourth, it is difficult to transfer agent knowledge and the control architecture for building agents for other applications. Fifth, it is difficult to maintain the individual rules in the agent rule base over time. Sixth is the problem of static agent knowledge (that is, knowledge that cannot be customized to individual user habits and preferences). Seventh, making random exploratory recommendations can frustrate and disappoint the user. Eighth, it is difficult to allow for serendipitous discoveries of user preferences. Ninth, it is difficult to obtain user trust when an interface agent is very sophisticated, qualified, and autonomous from the start. Finally, too much data is required in an online setting for typical learning methods (for example, reinforcement learning methods)

Practical Approach

We strongly believe that practical agent learning for online applications is possible by integration with human-supplied knowledge. We believe this because humans can provide a lot of help to assist agents in learning, even if people cannot perform the task very well. Humans can provide some initial successful trajectories through the space. Trajectories are not used for supervised learning, but to guide the learning methods through useful parts of the search

space leading to efficient exploration of the search space.

Online profiling agents can be bootstrapped from a human-supplied policy, which basically gives some sample trajectories. The purpose of the policy is to generate “experiences” for the agents. This policy can be hand-coded by domain experts, need not be optimal, and may be very wrong. The policy shows the agents “interesting” parts of the search space.

Our online profiling agent, HumanE, is based on the aforementioned approach, and it offers users the opportunity to find products that best meet their requirements. HumanE guides users through a product selection process. Users get to specify information about their individual requirements and restrictions by creating and refining their profiles. Based upon the profile (and initial policy if the profile is newly created), HumanE offers an initial selection of products. Users can then select from these matching products to view more detailed product information such as product features. HumanE also tries to be helpful by providing products that are newly added as well as products that are popular among other users.

To refine the profile, users can specify which features are desirable or undesirable through an intuitive and friendly interface, and HumanE offers a new selection of products matching the revised profile. If no matching products are found, users can backtrack to their previous profile.

Design Assumption

Our design approach assumes that the entire user experience is an iterative process of browsing and meaningful user feedback. The approach has in fact been adopted successfully by similar systems such as RentMe (Burke, Hammond, and Young 1997), CASA (Gao and Sterling 1998), and Apt Decision (Shearin and Lieberman 2001). As the user is actively involved throughout the entire profile-creation process, the user can react independently to every feature of the real estate offerings.

Domain Analysis

To test the feasibility of the proposed learning model, we chose the real estate domain. Because the agent needed to have built-in knowledge about the domain, we analyzed online and offline apartment advertisements to determine the standard apartment features for the local real estate domain. Next, we considered how people choose apartments. After examining the features, we concluded that some of them (for example, district, type, and price) were pivotal to the final choice of apartment.

That is, most people would reject an apartment if the value for a crucial feature was not to their liking. Other features (such as bridge, underpass, or swimming pool) were less pivotal. All this domain knowledge went into HumanE. In addition, we examined two destinations of apartment seekers, real estate websites and human real estate agents, to determine what knowledge we could glean from those interactions.

Many real estate websites adopt either the pure browsing metaphor¹ or the searchlike metaphor.² One problem is that users are expected to enter many specific details about their ideal apartment. Another problem is that they must enter their preferences when they visit a new site and each time they visit the site. This is because there is no option to save multiple sets of preferences for a single site.

In our examination of human real estate agents, we considered how people deal with the ambiguity and imprecision of real-world decisions. For example, when a customer interacts with a real estate agent, the customer may react in a variety of ways not limited by answers to explicitly posed questions. The agent's description typically contains many details not asked for originally by the customer. The success of the interaction is determined largely by the agent's ability to infer unstated requirements and preferences from the responses. Near-miss examples establish whether the ostensible constraints are firm or flexible. Good agents are marked by their ability to converge quickly on a complicated set of constraints and priorities.

Transferring Domain Knowledge. Much of the work done for HumanE would transfer well into any domain in which the user could browse the features of a complex object. That is, objects such as calling plans, mutual funds, homes, computers, vacation plans, or cars would work well, but simple consumer goods such as clothing or food would not. Transferring the agent into another domain would require the services of a subject matter expert who could identify salient features of the complex objects in the domain, alter the program to work with those features and determine which features were crucial to the final decision. After testing on a suitable list of objects, the "new" agent could be released.

Learning Approach

The proposed two-phase learning approach has been tested successfully in past research on robotics (Smart and Kaelbling 2002). Kaelbling and Smart found that robots using reinforce-

ment learning learned better when they were provided prior knowledge about their environment using a supplied initial policy. The policy generated example trajectories through the state-action space and showed the robot areas of high rewards and low rewards. After the robot had acquired a suitable amount of information through this initial phase of learning, the reinforcement learning system took control of the robot. Usually by this time, the robot had learned enough to make more informed exploration of the environment.

In our work, we adapt a similar approach when building an agent-based online real estate system. To do so, we consider each user decision as a trajectory in the search space much like the trajectories in the robot motion.

Initial Profile Versus Initial Policy

Initial profile refers to the profile that is created at the very beginning of the learning approach. The initial profile contains only the user-defined preferred district, desired apartment type, and price. *Initial policy* refers to the set of trajectory samples that show HumanE areas of high rewards and low rewards in the search space.

Constituents of a Profile

The main objective of HumanE is to create a user profile (or, simply called, a *profile*) to store user preferences and to assist the user to refine his or her profile intelligently using the supplied learning approach. In our scenario, a profile stores both static and dynamic (learned) user preferences in the form of desired and undesired apartment features. Examples of apartment features include "high floor," "near MRT (mass rapid transit)," "marble floor," and so on.

Overview of the Learning Approach

We adopted a two-phase learning approach for HumanE. In the first phase of learning, HumanE learns by reinforcement and observation and takes actions that arise from a supplied initial policy. This mode of learning lasts for one iteration of the profile-refinement process (that is, the iterative process of viewing apartments and selecting/ranking desired and undesired features). In the second phase, HumanE learns by reinforcement and observation. The content of the initial policy is dynamic as it is updated without human intervention from the actions taken by HumanE. Figure 1 depicts the workflow of the two-phase learning approach.

Phase-One Learning

In the first phase of learning, HumanE initially learns and takes action from a supplied initial policy. This occurs right after the creation of an

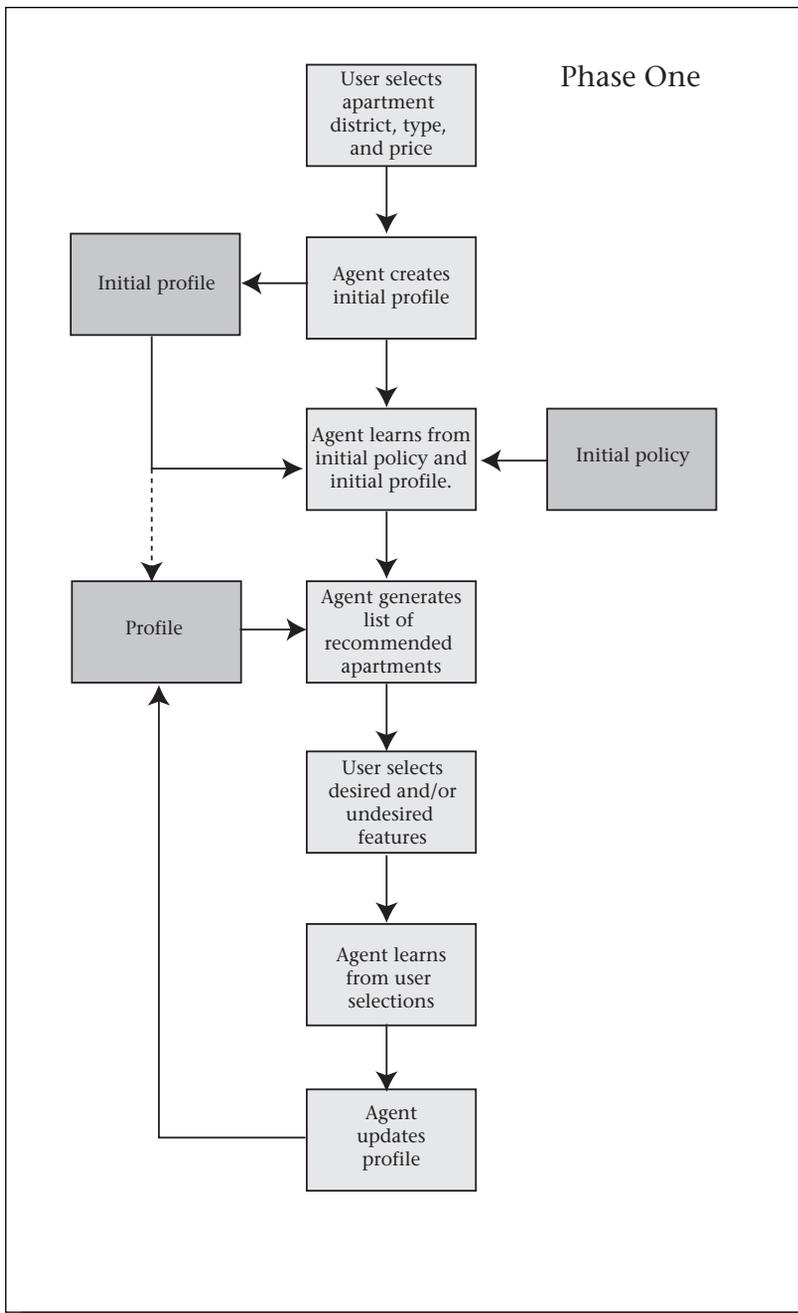


Figure 1. Workflow for the Two-Phase Learning Approach.

initial profile. Using the information stored within the supplied initial policy and initial profile, HumanE learns the locations of desired (high rewards) and undesired (low rewards) apartments in the search space that match the user preferences stored in the initial profile.

To emulate some of the inference power a human real estate agent might have, we have incorporated an initial policy to enhance the interactive learning ability of the agent. Basically, an initial policy is a XML file that stores information about which apartment features

are generally considered as desirable and undesirable. In HumanE, the initial policy is stored in a file named “Bootstrap.xml.” Figure 2 shows the content of this file.

The first piece of information encoded in the initial policy as shown in figure 2 is the list of features generally considered as desirable and undesirable. The attribute *id* denotes the attribute name and the attribute *value* refers to the value of the attribute *id*.

The second piece of information encoded is the list of top 10 most popular apartments per district. The attribute *id* denotes the attribute name, which in this case refers to the district *id*. The attribute *value* refers to the value of the attribute *id*, that is, the apartment *id* of the top 10 most popular apartments per district.

The third piece of information encoded is the list of top 10 most unpopular apartments per district. The attribute *id* denotes the attribute name, which in this case refers to the district *id*. The attribute *value* refers to the value of the attribute *id*, that is, the apartment *id* of the top 10 most unpopular apartments per district.

Coupling with the information stored in the initial profile, the initial policy generates certain “interesting” trajectories through the search space, showing HumanE areas of high (popular apartments) and low (unpopular apartments) rewards. These trajectories and associated rewards are then used in this first, passive phase of learning.

Consequently, HumanE is able to generate a larger but potentially interesting set of matching apartments. This gives users an opportunity to learn about apartments that do not exactly match their initial profile but may be of interest to them. In this way, HumanE allows for the serendipitous discoveries of new user preferences without the danger of random, unguided “exploratory” recommendations.

As shown in figure 1, the bootstrapping process occurs after the creation of the initial profile. Here, HumanE observes the states and rewards being generated and bootstraps this information into its memory. In our domain, the areas of high rewards are those apartments that have at least half of the desired features as specified in the initial policy.

On the other hand, the areas of low rewards are those apartments that have at least half of the undesired features as specified in the initial policy. We have labeled each feature as either “desired” or “undesired” based on common-sense rules.

In short, this process corresponds to a real-life situation in which a human real estate agent always has in mind a small number of re-

al estate properties that he or she knows are popular or unpopular by virtue of their features. The initial policy shows HumanE the locations of potentially “desired” and “undesired” apartments based on the profile. For example, if the user has specified in the profile that the features desired are “MRT,” “Schools” and “High Floor,” then HumanE searches for matching apartments that have a combination of the following criteria: (1) all desired features as stated in the profile (“MRT,” “Schools” and “High Floor”). (2) More than half of the desired features as stated in the initial policy (such as “MRT,” “Bus Stop,” “Lift Level,” “Mid Floor,” “Good View,” and “Windy”). (3) Less than half of the undesired features as stated in the initial policy (for example, “Playground,” “Rubbish Dump,” “Low Floor,” “Blocked View,” and “Facing Afternoon Sun”).

After HumanE has generated a list of recommended apartments, it adopts reinforcement learning as the next learning technique to learn user preferences. It learns a multidimensional utility function on states (or histories) and uses it to select actions that maximize the expected utility of their outcomes. The reinforcement learning approach is used through the entire profile refinement process. In this way, the multidimensional utility function is able to capture past profile changes (that is, the agent remembers history information) and incorporate the knowledge learned into a simple representation to be used by the matching algorithm.

Learning by Observation. To augment the serendipitous discoveries of apartments that can be of potential interest to the user, we have implemented the “favorites” and “views” functions. First, the user can specify an apartment to be added to a “favorites” list for a particular profile. Second, the user can select an apartment to develop the profile or simply to view more details about it. As part of the matching process, HumanE selects the top 10 apartments, which are the top “favorites” (figure 3) and widely “viewed” (figure 4). This selection encourages users to make more serendipitous discoveries of apartments that they may be interested in. The assumption taken here is that there is a high possibility that a typical user may be interested in apartments that are generally considered by other users to be “good.”

Matching Algorithm. HumanE employs a matching algorithm that is based on the concept of property relaxation. It uses a sequence of built-in rules when searching for matching apartments. If no matching apartments can be found, HumanE displays the apartments listed in the “top 10 most popular apartments per

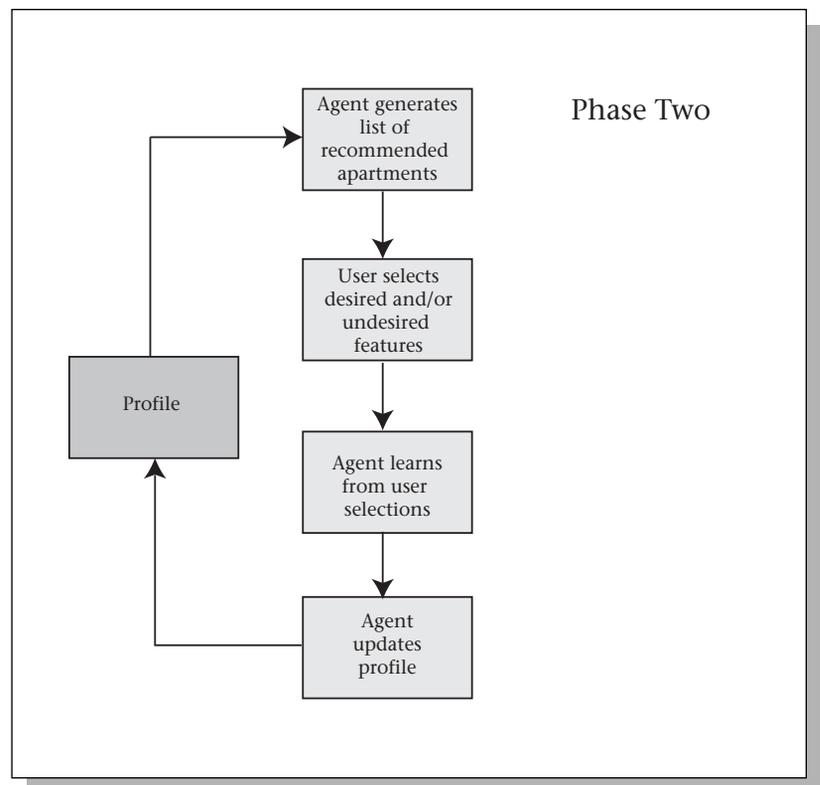


Figure 2. Initial Policy Used in HumanE.

district” information contained in the initial policy for the district specified in the profile.

User Interface. We do not expect an average user to have a high degree of computer skills. Hence, we have paid extra attention to the design of the agent interface. We have made several changes to the HumanE agent interface in the hope that it will be intuitive and responsive to users’ actions.

After trying out a few approaches and gathering some useful user feedback, we decided to use two list boxes, “desired,” and “undesired,” to enable users to specify desirable and undesirable features. A user clicks the left arrow button to add a feature to the “desired” list box or clicks the right arrow button to add a feature to the “undesired” list box. Furthermore, users can rank the features in each list box in accordance to their liking. The features at the top of each list box correspond to those features liked most or liked least.

One advantage of using the present user interface is that users are encouraged to consider carefully the features they like most or least, as indicated in the list boxes; their selection helps speed up the process of discovering unstated user preferences. Figure 5 shows the interface of the current version of HumanE.

Street/Block/Unit	District	Type	Area	Price	#Favs	#Views		
Blk 656 Jln Tenaga #05 and above	Bedok / Chai Chee	5I	0	\$320,000.00	98	80	[Fav]	[Details]
Blk 51 Bukit Merah View	Bukit Merah / Queenstown / Telok Blangah	4S	0	\$420,000.00	98	49	[Fav]	[Details]
Blk 30 Jalan Klinik #03	Bukit Merah / Queenstown / Telok Blangah	3	0	\$335,000.00	98	50	[Fav]	[Details]
Blk 219 Spottiswoode Park Rd #02	Choa Chu Kang / Bukit Panjang	3A	0	\$320,000.00	98	94	[Fav]	[Details]
Blk 485A Choa Chu Kang Ave 5 #08	Choa Chu Kang / Bukit Panjang	4A	0	\$320,000.00	98	71	[Fav]	[Details]
Blk 112 Hougang Ave 1 #04 and below	Hougang / Serangoon / Yio Chu Kang	4NG	0	\$380,000.00	98	90	[Fav]	[Details]
Blk 510 Hougang	Hougang / Serangoon / Yio Chu Kang	5I	0	\$245,000.00	98	29	[Fav]	[Details]
Blk 128 Lorong Ah Soo	Hougang / Serangoon / Yio Chu Kang	EA	0	\$320,000.00	98	43	[Fav]	[Details]
Blk 101 Serangoon North Ave 1 #10 and above	Hougang / Serangoon / Yio Chu Kang	4	0	\$290,000.00	98	29	[Fav]	[Details]
Blk 317 Tampines	Tampines / Simei	5I	0	\$170,000.00	98	96	[Fav]	[Details]

Figure 3. Most Popular Apartments.

Street/Block/Unit	District	Type	Area	Price	#Favs	#Views		
Blk 130 Clarence Lane	Bukit Merah / Queenstown / Telok Blangah	5	0	\$480,000.00	55	98	[Fav]	[Details]
Blk 52 #08 Kent Rd	Central / City	4NG	0	\$210,000.00	38	98	[Fav]	[Details]
Blk 51 #10 Kent Rd	Central / City	4NG	0	\$201,000.00	89	98	[Fav]	[Details]
Blk 287 Choa Chu Kang	Choa Chu Kang / Bukit Panjang	5EM	0	\$400,000.00	64	98	[Fav]	[Details]
Blk 14 Clementi	Clementi / West Coast	5S	0	\$250,000.00	19	98	[Fav]	[Details]
Blk 8 Jalan Batu #09	Marine Parade	3	0	\$400,000.00	48	98	[Fav]	[Details]
Blk 185 Pasir Ris St #09 and below	Pasir Ris / Changi	5I	0	\$255,000.00	51	98	[Fav]	[Details]
Blk 867A Tampines	Tampines / Simei	5EA	0	\$375,000.00	19	98	[Fav]	[Details]
Blk 655 Woodlands Ring Rd	Woodlands / Sembawang	5I	0	\$220,000.00	8	98	[Fav]	[Details]
Blk 128 Marsiling Rise #06 and above	Woodlands / Sembawang	EA	0	\$260,000.00	33	97	[Fav]	[Details]

Figure 4. Widely Viewed Apartments.

ITEM DETAILS :::::..

Street/Block/Unit: Blk 398 Yishun Ring Rd

District: Yishun

Type: 4A

Room: 0

Area: 0

Price: \$180,000.00

CURRENT PREFERENCES :::::..

Desired Features:

Aircon
Marble Flooring

(Max 10 feats)

Features:

Shopping Centre
Sports Complex
Aircon
Fixtures

(Max 10 feats)

Undesired Features:

Partially Renovated
Fixtures

(Max 10 feats)

Figure 5. Current Version of HumanE Interface.

Phase-Two Learning

The purpose of having the initial policy in phase-one learning is simply to generate experiences of the world, which is tantamount to incorporating prior knowledge into HumanE. After a suitable amount of information has been acquired during the bootstrapping process, the second phase of learning takes over. During this phase, HumanE learns primarily by using reinforcement learning and learning by observation. Usually by this time, HumanE is more “knowledgeable,” which allows for more informed exploration of the search space.

Benefits of Proposed Learning Approach

One of the primary reasons why many reinforcement learning implementations fail to achieve much success for complex goods is that it is assumed that the agent developed knows nothing about the environment to begin with and thus must gain all of its information by exploration and subsequent exploitation of learned knowledge.

When dealing with a real, complex online system such as a large-scale real estate listing and brokerage application, however, this approach is simply not practical. Typically, the search space is too large to explore satisfactorily within the lifetime of the agent (much less within the attention time-span of typical online users). Worse still, making “random” exploratory recommendations can frustrate and disappoint the user, potentially causing the user to abandon the system totally.

For example, Apt Decision (Shearin and Lieberman 2001) suffers from the possibility that the user may get frustrated and disappointed if no suitable recommendations are found during the initial use of the system. This scenario is possible because the Apt Decision agent has no prior knowledge about the real estate domain and cannot initially make good recommendations. Moreover, the agent needs time to learn the user’s preferences from scratch. The time taken by the agent could be significantly long enough to cause the user to give up on the agent.

Another example is the SurfJet Agent,³ an intelligent assistant (much like HumanE) that acts as an autonomous learning agent. SurfJet is not web-based and uses an interest profile to perform searches on the Internet for articles on the user’s behalf. When compared to traditional searching techniques, SurfJet is able to make more accurate and useful searches because users can give it a profile describing many of their interests, including how interesting (or uninteresting) each item is and how they relate to each

other. However, SurfJet does not store any prior knowledge and relies solely on the iterative process of user feedback and profile refinement to make increasingly accurate recommendations. Making good recommendations in the early stages of learning can be difficult, and as with Apt Decision, a considerable amount of time can be spent training SurfJet to understand a user's stated and unstated interests. It is likely that many users are unprepared to commit the considerable time and effort needed to train the agent to make fairly good recommendations.

Accumulated knowledge, in the form of memories and experiences, allows humans to perform daily tasks. In the real world, we often go to a human real estate agent for assistance in selling or acquiring real estate properties. We naturally expect the agent to be an expert in his or her chosen field and hence able to offer suitable advice and recommendations. Certainly, we do not expect the real estate agent to be ignorant about the real estate domain.

As a consequence, in order to take our prior knowledge (which is often implicit) and incorporate it into a reinforcement learning framework, we have examined the idea of supplying HumanE with an initial policy about the real estate domain. In this section, we have briefly described the learning approach, which we are confident can aid agents in making better and faster recommendations faster, and ultimately give users greater satisfaction, confidence, and trust. We support our claims using experimental results, the details of which can be found in the next section.

Experimental Analysis

The following paragraphs outline the methodology used for the experiments conducted with HumanE and our testers. The results are provided in the last paragraph of the section.

There are four dimensions to measure HumanE's ability to increase customer satisfaction: (1) number of profile changes; (2) time taken to create a profile; (3) ease of use; and (4) performance. To ensure the realistic nature of the experiments to be conducted, we painstakingly created our test data set from more than 800 actual real estate ad postings from both offline and online sources.

Experimental Design

Basically, we wanted to test whether our proposed learning approach with the use of initial policy contributes to better performance for web profiling agents. Based on the findings in Kraemer (1991) and Newsted et al. (1997) and the Evaluating Websites site,⁴ we decided to use

survey research in our experimental design. In addition, our experimental design is also strongly influenced by the findings from Pinsonneault and Kraemer (1993)—especially in the area of web page evaluation.

We invited 150 genuine real estate buyers to evaluate HumanE based on the chosen metrics. Most testers were of age between their late 20s to 50 years old. This age span coincided with the age demographic of most people who consider buying apartments and would genuinely be interested in using HumanE as an intelligent assistant during the searching and selection of apartments.

We also took into consideration the occupational profiles of the testers. We wanted to avoid having many information-technology (IT) professionals as our testers because they would probably be more IT-savvy and might be inclined to rate HumanE more favorably due to its sophisticated mechanics. Another factor we considered was the size of the test groups. The size of each test group needed to be sufficiently large to allow for more precision in the analysis of the test results. On the other hand, we did not want the test groups to be overly large because the returns in terms of the accuracy of the test results could be diminishing as the test group size grew.

The evaluation process consisted of three tests. The first test was to test HumanE without the learning approach. In the second test, we tested HumanE with the learning approach but without the initial policy. In the third test, we tested HumanE with the learning approach and with the initial policy. We assigned 50 different testers to each test. We could not repeat the three tests for the same group of testers, as they might be influenced by the earlier test data. To obtain consistent feedback from the three groups of testers, we gave the testers some guidelines to follow when giving answers. For example during the measurement of the "ease of use" and "performance" metrics, we instructed the testers to give their answers based on the scales given in tables 1 and 2.

Before the actual evaluation took place, we gave the testers a quick introduction on how to use HumanE. To ensure the objectiveness of the testers' assessments, we chose not to be directly involved throughout the evaluation process (except the scalability test). Instead, a test coordinator with adequate knowledge of HumanE was asked to conduct the experiment.

Typically, the main method to show that a variable affects the outcome is to hold all other variables constant while changing only that variable. Preferably, the experiment should be conducted in such a way that the users do not

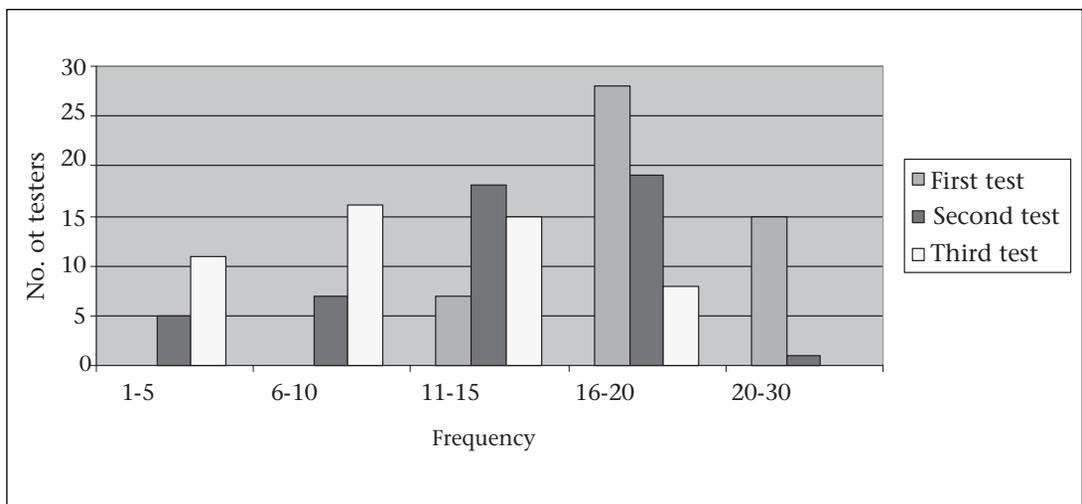


Figure 6. Test Result Summary for "Number of Profile Changes" Metric.

Scale	Requests for Help
Very Bad	>10
Bad	10 - 8
Neutral	7 - 6
Good	5 - 3
Excellent	2 - 0

Table 1. Scale Definitions for "Ease of Use" Metrics.

The values in the Requests for Help column represent the number of times a tester had to request help or ask a question on the use of HumanE.

Scale	Elapsed Time
Very Bad	>60
Bad	60 - 30
Neutral	29 - 21
Good	20 - 11
Excellent	10 - 0

Table 2. Scale Definitions for "Performance" Metric.

The values in the Elapsed Time column represent the number of seconds taken to return matching apartments.

know the state of the variable, so that they are unable to influence the result even if they want to. Thus, to ensure the accuracy of the test results, an identical interface was used to test HumanE (1) without the learning approach; (2) with the learning approach (excluding initial policy); and (3) with the learning approach (including initial policy) while the user was not informed of whether HumanE was learning or not.

The objective of each test was to allow the tester to compile a satisfactory profile. Each tester was asked to select his or her desired apartments using HumanE's web interface. The test was deemed to be completed when the user declared that she or he was satisfied with the list of desired apartments stored in the "favorites" list. The user was then allowed to keep the profile created by printing out a hard copy of his or her "favorites" list. In each of the three tests, the user was not told whether HumanE was used in helping him or her develop the profile.

The user went through the entire profile-creation process without much intervention from the test coordinator. The only assistance provided by the test coordinator was to answer questions asked by a few users about navigation and program operation.

At the end of each test, the values of the five metrics were recorded. For the "ease of use" and "performance" metrics, each tester was asked to rate them from a scale of 1 to 5 (that is, 1: very bad; 2: bad; 3: neutral; 4: good; and 5: excellent) for the three tests. For each test, the values for the "time taken to create a profile" and "number of profile changes" metrics were recorded and computed by HumanE. Since HumanE recorded the login time and lo-

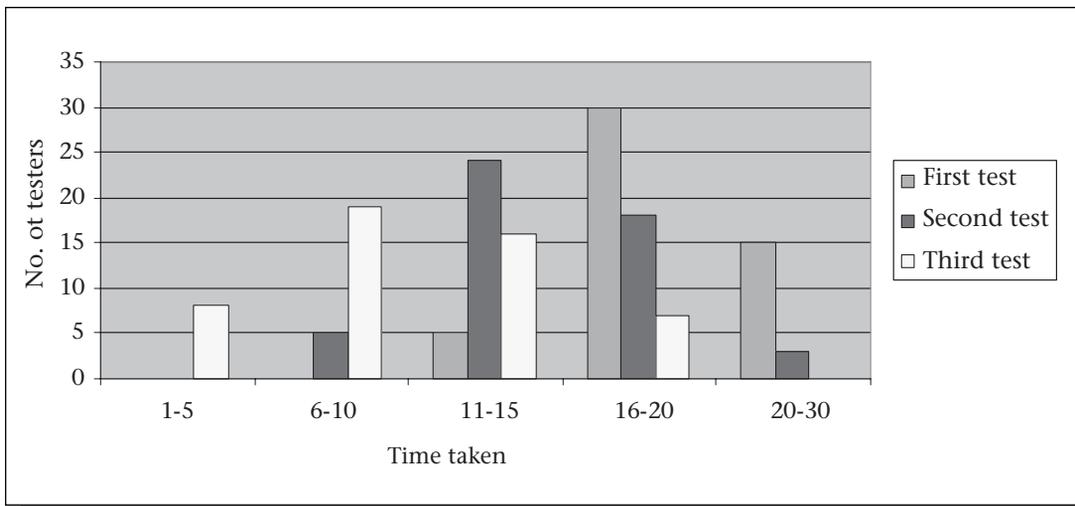


Figure 7. Test Result Summary for "Time Taken to Create a Profile" Metric.

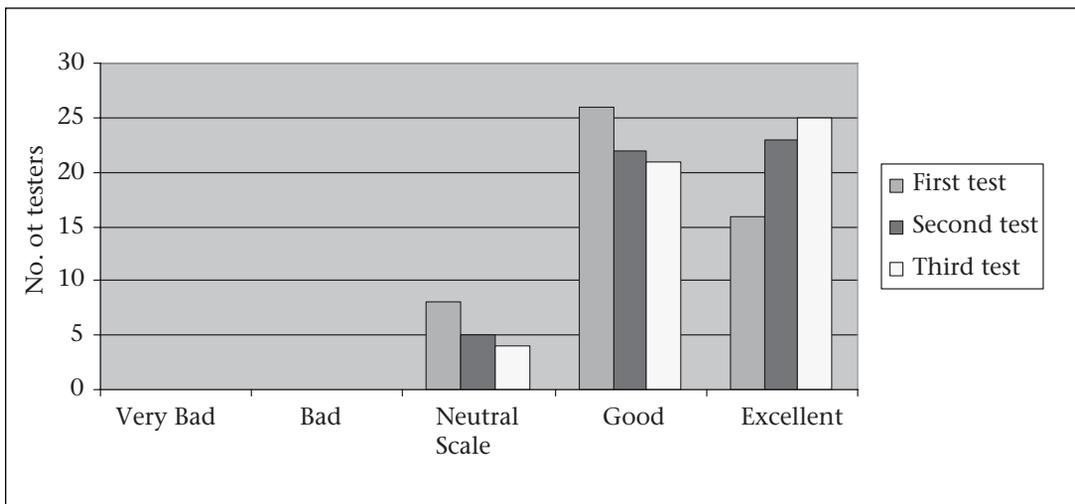


Figure 8. Test Result Summary for "Ease of Use" Metric.

goff time for each tester, HumanE was able to compute the value for the "time taken to create a profile" metric by subtracting the logoff time from the login time. And because every profile modification was recorded, HumanE was able to provide the value of the "number of profile changes" metric for each tester.

The results from the experiments conducted are tabulated in figures 6 through 9.

Discussion

The test results for the first test for the number of profile changes metric showed that most testers took 11 to 25 profile changes before converging on a satisfactory profile. The test results for the second test for the number of profile changes metric showed some improve-

ment as most testers took 11 to 20 searches. The test results for the third test for the number of searches metric showed further improvement as most testers took 1 to 15 searches. Thus, it is evident that testers tend to make a lesser number of profile changes with HumanE's assistance and an even lesser number when HumanE becomes more intelligent with the supply of the initial policy.

Similarly, the time taken to create a satisfactory profile decreased as we introduced a more intelligent HumanE with each test. The test results for the first test for the time taken to create a profile metric showed that most testers took 16 to 30 minutes while most testers in the second test took less time—from 11 to 20 minutes. However, the testers from the third

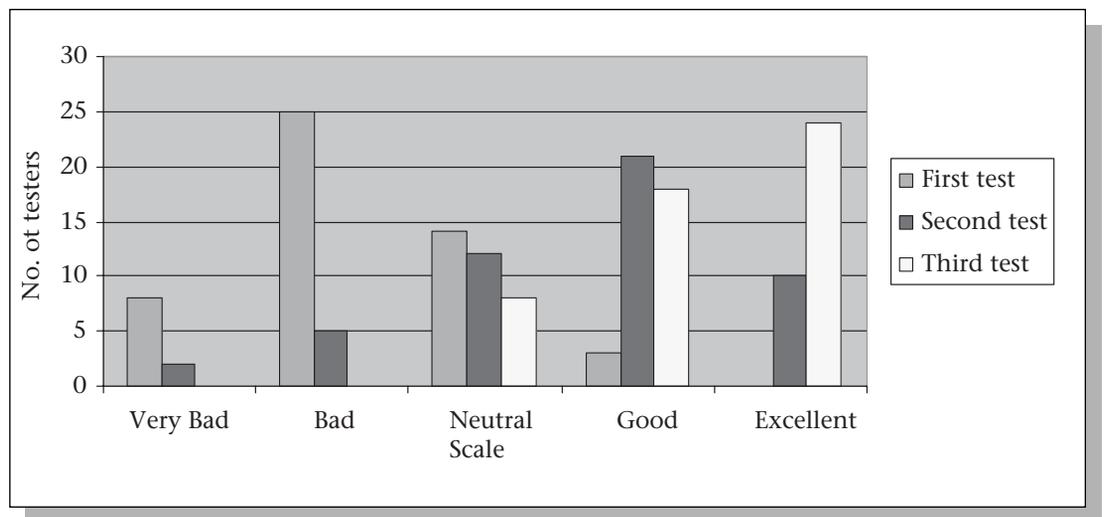


Figure 9. Test Result Summary for "Performance" Metric.

test took the least time as most of them spent 6 to 15 minutes. Thus, it is clear that HumanE can reduce the time taken by users when creating and refining their profiles.

The test results for the three tests for the ease of use metric are quite similar, indicating that almost all of the testers are happy with using HumanE regardless of whether the learning system with or without the initial policy was present or not. Hence, it is safe to say that using HumanE can result in increased customer satisfaction during the apartment-selection process.

The test results for the performance metric for the first test apparently showed that a majority of the testers were not satisfied with the average quality of the recommended apartments shown to them for selection and the average response time taken to display an apartment listing. Quite a number of them perceived HumanE as a search engine for apartment listings, and they were not satisfied with the perceived browsing metaphor that is offered by typical search engines. Although many testers were fairly happy that they were given complete control over the entire profile-creation process, they also voiced their displeasure over having to make many tedious profile changes before converging on a good profile.

On the other hand, the test results for the second test and the third test showed that the majority of testers preferred allowing HumanE to assist them during the apartment-selection process. Obviously, the use of HumanE can increase customer satisfaction.

In summary, the experimental results showed that the use of HumanE for complex

multidimensional domains such as real estate can result in higher customer satisfaction as it can learn faster through a supplied initial policy and is able to elicit trust from users through its user-friendly interface, quality recommendations, and excellent performance.

Future Work

The development of HumanE will continue to evolve particularly in a different domain, that is, vacation plans. In future versions of HumanE, we would like to incorporate some of the following features to improve its usefulness. First, we would like to refine the initial policy-refining algorithm based on the results obtained using more sophisticated data-mining tools. Second, we would like to incorporate the ability to ask the user questions in natural language, allow the user to enter the response in natural language, and, finally, understand the response obtained for profile refinement. Third, we would like to add the ability to seek advice from users with similar profiles by e-mail and to interpret the reply so as to refine the profile. Finally, we would like to incorporate the ability to submit user profiles to multiple domain-specific websites and to show the user the results online. The agent will also need to parse and understand the listing obtained for profile refinement.

Conclusion

HumanE addresses the problem of poor learning when implementing online large-scale au-

tonomous agent-based recommender systems for several complex domains through the use of a supplied initial policy that allows it to make more “knowledgeable” exploratory recommendations. We feel that existing implementations of the interactive learning method for online systems are simply impractical, as the state-action space is simply too large for the agent to explore within its lifetime. This problem is further exacerbated by the short attention time-span of typical online users. It seems easier and more intuitive for the application developer to specify what the agent should be doing and to let it learn the fine details of how to do it. The key strength of our approach is that, by incorporating an initial policy or prior knowledge, HumanE is able to provide better recommendations within a shorter time span because the initial policy has generated some experiences or knowledge about the real estate domain that HumanE can use throughout the interactive learning process. No longer does the user need to face an agent that does not know anything about the task to be completed. We believe that this approach is far more practical and effective than current implementations (Brooks 1986, Franklin and Graesser 1996, Turban and Aronson 1998, Schneiderman 1983).

We also postulate, contrary to the experimental results obtained from past research (Smart and Kaelbling 2002), that a good initial policy is critical to the success of HumanE from a reward perspective, as the user usually takes less time to build a good profile. Good initial policies head directly for the goal state, and they typically do not expose much of the state-space, since their trajectories through it are much more directed. This behavior is actually quite desirable, as most online users generally have little patience and want to see a good profile built quickly.

Finally, transferring the work done here to another different domain such as vacation plans, insurance, mutual funds, and mortgages would not require a “rocket scientist.” The main requirement would be to find a domain expert who would be able to identify the key features of the complex ob-

jects in the domain. Creating an initial policy would require the identification of good and bad features and the classification of features into loosely connected groups.

Notes

1. An example is Singapore Residential Properties, www.singapore-residential-properties.com/.
2. An example is Singapore-Real-Estate, www.singapore-real-estate.com/.
3. www.leptonicsystems.com/surfjet/.
4. trochim.human.cornell.edu/WebEval/webeval.htm.

References

- Brooks, R. A. 1986. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation* 2(1): 14–23.
- Burke, D.; Hammond, K. J.; and Young, B. C. 1997. The FindMe Approach to Assisted Browsing. *IEEE Expert*. 12(4): 32–40.
- Franklin, S., and Graesser, A. 1996. Is It an Agent, or Just a Program? A Taxonomy for Autonomous Agents. In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Lecture Notes in Artificial Intelligence, Volume 1193. Berlin: Springer-Verlag.
- Gao X.; and Sterling L. 1998. Classified Advertisement Search Agent (CASA): A knowledge-based Information Agent for Searching Semi-Structured Text. Technical Report 1/1998, Department of Computer Science and Software Engineering, the University of Melbourne, Melbourne, Australia.
- Kraemer, K., ed. 1991. *The Information Systems Research Challenge: Survey Research Methods*, 299–315. Boston: Harvard Business School.
- Newsted, P. R.; Chin, W.; Ngwenyama, O.; and Lee, A. 1997. Resolved: Surveys Have Outlived Their Usefulness in IS Research. Panel presented at the 1996 International Conference on Information Systems, Cleveland, OH, December 17.
- Pinsonneault, A.; and Kraemer, K. 1993. Survey Research Methodology in Management Information Systems. *Journal of Management Information Systems* 10(2): 75–105.
- Schneiderman, B. 1983. Direct Manipulation: A Step Beyond Programming Languages, *IEEE Computer* 16(8): 57–69.
- Shearin, S.; and Lieberman, H. 2001. Intelligent Profiling by Example. In *Proceedings of the Sixth International Conference on Intelligent User Interfaces*, 145–151. New York: Association for Computing Machinery.

Smart, W. D.; and Kaelbling, L. P. 2002. Effective Reinforcement Learning for Mobile Robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Volume 4, 3403–3410. Piscataway, NJ: Institute of Electrical and Electronic Engineers.

Turban, E., and Aronson, J. E. 1998. *Decision Support Systems and Intelligent Systems*. Upper Saddle River, NJ: Prentice Hall.



Oh Chin Hock Danny obtained a B.Sc. in computer and information systems in 1998 and a M.Sc. in computer science by research from the National University of Singapore in 2004. His primary research interest

is in developing effective agent-based learning techniques for large, complex, and adaptive autonomous agent systems for complex goods. He will be pursuing his doctoral degree in computer science soon, and he plans to develop a set of more robust learning techniques for online profiling agents for complex goods such as vacation plans, insurance, mutual funds, and mortgages.



Chew Lim Tan is an associate professor in the Department of Computer Science, School of Computing, National University of Singapore. He received the B.Sc. (honors) degree in physics in 1971 from the

University of Singapore, the M.Sc. degree in radiation studies in 1973 from the University of Surrey, UK, and the Ph.D. degree in computer science in 1986 from the University of Virginia. His research interests include document image and text processing, agent technology, neural networks, and genetic programming. He has published more than 200 research publications in these areas. He is an associate editor of *Pattern Recognition* and has served on the program committees of many international conferences and workshops, including the International Conference on Document Analysis and Recognition (ICDAR) 2005, the International Workshop on Graphics Recognition (GREC) 2005, and the International Conference on Pattern Recognition (ICPR) 2006.



Artificial Intelligence and Interactive Digital Entertainment

Save the Date for AIIDE-06!

Following the success of the first event this past June, the Second Conference on Artificial Intelligence and Interactive Digital Entertainment will be held in Marina del Rey, California, June 13–16, 2006. AIIDE is the definitive point of interaction between interactive entertainment software developers interested in AI and academic AI researchers.

Sponsored by the American Association for Artificial Intelligence, the AIIDE conference is targeted at both the research and commercial communities, promoting AI research and practice in the context of interactive digital entertainment systems with an emphasis on commercial computer and video games.

AIIDE-06 will include invited speaker sessions, paper presentation sessions, demonstration sessions, exhibits and a full day of tutorials. We invite researchers and developers to share their insights and cutting-edge results, and encourage the presentation of results from core AI research areas applicable to interactive digital entertainment and AI approaches developed and fielded in commercial systems. Please see www.aaai.org/Conferences/AIIDE for more details and the complete call for papers.

Conference Chair

John Laird, *University of Michigan*

Program Chair

Jonathan Schaeffer, *University of Alberta, Canada*

Demos and Exhibits / Local Arrangements Chair

Michael van Lent, *University of Southern California/ICT*

Publicity Chair

Ian Lane Davis, *Mad Doc Software*

Tutorials Chair

R. Michael Young, *North Carolina State University*