Trip Router with Individualized Preferences (TRIP): Incorporating Personalization into Route Planning

Julia Letchner
University of Washington
letchner@cs.washington.edu

John Krumm
Microsoft Research
jckrumm@microsoft.com

Eric Horvitz
Microsoft Research
horvitz@microsoft.com

Abstract

Popular route planning systems (Windows Live Local, Yahoo! Maps, Google Maps, *etc.*) generate driving directions using a static library of roads and road attributes. They ignore both the time at which a route is to be traveled and, more generally, the preferences of the drivers they serve. We present a set of methods for including driver preferences and time-variant traffic condition estimates in route planning. These methods have been incorporated into a working prototype named TRIP. Using a large database of GPS traces logged by drivers, TRIP learns time-variant traffic speeds for every road in a widespread metropolitan area. It also leverages a driver's past GPS logs when responding to future route queries to produce routes that are more suited to the driver's individual driving preferences. Using experiments with real driving data, we demonstrate that the routes produced by TRIP are measurably closer to those actually chosen by drivers than are the routes produced by routers that use static heuristics.

Introduction

Since the introduction of commercial route planners over a decade ago, people have grown to rely upon them for everything from finding their way to local businesses and friends' houses to planning cross-country road trips. Route planners are available today in cars as well as on the web, where drivers can choose from any of a number of planners, including those associated with Windows Live, Yahoo!, and Google. Although these planners are increasingly reliable in their knowledge of such details as one-way and otherwise quirky streets, they all share the same static-world assumptions. In particular, they are all built around the assumptions of *constancy* and *universality*—respectively, the notions that an optimal route is independent of the time and day of the actual journey and of the detailed preferences of drivers.

In reality, constancy and universality are poor assumptions. Most urban commuters can verify that the best route between work and home at midnight is not necessarily the best route to take between the same locations at, say, 8AM. Similarly, different drivers may choose different routes to carry them between the same start and destination points. While differences in knowledge may play a role in these divergent choices, in many cases drivers simply have different preferences about the types of routes they like to take. For example, one driver may avoid highways or particularly difficult merges, or is willing to extend the duration of her journey by a few minutes in order to follow a scenic coastal road, while another driver simply wants to arrive as quickly as possible or to traverse the shortest distance.

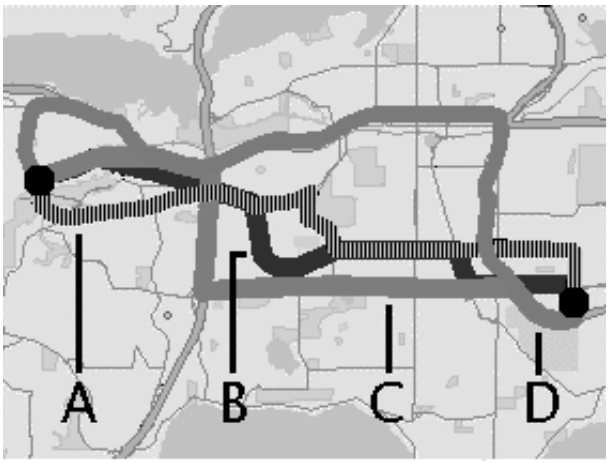


Figure 1: Four routes for a driver's morning commute computed using four different cost metrics. A: Driver's usual route (striped); B: Shortest-distance route; C: Fastest route (according to GPS-based estimates of road speeds, explained later); D: MapPoint's route.

As an example, consider Figure 1, which depicts a real driver's regular morning commute. It follows a route that defies the plans of three route planners that each use different but reasonable metrics, indicating that this driver has preferences, assumptions, and/or knowledge that differs from the knowledge and assumptions implicit in the three planners. In fact, data from our study shows that, on average, drivers take the fastest route for only 35% of journeys. We conclude that a spectrum of factors influences drivers' route choices.

This paper presents methods used in a prototype automated route planner named TRIP (an acronym for Trip Router with Individualized Preferences). TRIP produces route plans that more closely match the routes chosen by people who have extensive experience traveling within a region. The goal of the methods embodied in TRIP is to leverage this knowledge and use it to generalize both about

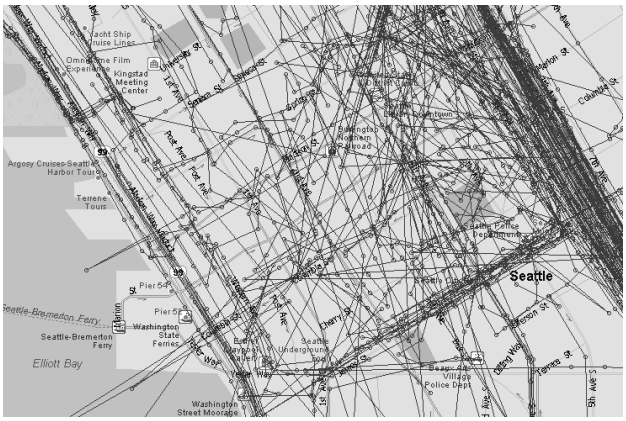


Figure 2: A zoomed view of the MSMLS data set for downtown Seattle. Although the measurements were taken from cars, GPS noise causes most to fall on off-road locations.

traffic conditions in an area and about the subtle preferences of individual drivers, and to use this information to provide higher-quality route suggestions to these drivers.

TRIP improves routing in two ways, the first of which is to incorporate time-variant road speeds learned from large amounts of driver-collected GPS data. To our knowledge, this paper is the first to describe the use of GPS-derived road speeds for routing; however, it is not the first to consider traffic speeds or flows. Handley *et. al.* [1] use contextual features (time/day, traffic flow) to predict the duration of travel along fixed routes, but do not generalize their estimation to the duration or planning of new routes. Fawcett & Robinson [2] estimate road speeds for fine-grained time slices, and like TRIP they use these estimates to provide time-variant optimal routes, but the quality of these speed estimates is unevaluated. Additionally and importantly, both of these related works rely upon a sparse set of fixed-location traffic sensors. TRIP takes a different approach by using as its input a large set of GPS traces collected by individual drivers. This approach is justified by the work of Oda *et. al.* [3], who demonstrate that estimating traffic speeds from real-time, car-attached sensor data (infrared sensors in their case) is greater than 90% accurate.

The second way in which TRIP's routes are improved relative to traditional route planners is that TRIP leverages a driver's route history to learn individual preferences that it then applies to future planning scenarios involving that driver. This paper is not the first to discuss personalized routing, though it is the first to do so with real data. Other work in this area has been confined to case-based approaches, developed as improvements to the efficiency of route planning with only a secondary emphasis on the incorporation of individual preference. A series of papers by McGinty & Smyth [4] introduce the idea (also embraced by TRIP) that preference is difficult to model explicitly; however, their evaluation uses simulated driver preferences. TRIP goes beyond this work by building and validating models based upon real user data; furthermore it does so without requiring any self-reporting of preferences. A similar pair of case-based papers by Haigh *et. al.* [5] uses the idea of "quality" parameters (assigned to individual roads as opposed to TRIP's assignment to users). The evaluation in [5] is based upon users' offline ratings of routes, while TRIP is evaluated on its ability to propose routes that local drivers actually choose.

TRIP extends beyond the scope of prior work by incorporating non-simulated user-preferences into routing, and by learning these preferences from GPS histories of individuals' actual driving behaviors. It also presents new approaches to the estimation of time- and day-specific road speeds from GPS logs, and for the leveraging of these dynamic speeds in the planning of routes. These extra considerations result in route recommendations that are superior to those provided by traditional route planners.

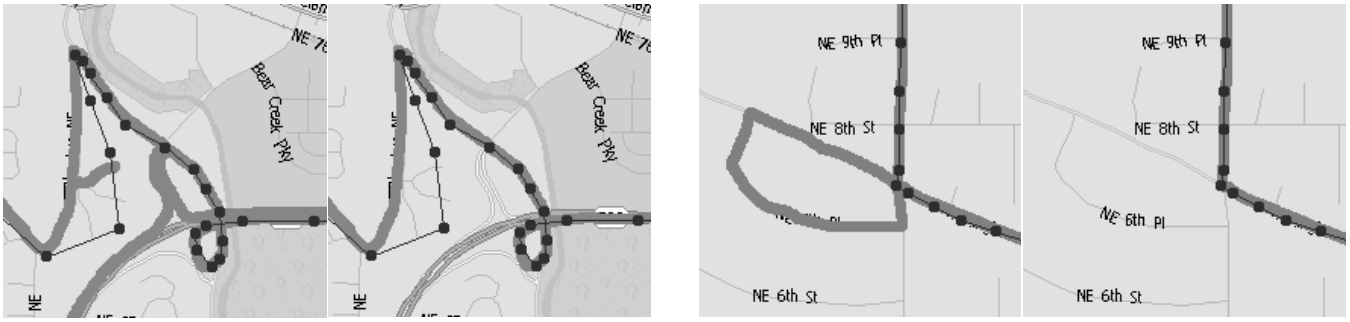
Data & Representations

The route planning work presented in this paper was enabled by the availability of a new repository of GPS traces collected from over 100 people driving within the Seattle metropolitan area. We introduce the data and explain our map representation here to give the reader a sense of the scope of this project, and also to motivate the following section addressing the subtle but challenging task of segmenting and aligning the raw GPS into sequences that are useful for reasoning about route planning.

GPS Data Set: The MSMLS Corpus

The Microsoft Multi-Person Location Survey (MSMLS) data[6], which provides the basis for the analyses in this paper, is a series of GPS data logs (Figure 2). Forty GPS devices were used to collect time-stamped (latitude, longitude) coordinates. Each device was placed for a two-week period on the car dashboard of a different, consenting Seattle-area resident. This process was repeated with new subjects over several collection periods, resulting in two-week-long driving traces of 102 different individuals. Importantly, these drivers were not asked to alter their driving behavior in any way during the study. The GPS receivers were configured to record only while the vehicle was moving, so the drivers did not have to remember to turn the GPS on or off or attend to it in any way over the two weeks. Each driver's data is therefore a snapshot of his or her natural driving routines.

During data collection, the GPS devices stopped recording after several minutes of immobility (*e.g.* when the car was parked) or several minutes after satellite signals were lost (*e.g.* when the car was in a garage). Recording continued again when normal travel conditions resumed. The result is that most—though not all—of the individual journeys contained in the two week logging period can be identified by the time gaps that appear in the GPS logs at each endpoint. Segmentation of the small



Naïve HMM Naïve HMM

(a) (b)

Figure 3: Two comparisons of naïve vs. HMM snapping. In (a), the incorrect snapping of a measurement onto a one-way exit ramp forces the naïve method to infer an extra loop onto/off the highway. In (b), a small amount of noise again forces the naïve method to incorrectly infer a loop.

number of more complex cases is discussed in the following section. After segmentation, the data set's 288,021 individual GPS measurements resulted in 2,517 separate journeys with a total mileage of 18,853.

Map Representation: MapPoint

TRIP represents road networks in the form of a graph: nodes in the graph represent intersections of roads, and graph edges are the roads themselves. Representation of a single road or highway often requires many edges, since each *road segment* (the smallest unbroken portion of a road between two intersections) is a separate edge in the graph. An alpha version of Microsoft's MapPoint software, on top of which TRIP is built, provides road networks in this graph format and additionally provides the physical geometry of each road/edge in (latitude, longitude) coordinates.

Snapping & Segmenting GPS Traces

In order to leverage the knowledge of road networks exposed by MapPoint when reasoning about the traces in the MSMLS data set, TRIP must identify individual trips within the traces and infer the sequence of roads that a driver traversed on each journey. Only when the GPS traces are described in terms of routes on the road network can they be compared and the information they contain be aggregated and leveraged.

Journey Segmentation

As mentioned previously, each two-week GPS trace is split into separate journeys during a preprocessing phase. Locations at which a car remained for more than 5 minutes are considered to be destinations, and the GPS trace is segmented at these locations into separate journeys.

We note that some destinations, however, go undetected by this destination-identification criterion. For example, the procedure misses locations at which a driver stops for only a few seconds to drop off a passenger.

Detection of such "drive-by" destinations is desirable, since they can significantly alter TRIP's concept of a best route; however, such destinations are extremely difficult to detect since Seattle traffic, like traffic in many other major cities, is often slowed significantly by congestion. The GPS measurements recorded by a car stuck in slow traffic are generally indistinguishable from those generated by a driver slowing or stopping briefly to run an errand. However, there are some instances in which detection is possible; namely, when the driver's route makes a loop. If the on-road route generated from a snapped GPS trace contains a loop, then TRIP segments the trace into two separate journeys. The segmentation point is chosen to be the point in the loop that is physically farthest away from the point where the loop closes.

Snapping To Roads

The task of aligning a raw GPS sequence to a set of corresponding road segments is nontrivial, and will likely be a challenge to others working on related methods. Before moving on, we pause to describe our method for snapping GPS data onto a road network.

The largest problem with mapping GPS readings onto road maps is that GPS data is noisy. We have found that the error in our GPS signals can be reasonably modeled as a zero-mean Gaussian with a standard deviation of 10 meters. Errors increase significantly beyond the expected ranges, however, in urban areas and under bridges or in tunnels where GPS satellite coverage is sparse. Additionally, policies embedded in specific GPS systems can bias data; for example, the GPS receivers used to collect the MSMLS data assume linear trajectories during moments when GPS satellite signals are lost.

A secondary problem with the mapping of GPS onto roads is that map representations themselves are imperfect; two different maps will pinpoint a particular latitude/longitude position at slightly different locations relative to surrounding features in each representation. The combined result of these various sources of error is that, despite having been collected entirely from on-road

locations, nearly all GPS points in our data set fall onto off-road map zones, *e.g.* the GPS points in Figure 3a.

In the best of circumstances, the noise problem is straightforward to fix: each GPS point can be "snapped" to the on-road location nearest to it, and the overall GPS trace can be reconstructed by connecting these on-road locations together via the shortest on-road paths between each consecutive pair. Unfortunately, the level of noise is too high and the network of Seattle roads too dense to make this naïve solution feasible (Figure 3).

TRIP solves the snapping problem with a Hidden Markov Model [7]. Informally, the HMM considers many potential snaps for each raw GPS observation o_i, and selects the best on-road snap s_i for each such that the resulting sequence of on-road locations is as smooth as possible, while still maintaining proximity between the raw and snapped location of each measurement. The set of possible snap locations for a single raw GPS point o_i is created by collecting, for each road segment within 150 meters of o_i, the single location s_i on each distinct road segment that is closest to o_i. This results in many candidate s_i locations in areas where the road network is dense, but a small number of candidates in areas with sparse road coverage.

Formally, the HMM defines the following joint probability over sequences of raw and snapped GPS locations (O and S, respectively), from which the maximum-probability sequence of snapped locations for a given GPS log can be deduced using the standard Viterbi algorithm (see Rabiner [7] for details),

$$P(S,O) = \prod_i P(o_i|s_i)P(s_i|s_{i-1})$$

The first term in the product is the observation probability: how likely would the observation o_i have been if the GPS receiver had actually been at on-road location s_i? This probability is given by the error characteristics of our GPS receivers, modeled as a zero-mean Gaussian with a standard deviation of 10 meters.

The second term in the product is the transition probability: how likely would the on-road location s_i have been if the GPS device was known to be at location s_{i-1} when the previous measurement was recorded? We define this as the fraction a/b where a is the straight-line distance from s_{i-1} to s_i and b is the length of the shortest on-road path between the same. This definition penalizes transitions between points that are physically close but not directly connected by roads, reflecting the fact that because measurements are recorded so frequently (at least every 6 seconds), long or roundabout paths driven between two measurements are unlikely. Transitions between distant snap locations are also penalized, since generally there is no on-road, straight-line path between them. We note that $a/b < 1$, and we normalize such that the transition probabilities emanating from each node of the HMM sum to one.

Two alternative models of transition probability include the use of fixed values (one for same-road transitions and another for transitions spanning intersections); or the assignment to each transition of a value inversely proportional to the number of road intersections spanned by the transition. In experimenting with these approaches on our data, we found both to be overly sensitive to the values/functions chosen to define the probabilities, providing either too little or too much smoothing of the GPS trajectory. Neither approach worked as well on our data as did the solution described above.

Unfortunately, ground truth is not available for any traces in the MSMLS data set; thus we cannot validate the success of the HMM snapping approach on this data. We did, however, examine the snapped routes computed by the HMM for a subset (~10%) of the traces. For these traces, the HMM solution always matched the path inferred by the human observer, while the naïve snapping solution matched only for a very small number of very short traces (Figure 3).

With the GPS data converted to sequences of road segments, we could now characterize routes from our data and compare actual routes to routes planned by a conventional router and those planned by TRIP.

Improving Routing Through Experience

Any driver knows that particular roads or routes may take longer to traverse at rush hour than they do at noon or midnight. Similarly, drivers expect that rush hour traffic patterns will repeat each weekday, but they expect entirely different traffic, if any, on the weekend. This dynamic aspect of road speeds is ignored by traditional route planners, which answer queries without regard to the time at which the resulting journey is expected to take place. Existing routers similarly ignore the identity of the driver for whom the route is intended, providing the same route for everyone when in fact some users might prefer routes that bias for or against the use of highways, scenic roads, *etc.* We describe TRIP's approach to both of these problems in this section.

Predicting Travel Time

Ideally, a route planner would receive its road speed information from real-time traffic sensors placed physically across a region. At the very least, incorporation of real-time traffic data would allow routers to better predict the duration of the routes they propose; at best, the routers could propose different routes under different conditions, to better circumvent delays. Unfortunately, most cities are not yet equipped with such sensors at an appropriately dense level of coverage.

Even if such data were uniformly available, however, using its real-time values as the basis for route planning presumes that driving conditions at the time of a user's journey match those at the time of the query. Many hours or even days can pass between the query and traversal of a route, so even planners with highly sophisticated real-time input must be able to plan routes starting at any time.

Specifically, an intelligent router must be able to plan routes starting at times in the future for which exact traffic conditions are not yet known.

TRIP provides such routes by learning time-dependent traffic speeds for roads (Figure 4). It breaks day-of-week into two categories: weekday and weekend. Both categories are further broken down into 96 time slices: 15-minute chunks of time covering all 24 hours of the day. For each road segment in the system, TRIP learns a separate average speed for each time-of-day and weekday/weekend breakdown. It does so by examining each pair (A, B) of consecutive GPS points in our snapped traces. The average speed of the driver between each pair is easily calculated, and the speed added to a running average for every road segment traversed to get from A to B. Speed measurements are applied to the running average associated with the time chunk whose time features match those of the GPS timestamps involved in the speed calculation.

Of course, even the most frequently traversed roads in our data set are not traversed during every time slice. For road segments and time segments where no data is available, the speed calculated for the same road at an adjacent time slice is used. If neither adjacent time slice contains data, TRIP estimates the segment's speed from the system-wide average of the speed of drivers at the given time on all other "similar" roads, where similarity is defined by road class (this is in turn defined by MapPoint, which identifies classes such as highway, arterial, on-ramp, *etc*.). The speed of road segments at times for which even a system-wide average is unavailable is taken simply to be its speed limit.

Incorporating Driver Preferences

Route planning decisions that vary based on expectations of road speeds at different times will produce routes that are, in expectation, faster to traverse. Nevertheless, drivers are not necessarily concerned only with speed; their utility functions may involve other variables. Ideally, a route planner should incorporate these variables into its planning so that the personalized routes it proposes can maximize the implicit utility function of each driver.

One approach to doing this is to explicitly identify the space over which preferences can range. A planner might then model preferences for avoiding highways, minimizing turns, or favoring scenic roads. Such an approach would require the explicit identification of variables affecting preference as well as learning, for each driver, the driver's utilities as a function of these variables.

As a simpler approach, TRIP instead learns and manipulates preference implicitly. TRIP does not model factors affecting preference (e.g. road quality, scenic value, etc.). Instead, TRIP treats each journey in a driver's trace set as a statement of preference. In particular, it assumes that the route a driver actually takes is preferred by that driver over any other route he could have taken between the same endpoints. As the drivers in our set are all local residents driving familiar areas, we believe that the number of routes for which this assumption does not hold is very small.

As a step toward characterizing a driver's implicit preferences, TRIP examines each of the driver's traces in turn and calculates its *inefficiency ratio* r—the ratio of the duration of the fastest route (in expectation) between the trace's endpoints, as determined by our own A* route planner relying upon the time-varying road speeds discussed previously, and the actual duration of the user's trip. Thus r's value is always a fraction between 0 and 1; in the rare cases where a driver's actual time was smaller than the expected-fastest time, we cap r at 1.0. The meaning of r is most easily understood in terms of its inverse, which is a value between 1.0 and infinity and represents the proportion of time by which a driver has extended his/her journey beyond the shortest possible time in order to satisfy preferences unrelated to efficiency.

For each driver, TRIP calculates a personal inefficiency parameter $\bar{r}$ by averaging the individual r values computed from each of the driver's GPS traces. Like the r values, $\bar{r}$ is always between 0 and 1. A value of 1.0 indicates that the driver generally takes the most efficient route, while lower values imply a higher willingness to sacrifice efficiency for other preferences.

TRIP uses $\bar{r}$ in the following utility function defining the driver-specific cost of traversing a particular road segment i:

$$c_i = \begin{cases} \bar{r}t_i \text{ if } i \text{ previously traversed} \\ t_i \text{ if } i \text{ not previously traversed} \end{cases}$$

where t_i is the estimated time required to traverse of segment i (dependent upon the time of travel). The result is that a path using only non-traversed (non-preferred) edges and requiring x seconds to traverse is equivalent in cost to a path using only preferred (previously traversed) edges and requiring $(1/\bar{r})x$ seconds to traverse (recall that $\bar{r} < 1$), since the discount of $\bar{r}$ to the latter path will result in a cost of $\bar{r}(1/\bar{r})x = x$. This equivalence is consistent with our assertion that the user is willing to extend the duration of his trip by up to a factor of $1/\bar{r}$ in order to satisfy preferences not related to efficiency. The cost function also reflects the interpretation of driver histories as statements of preference by allowing TRIP to reduce the cost of previously-used (preferred) edges.

Two major assumptions inherent in this approach to preference modeling are as follows:

1) Drivers in the MSMLS data set are making informed choices; they are not extending the duration of their journeys out of ignorance.
2) Drivers prefer roads that they have taken before.

Although these assumptions were used to guide the development of TRIP and are used in this section as a motivating story in the description of TRIP's process, we recognize that they may not necessarily be true for all trips. In particular, the difference between routes taken out of preference and those taken out of ignorance (*e.g*. a driver did not know that a different route was faster) is indistinguishable in our data set. Nevertheless, even with

these potential problems, the results in the next section demonstrate that routes generated from these assumptions can accurately match routes that drivers themselves would choose.

Experimental Results

In this section we present an experiment in which the routes generated by TRIP are compared with the routes actually taken by drivers in the data set, via a take-one-out cross-validation. One at a time, each individual journey in the MSMLS set was removed. Using the remaining traces, TRIP calculated the driver's inefficiency ratio according to the process described previously. TRIP was then queried for a route whose endpoints matched the endpoints of the removed trace, and the trial was considered a success if the proposed route matched the route actually taken by the driver. A match is defined as overlap between 95% of the distance in both routes. In all trials, the removed route was reinserted into the set before the next trial.

Over all traces in the MSMLS set, TRIP's proposed route matched the driver's actual route in nearly half—46.6%—of cases. To put this in context, consider the fact that only 34.5% of actual driver routes follow the path that, according to the aggregated road speed data, is the fastest; the incorporation of driver preference improves performance. Furthermore only 30% of the actual driver traces match the routes suggested by a traditional (static) fastest-route planner, demonstrating that the use of dynamic road speeds together with driver preferences improves performance well over that which can be achieved by traditional planners.

Another interesting contextual note is the fact that only 10.8% of routes in the MSMLS data are duplicates, where a duplicate is defined as a trace sharing the same driver, start point, and end point—regardless of whether the route taken between the two endpoints is the same. This means that on 35.8% of the test trips, TRIP was constructing *new* optimal routes (routes not seen in the training data) by piecing together preferences from a set of independent training journeys. Thus we see that TRIP can compute routes preferred by a driver without having ever seen an instance of the particular route being queried.

Conclusions

TRIP is a route planner that uses real-world GPS data to estimate both time-dependent road speeds and individual driver preferences. We demonstrated through experiments that by using route planning methods that include these two dimensions, we can generate routes that are significantly closer to those chosen by local drivers than are the routes produced by traditional, static planners.

We also believe that the techniques applied in TRIP are a simple and effective approach to integrating personalization into route planning. Fielding TRIP's methodology is quickly becoming feasible as the accuracy and affordability of GPS sensors makes them increasingly ubiquitous, and as the popularity of in-car navigation systems grows.

Directions for future work include development of solutions for drivers with little or no driving data, via automatic clustering of drivers. This would allow users with sparse data to identify with other users (perhaps via brief online entry of a few often-traversed routes), thereby allowing TRIP to use the data of other, similar users to guide its recommendations. Alternatively, routers could generalize about classes of roads instead of individual road segments, allowing personalized help to be provided in areas—and even cities—where a driver has not yet traveled.

Additional research directions include addressing the problem scenario in which drivers have collected so much GPS data that every road (in the hypothetical limit) has been traversed. In this case, TRIP's current system degenerates into a fastest-route planner. Two potential approaches to this problem include intelligent pruning of the training data, or the inclusion of the frequency of traversal into the edge discount policy.

We believe that route planners stand to benefit from the current technological trend toward personalization. We have provided an initial approach to such personalization, leveraging a large corpus of trips in a metropolitan area.

References

1. Handley, S., P. Langley, and F.A. Rauscher. *Learning to Predict the Duration of an Automobile Trip*. in *Fourth International Conference on Knowledge Discovery and Data Mining*. 1998. New York: AAAI Press.
2. Fawcett, J. and P. Robinson, *Adaptive Routing for Road Traffic*. IEEE Computer Graphics and Applications, 2000. **20**(3): p. 46-53.
3. Oda, T., et al. *Evaluation of Measured Travel Time Utilizing Two-way Communication in UTMS*. in *Third World Congress on Intelligent Transport Systems*. 1996. Orlando, Florida.
4. McGinty, L. and B. Smyth. *Turas: A Personalised Route Planning System*. in *Sixth Pacific Rim International Conference on AI, (PRICAI'00)*. 2000. Melbourne, Australia: Springer-Verlag.
5. Haigh, K.Z., J.R. Shewchuk, and M.M. Veloso, *Exploiting Domain Geometry in Analogical Route Planning*. Journal of Experimental and Theoretical AI, 1997. **9**(4): p. 509-541.
6. Krumm, J. and E. Horvitz, *The Microsoft Multiperson Location Survey*. Microsoft Research Technical Report MSR-TR-2005-103, August 2005, Microsoft Research.
7. Rabiner, L.R., *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE, 1989. **77**(2): p. 257-286.