Modeling learning and adaptation processes in activity-travel choice: a framework and numerical experiments

Citation for published version (APA):

Arentze, T. A., & Timmermans, H. J. P. (2003). Modeling learning and adaptation processes in activity-travel choice: a framework and numerical experiments. *Transportation*, 30(1), 37-62.
<https://doi.org/10.1023/A:1021290725727>

DOI:

[10.1023/A:1021290725727](https://doi.org/10.1023/A:1021290725727)

Document status and date:

Published: 01/01/2003

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Transportation 30: 37–62, 2003
© 2003 Kluwer Academic Publishers. Printed in the Netherlands.

Modeling learning and adaptation processes in
activity-travel choice
A framework and numerical experiment

THEO ARENTZE* & HARRY TIMMERMANS
Eindhoven University of Technology, Urban Planning Group, P.O. Box 513,
5600 MB Eindhoven, The Netherlands
(*Author for correspondence: E-mail: t.a.arentze@bwk.tue.nl)

Key words: activity-travel choice, adaptive choice, reinforcement learning

Abstract. This paper develops a framework for modeling dynamic choice based on a theory of reinforcement learning and adaptation. According to this theory, individuals develop and continuously adapt choice rules while interacting with their environment. The proposed model framework specifies required components of learning systems including a reward function, incremental action value functions, and action selection methods. Furthermore, the system incorporates an incremental induction method that identifies relevant states based on reward distributions received in the past. The system assumes multi-stage decision making in potentially very large condition spaces and can deal with stochastic, non-stationary, and discontinuous reward functions. A hypothetical case is considered that combines route, destination, and mode choice for an activity under time-varying conditions of the activity schedule and road congestion probabilities. As it turns out, the system is quite robust for parameter settings and has good face validity. We therefore argue that it provides a useful and comprehensive framework for modeling learning and adaptation in the area of activity-travel choice.

1. Introduction

An examination of the literature on activity-based analysis in transportation research suggests a shift from the analysis and modeling of activity-travel *patterns* to the analysis and modeling of the *dynamics* of transport systems and traveler behavior. The problem of how travelers learn in uncertain and constrained environments, and how such learning affects their adaptation offers a new challenge to travel behavior research. Transport systems and urban environments are highly dynamic, non-stationary and uncertain. Moreover, travelers’ information is limited and perhaps imperfect and biased. Consequently, travelers need to make activity-travel decisions based on their day-to-day experiences with the system that result from repeated previous choices. These experiences constitute the ingredients by which travelers learn about their environment and develop adaptation strategies to cope with the dynamic, uncertain, non-stationary system.

The scarce work carried out along these lines to date tends to focus on expectations of travel times related to route choice and departure time decisions. For instance, Horowitz (1984) suggested a weighted sum of previous experiences and a learning error term. Because the implied assumption of unlimited memory is not cognitively plausible, more recent work has adopted either the (exponentially weighted) moving average model (Mannering & Chang 1986; Ben-Akiva et al. 1991; Iida et al. 1992; van Berkum & van der Mede 1993; Axhausen et al. 1995; Emmerink 1996; Nakayama et al. 1999, 2000, 2001; Polak & Oladeinde 2000; Fujii & Kitamura 2000) or a distributed lag model (Polak & Hazelton 1998).

Despite the appeal of these models, they are limited in that they concentrate on single as opposed to more complex, multi-faceted decision-making. Also, the dynamics of the heuristics that underlie activity-travel choice are not considered.

In the present paper, we develop a new theoretical framework for modeling learning and adaptation processes in activity-travel behavior. We build on previous work exploring components of a more comprehensive framework (Timmermans et al. 1999; Arentze & Timmermans 2000). In the following, we will first present the key concepts and their relationships and introduce possible model specifications. Then, we illustrate the performance of a model by means of numerical simulations. We complete by drawing conclusions and discuss avenues of future research.

2. The reinforcement learning approach

We assume that individual choices in complex environments are driven by rules that are formed and continuously adapted through learning while the individual is interacting with the environment. Our model system is based on reinforcement learning theory in machine learning, artificial intelligence, and behavioristic and cognitive psychology. We will not attempt to review this field in any detail. An excellent recent introduction can be found in Sutton and Barto (1998). The purpose of this section is to introduce the concepts underlying the model system elaborated in the next section.

Memory and search play a key role in reinforcement learning. Through search an individual explores choice opportunities in his or her environment and keeps a memory record of the varying rewards associated with the actions. Actions that produce positive rewards are reinforced and have a higher probability of being repeated in future choice situations under similar conditions, while actions with negative outcomes tend to be avoided. A dilemma for any individual that has limited knowledge about current circumstances, for example because he or she has entered a new life cycle (e.g., getting married

or children) or moved to another city or country, is the choice between exploration and exploiting current knowledge. Selecting actions that have not been tried before gives the opportunity of discovering new choices that yield higher rewards than the currently best action. However, this comes with the risk of negative experiences. Individuals who wish to avoid such risks may stick to the currently best choice. The choice for engagement in search needs to be reconsidered each time changes in circumstances reduce the value of current actions or provide new opportunities for alternative choices. Thresholds for reconsidering current choices and the thoroughness of search will vary depending on the individual's tendency to take or avoid such risks.

Reinforcement learning differs from the situation where an individual learns from instructions. In the latter case, the individual is provided with the right answers for a representative sample of cases, and the problem is to discover rules that allow him or her to respond adequately to new cases within the same domain. Much work in artificial intelligence, machine learning, and psychology has focused on the problem of supervised learning and several techniques now exist to model this form of learning. Artificial neural networks and decision rule or tree induction programs are the most important examples. However, the travel and activity choices that individuals make in everyday life are unsupervised. The right actions in a given choice situation is undefined, let alone provided by an instructor. The only information available is feedback on actions in terms of a varying amount of reward. Mimicking the actions or listening to advice from others may introduce some elements of supervised learning though, especially in an exploration phase. But even in such cases the ultimate test of behavior is the reward given by experience. Thus, we assume that reinforcement learning forms the most important basis of dynamic choice that we intend to explain and predict.

Existing systems for modeling reinforcement learning have a number of elements in common. First, the reward function simulates outcomes of implemented actions in terms of a numeric value that indicates the extent to which the outcome satisfies the individual's objectives. Assuming some degree of rational behavior, the goal of the learning agent is to maximize the amount of received reward over some time horizon. Except that it is dynamic, this concept is comparable to a utility function in existing utility-based models of choice behavior. Rewards represent the immediate feedback on an action but may change over time or be subject to random variation. The value function is a second component that models how individuals integrate rewards received in the past to assess the value of an action. Third, a policy defines the choice of an action given a perceived state of the environment and action values. Stimulus-response rules or associations are the typical form of a policy model in behavioristic psychology. Additionally or alternatively, a policy may include complex heuristics for search and problem solving. Finally, some systems

also include a model of the environment for simulating actions to predict their consequences before making a choice.

3. The model system

Existing approaches to specify components of learning systems reflect the diversity of application areas ranging from instrumental conditioning of animals in laboratory settings to highly computation intensive learning machines and rather limited game settings. Considering our focus, only those approaches that can account for the degree of rational behavior and cognitive constraints that characterizes human search, problem solving and learning are relevant. In this section we will propose a model system that combines elements from several fields. Still, the model is a framework in the sense that it leaves open choices of specifications of the components.

To develop the system, this section is structured as follows. First, we consider the basic unit of memory, which we call events. Second, we discuss value functions and policies in the context of action selection decisions. The proposed action-selection methods assume that the system has identified relevant condition states in the environment. The final section discusses the proposed method for inducing condition states from events in memory.

3.1. Events

Any unit of experience that produces some reward and is stored in long-term memory is called an event. For each event, the memory of the system stores attribute information together with the reward value received. Formally, an event is represented as the list

$$e = (x, r, t, w, \rho) \quad (1)$$

where

- x is a vector of values of attributes x_i , $i = 1, \dots, I$
- r is a vector of reward values r_g related to objectives $g = 1, \dots, G$
- t is an index of the time moment when the event took place ($t = 1, 2, 3, \dots$)
- w is a vector of weights w_g related to rewards $g = 1, \dots, G$
- ρ is the retrievability of the event from memory ($\rho > 0$)

The state of memory at time t is represented as a set of events E_t . With respect to the dimensions of events we make the following assumptions.

Attributes

The number of features that an individual could use to describe an event is virtually infinite. Human perception generally focuses on the subset that is believed to be relevant or potentially relevant for predicting rewards of events. Although a complete learning model should incorporate a model of perception, we will currently assume that the selection of attributes is given and invariant across time. Furthermore, we assume that the same attributes are used to describe events of the same class. Events may relate to trips, activities, or even complete activity patterns. Relevant attributes typically include choice facets of the trip, activity or pattern as well as conditions given by the environment and household setting in which the action took place.

Rewards

A reward is a numeric value indicating the intrinsic desirability of an event, much like the utility in conventional choice models. In a previous paper (Arentze & Timmermans 2000) we used a unidimensional reward concept. However, it is reasonable to assume that individuals have multiple objectives and are able to evaluate an event in relation to each objective separately. For example, objectives may relate to the need for the activity and aspects such as ease, comfort, safety, monetary costs, and others.

The individual has no control over the function that generates rewards. Neither is the function open to introspection. In that sense the reward function is part of the environment from the perspective of the actor. In the proposed system, reward functions have the following general form

$$r_g = f_t(\mathbf{x}_g^*) + \epsilon_{tg} \quad g = 1, \dots, G, \quad t = 1, \dots, T \quad (2)$$

where

- f_t is a time-varying and individual-specific function;
- $\mathbf{x}_g^*$ is a vector of event attributes relevant for dimension g .
- ϵ_{tg} is a stochastic error component.

A time-varying function reflects the idea that preferences of the individual or event-reward contingencies in the environment may change over time. Rewards are measured on a continuous scale where $r_{ig} > 0$ represents a positive outcome, $r_{ig} < 0$ a negative outcome and $r_{ig} = 0$ a neutral outcome. The scale of the part-worth rewards may differ across the objective dimensions they relate to. Therefore, the overall reward value is some weighted function of part-worth rewards. We use the star superscript for the attribute variables to indicate that the “true” set of relevant attributes may differ from the set of attributes perceived as relevant by the individual and stored with the event. Moreover, there may be a discrepancy in corresponding attributes as a consequence of perception error.

Weights

The weight vector represents for each dimension g the perceived strength of the association between the event and reward received. It expresses the individual's belief of the extent to which the reward value follows logically from attributes of the event or is atypical. If standardized, weight values vary between zero and one. A zero value represents the belief that the reward value is not systematically related to attributes and a value of one represents the belief that the same reward value would be obtained were the event repeated. In general, the model assumes that this amount of confidence is a decreasing function of the deviation between a predicted and the actual reward

$$w_g = g(\Delta) \quad (3)$$

$$\Delta = |f'_i(\mathbf{x}_g) - (f_i(\mathbf{x}_g^*) + \varepsilon_{ig})| \quad (4)$$

where f'_i represents the individual's mental model for predicting rewards based on perceived relevant attributes $\mathbf{x}_g$ and f_i represents the "true" function, as before. At present, the framework does not incorporate mental models for prediction.

Retrievability value

The memory of an event may decay over time. A retrievability parameter represents the strength of the memory trace and hence the ease with which it can be retrieved from memory. The initial value may differ between events dependent on characteristics of the event such as saliency, primacy, novelty, etc. Limited memory retention capacity is simulated in the system by a parameter that determines rate of decay over time. Each time an event is retrieved the retrievability is reset to the initial maximum value to simulate recovery of the memory trace. Thus, the incremental update retrievability function has the general form

$$\rho_t = f(\rho_{t-1}, e, h) \quad (5)$$

where, as before, e represents the set of attributes of the event and received reward, and h is the time moment at which the event was added to the memory or last retrieved from memory. A possible implementation of this function assumes a constant decay rate as follows:

$$\rho_t = \begin{cases} \rho_0(e) & \text{if } t = h \\ \delta(e)\rho_{t-1} & \text{if } t > h \end{cases} \quad (6)$$

where $(\rho)_0 e$ is the initial value and $0 < \delta(e) \leq 1$ is a memory decay rate both defined as functions of characteristics of the event.

3.2. The problem

At the moment the individual plans an event, part of the attributes may be free to choose, whereas other attributes may be given by previous decisions or the environment. For example, when the individual decides to engage in an activity, the available time for the activity and time of day may be given by previous decisions, whereas the location, travel mode and route may be free to choose. This section introduces the elements of the learning system that will be detailed in subsequent sections. For reasons of simplification, we will consider the case where rewards, r , represent an overall desirability of events aggregated across objective dimensions, g . Furthermore, in the system each event coincides with a time step, so that the index of time t can be used as an index for events as well.

At the decision moment we have the classification of attributes $\mathbf{x} = (\mathbf{c}, \mathbf{a})$ where $\mathbf{c}$ is a vector of given attributes and $\mathbf{a}$ is a vector of attributes that are free to choose. The objective of the learning system can be described as

$$\max_{\{a_1, \dots, a_T\}} R_T = \sum_t^T r_t \quad (7)$$

where T defines a sufficiently long time horizon, r_t is the (aggregated) reward value for the t -th event and $\mathbf{a}_t$ is the vector of chosen levels for the free attributes. An action selection method defines the probability of undertaking action $\mathbf{a}$ when in state $\mathbf{c}$. In pursuing the objective (7), the system selects an action based on the expected reward. We define the value of an action, denoted as $Q(\mathbf{c}, \mathbf{a})$, as the expected return when action $\mathbf{a}$ is chosen in state $\mathbf{c}$. Action values are updated each time a reward is received. Assuming that the choice of an action does not affect the probability of future states, the value update function has the general form

$$Q_{t+1}(\mathbf{c}, \mathbf{a}) = f\{Q_t(\mathbf{c}, \mathbf{a}), w_t, r_t\} \quad (8)$$

Thus, the system incrementally updates the state-dependent values of the concerned action based on the value and weight of a received reward each time an event takes place and, thus, is able to improve the selection of actions as experience accumulates.

In the case of problems where the number of attributes and levels of attributes is small, it would be feasible to initialize and update an action value for each possible combination of levels of attributes. However, in real-world problems, the set of potentially relevant attributes tends to be large. Thus, determining the value of an action based on only those earlier events that fully match the current event in terms of the fixed and free attrib-

utes severely slows down learning. Moreover, considering all attributes to predict rewards is an implausible model. Therefore, the system employs the following reductions of the condition space and action space.

First, the reduction of the condition space is based on the notion that some attributes may be irrelevant or only conditionally relevant for the size of rewards. The system derives from the events stored in memory (E_t) a decision tree that describes the minimum set of attributes (nodes) required to identify the class to which an event belongs. Assuming that the system has no a-priori knowledge about a given domain, the initial tree consists of a root node only and every event is classified into the same single category. Each time an event is added to memory, the system considers splitting branches of the tree to create (more) homogeneous groups of events in terms of rewards. The proposed incremental induction method will be discussed in Section 3.5. Here, the function of the decision tree in the larger system can be established as

$$D_t(\mathbf{c}) = 1, \dots, n_t \quad (9)$$

where n_t is the number of leaf nodes or classes of events. Thus, the decision tree reduces the size of the condition space effectively from the number of unique attribute profiles to a minimum of n_t (composite) states.

Second, to reduce the action space, the system assumes that decisions are made sequentially in a predefined order of free attributes. When a choice is made on the first attribute, it is considered given and fixed for the choice on the second attribute, and so on. Thus, for the k -th decision, $\mathbf{c} = (c_1, \dots, c_m, a_1, \dots, a_{k-1})$ and $\mathbf{a} = (a_k)$ in the definitions of Q_t and D_t above, where m is the number of fixed attributes.

To summarize, learning proceeds through two parallel processes. The first process incrementally updates values $Q_{i(t,k),a(k)}$ representing the expected return when level $a(k)$ of attribute k is chosen in state $i(t,k)$. The second process continuously updates decision trees, $D_{t,k}$, defining states $i(t,k)$ based on fixed attributes.

Finally, it is worth noting that, as implied by the sequential decision process, action values are updated at the choice facet level, whereas feedback is received at the level of the integral event. The simulation study described below will show that even though interactions between choices are not explicitly represented, they tend to emerge from the system operations.

3.3. Action-value functions

This section discusses possible specifications of incremental action update value functions of type (8) for a given action facet k . For clarity of presentation, we drop the subscript k .

There are different ways in which expected rewards can be estimated. The mean reward across events in memory of a given state is the most obvious candidate calculated as

$$Q_{t, i(t), a} = \frac{w_1 r_1 + w_2 r_2 + \dots + w_t r_t}{w_1 + w_2 + \dots + w_t} \quad (10)$$

where as before $r_1, \dots, r_t$ are all the rewards received following events with action a under state $i(t)$ prior to time t and $w_1, \dots, w_t$ are the weights assigned to each outcome (given by (3)). This formula, however, implies a computation intensive procedure since the individual would need to retrieve every event (matching the state) from long-term memory and perform the calculation. Incremental update formulas are employed instead as more realistic. An incremental formula for the mean reads as follows

$$Q_{t+1, i(t), a} = Q_{t, i(t), a} + \frac{w_t}{\sum_{j=1}^t w_j + w_t} (r_t - Q_{t, i(t), a}) \quad (11)$$

This formula requires memory only for $Q_{t, i(t), a}$ and sum $\sum_{j=1}^t w_j$ and simply adds the second term on the right-hand side of the equation each time an event of state $i(t)$ is added. This simulates an adjustment of the state-dependent value of the action based on the received reward.

The mean is only one of several candidate value functions. As the formula makes clear, the adjustment caused by each added reward decreases over time. If the reward function is stationary, this assumption would be rational. However, the dynamic character of environments and preferences generally implies that reward functions should be non-stationary. A better strategy, then, assigns higher weight to more recently received rewards. A widely used parameterized incremental update function based on this principle is

$$Q_{t+1, i(t), a} = Q_{t, i(t), a} + w_t \alpha (r_t - Q_{t, i(t), a}) \quad (12)$$

where α is a step-size parameter and w_t the relative weight of the event. This function applies a constant factor of the difference between the received reward and the existing value rather than a decreasing factor. It can be rewritten as

$$Q_{t+1, i(t), a} = (1 - w_t \alpha) Q_{t, i(t), a} + w_t \alpha r_t \quad (13)$$

This implementation shows that the step-size parameter determines the weight of the new reward value relative to the existing value. Thus, as the parameter approaches one and keeping weights w constant, the value tracks the moving target r closer. Although this strategy should perform better in non-

stationary environments, it is less robust in case of stochastic reward functions. If values of r have a random component, the $\alpha = 1$ value function tracks random movements rather than just the systematic changes over time. The value of the α parameter, therefore, should be in balance with the dynamic and random nature of reward functions.

The value functions considered so far concern alternative estimates of an expected reward value. Individuals may, however, also be risk averse and maximize the minimum reward across events or risk taking and maximize the maximum reward. To support maxmin and maxmax strategies, incremental update value functions representing the minimum or maximum reward received for the action under that state can be defined. Another possibility is to update a measure of variance (such as entropy, for example) of received rewards reflecting an objective to pursue reliable actions (i.e., actions that tend to produce consistently the same reward). These and other value functions do not need to be competing, but may represent multiple criteria of varying importance to the individual dependent on his or her preferred strategy. Therefore, to cover a multitude of possible strategies, the system adds a criterion dimension to action values.

Finally, the values need to be updated not only in response to adding an event, but also when new states are created (by the induction system discussed in Section 3.5). Assume for example that at time t , an existing state i is split into two new (sub)states, say i_1 and i_2 . Then, the system needs to establish new values $Q_{t+1, i_1, a}$ and $Q_{t+1, i_2, a}$ for action alternatives a to represent the new situation. This cannot be handled by an incremental update function, because the split has led to a change in classification of all events currently stored. Therefore, the system applies non-incremental versions of the relevant measures to recalculate the values from scratch based on the relevant subsets of events. Other more heuristic types of methods may be considered such as for example initializing the variables with their current values before the split.

3.4. Action-selection methods

Having defined the value functions, the next question then is how to select actions based on action values. The most straightforward (and commonly used) method is to select the action that maximizes the value function (under the given state). This so-called greedy method, however, fails to satisfy a general need to explore. Alternative actions need to be sampled at least once or once in a while (in case of stochastic or non-stationary reward functions) to make sure that the greedy action is indeed the best action. A well-known extension of the greedy method, therefore, selects a random action every once in a while. The probability of making a random choice is handled by a parameter,

say ϵ . The problem of choosing a level for this parameter demonstrates the earlier mentioned choice dilemma between exploration and exploitation. The more the system explores, the higher the probability of discovering highly rewarding actions. At the same time, however, random choice of actions will also produce negative rewards more often. Different individuals may make different trade-offs. In the system, the choice of the parameter value should reflect the actual tendency to explore by the individual simulated.

An advantage of the ϵ -greedy method is its simplicity. However, it may not represent the best strategy. In the exploration mode, the method assumes equal selection probabilities across alternatives, whereas the chance of selecting a better alternative may be increased by taking the current value distribution across alternatives into account. Methods based on this principle are called softmax action-selection rules. A commonly used softmax method assumes a Boltzmann distribution and selects action a with probability

$$\Pr(a) = \frac{e^{Q_{t,i(t),a}/\tau}}{\sum_{a'} e^{Q_{t,i(t),a'}/\tau}} \quad (14)$$

where τ is a parameter usually called the temperature. The higher the temperature, the more evenly probabilities are distributed across alternatives and, hence, the higher the system's tendency to explore. As the temperature decreases, the system assigns increasingly higher probability to the highest valued action, and, hence, the lower the tendency to explore. The value of the temperature parameter (as well as ϵ) might be a function of time rather than a constant. Then, the system can simulate a tendency to increase exploration in new environments (small t) and decrease this tendency as experience is accumulating.

Furthermore, we note two other possible ways of influencing the system's tendency to explore. First, the choice of the initial value for the value function is relevant. When initial values are set to a high level relative to what can be expected, the system will, even for $\epsilon = 0$ (in the greedy method) or low temperature (in the Boltzmann softmax method), display a high rate of exploration in an early stage and stop searching when all alternatives have been tried or the best alternative tried so far exceeds the initial level. This strategy may give good results (and simulate behavior adequately) especially in stationary, non-stochastic environments. Second, the system may incrementally update an aspiration level (under each state) and switch to an exploration mode each time the currently best alternative (under the concerned state) drops below the aspiration level. For example, a state-dependent expected value of type (11) or (12) may serve as a reference reward. The last method is more flexible in that it takes possible non-stationarity into account.

The selection methods considered so far assume only a single value

dimension. In the previous section, however, we argued that multiple criteria may be considered in some strategies. For instance, individuals may take uncertainty, in the form of a high variability of rewards, into account and display a preference for low-risk alternatives among more or less equally good alternatives. A multitude of alternative multicriteria selection rules might be investigated. At present, the system incorporates a rule that combines the logic of the eliminating-by-aspects heuristic (Tversky 1972) and a need to explore as follows (dropping subscripts i and t):

Focus on the most important criterion first, say Q_1 . Determine $Q^* = \max_a Q_{1,a}$. If Q_1^{ref} represents a reference reward value and $Q^* < Q_1^{\text{ref}}$ then select at random an action that has not been selected under this state before. If every action has been selected at least once, then eliminate the alternatives a' for which $Q_{1,a'} < Q^*$. If there is more than one alternative left, then focus on the second most important criterion, say Q_2 . Determine $Q^* = \max_a Q_{2,a}$ and eliminate every alternative a' for which $Q_{2,a'} < Q^*$. Continue until a single choice has been identified.

This heuristic in effect elaborates the exploitation part of a greedy strategy. To make sure that the system keeps some tendency to explore also after an initial stage, this heuristic is combined with a random mutation probability. Note that mutation differs from random selection (employed by the ϵ -greedy method) in that it assigns zero probability to the current greedy action. It simulates an individual that every now and then changes its currently preferred choice in a random manner.

3.5. *Inducing condition states*

The previous sections assumed that a differentiation in condition states is given at any moment t . The part of the system considered in this section is concerned with deriving and incrementally updating a decision tree $D_t(c)$. This process simulates an individual who learns to discriminate between contexts that tend to produce different rewards (under current action selections). Having distinguished states, the system can explore and exploit action selections dependent on the context and improve the overall result. A proposed induction method that this section discusses assumes a discrete reward variable with reward values ranging from r_1 (maximum reward) to r_n (minimum reward). The problem can then be formulated as finding the (minimum) groupings of events in E_t that maximizes homogeneity of events within groups in terms of reward value. Heuristic methods for solving problems of this type already exist in the form of decision tree induction methods. These methods build a tree by recursively splitting a sample on predictor variables (in our case the fixed

attributes). In each step the best possible split among available attributes is identified based on some split criterion. For example, the program CHAID uses a Chi-square-based measure of significance of difference in distributions between groups as a criterion (Kass 1980). The process of splitting stops if no further significant splits are possible or group size within partitions would become too small according to some user-defined minimum.

Figure 1 gives an example that originates from the case study discussed in the next section. The case concerns mode, route, and destination choice for implementing an activity in the context of an activity schedule where the time of day and the available time for the activity are given. The decision tree relates to the mode choice whereby the route and destination choices are given by previous decisions and, hence, available for splitting. Each path from the root to a leaf node represents a condition state distinguished by the system. For example, the fourth state describes the situation where route-destination is AA1 or AA2, available time is Short and time-of-day is Peak. Between the states the reward distributions differ maximally in terms of a Chi-square-based measure of significance and within states no further splits are possible in terms of this criterion. Only under the fourth state the variance in reward values is completely reduced. The remaining variance under the other states may be random or related to the action variable, which in this case is mode choice.

Figure 1. Example of a decision tree induced from past events.

Incorporated in the system is an incremental update variant of a CHAID-based induction method. Recall that a tree defines a classification of events in memory. At $t = 0$, trees $k = 1, \dots, q$ are initialized with a single root node. Each time an event is added to the memory, the trees are updated. Updating a tree involves classifying the event based on the current tree and evaluating the possibility of splitting the group to which the event was assigned. There are two parameters in the system that determine the height of a threshold for implementing splits. These include the significance level of the Chi-square measure and a minimum group size for each partition. If the threshold is low trees tend to become overly complex, whereas if the threshold is high splits that are significant in reality may be overlooked. Through this, the threshold level has impact on the speed of learning as well as the size of cumulated rewards over time. By manipulating this parameter we can simulate different behavioral tendencies related to inductive learning such as over- and undergeneralizing from experience and observe the impact such tendencies have on learning and choice behavior.

So far, we have assumed that at any moment in time every event in memory is available for induction irrespectively of recency and weight of the event. In reality, however, memory traces of events will fade over time and event weights are differentiated. First, to take weights into account, the system sets a minimum level and includes only events with sufficient weight (i.e., confidence in the reward received) in the calculations. Second, to take limited memory into account, the induction system considers only those events whose memory strength parameter exceeds the given threshold value P . As a consequence, the size of the effective event base is limited. Moreover, in non-stationary environments, distributions of rewards within partitions may change over time. Some condition states that were useful in the past may no longer be relevant in the present. To take such dynamics into account, the induction system should not only evaluate possibilities of splits but also opportunities to simplify the current tree by pruning branches that are no longer significant. Existing pruning methods (e.g., Quinlan 1993; Breiman et al. 1984) are suited for this purpose.

Finally, we point out the role of the t variable, which defines the moment in time the event took place. Consistently, the system considers the t variable as a candidate for splitting in addition to the condition variables. If condition-reward contingencies change at some point in time, the system can resolve arisen variance by splitting condition states on the t variable in a before- and after-change period. Due to the constant decay of memory, the before-change branches of the tree eventually will fade away simulating the process of forgetting how contingencies were like in earlier days. Thus, limited memory retention and availability of the t -variable for splitting provide two mechanisms by which the induction system is able to take non-stationarity into account.

4. Numerical experiments

A series of numerical experiments was conducted based on a hypothetical choice situation. The case considered is relevant for travel demand modeling and concerns the repetitive choice of a destination, transport mode, and route for conducting a given activity under varying conditions of available time, time-of-day, and road congestion. Before discussing the results, this section first briefly describes the implementation of the system and the (hypothetical) data of the case.

4.1. The system implementation

The main loop of the system describes the passage of time in discrete steps. In each step an event is generated and the choice behavior of a given (hypothetical) individual is monitored. This main loop can be written as follows

```

Initialize decision trees  $D_k$  for decision dimensions  $k = 1, \dots, q$ 
Initialize memory:  $E_0 = \emptyset$ 
for  $t = 1$  to  $T$  do
  Initialize a new event  $e_t$ 
  Determine the values of fixed attributes  $\mathbf{c}$  of  $e_t$ 
  For dimensions  $k = 1$  to  $q$ 
    Select action  $a_k$  of  $e_t$ 
    With probability  $\varepsilon$  mutate  $a_k$  random
  Append  $a_k$  to vector  $\mathbf{c}$ 
  Determine reward  $r_t$  and weight  $w_t$  of  $e_t$ 
  Add the event to memory:  $E_t \leftarrow E_{t-1} + e_t$ 
  Update decision trees  $D_k$ 
  Update action values  $Q$ 
  Update the retrievability parameter of events in  $E_t$ 
  If  $t = T_1$  then switch to another reward function

```

The condition state for each new event is determined by a random generator. A stochastic component is also included in the reward function. Furthermore, as the last line indicates, a non-stationary environment is simulated by switching to another reward function at some pre-defined moment T_1 .

4.2. The case

Events simulated for the hypothetical individual concerns an activity of some kind (e.g., shopping) with the following attributes:

Table 1. Reward function.

Available time	L		S									
	AB		AA1		AA2		AB		AA1		AA2	
Route-destination	S1 or F1	F2	S1	F1	F2	S1 or F2	F1	S1 or F1	F2	S1 or F2	F1	
Transport mode												
Time-of-day												
Strong positive	0	0	0	$1 - p_1$	$1 - p_2$	0	0	0	0	0	0	0
Positive	0	0	0	0	0	1	0	0	0	0	$1 - p_1$	$1 - p_2$
Neutral	1	0	0	0	0	0	0	1	0	0	0	0
Negative	0	1	1	p_1	p_2	0	1	0	1	1	p_1	p_2
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
											R13	R14

p_1 probability of congestion during off-peak hours.
 p_2 probability of congestion during peak hours.

1. A marker indicating the moment of a change of some kind that causes changes (in unknown ways) in peak and off-peak road congestion probabilities.
2. Available time for the activity in the given activity schedule with levels “long” (L) and “short” (S).
3. Time of day when the activity is to be conducted with levels “off-peak” (OP) and “peak” (P) period.
4. Combination of route and destination where the activity can be conducted with levels “location near home” (AB), “location farther away via shortest route” (AA1) and same location as AA1 via longer route (AA2).
5. Transport mode to travel to the location with levels “slow” (S1), “car” (F1) and “public transport” (F2).

Attributes 1–3 are given and attributes 4 and 5 are free to choose each time the event takes place. The reward variable is discrete and can take on values ranging between 2 (strong positive) and -1 (negative experience) (Table 1). The underlying notions of this highly discontinuous reward function are as follows. If there is enough time in the schedule, the more distant destination via route AA1 or AA2 yield higher rewards than the nearest destination via route AB. The use of the slow mode for routes AA1 or AA2 generates a negative reward in any case, because then the time left for the activity at the location is too short, even if there is more time available. Public transport mode is generally slower than car and only available on route AA1. Unlike AB and AA2, route AA1 is sometimes congested. When congested, travel time by car is too long and rewards are negative even if the time window is relatively long. Public transport is not sensitive to congestion (e.g., uses a separate lane), and, therefore, provides an alternative on route AA1. Route AA2 takes longer by car, but is not sensitive to congestion. In sum, Table 2 shows the highest reward choices dependent on time available and congestion condition. Because the example assumes that there are no intrinsic preferences regarding route or mode, different modes yield equal rewards under some of the conditions. The system manipulates probabilities of congestion on the AA1 route in off-peak and peak period respectively. The individual has no information of the state of traffic before each event or of the congestion probabilities in general. He may only induce from experience the size of the congestion probability by time of day.

4.3. Results

In all the numerical experiments, the congestion probability of AA1 was set to zero for the off-peak period. For the peak period, however, the probability was set to zero before time step $T_1 = 100$ and to 0.67 after T_1 simulating a

Table 2. Highest reward actions.

Condition		Action		Reward
Available time	Congestion	Route-destination	Mode	
L	No	AA1	F1	+2
	Yes	AA1	F2	+1
	Yes	AA2	F1	+1
S	No	AA1	F1	+1
	Yes	AB	S1 or F1	0

change in the environment at T_1 . This section first considers impacts of various parameter settings on learning and next considers the evolution of choice over time under a typical setting.

Basic settings

As described above, a variety of value functions and action-selection methods is possible and worth exploring. As a first step, the experiments investigate choice under a basic specification. That is to say, the incremental value update function (13) is used in combination with the reference-based action selection heuristic (see section 3.4). Furthermore, the memory decay parameter δ (in 6) is set to one, so that the ability to retrieve events from memory is unlimited. Weights w are set to one for each event too. With regard to the threshold level for splitting condition states, the minimum number of events per state is set to 5 and the significance level to 0.05.

Experiment 1

This experiment considers the impact of action-mutation probability on learning. Recall that this parameter defines the probability of selecting a random alternative for the currently best choice. The α parameter in the incremental update value function (13) was set to 0.2 and the initial and reference reward level to zero. Figure 2 shows the results for mutation probabilities of 0, 0.05, 0.1 and 0.2. The y-axis represents for each time step the mean reward value across the last 20 events. To reveal trends, each run was repeated 5 times and the averages across 5 runs are shown in the figure. Events involving a mutated action (on at least one of the two dimensions) were left out, so that the x-axis shows the non-random choices only, in the following referred to as *structural steps*.

A common trend in all four curves is clearly visible. The mean reward increases very fast and reaches a maximum already after approximately 25 structural time steps. After 75–100 structural time steps (dependent on the

Figure 2. Impact of action mutation probability.

mutation probability) the change is implemented so that from then route AA1 has a probability of 0.67 of being congested. At this point, the mean reward starts to decline and reaches a minimum approximately 40 time steps later. From then mean reward starts again to increase indicating that the system is successfully adapting to the new situation. The mean reward does not reach the same high level as before indicating that the introduction of congestion reduced opportunities to realize preferred choices (in particular, choice of the AA1 route with car).

Aside from this common trend, the graph clearly shows differences in speed of learning and adaptation between the mutation rates. For the relatively high mutation rates of 0.1 and 0.2, discovery of high rewarding choices is much faster and reaches a higher maximum. Also, learning how to adapt to the changed situation is clearly faster and in the end more successful. In the zero case, successful adaptation seems to be absent. Note, however, that under zero adaptation the mean reward would drop below zero in most cases (where the original pattern was AA1-car). So, the fact that the system is able to maintain a positive mean reward indicates effective adaptation. In sum, learning and adaptation occur over the total range of mutation rates, but it is faster and more successful for the higher rates. However, the negative immediate effects of random action are not shown in the graphs where the mutated action steps were left out. The optimum in terms of the overall reward received, therefore, may lie somewhere in between a high and low exploration rate.

Experiment 2

This experiment focuses on the impact of the reference level of reward. Recall that this reference level is used by the system to determine whether or not to start exploring. Each time the value of the currently best action is below

Figure 3. Impact of action value reference level.

the reference value, the system will select a (random) untried alternative under the given state, if any. Again, in this experiment the α parameter in (13) was kept constant at the level of 0.2. Because the reference-value method provides an alternative mechanism for exploration versus exploitation choices, the mutation rate was set to zero.

Figure 3 displays the results for 4 levels of the reference reward ranging from 0 to 2.5. Note that the zero curve in this case is generated under the same settings as the zero curve in the previous case. Furthermore, note that the theoretical maximum action value equals 3.6 given the maximum reward value of 2 and α value of 0.2 (13). As it appears, the impact of reference value is not monotonous. Compared to the zero case, a level of 0.5 gives virtually no improvement, a value of 1.5 a strong improvement, and the 2.5 value a strong deterioration. This holds for learning in the first phase as well as adaptation in the second phase. The maximum reached in both phases are, however, lower than in the high mutation rate case in the previous experiment. To conclude, both a too high and too low reference reward value is counterproductive. A well-chosen reference point improves learning, but does not fully compensate the negative effect of a zero mutation rate. The positive effect of a well-chosen reference point disappears if mutation is re-introduced at a sufficient rate (not shown here).

Experiment 3

In this experiment, we consider the impact of the α parameter (in 13), which determines the relative weight of a new reward in the action value update function. The mutation rate was kept constant at a level of 0.2 and the reference reward value was set to zero.

Figure 4. Impact of step-size parameter, α .

Figure 4 displays results for levels of α varying from 0.2 to 0.8. Included for comparison is also the mean value function (equation 11). As it turns out, learning and adaptation behavior is rather robust for differences on this parameter. In this context, it is relevant to note that the first phase (before change) represents non-stochastic reward conditions and the second phase (after change) stochastic reward conditions in the form of a probability of road congestion. In the first phase differences are less pronounced than in the second phase. In the first phase, only the 0.8 setting clearly impairs learning, but there is no clear optimum level of α . In the second phase, the 0.8 level still performs very poor, but here learning seems to increase monotonically with decreasing levels of α . So, assigning a low weight to new rewards improves learning under stochastic reward conditions. This also explains the relatively good performance of the mean-reward method, where the weight of new events decreases with a constant rate as time passes (events accumulate).

Choice

The course of a mean reward in time gives an indication of how well the system is able to learn and adapt to the environment. In this section we focus on the time path of the underlying choice. Figures 5–7 show the course of choice probabilities (P) for the route-destination and mode dimensions in one representative run. In this run, an α level of 0.2, mutation rate of 0.2, and reference level of 0 were applied assuming that these settings are representative of rational choices. We expect that through learning the system will develop separate courses of behavior under different states determined by the combination of available time and time-of-day. Figure 5 shows the overall

Figure 5. Route-destination and mode choice probabilities: all events.

Figure 6. Route-destination and mode choice probabilities: short-time, peak period events.

Figure 7. Route-destination and mode choice probabilities: long-time, peak period events.

picture across states. As an example, Figures 6 and 7 show choices under the “short time and peak period” state and “long time and peak period” state, respectively. The other two possible states related to the off-peak period are not shown because of limited space.

As it appears from Figure 5, in the phase before the change, the combination AA1 and car is chosen unconditionally, which is rational since this

combination yields the highest reward under every state. At time step 82, the change that causes the 0.67 congestion probability for AA1 during the peak period occurs. At this point choice probabilities become differentiated indicating that the system is trying to adapt to the new situation. The result is that in the last time step, AA1 and AB have choice probabilities of approximately 0.75 and 0.25. At some point, AA2 reached a probability of 0.6 but was later re-substituted with AA1. The mode choice pattern, on the other hand, appears to be closely linked. At first slow mode increases as a substitute of car and later public transport achieves a certain share. In the end (the last step), car is selected with probability of 0.6 and slow and public transport mode both have a share of 0.2.

Monitoring behavior per state provides an explanation for this result. As Figure 6 shows, the AA1 choice is gradually substituted by AB in the “short time – peak period” state. At the same time, the car mode decreases and the slow mode increases until both reach approximately an equal share of 0.5. Apparently, the individual randomly alternates between AB-Slow and AB-Car combinations after adaptation. Recall, that the case indeed assumed that these choices yield the same (and under this condition, maximum attainable) reward. In that sense the substitution of car by slow mode was not needed. The explanation of the decline of the car is that the system at first did not differentiate between peak and off-peak periods so that the value of the car decreased with each negative feedback. A similar effect is even more clearly visible in the off-peak period (Figure 7). In the before-change phase, the AA1-Car combination is chosen with probability 1. Immediately after implementing the change, AA1 is substituted with AB. But, then, after some point the system switches back to AA1. For mode choice, this is accompanied by a substitution of Car with Slow, which is later undone. This behavior indicates that the system discovered the relevance of the distinction between peak and off-peak period only after some time. Before this discovery the AB-Slow combination gave the best overall results. Only after distinguishing the states – peak or non-peak – it was possible to (safely) return to the preferred AA1-Car alternative in off-peak period events.

Besides the impact of learning to discriminate between condition states, this example clearly shows the strong interactions between route-destination and mode choices even though the system makes the decisions in a sequential manner. Comparison of right-hand and left-hand graphs shows that switches in choices between the dimensions tend to be synchronized. Note that this is an emergent property of the system. In making a route-destination choice, the system – by its action-selection mechanism – anticipates the subsequent mode choice, and, vice versa, the system makes mode choices conditionally upon the previous route-destination choice.

5. Conclusions

In this paper, we have proposed a framework for modeling learning and adaptation behavior based on reinforcement learning theory. According to this theory, individuals explore their environment and learn from experience. The proposed framework incorporates methods for updating values of actions, selecting actions and discovering relevant condition states. A series of numerical experiments revealed learning and adaptation of the system under varying parameter settings. As it turns out, the system is rather robust within ranges for the parameters investigated. Learning and adaptation is fast and most successful if the system displays some minimum rate of exploration, chooses realistic reward reference levels and assigns only a modest weight to new rewards in updating action values. For given parameter settings, the system simulates choice behavior through time on multiple choice-facets simultaneously. The ability to discover relevant condition states is a distinct feature of the system and plays a crucial role in the course the behavior will take. Depending on the complexity of the reward function, choice can become highly differentiated between condition states.

Although the case on which the simulations were based is hypothetical, the system is designed to handle the full complexity of real-world situations. It assumes a sequential decision making process so that the dimensionality of choice behavior is not restricted by computational resources. The present case included only three choice facets, but there is no reason to assume that the system would perform less realistically when this is increased to the full set of choice facets at the activity or even activity pattern level. The case study showed that the system is able to reproduce interactions between choice facets despite the sequential nature of the decision making process. On the condition side, the dimensionality of the search space can also be increased. The incremental induction method incorporated does not split up the condition space more than is needed to explain differences in rewards and uses a heuristic to search for relevant condition variables. The modeler does not need to make a pre-selection, but can simply supply the system with all potentially relevant condition variables for the case at hand. Finally, the system is able to operate in stochastic, non-stationary environments.

The proposed system provides a framework for modeling learning and adaptation behavior. In contrast to the dominant modeling approach based on random utility, the present framework takes exploration and adaptation in choice behavior into account. At this stage, we have not put much emphasis on the empirical validity of the component models of the framework. There are many specifications possible for the value function, action selection method and induction methods that make up the system and it is worth while to further develop and empirically test model specifications. In addition, parameters of

these components and the reward function need to be estimated from data. Interactive computer experiments seem to provide the most suited tool for collecting data for estimation. The computer program controlling the experiments should simulate the land-use system, transportation system and traffic for a given study area (for example, a city) in real time as realistically as possible, for example, by using Virtual Reality techniques (VR). A major advantage of VR is that subjects can experience the environment while experimental conditions, for example, in the form of congestion probabilities, opening hours of stores, etc. can be manipulated by the experimenter. The experiments need to be carefully designed, but may closely follow the logic of the learning model. That is, subjects are confronted with a series of events and for each event are asked to perform the activity in virtual reality and indicate how they evaluate the event afterwards. By introducing a change in conditions at some point in time, adaptation can be revealed. Such time series should provide the data points on which parameters can be fitted.

Finally, we point out a possible extension of the framework that is worth considering in future research. The present framework does not incorporate a model of the environment. Yet, individuals have previous knowledge of a domain and use this knowledge to make more informed guesses about alternative choices that may solve an encountered conflict. For example, if a slow mode produced a negative reward because of limited time, the hypothesis that a fast mode may produce better results under that state may direct subsequent search. A model of the environment would allow the system to simulate alternative actions and predict consequences before making a choice. Enhanced with such knowledge, the system would be able to search in a more intelligent way and learn faster.

References

- Arentze TA & Timmermans HJP (2002) An inductive learning approach to evolutionary decision processes in activity scheduling behavior: Theory and numerical experiments. *Transportation Research Record*.
- Axhausen K Dimitrakopoulou E & Dimitripoulos I (1995) Adapting to change: Some evidence from a simple learning model. *Proceedings PTRC* P392: 191–203.
- Ben-Akiva ME, De Palma A & Kaysi I (1991) Dynamic network models and driver information systems. *Transportation Research* 25A: 251–266.
- Breiman L, Friedman JH Olshen RA & Stone CJ (1984) *Classification and Regression Trees*. Belmont, CA: Wadsworth.
- Emmerink RHM (1996) *Information and Pricing in Road Transport*. PhD dissertation, Tinbergen Institute Research Series, Vrije Universiteit, Amsterdam.
- Fujii S & Kitamura R (2000) Anticipated travel time, information acquisition and actual experience: The case of Hanshin Expressway Route Closure, Paper presented at the 79th Annual Meeting of the Transportation Research Board, Washington, DC, USA.
- Horowitz AJ (1984) The stability of stochastic equilibrium in a two-link transportation network. *Transportation Research* 18B: 13–28.

- Iida Y, Akiyama T & Uchida T (1992) Experimental analysis of dynamic route choice behavior. *Transportation Research* 26B: 17–32.
- Kass GV (1980) An exploratory technique for investigating large quantities of categorical data. *Applied Statistics* 29: 119–127.
- Mahmassani HS & Chang G (1986) Experiments with departure time dynamics of urban commuters. *Transportation Research* 20B: 297–320.
- Nakayama S, Kitamura R & Fujii S (1999) Drivers' learning and network behavior: A dynamic analysis of the driver-network system as a complex system. *Transportation Research Record* 1676: 30–36.
- Nakayama S & Kitamura R (2000) A route choice model with inductive learning. Paper presented at the 79th Annual Meeting of Transportation Research Board, Washington DC, USA.
- Nakayama S, Kitamura R & Fujii S (2000) Drivers' route choice heuristics and network behavior: A simulation study using genetic algorithms. Paper presented at the IATBR Meetings, Gold Coast, Australia.
- Osby K, Dattu A & Kachroo P (2001) Modelling route choice behavior using stochastic learning automata. Paper presented at the 80th Annual Meeting of the Transportation Research Board, Washington DC, USA.
- Polak J & Hazelton M (1998) The influence of alternative traveller learning mechanisms on the dynamics of transport systems. *Transportation Planning Methods* 1: 83–95.
- Polak J. & Oladeinde F. (2000) An empirical model of travellers' day-to-day learning in the presence of uncertain travel times. Unpublished manuscript, Imperial College of Science Technology and Medicine.
- Quinlan JR (1993) *C4.5 Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers.
- Sutton RS & Barton AG (1998) *Reinforcement Learning: An Introduction*. London: MIT Press.
- Timmermans HJP, Arentze TA & Joh C-H (2000) Modeling learning and evolutionary adaptation processes in activity settings: Theory and numerical simulations. *Transportation Research Record* 1718: 27–33.
- Tversky A (1972) Elimination by aspects: A theory of choice. *Psychological Review* 79: 281–299.
- Van Berkum E & Van der Mede P (1993) *The Impact of Traffic Information: Dynamics in Route and Departure Choice*. PhD dissertation, Delft University of Technology.

About the authors

Theo Arentze received a MSc degree in cognitive psychology from Groningen University and a PhD in urban planning from Eindhoven University of Technology. He is now an Assistant Professor in the Urban Planning group at the same University. His main fields of expertise and current research interests are decision support systems, activity-based modeling, discrete choice modeling, knowledge discovery and learning-based systems with applications in urban and transport planning.

Harry Timmermans received a PhD in Spatial Sciences from the Catholic University of Nijmegen. He is Chair of the Urban Planning Group, Director of the European Institute of Retailing and Consumer Services and Dean of the Faculty of Architecture, Building and Planning at the Eindhoven University of Technology. His main fields of expertise concern behavioral modeling, consumer studies and computer systems in a variety of application contexts including transportation.