A User-Guided Cognitive Agent for Network Service Selection in Pervasive Computing Environments

George Lee   Peyman Faratin   Steven Bauer   John Wroclawski
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
{gjl,peyman,bauer,jtw}@lcs.mit.edu

Abstract

Connectivity *is central to pervasive computing environments. We seek to catalyze a world of rich and diverse connectivity through technologies that drastically simplify the task of providing, choosing, and using wireless network services; creating a new and more competitive environment for these capabilities.*

A critical requirement is that users are able to benefit from this rich environment, rather than simply being overloaded with choices. We address this with an intelligent software agent that transparently and continually chooses from among available network services based on its user's individual needs and preferences, while requiring only minimal guidance and user interaction. In this paper, we present an overview and model of the network service selection problem. We then describe an adaptive user agent that learns its user's network service preferences from a very minimal, intuitive set of inputs, and autonomously and continually select the service that best meets the user's needs. Results from preliminary user experiments are presented that demonstrate the effectiveness of our agent.

1. Introduction

Connectivity—ubiquitous, reliable, inexpensive, and essentially invisible wireless access—lies at the heart of many pervasive computing visions. We ask, how might we bring truly ubiquitous connectivity closer to reality?

This paper presents the design and implementation of an autonomous, cognitive personal agent for wireless access service selection. The agent performs a critical function within our larger research framework, the Personal Router. Our work is motivated by two visions: first, that widespread wireless access to the Internet can be catalyzed by a different economic model than the one currently in place, and second, that an open market for a new generation of wireless devices and applications can be created by such a wireless infrastructure.

The first vision is one that links economics and technology. Wireless access requires both a mobile device,[1] and a set of base stations with which to communicate. How can this infrastructure of base stations come into existence? The investment model today is that high-speed wireless LANs may be installed by private organizations within their own facilities, but in the wider area, service must be provided by a large-scale provider who blankets (or sprinkles) a region or a nation with towers.[2]

It need not be this way. An alternative is that *anyone*—small businesses or individuals—can easily put up local base stations and sell access to the Internet within small regions. With a suitable economic market, demand would trigger deployment. These small providers, together with today's large-scale providers and dedicated institutional networks, would create a rich, responsive, and competitive infrastructure for wireless Internet access. Our goal is to provide a technical and economic framework in which to explore this option.

Successful realization of this vision depends on a number of technical capabilities. First, it must be possible for the customer to move transparently and dynamically between different providers and service zones. This implies that transport layer connections, security associations, and the like survive the transition of the user from one provider to another. Although these capabilities are not supported gracefully in today's Internet, a number of current research and IETF efforts seek to address them.

More interesting is the need for this competitive, dy-

[1] Or perhaps a Mobile Ad-Hoc Network.

[2] We note that a WLAN hotspot service that offers only the same limited pricing models as a traditional cellular service is but a small step forward. What is central to our vision is creating richness of competition and business models, not the particular technology in use. Our work applies equally to a WLAN, 3G, or mixed environments.

namic service environment to be presented to customers in comprehensible terms. It is not enough that the user have a choice between wireless service providers. To truly benefit, the user, or their agent, must be able to make this choice simply and intuitively, and to re-evaluate the choice frequently, as service offerings change and the user moves about. Mechanisms that involve manual intervention, detailed understanding of application service requirements or network QoS offerings, and similar complexities, are too burdensome to succeed.

What is needed is an automated service selection mechanism, driven by a high-level, intuitive, unobtrusive capture of the user's current requirements. This mechanism, by considering the user's requirements as well as rules and service descriptions made available by providers, transparently and dynamically selects the most appropriate provider and offered service at any given time. The development of such a selection framework, together with its supporting interfaces, mechanisms, and economic models, is a central objective of our research. In this paper we describe our current prototype selection framework, and present some preliminary experiments that validate its abilities.

The paper is organized as follows. Section 2 describes the problem, and gives some background on our models of the user and the network. Section 3 presents the architecture of our agent and its computational learning model. We present our experimental results in section 4. We give a discussion of related work in section 5. Section 6 concludes with a brief discussion of ongoing work aimed at developing a more sophisticated decision agent.

2. The Service Selection Problem

This section provides a conceptual overview of the service selection problem and the structure of the system we have designed to address it. Figure 2 shows the general relationship between the main elements that impact the selection decision. The Personal Router (PR) is a physical device that, among other tasks, manages the network connectivity between a user's devices and the service providers in his environment. It communicates with service providers to obtain information about network services, and learns the user's preferences through an unobtrusive and intuitive user interface. When the environment changes or the user requests a different service, the PR makes a new service selection based on the information it has about the network and user. In the subsections below we describe the research challenges involved in these interactions and the assumptions we make about them. This lays the foundation upon which the agent architecture in Section 3 is based.

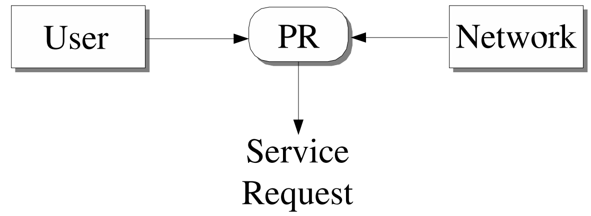

Figure 1. Conceptual Overview of Service Selection

2.1. Network Model

We can imagine many different mechanisms for advertising and selecting services, including negotiation and auction protocols. Regardless of the mechanism, we represent the set of services available to the PR with the variable S. This set of available services may change as the user moves to new locations or service providers change their offerings. Each of the available services $x \in S$ has an associated *service profile* $p(x)$ describing its features, including both performance and price information.

An ideal definition of a service profile is one for which the user's perception of a service's usefulness in a particular context is based solely on the features described by the service profile. If this were the case, then the PR could make accurate selections based only on information in the service profile. In reality, however, it is not possible to capture all the variables that affect the user's perception of a service. Therefore we choose a specific, minimal set of parameters that can be accurately characterized by providers and that are closely related to the user's perception of a service.

To describe the performance of a service, we use a two bucket profile [33], describing a service in terms of its short term and long term burst characteristics. A two bucket profile consists of a token rate ρ, a burst size β, and a refill interval T_{refill} (called *forced_off* in [33]) after which the bucket is refilled if it is empty. This type of profile can describe a wide variety of services useful for different application classes such as email, web browsing, and real-time voice or video conferencing. Importantly, the two bucket profile description corresponds well to the quality perceived by users for common activities such as web browsing that exhibit an on/off heavy-tailed traffic distribution.

Along with quality attributes, service profiles contain information about the price of a service. Service providers may choose complex pricing models with different initial costs, pricing depending on congestion, user-specific pricing, and the like. However, in the profile, we consider just two cost attributes, price per minute c_{min} and price per kilobyte c_{kb}. The total price c_{total} of a service to the user is calculated by applying these price attributes to the duration of usage t and the quantity of data transferred y according to the equation $c_{total} = c_{min}t + c_{kb}y$. This corresponds to a *linearization* of the provider's potentially complex pricing model over the lifetime of a service profile; if the linearization becomes too inaccurate, the provider can advertise, and the PR can respond to, a new profile with changed pricing. Combining the quality and cost features described yields a service profile $p(s) = (\rho, \beta, T_{refill}, c_{min}, c_{kb})$ for a service s.

An important point is that service providers might not advertise services accurately and truthfully. The task of validating or estimating accurate service profiles is important to the success of the PR, but is beyond the scope of this paper. In the present work, we assume that the agent has access to accurate service profiles. We also assume that the PR can seamlessly switch between available services without disruption using a mobility solution such as mobile IP [24].

2.2. User Model

Users do not evaluate services purely in terms of the performance and price features of service profiles. Instead, a user is assumed to perceive services subjectively in terms of two possibly conflicting objectives: quality and cost. These objectives are also assumed to be dependent on the user's goals and context. For example, a user may consider a high bandwidth, high latency service to have high quality for bulk file transfer but low quality for videoconferencing.

To accurately select services, we must identify the user's *context*. The correct service for the user depends on what applications are running and the activity of the applications. In order to make the space of user contexts manageable and to improve performance, we presently take note only of the currently active application. We assume that a user only cares about the network performance of the current foreground application and define a user context g as this application.[3]

Another factor that influences a user's preferences is their current goals and mindset. If the user just wants to surf the web and read the news, then perhaps they prefer a low cost service. On the other hand, if they urgently need to transfer a large file, then they may want a high quality service even if it has high cost. We model these changing goals with a trade-off between quality and cost objectives, represented by $w \in [0, 1]$, where $w = 0$ means the user only cares about cost, $w = 1$ means the user only cares about quality, and values in between represent intermediate weightings.

[3]This assumption is a simplification. A more complete approach would consider different activities occurring within the same application, and might consider the needs of background applications as well as the user's foreground activity. Note, however, that adopting a more complex model of context can be done without modifying the work described in this paper.

We make several (decision theoretic) assumptions about user preferences. First, it is reasonable to assume that users evaluate services based on subjective quality and cost in a given context. That is, any two services which they perceive to have the same quality and cost in that context are perceived as equivalent. We assume that their orderings of services over quality and cost are *complete*—given any two services, the user can decide which one has higher quality or that they are indifferent, and similarly for cost. Additionally, users' cost and quality preferences are *transitive*—if service A has higher quality than service B, and B has higher quality than C, then they perceive A to have higher quality than C. These assumptions allow us to represent their perceptions of the quality and cost of a service s in context g with quality and cost functions $q(g, s)$ and $c(g, s)$, representing the user's orderings over quality and cost. The user perceives service s_i to have higher quality than s_j in context g if and only if $q(g, s_i) > q(g, s_j)$, and similarly if s_i is cheaper than s_j, then $c(g, s_i) > c(g, s_j)$.

We model a user preferences with a utility function $u(q(g, s), c(g, s), w)$. This efficiently represents user preferences and enables the agent to reason about the quality and cost of services in different contexts with different user quality/cost trade-offs. The elicitation and construction of such a function is at the core of not only the service selection problem, but *any* decision problem. Classic solutions to this problem (e.g conjoint analysis) have attempted to achieve this by asking laboratory subjects pairwise questions over a large set of choices. However, as shown in [11], the dynamicity and combinatorial size of the service selection problem means we cannot use classical solution concepts for our problem. In fact, the best we can do, given our concern with usability, is to construct a model of the user's utility function using sub-optimal information. As we will show in the next section, these constraints impact the design of the user-interface that carries information from the user to the agent and vice-versa.

Finally, as mentioned earlier, the service profile does not capture all the factors that influence user perceived quality and cost. However, service profiles are still useful for predicting the value of new services if we assume the service profile features are partially correlated with $q(g, s)$ and $c(g, s)$.

3. Agent Architecture

We decompose the service selection problem into four parts: 1) Devising an intuitive, effective user interface to elicit feedback useful for agent decision-making; 2) Accurately evaluating services in terms of their user perceived characteristics to help learn the value of the agent's actions; 3) Deciding when to change services and which service to select based on the user's preferences, context, and goals;

and 4) Correctly predicting the value of new, previously unseen services based on previous observations. The solutions to each respective subproblem are then modularized in to the following set of components, implemented as the agent architecture (see Figure 2): 1) the user interface (UI), 2) a service evaluator, 3) a service change controller, and 4) a service value predictor. Together, these components allow the PR to learn the value of services from user feedback, adapt to changing user needs, and also estimate the value of new services.

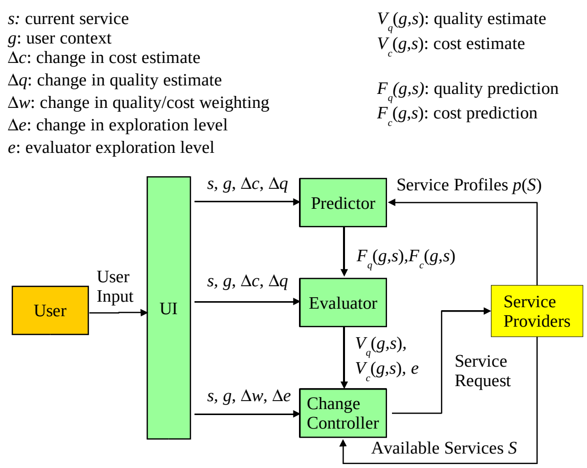

Figure 2. Agent Architecture

3.1. User Interface

As mentioned above, the design of the user interface is constrained by human factors such as ease of use and intuitiveness. The simplest approach is to only have one input: satisfaction or dissatisfaction with the current service. This has the advantage of making service evaluation easier since the PR does not need to track quality and cost separately. However, it fails to capture the user's dynamic quality/cost weighting. Thus we allow the user to not only express whether or not they are satisfied with the current service, but also whether they desire a higher quality or lower cost service. If the user is dissatisfied with a service, it is either because it has low quality or high cost. Therefore the UI provides two buttons for the user to express their feedback r about the current service: a *better* button to indicate $r = \texttt{better}$, expressing dissatisfaction with the current service's quality level and requesting a higher quality service; and a *cheaper* button for $r = \texttt{cheaper}$, expressing dissatisfaction with the current service's cost and requesting a lower cost service. If the user is satisfied with a service, they need not do anything ($r = \texttt{sat}$), expressed as no button presses for some period of time. We assume that the longer the user waits before pressing a button, the more likely they are to be satisfied they with the quality and cost of that service.

If the user inputs $r = \texttt{better}$, it is due to one or both of the following reasons: either the user's perceived quality of the service q is lower than the PR's estimate, or the user's quality/cost weighting w has increased. To the extent that the button press is due to low q, the evaluator needs to update its quality estimate for the service. To the extent that w has increased, the change controller must choose higher quality services. Similar reasoning applies to $r = \texttt{cheaper}$.

The user's willingness to try new services may change as well. The UI may provide a means for the user to express this change via *explore more* and *explore less* buttons, a slider, or attempt to infer it from their behavior. Therefore from an input r the UI needs to generate four outputs, Δq, Δc, Δw, and Δe, the amount to change the quality estimate, cost estimate, quality/cost weighting, and exploration level, respectively.

To help the user make their decisions, the UI also needs to give feedback to the user about critical, but otherwise hidden, parameters such as the cost of a service. Too much feedback is detrimental; giving detailed technical information about each service is likely to distract and confuse the average user. On the other hand, without any feedback at all the user has no basis on which to compare services and decide which one they prefer out of the choices available.

The user must be able to at least identify which service is currently in use and how much it costs. The current service may be indicated using a name, description, or other identifier. There are many possible ways to give feedback about cost, including cumulative cost, estimated cost per unit time, and estimated cost per byte.

3.2. Service Evaluator

The function of the evaluator module is to learn individual user preferences in many different contexts, based on user feedback. In particular, the task of the evaluator is to estimate this perceived quality and cost for each service and activity. For each service s and user context g experienced by the PR, the evaluator maintains quality and cost estimates $V_q(g, s) \in [0, 1]$ and $V_c(g, s) \in [0, 1]$, where 0 corresponds to the worst possible quality or cost (slow or expensive) and 1 is the best (fast or cheap). Since these estimates should be based purely on the user's perception of the service, they are calculated from the Δq and Δc UI outputs and not on any information in the service profile. The evaluator is adaptive, meaning its estimates of cost and quality improve as it receives more user feedback. We implement this adaptation with reinforcement learning [15]. The evaluator's confidence in its learned estimates can be communicated to other agent modules through an exploration value e, allowing it to request more exploration (from the change controller module (see below) when untried services exist and less exploration when it has greater confidence in its estimates.

3.3. Change Controller

The function of the change controller module is to decide when to switch services and which services to select given information from the service evaluator and from the user interface. This switching choice is regulated by the user's context, the user's weighting between quality and cost, as well as the amount of exploration the user will tolerate. Since user utility is a function of perceived quality and cost and a quality/cost weighting w, the change controller must have an estimate of w and an approximate utility function to select services. The change controller estimates the quality/cost weighting based on Δw inputs and then applies an additive linear utility function to the perceived quality and cost of each available service. It then makes selections based on a stochastic function of these utilities. Since the consequences of sub-optimal decisions are minimal (given micro-payments) the change controller may occasionally select a service with lower estimated utility in order to improve the evaluator's estimates. Some amount of exploration accelerates the learning process, but too much exploration results in suboptimal selections [3]. The exploration level of the service evaluator and Δe from the UI affect this balance.

3.4. Service Value Predictor

Finally, to improve performance when previously unseen services are encountered, the PR forms a model of user utility to predict the value of new services based on the Δq and Δc outputs from the UI and the current service profile. The task of the predictor is to approximate user perceived quality and cost as closely as possible given a limited number of observations about their behavior and service profiles. This assumes that there exist functions $f_q(g, p(s))$ and $f_c(g, p(s))$ correlated with the user perceived quality $q(g, s)$ and cost $c(g, s)$ of a service s. The predictor attempts to approximate these functions f_q and f_c based on previous observations. Since the predictor is only used for the initial estimate, predictions need not be completely accurate. As long as its estimates allow the PR to make better than random selections, the predictor can improve performance. We chose a multi-layer neural network (MNN)[13] to compute the solution to this approximation problem. MNNs were chosen because there exists a tractable and optimal training algorithm (back-propagation) which can approximate any arbitrary utility function. Therefore in our implementation, when the PR encounters new services it attempts to approximate the user's utility function using a two-layer feed-forward neural network. The PR trains the predictor on the Δq and Δc UI outputs, the current service profile, and the current activity. As the neural network receives more training data, its predictions will improve. Since the neural network requires a substantial amount of training data before it becomes useful, the PR does not use it until it has accumulated enough observations. Once the predictor is activated, the evaluator can use it to initialize its quality and cost estimates for a given service and activity.

4. User Experiments

To evaluate the adequacy of our approach, we tested the agent's ability to learn user preferences in a static and dynamic service environments in a series of short term controlled user experiments. These preliminary experiments had two objectives. The first was to determine if the PR performs its function correctly, learning user preferences and selecting appropriate network services. Secondly, we assessed the usefulness of automatic service selection in the PR compared to manual selection.

Our experiments confirm that the PR can learn user preferences and select services effectively. The agent also performed similarly to manual selection on average, but required less user interaction to select a good service and reduced variance. The data we collected also gave us valuable insight into how users interact with the PR in realistic situations, helping us further tune and improve the system. More studies are needed to conclusively determine the usefulness of the PR, however. The goal of these experiments instead were to identify important general trends that can then be formulated as hypothesis which test a causal model of interactions [10]. The experimental setup, procedures and results are discussed below.

4.1. Experimental Setup: Network and Users

The set of available services were generated using traffic shaping in a software router. Services were defined by four features: average data rate, cost per minute, and cost per kilobyte. These features were chosen because we assumed they are the easiest for users to perceive in the short time frame of the experiment. The values of these features were chosen to mimic a range of realistic network services, from inexpensive low quality services to expensive high quality services. We chose to use seven quality levels, corresponding to bandwidth levels commonly encountered by users in current 802.11b and broadband, modem, and cellular data networks: 11Mbps, 1Mbps, 384Kbps, 128Kbps, 56Kbps, 28.8Kbps, and 9600bps. Costs were set so that the user must choose services carefully to avoid expending all their credits.

Eight services were available in three simulated locations. For each location there was exactly one optimal service that allowed the user to complete the experiment objectives. All other services were either too costly or too slow. Eight services were chosen because it is a large enough number to make the task of correct service selection hard enough for the subjects while enabling the PR to learn service values within the time frame of the experiment.

Subjects were 17 students and staff of the MIT Computer Science and Artificial Intelligence Laboratory. Subjects were rewarded for their participation with $10 to $20 based on their performance.

4.2. Procedure

In order to evaluate the ease of use and effectiveness of the agent and user interface the performance of the PR was benchmarked against a manual selection policy. Subjects were randomly assigned to one of two groups: 1) the control group where subjects had to choose between services manually by selecting from a menu displaying the available services and their features and 2) the test group where subjects used the PR to select between services, requesting services using the *better* and *cheaper* buttons described earlier. The final distribution of subjects to groups was 8 control and 9 test conditions.

An experiment consisted of three phases. The first phase controlled for task learning effects. Subjects were given ten minutes to become familiar with the user interface, the procedure and the available services. The second phase consisted of a *static* configuration of all eight services for a particular location and tested how well the PR can learn an estimate of user preferences (estimation tests). The third phase was identical to the second phase but tested for the adequacy of the selected choices when the set of available services *changed* from the set available in the second phase (prediction tests). Within each phase the subjects had to complete one or more tasks. For each task, the subjects had to use a web browser to fully load a series of ten web pages within five minutes while selecting a service using the mechanism designated for their group. Each web page contained four large image files. We chose this task because it approximates the network usage of a typical subject shopping or looking for information on-line and makes use of the network service in a realistic and familiar way. Subjects were instructed that they would be charged for their network usage based on the cost of the current service. The task performance of the subjects was given by the dependent variable *score*, measured as the number of credits expended during that task.

In each phase the simulated location and the set of available services changed. In Phase 1 the subject was placed in Location 1 and was asked to perform their downloading task twice for practice. In Phase 2, the PR was reset and the subject chose services from Location 2. The subject was given two attempts at their task and was instructed to try to minimize their score. Finally in Phase 3, the user attempts to minimize their score in Location 3.

4.3. Results

Table 1 shows the summary statistics for experiments that measured the adequacy of the agent's estimation and prediction learning mechanisms. Correlations were measured between independent variables adr and cpm (measuring the amount of bandwidth and cost per minute respectively) and dependent variables q and c, the quality and cost valuation as perceived by the user in Phases 2 (estimation) and 3 (prediction) trials. The high correlations between the variables across both phases suggests that the PR can indeed learn *both* the quality and cost values based on user feedback (in Phase 2) as well as predicting, although less accurately in Phase 3, the value of new services as they become available (in Phase 3). We expect that cost predictions would take longer to learn in Phase 3 since there are three service profile features affecting cost while only data rate affects quality. These results suggest that the predictor can provide useful estimates when services change, but may require more time to learn before it can produce more reliable results.

Table 1. Correlation coefficients for learning service estimates and prediction

	Estimation	Prediction
q vs. adr	$r^2 = 0.793$	$r^2 = 0.946$
c vs. cpm	$r^2 = 0.931$	$r^2 = 0.438$

Figures 3 and 4 show the sorted and ranked distributions of score percentiles across PR and manually selected services in phase 2 trials. The value plotted is the score under which a given fraction of the subjects scored. The left-most data points are the lowest score for that try (hence better, since the goal of the subject was to minimize expenditure) and the right-most data point is the highest score.

The results suggest that compared to manual selection (Figure 4) the PR achieves comparable scores in Phase 2 but with lower variance (Table 2). Figure 3 shows that the performance of subjects who used the PR improved in the second task of Phase 2. In Phase 2, the PR improves the score achieved at almost all percentiles. Figure 4 shows that there is no such improvement within Phase 2 with manual selection, suggesting that the change results from better estimates. Given more time to learn, we expect that the performance of the PR will improve further.

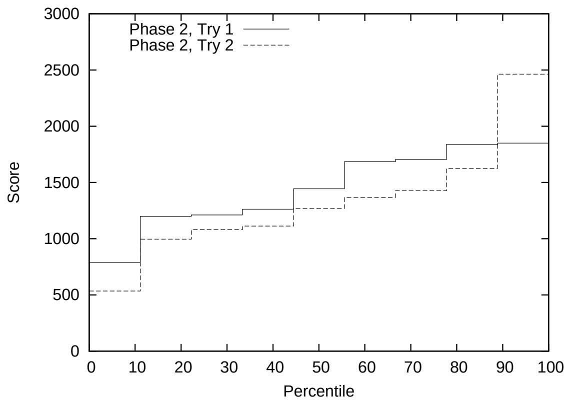

Figure 3. Ranked and sorted distribution of scores in Phase 2, PR selection

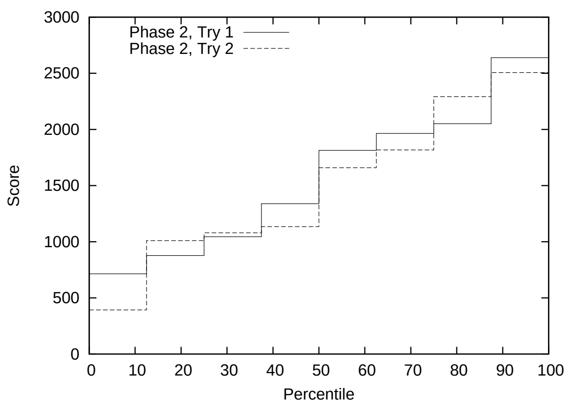

Figure 4. Ranked and sorted distribution of scores in Phase 2, manual selection

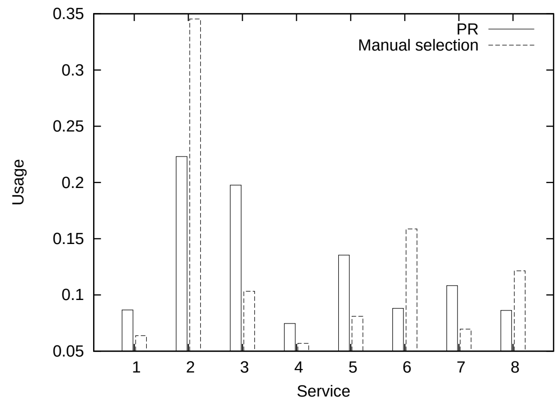

Figure 5. Service usage duration, Phase 2

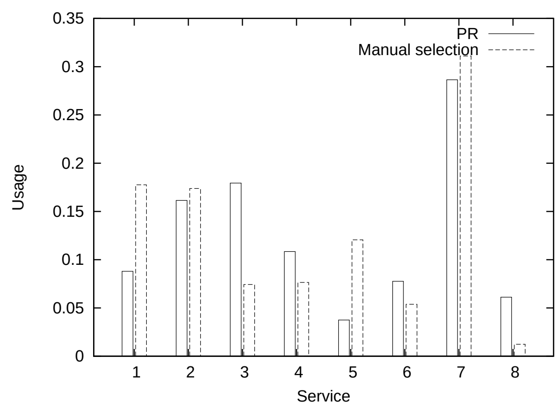

Figure 6. Service usage duration, Phase 3

Figures 5 and 6 show the observed duration of usage of each of the available services in Phases 2 and 3 as a fraction of total usage time respectively. The data shows that both the subjects who used an agent and subjects who used manual selection chose the best service more frequently than other services. In Phase 2, using service 2 results in the best score. In Phase 3, service 7 is best. The figures show that both approaches are capable of identifying and selecting the best service. However, the data suggests that the agent does not choose the optimal service as frequently as under manual selection. Two possible causal hypothesis are: 1) the agent's learning is suboptimal for the given limited number of trials (therefore performance should increase with more trials) and 2) the reward or utility models of the change controller are incomplete or inaccurate.

Figures 7, 8, and 9 show the observed distribution of scores by percentile in Phases 2 and 3 for both the PR and manual selection. The observed mean of scores with PR selected services for Try 1 and 2 in Phase 2 were 1480 and 1430 respectively, with standard deviations of 361 and 616. Conversely, the observed mean of scores for manual selected services for Try 1 and 2 in Phase 2 were 1680 and 1600 respectively, with standard deviations of 722 and 747. The observed mean of scores for PR and manually selected services in Phase 3 were 1680 and 1090 respectively, with standard deviations of 1120 and 713. Table 2 summarizes these statistics. The data shows that the average score achieved by subjects who used the PR is slightly better than that of subjects who used manual selection in Phase 2. In both tries of Phase 2 PR using subjects significantly reduced the variance in their scores, with scores in low and high percentiles closer to the mean. This suggests that service selection guided by the agent provides a more predictable and consistent experience across subjects. In Phase 3, however, the score varied widely for different sub-

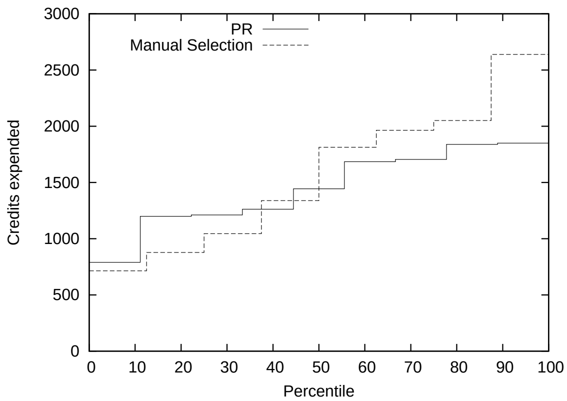

Figure 7. Score distribution, Phase 2 Try 1

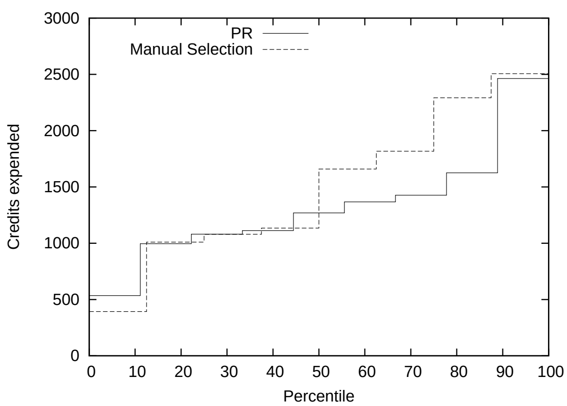

Figure 8. Score distribution, Phase 2 Try 2

jects, with some subjects scoring extremely well and others doing poorly. An examination of the data log revealed that this disparity results from differences in usage during Phase 2. Some subjects were able to train the PR better than others during Phase 2, enabling the agent to make better selections in Phase 3. This suggests that as subjects become more experienced at using the PR, the agent's performance improves substantially. More experiments are necessary to conclusively determine the usefulness of the PR compared to manual selection.

Verbal feedback from subjects regarding their experience was also collected. Subjects of the manual selection mechanism remarked that it was difficult to remember the features of the different services. This suggests that even if the PR performs similarly to manual selection, users may prefer it simply because it requires less cognitive effort to use. In this experiment we used only eight services per location, but in locations with many more services available we expect users would have an even greater preference for automatic service selection.

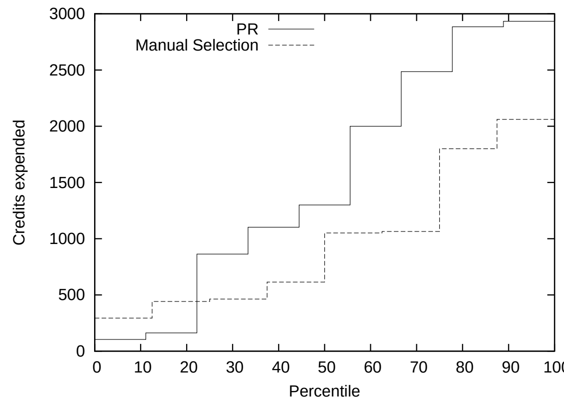

Figure 9. Score distribution, Phase 3

Table 2. Score statistics

	Try 1		Try 2	
	PR	Manual	PR	Manual
Mean	1484	1676	1434	1600
Median	1564	1813	1318	1659
SD	360.7	722.0	616	747

5. Related Work

Our work in network service selection spans several fields of research, including mobile and wireless networks, network quality of service (QoS), intelligent agents, machine learning, and user modeling. Our research synthesizes ideas from these fields to produce a novel machine learning service selection approach.

Previous service selection researchers have developed systems that use an agent to select network services for the user, but they do not attempt to learn user preferences with a simple and unobtrusive interface. In [2], the authors describe a mechanism for users to manually choose among a fixed number of priority-level/price pairs. They study the simulated behavior of multiple users on the load and performance of such a network. Their work assumes that users can accurately and rapidly choose the appropriate priority level as the observed network performance changes. In this paper, services are described by features more complex than a fixed set of priority levels and prices. Furthermore, our focus is not on the efficiency of the network but on ease of use and user satisfaction.

The authors of [20] propose a user agent to select among services in a diverse network environment. They describe the framework for an agent that selects services based on user expectations and application requirements, measures the performance of the current service, rates services, and shares this information with other agents. They do not implement this system, describe the user interface, or give detailed algorithms for selection and rating, however.

There has been previous research in wireless and mobile networks on deciding how and when to switch network service. While the 802.11 specifications [14] do not specify a procedure for when handoffs should occur, a variety of common approaches are taken. These include STA initiated handoffs based upon signal strength information. Access points may also initiate handoffs based upon their load. Commercial products employing multi-mode wireless access technologies typically just prioritize the radios and employ the highest priority radio at any point in time [22]. Prior research has suggested that handoff between wireless technologies ("vertical handoff") [27] and systems employing a variety of wireless networks and technologies [17] are feasible. In [31] the authors argue the need for policy-enabled handoffs across such heterogeneous wireless networks. [29] describes a measurement based prioritization scheme for handovers in mobile cellular networks. In Cellular IP [7], handoff is partly based on clients' signal strength measurements. In Mobile IP [23] handoff decisions are based upon foreign agent advertisements. Katz has studied mobility in wireless overlay networks [16, 17]. Research on seamless and transparent handoffs is similarly voluminous [4, 8, 12, 24, 26, 30]. Software radios have a similar type of problem of determining which radio configuration to employ [9]. None of these approaches account for user preferences, however.

In this paper we employ several well-studied user modeling techniques. The authors of [34] give an overview of different ways to model and predict user behavior, including a discussion of content-based and collaborative learning, linear models, Markov models, neural networks, and Bayesian networks. In [32], the authors discuss the challenges of machine leaning for user modeling, including the need for large data sets, the need for labeled data, concept drift, and computational complexity. Most of these approaches require a great deal of explicit feedback from the user, however. In order to make our system unobtrusive and easy to use we do not explicitly ask for ratings from the user and use implicit rating instead, as described in [21]. Nichols examines methods for implicitly learning user preferences from their behavior. We also incorporate aspects of previous approaches to measuring user perceived quality that make use of feature-based models. The authors of [1, 25] describe several methods to make recommendations based on the features of items users previously rated.

We make use of AI techniques to learn user preferences. The traditional approaches to modeling user utility include conjoint analysis, in which users are asked to specify their preferences over different goods [18, 19]. We reject this approach since users are unable to accurately evaluate a service based on a description of their features; they must experience it before they can determine its utility. Instead we use ideas from reinforcement learning to learn user preferences based on their behavior. Kaelbling gives an overview of the field in [15]. Sutton and Barto provide an excellent reference for reinforcement learning in [28].

The abstract model of service selection employed by our agent resembles a partially observable Markov decision process (POMDP) [6, 5]. This gives us a framework to reason about nondeterminism and unobservability in the network and user and allows us to take advantage of the previous work in reinforcement learning with Markov models.

6. Conclusions and Future Work

A major challenge in any communication-rich environment is selecting the best network access service for a user's needs. To address this usability problem in service selection we introduced the concept of the Personal Router agent. AI based techniques were used to design an autonomous adaptive agent for learning user preferences in realistic network environments and for different activities. The developed agent was then empirically tested in a series of exploratory experiments that assessed the learning capabilities as well as the comparative performance of users using the PR and manual service selection in a number of experimental settings. We found positive learning effects and statistical and cognitive benefits for autonomous service selections performed by the PR.

We plan to refine the work presented here in several dimensions. First, we continue to empirically evaluate different learning algorithms for more complex models of the PR in environments with different numbers of services available, changing user contexts, more complex network services, uncertainty in network services, and changing user goals. We plan to perform long-term user studies to determine if performance improves as users become more skilled at using the PR and the agent collects more observations about user preferences. Finally, after achieving a satisfactory level of performance for the agent we plan to extend the single agent decision mechanism to multi-agent systems (MAS) using peer-to-peer networks, including the exploration of distributed reputation, gossiping, epidemic, and collaborative filtering mechanisms. This information can help the PR initialize and verify the accuracy of service profiles or estimate user preferences.

Finally, we note that our approach of using an unobtrusive learning agent to make decisions for the user facing a complex environment applies to problems beyond wireless network service selection. We believe that as computing environments becomes more rich and pervasive, users will increasingly face this situation. We see a growing need for intelligent agents similar to the PR that can both learn individual user preferences and act on a user's behalf without disrupting or distracting them, and believe that the results of our research will provide useful progress in this direction.

References

[1] J. Alspector, A. Kolcz, and N. Karunanithi. Feature-based and clique-based user models for movie selection: A comparative study. In *User Modeling and User-Adapted Interaction*, volume 7, pages 279–304. Kluwer Academic Publishers, 1997.

[2] J. Altmann, H. Daanen, H. Oliver, and A. S.-B. Suárez. How to market-manage a QoS network. In *IEEE InfoCom2002*, 2002.

[3] M. Balabanović. Exploring versus exploiting when learning user models for text recommendation. In *User Modeling and User-Adapted Interaction*, volume 8, pages 71–102. Kluwer Academic Publishers, 1998.

[4] H. Balakrishnan, S. Seshan, and R. H. Katz. Improving reliable transport and handoff performance in cellular wireless networks. *ACM Wireless Networks*, 1(4), 1995.

[5] G. Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of American Association of Artificial Intelligence*, pages 239–246, Edmonton, Alberta, Canada, 2002.

[6] G. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[7] A. Campbell, J. Gomez, S. Kim, A. Valko, C. Wan, and Z. Turanyi. Design, implementation, and evaluation of cellular IP, 2000.

[8] R. Cceres and V. N. Padmanabhan. Fast and scalable wireless handoffs in support of mobile internet audio. In *Mobile Networks and Applications*, pages 351–363. 1998.

[9] A. Chiu. Adaptive channels for wireless networks, 1999.

[10] P. R. Cohen. *Empirical Methods for Artificial Intelligence*. MIT Press, Cambridge, MA, 1995.

[11] P. Faratin, G. Lee, J. Wroclawski, and S. Parsons. Social user agents for dynamic access to wireless networks. In *Proceedings of AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments*, Stanford, CA, US, 2003.

[12] D. Forsberb, J. T. Malinen, J. K. Malinen, and H. Kari. Increasing communication availability with signal-based mobile controlled handoffs. In *IPCN*, 2000.

[13] S. Haykin. *Neural networks: a comprehensive foundation*. McMillan, N.Y., 1994.

[14] IEEE Standard. 802.11 1999 edition (1999).

[15] L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[16] R. H. Katz. Adaptation and mobility in wireless information systems. *IEEE Personal Communications*, 1:6–17, 1994.

[17] R. H. Katz and E. A. Brewer. The case for wireless overlay networks. In Tomasz Imielinski and H. F. Korth, editors, *Mobile Computing*, pages 621–650. Kluwer Academic Publishers, Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1996.

[18] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives*. John Wiley and Sons, New York, 1976.

[19] R. L. Keeney and H. Raiffa. *Design and Marketing of new products*. Prentice-Hall, Cambridge, UK, 1980.

[20] B. Liver and J. Altmann. Social carrier recommendation for selecting services in electronic telecommunication markets: A preliminary report. Technical Report TR-97-033, ICSI, Berkeley, CA, 1997.

[21] D. Nichols. Implicit rating and filtering. In *5th DELOS Workshop on Filtering and Collaborative Filtering*, Budapest, Hungary, November 1997.

[22] PadCom, Inc. `http://www.padcomusa.com`, 2003.

[23] C. Perkins. IP mobility support request for comments(standard). In *Internet Engineering Task Force*. 2000.

[24] C. Perkins and K. Wang. Optimized smooth handoffs in mobile IP. In *Proceedings of the IEEE Symposium on Computers and Communications*. 1999.

[25] B. Raskutti, A. Beitz, and B. Ward. A feature-based approach to recommending selections based on past preferences. In *User Modeling and User-Adapted Interaction*, volume 7, pages 179–218. Kluwer Academic Publishers, 1997.

[26] S. Seshan, H. Balakrishnan, and R. Katz. Handoffs in cellular wireless networks: The daedalus implementation and experience, 1996.

[27] M. Stemm and R. H. Katz. Vertical handoffs in wireless overlay networks. *Mobile Networks and Applications*, 3(4):335–350, 1998.

[28] R. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, Cambridge, MA, 2002.

[29] S. Tekinay and B. Jabbari. A measurement based prioritization scheme for handovers in mobile cellular networks. In *IEEE Journal on Selected Areas in Communications*, 1992.

[30] N. Tripathi and J. VanLandinoham. Handoff in cellular systems. *IEEE Personal Communications*, 5:26–37, 1998.

[31] H. J. Wang, R. H. Katz, and J. Giese. Policy-enabled handoffs across heterogeneous wireless networks. In *WMCSA*, 1999.

[32] G. I. Webb, M. J. Pazzani, and D. Billsus. Machine learning for user modeling. In *User Modeling and User-Adapted Interaction*, volume 11, pages 19–29. Kluwer Academic Publishers, 2001.

[33] X. Yang. Designing traffic profiles for bursty internet traffic. In *Proceedings of IEEE Global Internet*, Taipei, Taiwan, 2002.

[34] I. Zukerman and D. W. Albrecht. Predictive statistical models for user modeling. In *User Modeling and User-Adapted Interaction*, volume 11, pages 5–18. Kluwer Academic Publishers, 2001.