Interactive Refinement of Route Preferences for Driving *

Seth Rogers and **Pat Langley**
(rogers, langley)@rtna.daimlerbenz.com
Daimler-Benz Research and Technology Center
1510 Page Mill Road, Palo Alto, CA 94304-1135

Abstract

Generating satisfactory routes for driving requires data about the road network and an individual's relative weighting of available factors. We describe an interactive planning system that generates routes with the help of a driver and refines its model of the driver's preferences through interaction. Results of a study indicate that it is possible to model drivers through feedback about relative preferences, but a richer description of the road network can improve accuracy. Our adaptive route advisor unobtrusively collects data on preferences in relevant areas, provides its user with a useful service, and improves its performance as it updates its user model.

Introduction

Generating routes for drivers is a challenging problem for several reasons. First, driving occurs in a rich environment where many factors influence the desirability of a particular route. Second, many relevant factors are not currently available in digital form, such as average congestion on streets. Additionally, some factors are virtually impossible to encode, such as whether a route is "scenic." The relative importance of these factors varies among individuals, and drivers may not know themselves what they value most in routes.

In this paper, we report a planning system that flexibly combines a number of factors into a single utility function, which it uses to plan a path from a source node on a road network to a destination node. The system also incorporates a simple heuristic for approximating unobservable factors. The combination function is a simple weighted sum of factors, and the system personalizes the weights to individual drivers based on direct feedback through user interaction.

* Proceedings of the Spring Symposium on Interactive and Mixed-Initiative Decision-Theoretic Systems, Stanford, CA, pp. 109–113, AAAI Technical Report SS-98-03, March 25, 1998, http://pc19.rtna.daimlerbenz.com/~rogers/ss-98.ps.

The planner finds a path to the destination that minimizes the utility function and describes it to the user in terms of a sequence of street names. If the user model is not accurate, the initial route presented to the driver may not be acceptable. In this case, mixed-initiative planning is necessary to further specify the driver's requirements for the planning task. Moreover, the planner can attempt to generalize the interaction that led to a satisfactory route so as to improve the user model for future tasks.

The pages that follow describe our approach in more detail and present the results of an experiment in personalizing the user model. First we describe the planning algorithm and the style of interaction, followed by a description of the adaptation method the system uses to refine the user model. We report on our experiment with human subjects and its results in the next section. The following sections present our approach to handling hidden attributes and outline planned improvements to the system. The final section summarizes our system and describes its relevance to more general planning problems.

The Routing Algorithm

The core of the system is the routing algorithm that plans a path through a digital map from a starting point to a destination. The planner represents the digital map as a graph, where the nodes are intersections and the edges are parts of roads between intersections. The routing algorithm finds a path from a designated source node, usually the current position, to a designated destination. The cost of an edge is the weighted sum of its attributes,

$$c = \sum_i (w_i \cdot a_i).$$

The weight vector plays the role of a user model that defines the relative importance of the attributes. The system uses Dijkstra's shortest path algorithm (Cormen, Leiserson, & Rivest 1990) to find the path with the minimal sum of the costs for each edge in the path.

Our digital maps provide three attributes for each edge: length, driving time, and turn angle to connected edges. The planner refers to these digital maps to minimize the weighted sum of the driving time, length, number of turns, and number of intersections.

In the current implementation, the user enters a starting intersection in terms of the cross street names, a destination intersection, and the weights for each attribute. The weights must be non-negative, because negative edges cause the planner to enter an infinite loop. After computing a path, the planner displays it in terms of the sequence of street names, and the user has the option to replan with different parameters. We have also developed a graphical interface that lets the user define the task on a map, view the results, and drag sliders to set the weights.

Acquiring the User Model

Although weighting each edge attribute creates a flexible cost function for the planner, it is difficult and inconvenient for a user to specify his relative preference for each attribute, especially while driving. Instead, our system induces driver preferences from interaction with the driver. We have implemented a perceptron-style training algorithm (Nilsson 1965) that processes a sequence of interactions with the planner and produces a weight vector that attempts to model the preferences expressed.

We define an interaction with the driver to be the presentation of a pair of generated routes and feedback from the user indicating which route is preferable. We feel that requiring a simple binary choice is a small burden on the user, and it is sufficient for approximating the true user model after many instances. For example, if a user prefers a route that optimizes time over one that optimizes turns, and the current user model prefers the other route, the adaptation method increases the weight for the time attribute and decreases the weight for the turns attribute.

The adaptation method represent routes with a vector $\vec{x}$ containing its four (currently) measurable attributes: estimated total time, total distance, number of turns, number of intersections. With an initial weight vector $\vec{w}$, we estimate the cost of a route to be their linear product, $c = \vec{w} \cdot \vec{x}$. If route $\vec{x}_1$ is rated better than route $\vec{x}_2$ and the cost of $\vec{x}_1$ is lower than $\vec{x}_2$, the weights are consistent and do not need modification. If the cost of $\vec{x}_1$ is higher than $\vec{x}_2$, we apply the perceptron update rule to $\vec{w}$ to decrease the cost of $\vec{x}_1$ and increase the cost of $\vec{x}_2$,

$$\Delta\vec{w} = \eta\vec{x}_2 - \eta\vec{x}_1 = \eta(\vec{x}_2 - \vec{x}_1).$$

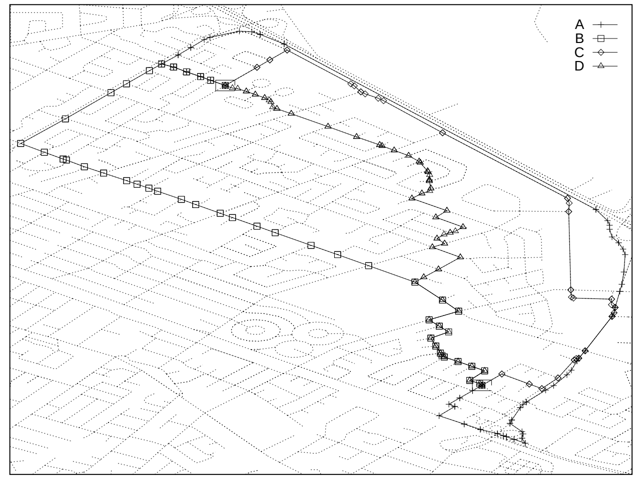

Figure 1: Sample task for the subjects. The starting point is the box at the upper left and the ending point is the box at the lower right. A is the route with fewest turns, B is the fastest route, C is the route with fewest intersections, and D is the shortest route.

For each pass through all available training data, the perceptron adds $\Delta\vec{w}$ to $\vec{w}$ and continues running through the training data until the weights stop changing or it exceeds a certain number of iterations. Although the system could potentially retrain the perceptron on the entire data set after each new training example, the experiment described in the next section trains on a fixed set of examples.

Testing the Adaptation Algorithm

The goals of interacting with the user are to generate a more satisfactory plan and to improve the user model. Although we do not yet have a way to objectively test the first goal, this section presents an experiment testing the second goal. Since the feedback portion of the planner is not yet implemented, we simulated a series of interactions on paper with human subject evaluations of planner output. We implemented the perceptron training method as a separate program. The test consisted of 20 tasks that involved trips between intersections in the Palo Alto area. For each task, we produced four routes using dummy user models with a unit weight for one attribute and zero for the rest, creating routes optimized for time, distance, number of intersections, and number of turns. We plotted the four routes, labeled A through D in random order, on a map of Palo Alto. We presented the tasks in a different random order for each subject. Figure 1 shows an example of one of the tasks and its four route choices.

We asked the subjects to evaluate the routes for each

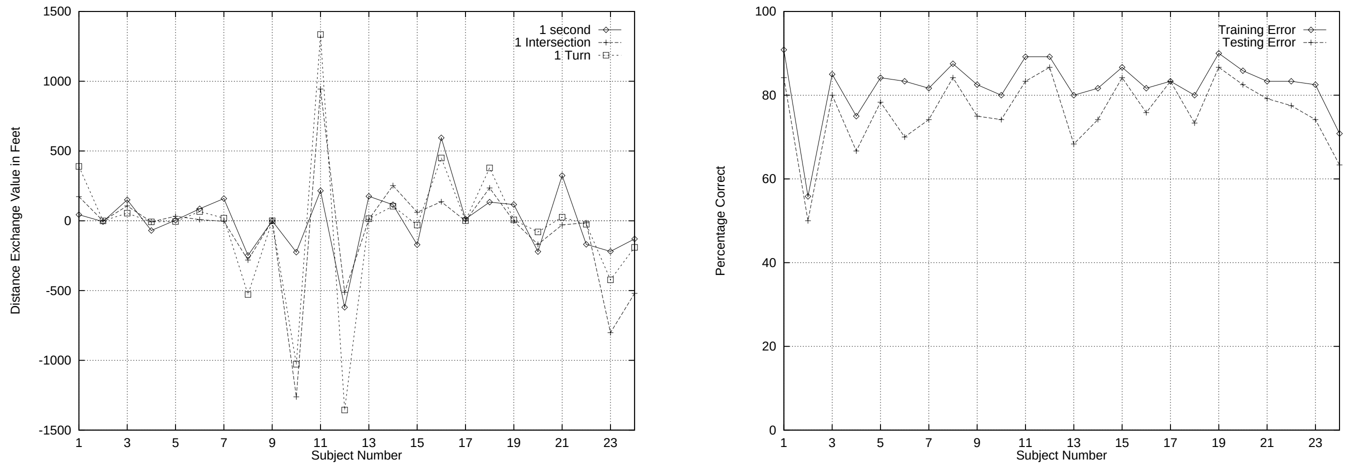

Figure 2: (left) Exchange rates for three of the attributes with respect to distance. High positive values for an attribute indicate that shorter distance is less important than reducing that attribute, near zero values indicate that shorter distance is more important, and high negative values indicate that longer distance is more preferable. (right) Percentage of correct predictions on pairwise route orderings. The training accuracy comes from using the entire set of training instances, and the testing accuracy comes from using ten-fold cross validation.

task and rank them in preference order, using 1 for best and 4 for worst, repeating for each task. Since a ranking of four routes gives six independent binary preferences (A better/worse than B, C, D; B better/worse than C,D; C better/worse than D), each subject provided $6 \cdot 20 = 120$ training instances.

We trained the perceptron for 100,000 epochs ($\eta = 0.001$) for each subject. The resulting user models varied widely. Since the cost of a route is a relative measure, the relative values of the weights is more informative than the absolute values. We will refer to the ratio of two weights as the *exchange rate* between the two attributes. For example, if the exchange rate between time and turn weights is 30, the driver is willing to drive up to 30 seconds longer to save one turn, but no more. Figure 2 (left) shows the exchange rates between distance and the other three attributes.

The results indicate that route preferences differ widely across people. Some subjects, such as 11 and 16, are apparently willing to go to great distances to improve their route on some other attribute. Other subjects, such as 9 and 17, would sacrifice other attributes to reduce the distance attribute. The most surprising results are that many subjects have *negative* exchange rates. For example, the distance/turns exchange rate for Subject 10 is -1027. This means that, given two routes A and B, if route A has one more turn than route B, it will have a lower cost if it is more than 1027 feet *longer* than B. Besides its intuitive contradictions, it is impossible to directly use these weights as a user model for planning because it means some edges could have a negative cost. We believe these negative weights come from the bias in the training data toward optimal routes on some attribute. For example, the fact that drivers prefer shorter routes, other factors being equal, is not explicitly represented in the training data. If we include this as background knowledge, it should eliminate the negative exchange rates.

Figure 2 (right) shows the training accuracy and, using ten-fold cross validation, testing accuracy for the pairwise ordering predictions. The accuracy is uniformly better than chance (50%), but far from perfect. Since the training accuracy is not 100%, the training data must not be linearly separable. Some possible sources for this model failure are that people are inherently inconsistent or use additional, unencoded attributes in their route preferences. For example, people may dislike a certain road or intersection, which affects the rankings for some tasks but not others. Future studies will include additional information about the routes and measure the subjects' consistency on redundant tasks.

Personalized Features

One source of error for the experiment was the sparse and impersonal nature of the route descriptors. Some descriptors are impersonal because they reflect estimates or averages over many individuals, such as the transit time for edges. Using traces from a Global Positioning System to record personal data for individual drivers lets the system personalize some attributes. For example, analysis of individual driving habits provides average speeds on edges taken by that driver.

Positing similar speeds for undriven edges with similar attributes generates a function that estimates the speed difference from the overall traffic averages for each edge. Although not yet implemented, this personalized transit time would replace the current estimated transit time attribute for each edge.

As Haigh and Veloso (1995) note, the descriptor set is sparse because it may not represent all factors relevant to a driver. In fact, there are many features of edges and entire routes intrinsically not representable by digital maps, including features only of importance to individuals. Although extensive interviews with a particular driver are not practical, we can assume that the routes a person drives, in general, are desirable by that person's true internal cost function.[1] The planner uses familiarity information, when available, as an additional attribute for each edge. Since we did not have familiarity information for the subjects, our experiment did not use the familiarity attribute.

With an additional assumption that sequences of familiar edges (subroutes) are more desirable than isolated familiar edges, we have developed a system (Rogers *et al.* 1997) that groups sequences of road edges between commonly used intersections into higher-level links, similar to macro-operators. A macro link between two intersections represents all distinct routes the driver has used between these intersections. Moreover, macro links can incorporate smaller macro links, until the largest macro link is an entire familiar trip between intersections, such as the trip from home to work. Including these macro links adds three properties to the planner: it uses sequences of familiar edges as primitives, it shortens the edge-by-edge description of the route by summarizing familiar sequences, and it biases the route description toward using familiar intersections.

However, the existence of a familiar route between the start and destination does not necessarily force the planner to include it in a plan. Since we represent familiarity as an additional attribute for an edge, the cost of a familiar edge depends on the weight for the familiarity attribute and the weights and values for the other attributes. For example, if familiarity were less important than time to a driver, the planner would prefer a fast, unfamiliar route over a slow, familiar route. Treating familiarity as an attribute gives the planner the flexibility to select familiar routes and edges when appropriate, and unfamiliar edges when they are more desirable on other attributes.

[1]Situations in which this assumption does not hold include cases where the driver is lost, where he is forced to take an undesirable road because it is the only route to his destination, and where he is following directions.

Directions for Future Work

The results of our experiment indicate that it is possible to learn a cost function that predicts driver preferences, although imperfectly. More important, this cost function serves as a user model for generating routes that will be satisfactory to the driver. The system can be made more powerful and useful through work in three key areas: better street descriptions, better interaction and user feedback, and better model induction.

We can improve street descriptions by accessing currently existing geographic databases and generating new geographic databases. Current databases provide information about the types of roads, the location and types of businesses, and demographic information. We will generate new geographic databases by collecting and analyzing a large set of Global Positioning System traces of car trips. Analyzing the trajectories of many cars along the same edge provides average speed models for different times of day, the location of traffic controls, and number of lanes.

We also intend to continue work on integrating the planner with a graphical user interface and use the interactions as feedback to improve the user model. We are planning a highly interactive, mixed-initiative system such as TRAINS (Ferguson, Allen, & Miller 1996). The driver will explore routes already generated while the planner generates more routes in the background. As soon as a new distinct route is finished, the planner will summarize it in terms of total time, distance, turns, and other relevant features. After the driver selects a route, he will display it on a map, view turn-by-turn directions, or expand macro links. He will also request the planner to generate a similar route with more or less emphasis on some attribute. The planner will accomplish this by changing the weight for that attribute and replanning. Since similar weight settings tend to produce the same route, the driver will only be able to increase or decrease an attribute's importance, and the planner will incrementally change the weight until the route itself changes. This interaction will continue until the driver and planner generate a satisfactory route or the driver starts a trip.

Besides letting the driver easily and quickly generate a satisfactory route to a destination, each interaction will provide feedback to the interface. If the driver requests a faster or more familiar route, the driver's personal profile can be updated with more weight on that feature. Unlike the CABINS project (Sycara & Miyashita 1994), which personalized schedules in a job shop by recording schedule repairs, we intend to store preference decisions over route summaries, in terms of total time, distance, number of turns, and other attributes. Also, the interface may automatically expand routes that score high on important features in the driver's profile.

Another form of feedback comes from observing the routes actually driven. If the driver does not take the route the user model predicted, the new route will be presumed to be better than the predicted route, and this will generate a new instance for the personalization module. This type of feedback includes more classification noise than direct feedback because there is no evidence that the driver liked his route or even that the driver was not lost. However, if the driver usually follows routes because of his own true preferences, the noise should cancel out after suitable training time. These indirect forms of feedback are less intrusive on a user than the approach in the experiment reported in this paper or the approach used in the Automated Travel Assistant (Linden, Hanks, & Lesh 1997), where the user explicitly lists his preferences for airlines, airports, and other plan components.

We are also exploring other inductive methods for adapting the user model, such as regression over the preference rankings, multi-layer neural networks, and principal component analysis. Results from any of these methods could improve with some background knowledge about the domain, such as a preference for routes that dominate others on all attributes. Adding more relevant attributes to the street descriptions should increase accuracy, as would more interaction traces from the user interface. We can improve our evaluation by determining the fraction of our modeling errors that are due to driver inconsistency. We will measure this by including some redundancy in the user surveys. Our final goal is a system with a flexible, usable interface that accurately adapts itself to its user over time.

Conclusion

Route planning for automotive domains is a knowledge-rich problem where the criteria for making decisions (attributes of the edges), the relative weight of the features (cost function), and the presentation of routes must be personalized. The ability to interact with the planner allows the system to generate a more satisfactory route than a single plan, and provides an opportunity to receive feedback from the driver reflecting his route and display preferences. Automatic feedback while driving provides personalized data for the digital map and additional training data for improving the user model. Although interaction is in the driver's best interest if he wants a satisfactory route, the system does not require it, and ideally it will become less necessary as the system better approximates the driver's true cost function. This light interaction requirement is crucial for in-car decision making where the driver's attention is necessarily focussed elsewhere.

We believe this approach of automatically acquiring value judgments by observing the user's actions in a domain, while utilizing user interaction as an additional source of value judgments, is a powerful and general method of personalizing a user model. The approach generates an optimal solution using its current user model, receives feedback from the user if its model is inaccurate, and corrects its model in areas relevant to the problem being solved.

Acknowledgments

The authors would like to thank Daniel Russakoff for preparing and running the experiment, and Renée Elio for many helpful comments and discussions.

References

Cormen, T. H.; Leiserson, C. E.; and Rivest, R. L. 1990. *Introduction to Algorithms*. Cambridge, MA: MIT Press.

Ferguson, G.; Allen, J.; and Miller, B. 1996. Trains-95: Towards a mixed-initiative planning assistant. In *Proc. Third Conference on Artificial Intelligence Planning Systems (AIPS-96), Edinburgh, Scotland*, 70–77. ftp://ftp.cs.rochester.edu/pub/papers/ai/96.Ferguson-Allen-Miller.AIPS96.TRAINS-95.ps.gz.

Haigh, K. Z., and Veloso, M. M. 1995. Route planning by analogy. In *Case-Based Reasoning Research and Development, Proceedings of ICCBR-95*. Springer-Verlag. http://www.cs.cmu.edu/afs/cs/user/mmv/www/papers/iccbr-95.ps.gz.

Linden, G.; Hanks, S.; and Lesh, N. 1997. Interactive assesment of user preference models: The automated travel assistant. In *Sixth International Conference on User Modeling*. http://www.cs.washington.edu/homes/glinden/UMPaperFinal.ps.

Nilsson, N. J. 1965. *Learning Machines*. New York: McGraw-Hill.

Rogers, S.; Langley, P.; Johnson, B.; and Liu, A. 1997. Personalization of the automotive information environment. In Engels, R.; Evans, B.; Herrmann, J.; and Verdenius, F., eds., *Proceedings of the workshop on Machine Learning in the real world; Methodological Aspects and Implications*, 28–33. http://pc19.rtna.daimlerbenz.com/~rogers/mlwkshp-97.ps.

Sycara, K., and Miyashita, K. 1994. Case-based acquisition of user preferences for solution improvement in ill-structured domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, WA*, 44–49.