The Personal Router: An Agent for Wireless Access

P. Faratin and **J. Wroclawski** and **G. Lee**
Laboratory for Computer Science
M.I.T
Cambridge, 02139, USA
peyman,jtw,gjl@mit.edu

S. Parsons
Department of Computer and Information Science
Brooklyn College, City University of New York
NY,11210, USA
parsons@sci.brooklyn.cuny.edu

Abstract

The Personal Router is a mobile personal user agent whose task is to dynamically model the user, update its knowledge of a market of wireless service providers and select providers that satisfies the user's expected preferences. The task of seamlessly managing the procurement and execution of short or long term connection for a mobile user is further complicated because mobile users performs multiple, concurrent and varied tasks in different locations and are reluctant to interact and provide subjective preference information to the agent. In this paper we present a detailed description and a formal model of the problem. We then show how the user modeling problem can be represented as a Markov Decision Process and suggest reinforcement learning and collaborative filtering as two candidate solution mechanisms for the information problem in the user modeling.

Introduction

The Personal Router (PR) project is a multi-disciplinary research effort whose overall goal is to analyze the technology and policy interactions that can occur in future wireless protocol and simultaneously assist in designing equitable multi stake-holder policies (Clark & Wroclawski 2000; Internet & Telecoms Convergence 2002). A crucial part of this goal is the technological infrastructure that supports mobile access to wireless services. Such an infrastructure poses a number of challenges along various dimensions including:

- network support for mobility and fast hand-off
- service description and advertisement mechanisms
- pricing policies
- determination of quality of service
- network traffic monitoring
- user modeling

The last problem is the main focus of this paper. We are interested in developing agents that model their user's requirements in order to select, or negotiate with, wireless service providers. In this paper we present a general description of the user modeling problem in a wireless access domain, concentrating on an in-depth problem description together with some preliminary single and multi-agent solutions. In particular, the underlying problem addressed in this paper is the problem of decision making with uncertain and incomplete information. The sources of scarcity and incompleteness of information in the wireless domain are due to: a) changing user preferences given different service and requirement contexts; b) the spareness of user preference data given the combinatorially large elicitation space and c) the variability inherent in the network itself resulting in uncertainties by both the buyers and the sellers of a service as to the guarantees that can be made over the quality of a service (QoS). One net result of such sources of complexities is our inability to use classical utility analysis techniques, such as conjoint analysis, to elicit user preferences (Keeney & Raiffa 1976; 1980). Furthermore, classical decision analysis techniques have a number of limitations such as "clamping" the decision environment to include a set of manageable and well behaved features (Doyle & Thomason 1999) .

The contributions of this paper are mechanisms to represent and assist agent's decision making in light of such classes of uncertainties. To achieve this we present a formal description of the wireless service selection problem that is implementable as a Markov decision process. We then present some initial contributions on how to integrate learning and social mechanisms within the MDP framework that model PR's incremental updating of individual user preferences given information uncertainty and incompleteness.

The paper is organized as follows. An example scenario is briefly described in the first section followed by its feature. We then present a formal model of the service selection problem that gives us a language for describing the problem in an unambiguous manner. We then show how this problem description can be computationally represented within a Markov Decision Model followed by how an agent might use the combination of decision mechanism of an MDP and other information mechanisms to develop a model of the user. In the penultimate section we informally touch on the possibility of modeling the interactions in the entire system, consisting of the user and the agent, as interactions in a Markov game. Finally, we presents our conclusions together with the directions of future research.

An Example Scenario

Consider a user who is going to meet a friend for lunch at a restaurant. However, the user does not know how to get to the restaurant, so on his way out of the office he uses his web browser on his PDA to find the location of the restaurant. The PDA notifies a device, which we will call the *Personal Router* (PR), that its current *activity* is PDA web browsing and requests network service. The PR is the interface between user devices and the Internet that, for wireless services at least, is currently organized in hierarchical layers consisting of base stations that provide wireless services to individual users who in turn receive services from upstream ISPs. Assume that the PR knows about three different available *service profiles*,[1] description of services provided by access providers: the wireless service provided by the user's company, Verizon's basic wireless service, and Verizon's premium service. Based on the user's past behavior, the PR knows that he prefers his company's service if it's available. The PR connects to his company's access point and authenticates itself. All of this happens in a fraction of a second. The user then uses his web browser to get directions to the restaurant. When he is done the web browser tells the PR that it no longer needs Internet service. The PR disconnects from the company access point.

Assume now that the user gets to the restaurant a little early, so he turns on his MP3 player and listens to some music. He likes what he hears and asks his MP3 player to download more songs by the current artist. The MP3 player requests that the PR select the best service for the current activity, bulk file transfer. While the user was walking, the PR was collecting service profile announcements from access points broadcasting their available services. The PR knows of three different service profiles in this area: the restaurant's wireless service and Verizon's basic and premium services. Assume that the user has never been to this location before, but other PR users have. The PR consults a mechanism that maintains and updates group preferences and selects from this information source Verizon's basic service. However, the user is dissatisfied by the current service, noticing that the music is downloading slowly, so he presses a button on the PR to indicate that he is dissatisfied with the service quality. Again the PR refers to the group preference database and determines that the restaurant's service is higher quality than Verizon's basic service. The PR switches to the restaurant's wireless service. However, the user is still dissatisfied with the performance and asks for a higher quality profile once again. The PR selects the premium service.

In general, the goal of the PR is to deliver services to the user that perfectly satisfy his/her requirements and minimize their interactions with the system. However, in the absence of perfect information about the user the PR is likely to select inappropriate services that causes the user to experiment with the attributes or features of a PR selected service by continually interacting with the system. The features of a service we consider important in our applications are both the perceived quality and the price of a service. Users are given an interface to manipulate these features as free variables via a *better* and *cheaper* buttons on the PR respectively. The assumption we make is that user will choose better or cheaper services if the current selected service is either of poor quality or high price respectively. This process of interaction with the PR may continue until the PR learns to select a service that satisfies the user's current tasks and goals.

[1]This knowledge is not embedded into the PR but is dynamically updated by the PR.

Features of the Scenario

We identify a number of important problem features in the above scenario that form the basis of the system requirements and constrain the space of possible system design. The problem feature are:

- **multi buyer seller marketplace** for J finite number of wireless services, where M different service providers (ISPs and/or individuals) sell service profiles with differentiated price and quality features to N number of buyers. Furthermore, the trading mechanism can either be negotiation or take-it-or-leave it, and can occur over both a spot market (short term) and a contract (long term). However, although a user may have a long term contract for a service s/he can also select/negotiate a service from the current short term market.
- **repeated encounters** between buyers and sellers. Whereas sellers are likely to be stationary, buyers on the other hand may visit *locations* repeatedly. One implication of this feature is that complicated incentive mechanisms may not be needed in this market to prevent gaming of the system by either a seller or a buyer. In turn, cooperation may become self enforcing through some reputation mechanism given encounters between buyers and sellers are repeated.
- **uncertainty** associated to not only the buyers and sellers decisions and actions, but also the agent's model of the user. Buyers may not be sure of their preferences for a service. For example, a buyer may not be sure whether she likes a service until she tries it. Conversely, due to the complexities of network model a seller may not be able to guarantee the quality of service that they advertise. Finally, users are unwilling and/or unable to extensively interact and communicate their preferences to the PR.
- **complex service profiles** advertised by service providers. ISPs may offer services with elaborate descriptions of quality and price. Quality may be described objectively in terms of bandwidth and latency, or with a subjective label such as "Gold Service". Service pricing is likely to be some complicated function of a number of factors. For example, cost may be given as a simple price per unit time, or it may depend upon factors such as time of day, amount of usage, and previously signed contracts. The PR must be able to understand these types of service profiles using some network and user models.
- **context** of a user, defined by the following state variables:
  - goals (or activities) of the user (e.g. arranging a meeting, downloading music). Users may have multiple concurrent goals/activities.

  - the class of application the user is currently running in order to achieve her goals (e.g. reading and sending emails, file transfer). Furthermore, different applications have different bandwidth requirements and can tolerate service degradation differentially (Shenker 1995).
  - the urgency of the user request (e.g. urgent, flexible, intermediate)
  - the location of the user. We distinguish two possible user location states: nomadic and stationary. In nomadic state a user moves through locations speedily (e.g in a taxi). Therefore overhead costs should be minimized during service provisioning given users require the services instantaneously. Therefore, PR needs to be reactive. Conversely, in a stationary state a user is in one location (e.g in a coffee shop). Therefore communication costs can be trade-off against negotiation/selection of better services. PR can therefore provision more resources in the course of service selection process. In either state the PR needs to be pro-active in updating its knowledge of service profiles.

- **rapidly changing context** of a user. The rate at which the PR switches between service profiles and the speed with which the user changes goals and applications makes it difficult to learn user preferences. The problem is further complicated because the different user activities and applications have several different temporal profiles. The PR needs to learn how much a user likes a service, but the user evaluates services based on the performance of the network over the last several seconds or minutes.
- **continuous service switching** by a user because of a combination of mobility and changing user goals. There also exists some switching costs associated with both dropping a profile and handover to another provider.
- **minimal user interface** between the user and the PR. The user interacts with the agent via an extremely simple user interface (UI) that consists of only three buttons: *better*, *cheaper*, and *undo*. These buttons provide feedback to the personal router about the current service and also request that the PR switch to a different service. The user does not express preferences explicitly; instead, the PR must learn as much as possible about user preferences from the way the user interacts with the UI. The PR in turn may provide feedback information to the user in terms of prices and possibly quality of a service, although feedback on the quality of a service is more likely to be based on the user's perception of how well the applications are running.
- **user tolerance** to suboptimal decisions in service selection. Because the operating costs of a single service for the seller is almost zero and the period of access demand for a buyer can be short, compared to non-wireless contract services, then prices for spot market wireless services are likely to be relatively smaller compared to prices for other types of commodity goods. Therefore users maybe more tolerant of suboptimal services (or noise in the service selection process) because the cost of decision errors are low. Another secondary implication of low prices is that coordination costs may be higher than services prices thereby creating a barrier to coalition among the buyers to drive down prices.
- **distributed group preference and network models** learned by the PR in order to improve the accuracy of service selection. The PR uses group preferences to infer user preferences. It also uses information about the network gathered by nearby PRs to improve its knowledge about which network services are available. The protocols used for learning the group and network models must be efficient and be able to deal with imperfect information and inaccessible PRs in the network. The limited resources of the PR makes it important that we identify the most useful data to retrieve and retain.

Service Selection Problem

Figure 1 shows the functional architecture for the service selection component of the PR. The functional components and their interactions that make up the PR are shown inside the dashed box. The aim of this work is to define representation and algorithms for the problems of service selection given individual and group models. The inputs to this selection function are:

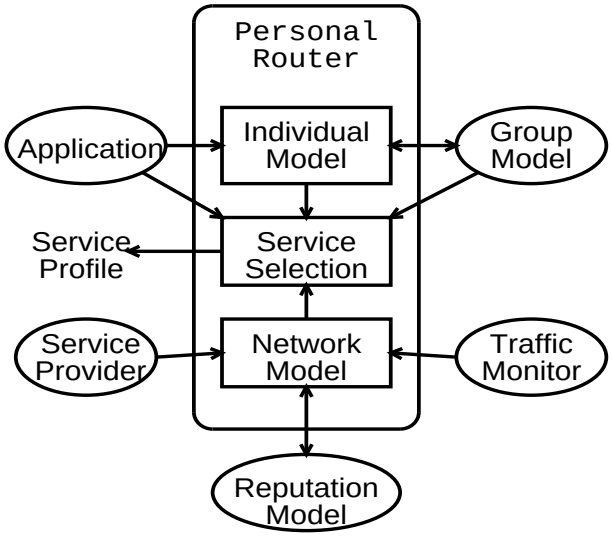

Figure 1: PR Architecture

- set of service profiles, derived from the network model. For current purposes we will assume the selection process has access to some well defined set of service profiles P derived from a set of mechanisms for modeling the network. However, due to the short range nature of the wireless communication the composition of the set in any instance of time can change because the location of the user changes. Furthermore, the complexity of a service description can vary according to a number of variables. One possible description scheme is in terms of the price and bandwidth of the service profile, where bandwidth itself can be described in a canonically in terms of other

network level service features such as peak rate, average burst, etc. The important point is that different users are likely to experience different subjective quality associated with the bandwidth.

- a model of the individual user and their preferences over the set of profiles available in the market. This input is described in detail below.
- a model of the group preferences
- application requirements. The selection of a wireless service is not only dependent on what the user requires but also the requirements of the application. However, since the behaviour of the application is under the control of the application designer its behaviour is therefore assumed to be more predictable than the user and can be generally modeled by an application's elasticity profile to bandwidth levels (Shenker 1995). For instance, text based applications such as email, ftp or telnet are called elastic applications because they can tolerate degradation of service quality but still operate, and their response profile exhibit decreasing marginal rates of improvement with increasing bandwidth. Conversely, inelastic applications, such as video conferencing, can only operate within a strict regions of bandwidth levels. We will concentrate on the user requirement problem in this paper.

A Formal Model of the Service Selection Problem

A formal model of the problem above is developed in this section. The goal of this model is not to commit to or specify domain level details but instead provide a language for specifying implicit and explicit objects that exist and relations that hold in the problem description above. By implicit objects we mean objects whose true values are inaccessible to the agent. For example, the agent may not know the goal or time deadlines of the user but as designers of agents we can develop agents that have probabilistic representations of and the ability to reason over such objects.

We condition each service selection process instance on the current context of the user. As mentioned above a user context includes the current running application set, the time deadline and the location of a user for current goal. We let C represent the set of all possible contexts and $C^g \subseteq C$ be the set of contexts that are partitioned by the user goal g. An element $c \in C$ is composed of the tuple $c = \langle \beta, \gamma, \delta \rangle$, where β, γ and δ represent the set of running applications, user deadlines and locations respectively. Then, a particular user context $c^g \in C^g$, partitioned by the goal g, is defined by the tuple $c^g = \langle \beta^g, \gamma^g, \delta \rangle$, where β^g, γ^g and δ represent the set of running applications compatible with current goal g, the user deadline for current goal g and the concrete location of the user respectively. The location of a user at any instance of time is represented by both the physical location as well as the temporal location.

Next we let $\mathbf{P}$ represent the set of all possible service profiles, where each element of this set $P \in \mathbf{P}$ is composed of n features f_i, $P = (f_1, \ldots, f_n)$. Because service profiles available at any time change due to both user roaming (given a nomadic user) and changes in service offerings (given service providers' uncertainty in the state of the network) then we assume the (physical and temporal) location δ of a user partitions the set of possible service profiles available. Therefore we let $P^\delta \in \mathbf{P}$ represent the subset of possible service profiles available to the user in location δ.

Let $\mathbf{R}$ represent the set of all requirements for all applications (elastic, inelastic or adaptive). Further let the set of all requirements of a given application, $R \in \mathbf{R}$, be given by the set of m service profile features or $R = (f_1, \ldots, f_m)$. We then let $R^{c^g} \subseteq \mathbf{R}$ represent the subset of all requirements of all applications $\mathbf{R}$ that are being currently used by the user in context c for goal g.

Next let the set of all user preferences be given by $\mathbf{U}$. We then let each element of this set, $U \in \mathbf{U}$ represent a unique orderings over all the possible pairs of service profiles $\mathbf{P}$, or $U = (P_i \succ P_j, \ldots, P_{l-1} \succ P_l)$[2] for all combination of l profiles. Similarly, the current user context and goal partition the set of all possible preference orderings, or $U^{c^g} \subseteq \mathbf{U}$.

The ordering generated by U can then be captured by a utility function u such that:

$$u(P_i) > u(P_j) \quad \text{iff} \quad P_i \succ P_j \tag{1}$$

One possible utility function is the simple weighted linear additive model:

$$u^{c^g}(P_i) = \sum_{j=1}^{n} w_{ij}^{c^g} v(P_{ij}) \tag{2}$$

where $u^{c^g}(P_i)$ is the utility for profile i in context c given user goal g. $w_{ij}^{c^g}$ in turn is the weight that the user attaches to feature j of profile i in context c and user goal g. Finally, $v(P_{ij})$ is a function that computes the value (or goodness) of a feature j of profile i.

Finally, we can also model the utility of switching to a new service profile as:

$$u^{c^g}(P_i \rightarrow P_j) = Eu^{c^g}(P_j) - (u^{c^g}(P_i) + cost(P_i \rightarrow P_j)) \tag{3}$$

where $Eu_{P_j}^{c^g}$ is the expected utility of switching to a new profile j and $cost(P_i \rightarrow P_j)$ is the (switching and monetary) cost of switching from profile i to profile j.

Representing the Problem as a Markov Decision Process

The aim of this section is to show how the above formal model of the PR problem can be described within the Markov Decision Process (MDP) modeling framework (Kaelbling, Littman, & Moore 1996; Boutilier, Dean, & Hanks 1999). An MDP is a directed acyclic graph composed of a set of nodes and links that represent the system states $\mathbf{S}$ and the probabilistic transitions $\mathbf{L}$ amongst them respectively. Each system state $S \in \mathbf{S}$ is specified by a set of

[2]The operator $\succ$ is a binary preference relation that gives an ordering. For example, $A \succ B$ iff A is preferred to B.

variables that completely describe the states of the problem. The value of each state variable is either discrete or continuous but with the constraint that each state's variable values be unique. In our problem each system state $S \in \mathbf{S}$ is fully described by the combination of:

- the user context ($c^g = \langle \beta^g, \gamma^g, \delta \rangle$) for goal g
- the set of profiles available in the current location (P^δ) and
- the user interaction with the PR, which we will represent by the variable I.

Therefore, a complete description of *a* system state at time t is:

$$S^t = (\beta^g, \gamma^g, t, loc^g, P, I) \quad (4)$$

where $\beta^g, \gamma^g, t, loc^g$ represent the context of the user for goal g. Note that for reasons to be given below we disaggregate δ, the user location and time, to two state variables t and loc^g, the location of the user in temporal and physical space respectively. We can also specify user goals g in a similar manner by a subset of system states $S^g \subseteq \mathbf{S}$.

The other element of a MDP is the set of possible actions $\mathbf{A}$. Actions by either the user, the PR or both will then results in a state transition, that change the values of the state variables (see tuple 4), to another state in the set of all possible states $\mathbf{S}$. In an MDP these transitions are represented by links $\mathbf{L}$ between nodes that represent the transition of a system state from one configuration to another after performing some action. Additionally, each link has an associated value that represents the cost of the action.

In our problem the set of actions A available to the user u are defined by the set $A^u = \{\Delta^{loc}, \Delta^{app}, \Delta^I, \phi\}$, representing changes in the user location, set of running applications, service quality and/or price demand and no action respectively.[3] The consequences of user actions are changes in values of state variables $\beta^g, \gamma^g, t, loc^g, P, I$; that is, changes in either:

- the user context (changes in running applications, the time deadlines for connections, the current time, the user location and/or price/quality demands, observed by interaction with the PR via better and cheaper responses) or
- the set of currently available profiles or
- the combination of the state variables.

Conversely, the set of actions A available to the PR are defined by the set $A^{PR} = \{\Delta^{P_i \to P_j}, \phi\}$ representing PR dropping service profile i and selecting j and no action respectively. The consequence of a PR action is a change in the likelihood of future user interaction I, where decreasing likelihoods of user interactions is more preferred.

[3]Note, that since time is an element of the state description then the system state always changes in-spite of no action by either the user or the PR or both. Furthermore, the granuality of time is likely to be some non-linear function of user satisfaction, where for example time is finely grained when users are not satisfied with the service and crudely grained when they are satisfied. However, the granuality of time is left unspecified in our model.

Additionally, in an MDP the transitions between states are probabilistic. Therefore there exists a probability distribution $Pr_{a_j}(S_k|S_j)$ over each action a_j reaching a state k from state j.

Finally, we can compute the utility of a service profile i in context c for goal g (or $u^{c^g}(P_i)$—see equation 2) as the utility of being in a unique state whose state variables $(\beta^g, \gamma^g, t, loc^g, P, I)$ have values that correspond to service i in context $c = \{\beta^g, \gamma^g, t, loc^g\}$. The utility of this corresponding state, say state m, is then referred to as $U(S_m)$. However, since in the formal model above a goal partitioned the set of all possible contexts, that in turn partitioned the ordering of profiles, so likewise the utility of a state m is computed by the function $U(P_i, c^g)$, the conjunction of both the the utility of a context given a user goal, $U(c^g)$ and the current profile given the context ($U(P_i|c^g)$). That is:

$$U(S_m) = U(P_i|C^g) \wedge U(C^g) \quad (5)$$

where $\wedge$ is the combining operator (Shoham 1997).

Example MDP of PR Service Selection Problem

A subset of the state space (nodes) and transition paths (vertices) is shown in figure 2 below. Bold and normal links represent the action performed by the user and the PR respectively. Assume the current state of the system at time t is given by the node S_i, representing a unique context, profile set and user demand $S_i = (\beta^g, \gamma^g, t, loc^g, P, I)$. At the next time step $t+1$ the PR may decide to select another service profile $P^{'}$ for goal g because the utility of state $S_k = 0.8$ is greater than $S_j = 0.2$. However, consequences of actions are in-deterministic in an MDP. Therefore assume that the PR's action $A^{PR} = \Delta^{P \to P^{'}}$ results in the state transition $S_i \to S_k$, corresponding to $S_k = (\beta^g, \gamma^g, t, loc^g, P^{'}, I)$, with probability 0.95 (or $Pr_{\Delta^{P \to P'}}(S_k|S_i) = 0.95$). However, due to noise in the selection process there is still a 5% chance that action $\Delta^{P \to P^{'}}$ results in another state S_j that corresponds to some other profile $P^{''}$ being used. Therefore the expected utility of state S_k, $EU(S_K)$, is computed as $EU(S_K) = Pr_{\Delta^{P \to P'}}(S_k|S_i)U(S_k) = 0.95 \times 0.8 = 0.76$. Next assume the user takes an action at the next time step

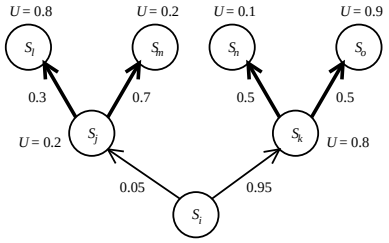

Figure 2: An Example of a Portion of a State Space

$t+2$ from either $S^{t+2} = S_j$ (PR selected $P^{''}$) or $S^{t+2} = S_k$ (PR selected $P^{'}$). The user may at any moment in time take

many simultaneous actions represented by the compound action $A^u = \{\Delta^{loc} \times \Delta^{app} \times \Delta^I\}$. Therefore for explanatory purposes we only consider the user price quality demand action, Δ^I. Further assume that states l to o represent changes in only quality demand profile (observed as "better" request). Then, the value of the links from S_k represent the belief model of the PR of what is the likelihood of the user requesting different service quality services (here S_n and S_o) given the current service profile $P^{'}$ at state S_k.

The problem of *estimating* and updating the (link) probabilities and (state) utilities of an MDP is described in the sections below.

Reasoning with MDPs

The MDP formulation of the service selection problem gives us a representational framework to model the user behaviour and preferences as the combination of state transition probabilities and utilities. Reasoning with an MDP (or user modeling in our problem) in turn is taken to mean *both*:

- solving an MDP and
- updating the transition and utility estimates over the state space.

However, we first consider the problem of intractability in the size of the state-space, a serious consideration in MDP problems, before returning to the problem of how to optimally reason with MDPs.

Pruning the State-Space

Theoretically each state of a system can access/transition to any other possible state. That is, we can represent all possible states of the system (or possible worlds) through construction of a graph of nodes and links of an MDP that are fully connected.

However, in practice computing optimal policies when there are exponentially large number of states and transitions is intractable. Therefore we use domain level structural assumptions to make the solution to the MDP problem tractable. The structural assumption we make is that natural environmental constraints limit the number of possible states that can be reached from any other state.[4] Firstly, we partition the transitions of an MDP into its temporal, spatial, application and user demand dimensions (or partitioning using elements of the context of the user). We then make the following assumptions about the structure of the MDP. Along the temporal dimension the MDP we consider is non-stationary but instead "unfolds" sequentially in time. The assumption we make about the spatial dimension of the problem is that the user cannot be in different locations at the same time. Therefore, states with different location but same time values are unreachable. Similarly, running of application/s is not instantaneous but unfolds in time. Therefore states where an application is running and simultaneously turned off are unreachable. Finally, users also require time to experience the services. Therefore, there are not only no transitions to states with different demand profiles in same time frame sanctioned, but also there is likely to be some increasing or decreasing likelihood of changes in demand over time given by some probability distribution. Finally, the transitions in the state-space can be further reduced by assuming that, for a given granuality of time, only states adjacent in time are directly connected and all other future states are reachable via an indirect path (i.e. future states cannot be reached in a single step).

[4]Assumptions essentially prune the transition/links between states and can be viewed functionally to be equivalent to infeasible regions in constrained optimization problems.

Solving an MDP

On each time step solving an MDP is simply defined by finding a policy π that selects the optimal action in any given state. There are a number of different criteria of optimality that can be used that vary on how the agent takes the future into account in the decisions it makes about how to behave now (Kaelbling, Littman, & Moore 1996). Here we consider the finite horizon model, where at any point in time t the PR optimizes its expected reward for the next h steps:

$$E(\sum_{t=0}^{h} r_t) \tag{6}$$

where r is the reward the PR receives which in our problem domain is the utility of the user, observed as interactions with the cheaper/better button. Therefore, the model allows the contribution derived from future h steps to contribute to the decisions at the current state. Furthermore, by varying h we can build agents with different complexities, ranging from myopic agents $h = 1$ to more complex agents $h > 1$.

Given a measure of optimality over a finite horizon of the state-space solving an MDP (or selecting the best policy) is then simply selecting those actions that maximize the expected utility of the user (see example in section above):

$$\pi = \arg\max E(\sum_{t=0}^{h} U_t) \tag{7}$$

Such a function is implemented as a greedy algorithm.

Estimating and Learning Probabilities and Utilities

The other component of reasoning with the MDP is how to form an initial estimate and subsequently update model parameters values (transition probabilities and utilities) that can be used algorithmically given the MDP representation.

One single agent solution to the problem of deriving the agent's initial beliefs over the state space is to simply use domain knowledge to represent the transition probabilities along each dimension of the MDP as some distribution with a given mean and standard deviation. For example, as a first case approximation we can assume that the probability of a user changing location increases with time. Likewise, an agent can form some initial belief over the utility of each state according to some permissible heuristic such as equal utility to all states.

An alternative solution to the belief and utility estimation problem is to use multi-agent mechanisms to specify missing or uncertain user information needed for the agent decision making. For example, collaborative filtering mechanisms have been used extensively to make individual recommendations based on group preferences (Resnick *et al.* 1994; Breese, Heckerman, & Kadie 1998). Similarly, we can use the user preference information from a large number of users to predict state values (for example predicting the perceived quality for a service profile based on the preferences of users with similar quality functions) or transition probabilities (for example likelihood of changing locations). Furthermore, such a mechanism can either be centralized or decentralized. In the former mechanism each PR send its user preference information to a centralized recommendation server (RS). Individual PRs can then query the RS for state information (such as quality estimates of service profiles) and The RS then attempts to identify users with similar quality functions and generates a quality estimate. Alternatively, in a decentralized mechanism (or gossiping/epidemic mechanisms (Pelc 1996; Hedetniemi, Hedetniemi, & Liestman 1988)) each PR communicates not with a center but with a small subset of individuals in a Peer-to-Peer manner. The choice of which mechanism is often dependent on the trade-offs involved in the system properties (such as flexibility, robustness, etc.) and the quality of the information content of the mechanism.

These initial beliefs over transitions and utilities, derived from a multi-agent mechanism, can then be subsequently updated using reinforcement learning. In classic reinforcement learning this is achieved by using the reward signal r to incrementally update the true estimate of the costs from each state to the goal state. Then the PR maximizes the expected reward given the beliefs. However, under the reinforcement mechanism the agent needs to not only know the goal of the user, but the mechanism also requires the goal context to be repeated in time so that the PR can learn the true costs of paths to the goal state in an incremental fashion. Unfortunately, these two assumptions cannot be supported by the service selection problem because of complexity in reasoning about the user goals (since user may not be able to formulate and/or communicate goals) and the low likelihood of user having same repeated goals for the PR to learn from. However, the PR does have access to the utility information at each state. Therefore, rather than using the value of the goal state as the reference point in the optimization problem we instead propose to use the value of each state explicitly.

Cooperative Extensive Games

The above model and tentative solution was framed from the PR problem perspective—how to model the user. We believe the system of both the user and the PR can be modeled as a game, or more precisely as a cooperative bi-lateral, stochastic extensive game between the PR and the user (Rubinstein 1982; Littman 1994). Work to date has looked at how non-cooperative and strategic games can be implemented with a MDP framework (Littman 1994). In these strategic models both players are non-cooperative, each making a single move simultaneously. Furthermore, each agent chooses a strategy from a set of possible strategies that is in (Nash) equilibrium. However, due to errors in execution the outcomes reached are stochastic. Our problem on the other hand is not only regulated by a sequential protocol of user-PR interactions (referred to as extensive games) but also the nature of the game is not adverserial meaning that the agents can agree to a course of action and be mutually committed to that agreement. Therefore the information set of the agent not only reflects the true moves by the user (i.e the user is not being strategic in its interactions with the PR) but also increases at each step of the game. Equilibria of extensive games have been modeled for non-cooperative games (as sub-game perfect equilibrium (Rubinstein 1982). However, the equilibria of these games attempts to remove unlikely threats that players can make at each step of interaction. Therefore, since the user is unlikely to be strategic then other equilibria solutions must be sought to analyze the steady state of the PR-user interaction.

Conclusions and Future Work

In this paper we described a user-modeling problem for the domain of wireless services. An agent, called a Personal Router, was proposed as a solution to this problem. We showed how the nature of the problem bounds the information set of the agent. We then presented a formal model of the service selection problem and showed how it can be captured in an MDP representation, defined by $\langle \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{C}, \mathbf{U} \rangle$, the set of all possible system states, user and PR actions, transitions between states, costs of action and utility of states. We also hypothesized on how we can solve the information problem using multi-agent and reinforcement learning mechanisms.

There are a number of future directions. Firstly, we are currently developing multi-agent and single agent mechanisms for capturing initial values for the parameters of the developed MDP. Next we aim to develop solution strategies that can tractably solve the MDP, for making real-time decisions. Strategies currently under consideration include not only state-space pruning strategies but heuristics for solving piecewise MDPs for time, location and applications. Once the resulting MDP has been solved optimally on a tractable decision space we seek to compare the efficiency of heuristic algorithms that can be scaled up to larger search spaces. Another long term goal is to assess the appropriateness of qualitative information within the developed framework. Finally, the overall goal of this project is to develop system prototypes and perform user-centered field trials.

References

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

Breese, J. S.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, 43–52.

Clark, D., and Wroclawski, J. 2000. The personal router whitepaper. *MIT Technical Report*.

Doyle, J., and Thomason, R. 1999. Background to qualitative decision theory. *AI Magazine* 20(2):55–68.

Hedetniemi, S. M.; Hedetniemi, S. T.; and Liestman, A. 1988. A survey of broadcasting and gossiping in communication networks. *Networks* 18:319–351.

Internet & Telecoms Convergence. 2002. `http://itc.mit.edu`.

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

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

Keeney, R. L., and Raiffa, H. 1980. *Design and Marketing of new products*. New York: Prentice-Hall.

Littman, M. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, 157–163. San Francisco, CA: Morgan Kaufmann.

Pelc, A. 1996. Fault-tolerant broadcasting and gossiping in communication networks. *Networks* 28:143–156.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstorm, P.; and Riedl, J. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, 175–186. Chapel Hill, North Carolina: ACM.

Rubinstein, A. 1982. Perfect equilibrium in a bargaining model. *Econometrica* 50:97–109.

Shenker, S. 1995. Fundemental design issues for the future internet. *IEEE Journal on Selected Areas in Communication* 13(7).

Shoham, Y. 1997. A systematic view of utilities and probabilities. In *International Joint Conference on Artificial Intelligence, IJCAI*, 1324–1329.