Intelligent Profiling by Example

Sybil Shearin, Henry Lieberman
MIT Media Lab
20 Ames St., E15-305D
Cambridge, MA 02139 USA
+1 617 253 9601
{sibyl, lieber}@media.mit.edu

ABSTRACT

The Apt Decision agent learns user preferences in the domain of rental real estate by observing the user's critique of apartment features. Users provide a small number of criteria in the initial interaction, receive a display of sample apartments, and then react to any feature of any apartment independently, in any order. Users learn which features are important to them as they discover the details of specific apartments. The agent uses interactive learning techniques to build a profile of user preferences, which can then be saved and used in further retrievals. Because the user's actions in specifying preferences are also used by the agent to create a profile, the result is an agent that builds a profile without redundant or unnecessary effort on the user's part.

Keywords

Profiling, electronic profiles, personalization, infomediary, user preferences, real estate, interactive learning

INTRODUCTION

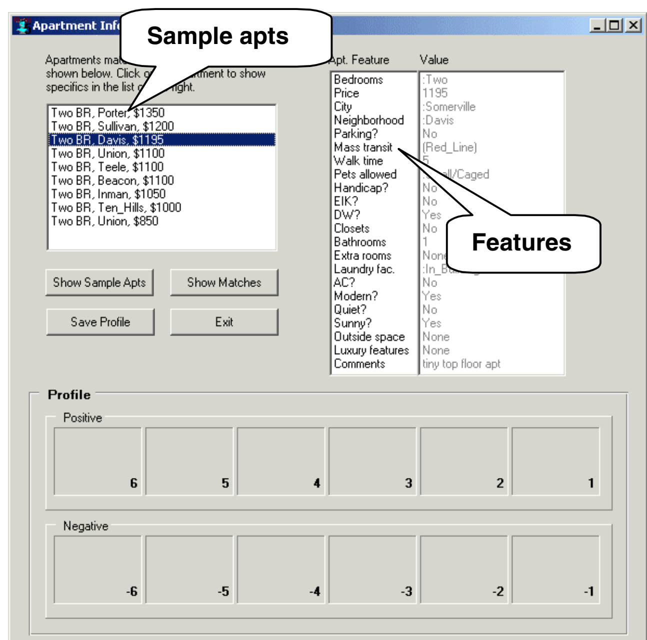
Electronic profiling is a popular topic recently, both for Internet startups and research efforts in the area of electronic commerce. In the rush to create profiles and make use of them, companies pay little attention to whether profiles are convenient for the user. Most profiles require considerable user effort, usually in filling out online forms or questionnaires. The technique of learning user preferences in order to build a profile has been used sporadically in autonomous agent development [10] to illustrate the learning behavior of an agent. However, it deserves individual attention because it is a technique that is quite useful for intelligently developing an electronic profile. Our alternative to complicated questionnaires is an agent like Apt Decision, which exposes the knowledge inherent in a domain (rental real estate), then learns the user's preferences in that domain and builds a profile without redundant or unnecessary effort on the user's part.

IUI'01, January 14-17, 2001, Santa Fe, New Mexico, USA.

HOW THE AGENT WORKS

Rather than adopt a purely browsing metaphor through the geographic space of homes, as in Shneiderman [12], or a search-like metaphor, such as the Boston Globe site [1], Apt Decision assumes that there will be an iterative process of browsing and user feedback. This work is most similar to systems such as RENTME [4]. Apt Decision's key feature is the ability for the user to react, not just to a particular apartment offering, but independently to *every feature* of the offering. Apt Decision exposes the profile creation process, and allows the user to interact directly with the various features of specific apartments. While we cannot yet give the agent the full inference power a human real estate agent might have, we can incorporate the principle of inferring preferences from the critique of concrete examples.

Using an initial profile provided by the user (consisting of number of bedrooms, city, and price), the agent displays a list of sample matching apartments in the Apartment Information window, shown below.



Up to twelve apartments matching the user's information are displayed in a list on the left side of the Apartment Information window. To ensure that the initial query is not too restrictive, Apt Decision uses commonsense measures in returning apartments.

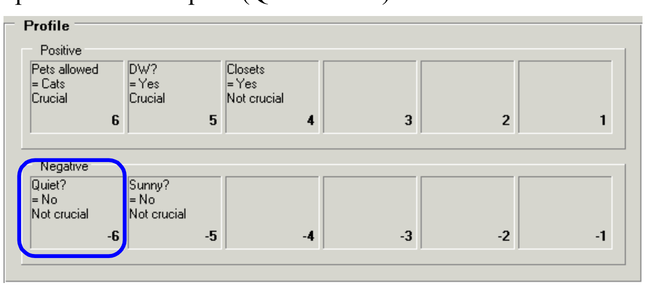
The price entered by the user is considered to be an upper bound; apartments having that price or less are returned. Apartments from all neighborhoods in the location specified are returned; if there are no apartments matching the user's specifications in that location, Apt Decision uses its knowledge of Boston to return apartments in nearby locations.

The user can browse through the apartments returned by highlighting each apartment in the left-hand list box. The features of the selected apartment are shown on the right side of the window.

Since each apartment listing contains far more information than was supplied in the initial query, the user has the opportunity to discover new features of interest. Perhaps one might not initially think of specifying secondary features such as laundry facilities or an eat-in kitchen, but once these attributes appear in specific examples, the user may realize their importance.

Each feature of an apartment in Apt Decision has a base weight, which is established as part of the domain modeling for the real estate domain. The user examines the features of each apartment, then reacts to a feature by dragging it onto a slot in the profile. Weights on individual features change when the user chooses to place them in (or remove them from) the profile. The new weight depends on which slot the feature occupies. The profile contains twelve slots: six positive and six negative. The slots are also weighted, with more important (higher weight) slots on the left and less important slots on the right.

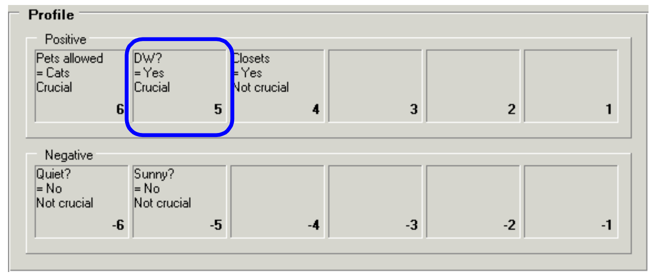
The resulting profile entry combines the user's opinion about a particular feature of an apartment with their reaction to that feature's value for the sample apartment currently being displayed. For example, the entry in the leftmost Negative profile slot below indicates that the user feels very strongly about the fact that this particular apartment is not quiet (Quiet? = No).



Crucial Features

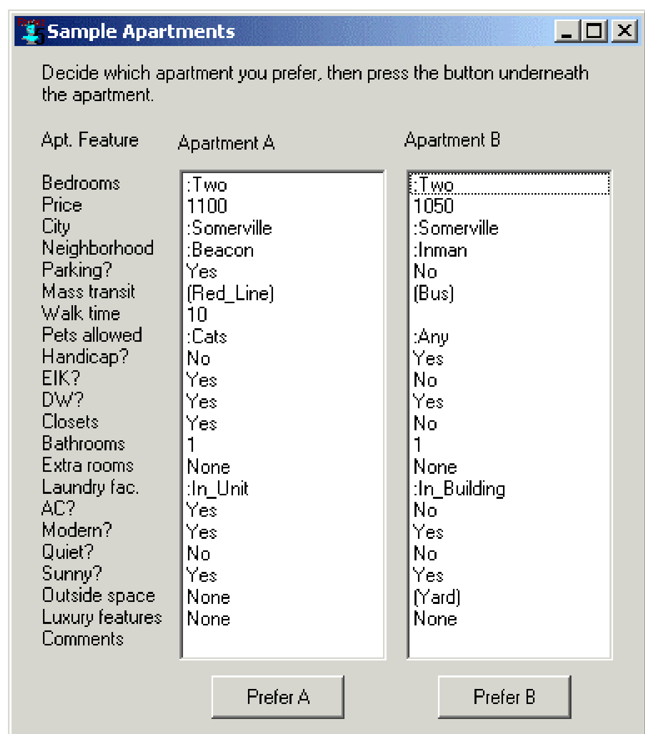
The user's reaction to a feature (measured by its position in the profile) differs from the knowledge about the real estate domain that is built into the agent. That knowledge specifies that some features are automatically crucial to the final decision: Parking, Pets allowed, Handicapped access, Bedrooms, Price, and City. (See the Domain Analysis section for more details.) The user can make other features crucial by dragging the same feature to the same profile slot again.

In the figure below, the user has chosen to make the 'DW?' feature (which indicates the presence of a dishwasher in the apartment) crucial by dragging it to the second Positive slot more than once.

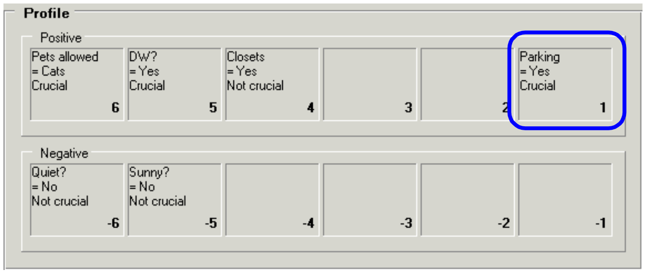


Profile Expansion

If the user does not want to choose further features manually, but still wants to develop the profile, he can use *profile expansion* to add items to the profile automatically by clicking the Show Sample Apts button. This button displays a dialog for the user to choose between two sample apartments.



When the user chooses between the two apartments by clicking the Prefer A or Prefer B button, the agent derives new profile information by examining the current profile and the apartment chosen by the user. The agent can fill up to three profile slots in this manner. New profile items are found by comparing the two apartments shown, and finding features that are unique to the chosen apartment but not currently present in the profile. New items are entered into the right side of the profile, as shown in the figure below; the user can drag the items to different slots in the profile if needed.



First, the profile expansion technique looks for crucial features to add to the profile, then tries non-crucial features if no crucial ones are available. In all instances, the features added to the profile are ones that are unique to the apartment chosen and which do not already appear in the profile.

Backtracking

An agent history window provides history and commentary on the user's actions as well as what the agent is learning. This process gives Apt Decision implicit information about user preferences, such as:

- Which apartments did the user choose to look at? In what order?
- Which features did the user think were important to comment on? In what order? How important were those features?
- How do the chosen features affect searching the space of apartments?

Each of these factors can be significant. Real estate agents know that showing a user the twentieth apartment is different than showing the first. Users may choose to explore the "best" choices before they explore less desirable choices. They may choose to comment on the attributes most important to them before they specify less important attributes. None of these heuristics is ironclad, but together they can contribute to a better understanding of user preferences.

The current version of Apt Decision uses these preferences to avoid overconstraining the choice of apartments. If a user creates a profile that matches fewer than three apartments, the agent offers the user four choices: remove the last item chosen, overwrite another profile slot with the last item chosen, or backtrack to an earlier version of the profile before adding the last item chosen. The fourth choice leaves the user profile unchanged, but advises the user that any further additions to the profile will result in very few matching apartments.

Matching Apartments

When the user is finished examining the sample apartments, he has a profile of apartment preferences that can be saved to a file. After the profile is complete, user searches no longer need to begin "from scratch", as is so often the case with web or database searches. The information contained in the profile provides a context for future searches. The profile can be used to retrieve matching apartments from the set provided with the agent, or taken to a human real estate agent as a starting point for a real-world apartment search.

Within Apt Decision, the user's actions in creating a profile alter the system's model of an "ideal apartment" for that user. As the user modifies the profile, the system updates the weights on its representation of the ideal apartment and re-orders the potential matches in the data set to reflect the new weighting.

DOMAIN ANALYSIS

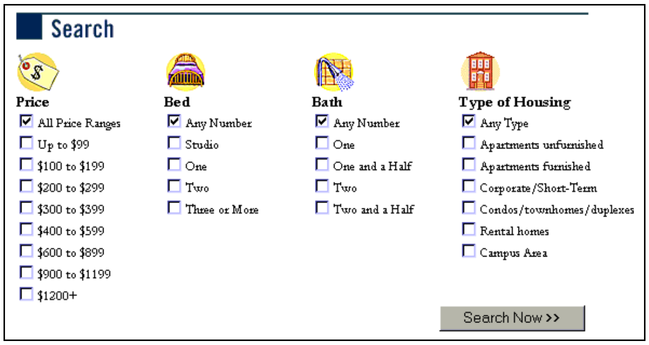
Before beginning development on the agent itself, we began by examining our chosen domain (rental real estate) carefully. The agent needed to have built-in knowledge about the domain. We quickly decided to focus on the Boston real estate market, since there are significant local and regional variations in the standard apartment features and rental rates. Next we analyzed apartment rental advertisements to determine the standard apartment features for the Boston area. Even though the Multiple Listing Service (MLS) database is a common real estate tool that we could have used to obtain features, we determined from speaking to local real estate agents that MLS data largely concerned properties for sale, not for rent.

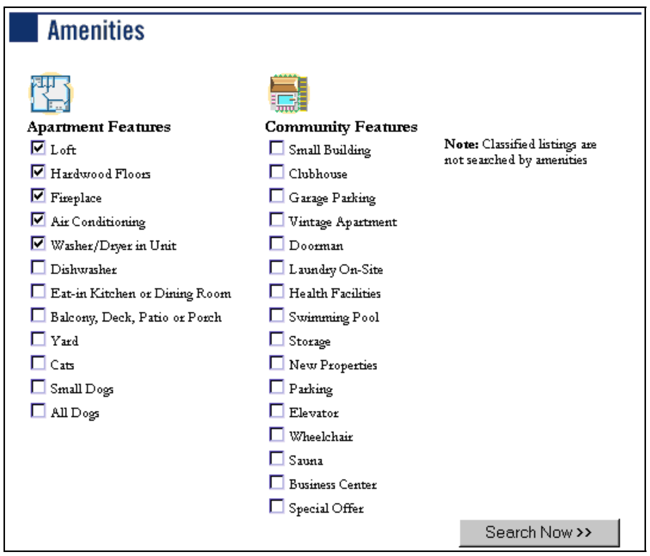
After the ad analysis, we had a list of twenty-one features commonly advertised in Boston real estate listings. Next, we considered how people choose apartments. After examining the features, we concluded that some of them (e.g., apartment size, availability of parking, whether pets were allowed) were pivotal to the final choice of apartment. That is, most people would reject an apartment if the value for a crucial feature were not to their liking. Other features (e.g., the presence of a dishwasher or an air conditioner) were less pivotal – some people would like them, some would be indifferent, some would dislike them. All this domain knowledge went into Apt Decision. In addition, we examined two destinations of apartment seekers: real estate Web sites and human real estate agents, to determine what knowledge we could glean from those interactions.

Real Estate Web Sites

Many real estate Web sites expect users to enter not only a price range and apartment size, but also many other specific details about their ideal apartment. One problem with these sites is that the apartment seeker must enter preferences separately at each site, each time he visits the site. There is also no option to save multiple sets of preferences for a single site.

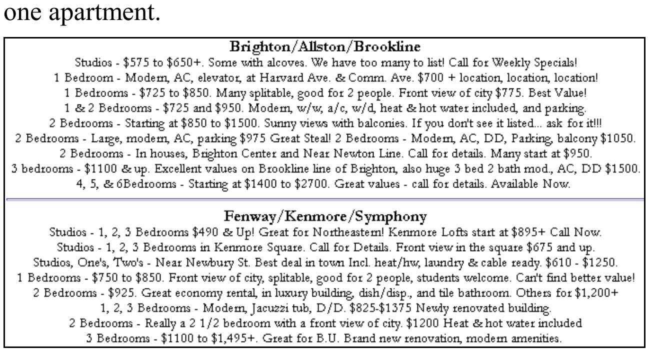
One type of real estate web site leads the user through several choice pages. The example below is from the Boston Globe site [1]. After choosing an area of Massachusetts (Boston) from the first page and a handful of Boston suburbs to narrow the search from the second page, the preference options shown below were displayed on the third page.





As you can imagine, selecting a specific set of checkboxes each time you visit this site would quickly become tedious.

The second type of site typically has only one set of choices or avoids choice pages altogether. Instead of endless pages of preferences, these sites display endless pages of listings. This is a sample from a local Boston real estate company's web site [9]. The sample shown here contains 17 distinct apartment "listings," each of which might refer to more than one apartment.


Brighton/Allston/Brookline

Studios - $575 to $650+. Some with alcoves. We have too many to list! Call for Weekly Specials!
1 Bedroom - Modern, AC, elevator, at Harvard Ave. & Comm. Ave. $700 + location, location, location!
1 Bedrooms - $725 to $850. Many splitable, good for 2 people. Front view of city $775. Best Value!
1 & 2 Bedrooms - $725 and $950. Modern, w/w, a/c, w/d, heat & hot water included, and parking.
2 Bedrooms - Starting at $850 to $1500. Sunny views with balconies. If you don't see it listed... ask for it!!!
2 Bedrooms - Large, modern, AC, parking $975 Great Steal! 2 Bedrooms - Modern, AC, DD, Parking, balcony $1050.
2 Bedrooms - In houses, Brighton Center and Near Newton Line. Call for details. Many start at $950.
3 bedrooms - $1100 & up. Excellent values on Brookline line of Brighton, also huge 3 bed 2 bath mod., AC, DD $1500.
4, 5, & 6Bedrooms - Starting at $1400 to $2700. Great values - call for details. Available Now.

Fenway/Kenmore/Symphony

Studios - 1, 2, 3 Bedrooms $490 & Up! Great for Northeastern! Kenmore Lofts start at $895+ Call Now.
Studios - 1, 2, 3 Bedrooms in Kenmore Square. Call for Details. Front view in the square $675 and up.
Studios, One's, Two's - Near Newbury St. Best deal in town Incl. heat/hw, laundry & cable ready. $610 - $1250.
1 Bedrooms - $750 to $850. Front view of city, splitable, good for 2 people, students welcome. Can't find better value!
2 Bedrooms - $925. Great economy rental, in luxury building, dish/disp., and tile bathroom. Others for $1,200+
1, 2, 3 Bedrooms - Modern, Jacuzzi tub, D/D. $825-$1375 Newly renovated building.
2 Bedrooms - Really a 2 1/2 bedroom with a front view of city. $1200 Heat & hot water included
3 Bedrooms - $1100 to $1,495+. Great for B.U. Brand new renovation, modern amenities.

Especially with a complex decision such as renting an apartment, people find it difficult to specify exactly what it is that they want. What they think they want may change in the course of their exploration of what is available; they may have firm constraints or weak preferences; they may have unstated goals, such as finding something quickly, or determining how reliable the agent is. Apt Decision represents the salient features of the domain and allows the user to quickly and easily ascertain preferences via a profile. It removes the cognitive burden of questions such as: What can I expect of apartments in Boston? What features are common and which are unusual? What is the range of rents I can expect to pay for a certain neighborhood? As a result, it allows the user to concentrate on questions not easily solved by technology, such as: Can I trust this broker? How does this landlord treat tenants? Who can I talk into helping me move?

Human Real Estate Agents

As a guide to how the online real estate experience might be improved, consider how people deal with the ambiguity and imprecision of real world decisions. Think about how a customer interacts with a real estate agent. The agent does not make the customer fill out a questionnaire containing all the possible attributes of houses, then search a database to present the customer with all the choices that fit the questionnaire!

Instead, the agent asks, "How may I help you?" and the customer is free to respond however he or she wishes. Typically, the customer will supply a few criteria; e.g. "I would like to rent a two-bedroom apartment in Somerville for about $1500." These criteria provide a rough "first estimate" for the agent. All of the criteria might be lies; the customer might very well rent something that fits none of the initial criteria. The real estate agent uses the initial guidelines to retrieve a few examples: "I've got a two-bedroom in Davis Square for $1500, but it has no yard; and a nice one-bedroom for $1300 in Porter Square that has a finished basement you could use as a second bedroom."

The agent then waits to see the customer's reaction. The key point is that the customer may react in a variety of ways not limited by answers to explicitly posed questions. The agent's description will typically contain 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. "Let's see the one in Davis Square" lets the agent infer assent with the initial criteria, but "What about my dog?" establishes a previously unstated requirement that the landlord must allow pets. Near-miss examples, such as "I've got a three-bedroom for $1500, but it is in Medford", "Would you pay $1700 if the apartment was in Cambridge, and right near a subway stop?" 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.

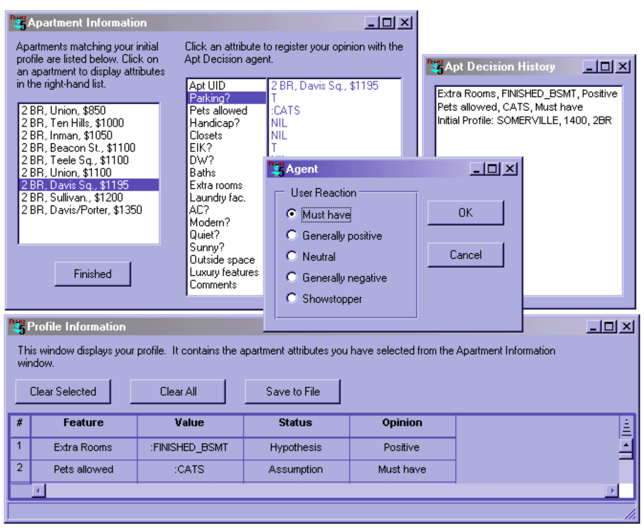
Transferring Domain Knowledge

Much of the work done for Apt Decision 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.

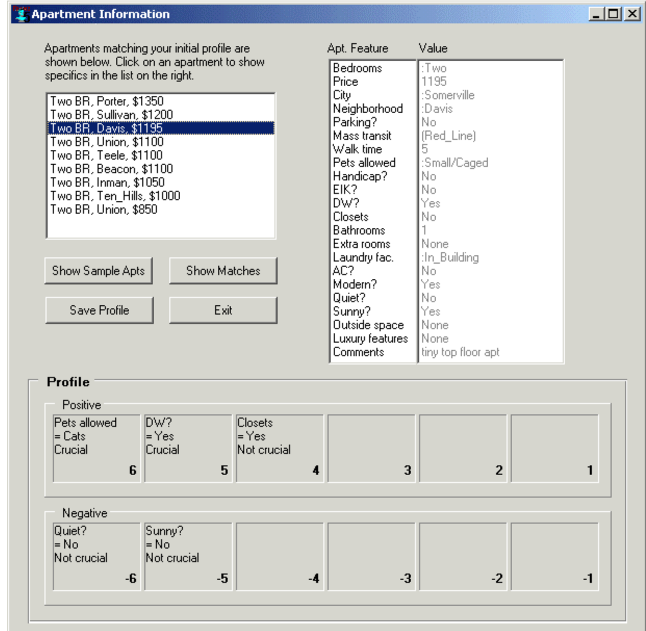
Transferring the agent into another domain would require the services of a subject matter expert who could identify salient features of the complex objects in the domain, alter the program to work with those features and determine which features were crucial to the final decision. After testing on a suitable list of objects, the "new" agent could be released.

ISSUES DURING DEVELOPMENT

Development on Apt Decision has been in progress since late 1999, and is nearly complete. Since we do not expect a high level of computer skill from our typical user, the design of the agent interface is of particular importance. The Apt Decision interface has gone through a number of iterations to make it more intuitive and responsive to the user's actions. Adding the drag-and-drop feature was crucial to this effort. The first version of Apt Decision (shown below) used three separate windows: one for the sample apartments and their features, one for the profile, and one for the agent history.



Each time the user chose a feature, a separate dialog would appear to register that feature in the profile. While it made the agent aspect of Apt Decision more obvious, that interaction did not work well when it occurred multiple times in rapid succession (i.e., as the users developed their profiles). Users tend to be familiar with drag-and-drop from popular business productivity software, so this familiar interaction provided continuity and reassurance in the unfamiliar context of a software agent. In addition, the decision to place the sample apartments (and their features) in the same window with the profile aided the transition to drag-and-drop.



We were also interested in making the agent learn from each interaction with the user. This interactive technique differs from many traditional machine learning approaches, which require test data to train on and acquire their knowledge via batch runs against large data sets. We made the assumption that every user action is meaningful, and indeed, designed the user interface with that assumption in mind. So while Apt Decision is running, it notes every user action and stores that knowledge. Currently, these observations are restricted to each individual user, but future versions of Apt Decision could well combine data from many users to (for example) derive a set of typical user profiles.

POTENTIAL USES

Apt Decision was originally conceived as a single-user agent. That is, an individual user would install the agent, then run it to find out about rental real estate in the local area, and build a profile to take to a human real estate agent. Several other scenarios are also possible. Roommate services (often used in the Boston area due to a large student population and high rents) could ask each customer for a profile and do simple matching to determine whether apartment expectations match. If a real estate office installed Apt Decision and entered their rental real estate listings into it, they could provide it as a decision-making service for their clients. Also in this scenario, if clients saved their profiles, the real estate company would be able to build up aggregate data on their customers, which could be used to advise potential landlords on desirable improvements to their rental property. That data would also provide useful information for real estate developers, for example, a trend toward larger households in Somerville (a Boston suburb).

RESEARCH AREAS

Profiling

Development of electronic profiles is currently dominated by "infomediaries" [8] who see their web-based services as the ultimate solution to collecting, managing, and distributing an individual's data. But how do the infomediaries obtain that data? The current solution is to have the user enter it by hand for their chosen infomediary. But, as [11] points out, "there is a clear need for some means of storing, representing, segmenting, organizing, and distributing [all of] an individual's personal data in a single electronic profile." As more and more personal data is added to the user's profile, the greater the chance that the data exist somewhere else electronically. But you, the user, are your profile. You know your interests, likes, and dislikes better than anyone else does. An agent that can learn those interests from the user's interactions with it is certainly useful in building profiles in a more intelligent way.

User Interface Design

Visualization. The visualization of the profile is important to Apt Decision's UI design. The user needs to be able to quickly scan the profile and see at a glance whether they have already put a feature into it. The original version of Apt Decision put the profile into a scrolling spreadsheet-type control, which was fine for small profiles but unwieldy for larger ones. In the redesign that led to the current interface, the field that indicated how the user felt about a given feature in the profile was entirely removed and incorporated into the interface itself. Thus arrived the current profile, with its positive and negative slots.

The profile displays information very simply: the apartment feature on the first line, the value for that feature (taken from the apartment displaying when the user dragged the feature into the profile) on the second line, and whether or not the feature is considered crucial on the third line. The user's opinion, not explicitly stated, is inherent in a feature's placement in the profile. If a feature is in the top row, the user feels positively about that feature/value combination; bottom row placement means that the user feels negatively about the associated feature and value. Drag-and-drop is fully enabled throughout the profile, so the user can change the placement of a feature at any time. The profile holds the agent's current knowledge about the user's preferences.

When the user expands the profile by choosing between sample apartments, the agent fills in one or two profile slots automatically, by analyzing the apartment chosen and the contents of the current profile.

User Constraint vs. User Discovery. Apt Decision illustrates the tradeoff that occurs between constraining user interaction and discovering the preferences of the user. Simply put, the more options you give the user at any moment in time, the more you can learn from which of those options the user chooses. Conventional interfaces that rely on rigid questionnaires cut off this possibility by constraining the user. They often do this to reduce the search space as fast as possible. First asking the user what city they want to live in cuts down possibilities rapidly, but eliminates the possibility of finding out whether they consider price or location more important. If they can specify either price or location to start, one could reasonably assume they would compromise the other attribute to get their desired goal on the primary attribute.

Our goal with Apt Decision was to relax constraints on order and feedback in the hopes of learning preferences more quickly. We believe that this will restore some of the flexibility that people find attractive in dealing with human real estate agents.

Interactive Learning

The Apt Decision agent takes an interactive learning approach, that is, it learns from each interaction with the user. Interactive learning makes the assumption that all the user's actions have some meaning, and the agent is designed so that this is true. Each time the user drags an apartment feature to the profile, the reinforcement learning algorithm changes the weightings on the features in the user's "ideal" apartment. This approach differs from traditional machine learning in several ways. First of all, it works with very small, but precise, amounts of data. Also, it is an interactive technique, in that the user is in constant contact with the agent; there is no batch processing of datasets.

Each feature of an apartment in Apt Decision has a base weight. Weights on individual features change when the user chooses to place them in or remove them from a profile slot. The new weight depends on which slot the feature occupies, whether the feature is crucial, and whether the slot was filled using profile expansion. Crucial features are weighted more heavily; features automatically added to the profile are weighted less heavily.

In addition, Apt Decision records the history of a user's interaction with the agent. If at some point in the profile-building process, there are suddenly no apartments that match the profile, the agent can offer the recourse of backtracking to a prior point in the interaction.

FUTURE WORK

The current version Apt Decision is almost finished, but there still remain some features that would improve future versions.

- the ability to partially order the apartment features using version spaces (for those that are not independent)
- the ability to compile profile information from multiple users and generate statistics to form aggregate profiles
- the ability to submit the user's profile to one or more real estate web sites and send listings that match the profile to the user via email

After the current version has been finished, we would like to perform the following experiment in-house to evaluate Apt Decision's performance and potential benefits. We would give subjects a specific task [e.g. find a two-bedroom apt in Somerville for $1500], and one of: the Apt Decision agent, the Boston Sunday *Globe* real estate section or a typical real estate Web site. Then we would compare objective measures such as how many apartments the user looked at and how long it took him/her to find an apartment. After subjects have found an apartment they like, we would present them with a questionnaire to find out how satisfied they were with the result and the process and how much they felt they had learned about the rental real estate.

We would also like to try real-world user testing, by making Apt Decision available to real estate agents so that their clients could use it and give feedback on its usefulness.

RELATED WORK

Work related to Apt Decision includes both shopping [6] and profiling [5] agents, as well as site search engines as discussed earlier, and query-by-example systems.

Gao and Sterling developed a Classified Advertisement Search Agent (CASA), which helped users search classified ads for real estate [7]. Their system was primarily used as a search engine, but there were several important points relevant to Apt Decision. First, it incorporated knowledge about the real estate domain. Second, the authors realized that all user preferences were not equally important. And third, the authors created a mechanism to allow users to refine their queries and resubmit them.

Shneiderman's HomeFinder system (discussed in [12]) used an interesting geographic visualization technique for displaying homes that had certain features or attributes such as garages or central air conditioning. However, the techniques in that paper focused on visualization and dynamic queries rather than the iterative profile-building approach we are using.

Williams' RABBIT system [13] was a query-building tool used to retrieve items from a database. Users could critique fields in example records via options such as "prohibit" or "specialize." The system would take the user's feedback, reformulate the query, and show another example record for the user to react to. RABBIT is interesting as an early example of relevance feedback, but Apt Decision focuses more on detecting user preferences than on strict query building.

Some case-based recommender systems, such as RENTME [2, 4] are quite similar to Apt Decision. The task is the same and the expectation of user goals is similar. Both systems begin their operation alike, in that they ask users to specify a location, a price, and a size for their desired apartment. RENTME, however, primarily uses critiquing examples as its fundamental interaction. Interaction is constrained to pre-defined "tweaks" such as "cheaper," "nicer," or "safer." Apt Decision derives much of its information from users' implicit critique of individual apartment features when the features are added to the profile.

REFERENCES

1. Boston Globe Online Apartment Search Page. Available at: http://www.apartments.com/search/oasis.dll?rgn1=114&page=Area&state=MA&partner=boston&prvpg=2. Accessed 19 May 2000.
2. Burke, D., Hammond, K. J., and Young, B. C. (1997). "The FindMe Approach to Assisted Browsing," *IEEE Expert*, pp. 32-40, July-August 1997.
3. Burke, R. (1999). The Wasabi Personal Shopper: A Case-Based Recommender System. In *Proceedings of the 11th National Conference on Innovative Applications of Artificial Intelligence*, pp. 844-849. AAAI, 1999.
4. Burke, R., Hammond, K., and Cooper, E. (1996). Knowledge-based navigation of complex information spaces. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 462-468. AAAI, 1996.
5. DigitalMe. Available at http://www.digitalme.com/. Accessed 19 May 2000.
6. Excite Shopping. Available at: http://www.jango.com/. Accessed 19 May 2000.
7. Gao X., Sterling L. (1998). Classified Advertisement Search Agent (CASA): a knowledge-based information agent for searching semi-structured text. *Proceedings of the Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, pp. 621-624.
8. Glave, J. The Dawn of the Infomediary. Available at: http://www.wired.com/news/business/0,1367,18094,00.html. Accessed 19 May 2000.
9. James Realty. Available at: http://www.tiac.net/users/jamsrlty/. Accessed 19 May 2000.
10. Lieberman, H. (1997). Autonomous Interface Agents. In *Proceedings of CHI 1997*, March 1997, pp. 67-74R.
11. Shearin, S., Maes, P. (1999). Representation and Ownership of Electronic Profiles. *CHI 2000 Workshop Proceedings: Designing Interactive Systems for 1-to-1 E-commerce*. Available at: http://www.zurich.ibm.com/~mrs/chi00/submissions/shearin.doc. Accessed 3 July 2000.
12. Shneiderman, B. (1994). Dynamic Queries for Visual Information Seeking. Available at: ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/93-01html/3022.html. Accessed 19 May 2000.
13. Williams, M. D. (1984). What makes RABBIT run? In *International Journal of Man-Machine Studies* 21, 1984, pp. 333-352.