INFESSION OF PUBLIC TRANSPORTATION TRIP DESTINATIONS USING FARE TRANSACTION AND VEHICLE LOCATION DATA: A DYNAMIC PROGRAMMING APPROACH

Gabriel E. Sánchez-Martínez
Massachusetts Institute of Technology
77 Massachusetts Ave. Room 1-235
Cambridge, MA 02139
Tel: 617-258-0699 Email: gsanmar@mit.edu

Word count: 5,197 words text + 4 tables/figures x 250 words (each) = 6,197 words

Submitted on August 1, 2016
Revised on November 14, 2016
ABSTRACT

Origin-destination matrices provide vital information for service planning, operations planning, and performance measurement of public transportation systems. In recent years there have been methodological advances for estimating origin-destination matrices from disaggregate fare transaction and vehicle location data. Unlike manual origin-destination surveys, these methods provide nearly complete spatial and temporal coverage at minimal marginal cost. Early models infer destinations based on the proximity of possible destinations to the next origin, disregarding the effect of waiting time, in-vehicle time, and number of transfers on path choice. This research formulates a dynamic programming model that infers destinations of public transportation trips based on a generalized disutility minimization objective. The model infers paths and transfers on multi-leg journeys and works on systems serving a mix of gated stations and ungated stops. It is being used to infer destinations of public transportation trips in Boston, producing better results than could be obtained with earlier models.
INTRODUCTION

Origin-destination matrices provide vital information for service planning, operations planning, and performance measurement of public transportation systems (1). The structure of demand in a network is a key driver of the planning of lines and corridors of service. Demand intensity drives the choice of mode, vehicle and infrastructure design, and frequency of service. In the context of an existing network, origin-destination matrices are relevant to quantifying performance in terms of crowding, reliability, journey times, and spatiotemporal distribution of demand.

Traditionally, origin-destination matrices could only be estimated through costly surveys, administered by handing out cards coded by location to each boarding passenger, collecting the cards from each alighting passenger, and sorting them by alighting location. Origin-destination matrices can also be estimated from counts, through methods such as bi-proportional fitting, although accuracy depends on the a-priori unscaled origin-destination matrix. The advent of automated data collection systems, particularly in the areas of fare collection, vehicle tracking, and passenger counting, have made it possible to estimate origin-destination matrices through inference models applied to nearly the full demand population, distributed across times and locations, and with minimal marginal cost.

While destinations are simply observed in systems requiring passengers to tap in and out, such as the subway system in London, they are unknown in systems requiring only tapping in. Many rail systems and most bus systems fall into the latter group. In recent years there have been methodological advances for estimating origin-destination matrices from disaggregate fare transaction and vehicle location data. These models infer origins by matching timestamps and identifiers in fare transaction records with those in vehicle and faregate location records, and they infer destinations by assuming they are close to (if not exactly at) the next recorded location. Barry et. al. (2) infer destinations of subway trips in New York City, and validate two key assumptions using travel diary surveys: passengers tend to return to the destination station of their previous trip to begin their next trip, and they tend to end their last trip of the day where they begin their first trip of the day. Zhao et. al. (3) extend this to capture bus location data when inferring rail trip destinations, for cases in which passengers make bus trips after rail trips. Trépanier et. al. (4) present a model to infer destinations of bus trips, and introduce the possibility of considering transactions of the following day or weekly travel patterns to complete missing information. Farzin (5) describes similar work in the context of São Paulo’s bus network, which requires integrating bus GPS data. Barry et. al. (6) infer destinations of both rail and bus trips in New York City, without making use of bus location data. They employ a shortest path algorithm to estimate rail destination times after using heuristics to infer destination locations. Wang et. al. (7) apply similar trip chaining heuristics to infer bus destination locations and times, using both fare transaction and vehicle location data, and they validate inference results with manually collected origin-destination survey data. Li et. al. (8) do similar work to infer bus trip destinations in Jinan, China, and validate results with boarding and alighting counts of a bus route, demonstrating that the inferences are reliable. Munizaga and Palma (9) do similar work in the context of Santiago de Chile, inferring destinations for 80% of boarding transactions in both bus and rail. For bus, they propose minimizing generalized time rather than distance to avoid inferring that passengers take a long tour to avoid a short walk. For rail, they infer destinations by minimizing distance, and then use a shortest-path algorithm to infer the destination time. Gordon et. al. (10) apply trip chaining heuristics to infer bus origins and destinations, and then infer transfers in the multimodal transit network, with a full scale application in London. Transport for London has since implemented their model to make inferences on a daily basis, which has significantly enhanced analysis capabilities. Dumas (11)
adapts Gordon’s model to make inferences in Boston’s public transit network, in which there are
no exit fare transactions in rail, and demonstrates the use of inferred origin-destination data for
equity analysis.

Early models infer destinations based on the proximity of possible destinations to the
next origin location, disregarding other journey components that contribute disutility and affect
path choice, such as waiting time, in-vehicle time, and number of transfers. Some use shortest
path algorithms to infer destination times, but not destination locations. The inference model by
Munizaga and Palma is the only one that captures the combined disutility of in-vehicle time and
walking, but only for bus trips. This research formulates a dynamic programming model
capturing generalized costs of bus and rail trips. It is the first model that infers paths and
transfers on multi-leg journeys and works on systems with vehicles serving a mix of gated
stations and ungated stops. It is already being used to infer destinations of public transportation
trips in Boston, producing better results than could be obtained with earlier models.

The next section presents a framework for inferring origins, destinations, and transfers
based on disaggregate fare transaction and vehicle location data. Following this, this paper
presents the formulation of the dynamic programming algorithm for inferring destinations based
on generalized disutility, and then describes the application of the framework and model to
Boston. The last section summarizes findings and lists opportunities for future research.

FRAMEWORK

The process of inferring trip patterns from disaggregate fare transaction and vehicle location data
can be divided into three steps: origin inference, destination inference, and transfer (i.e.
interchange) inference (10). Gordon coined the acronym ODX, which refers to the whole
process with a letter of each of the steps.

Origins are inferred by matching fare transactions to locations through transaction
timestamps and card reader identifiers. When the card reader is installed at a fixed location, such
as a faregate in a station, the reader’s location is simply assigned to the transaction. When the
reader is installed in a vehicle, the transaction timestamp is compared with vehicle location data
to determine boarding stops. Erroneous or incomplete vehicle location data may lead to
uninferred or incorrectly inferred origins. Generally, origins are inferred for each transaction
before proceeding to destination inference.

On systems in which passengers tap in and out, the same procedure is followed to infer
destinations, using tap out transactions. For instance, Gordon et. al. apply ODX in London,
where passengers tap in and out of subway stations. Although it is less common, some bus
services also require tapping in and out. However, many urban rail and bus systems only require
passengers to tap in, so there are no transactions indicating destination times and places. In this
case, the location of the next transaction is taken as a target location, and it is assumed that the
passenger’s actual destination is near the target location. The target location of each card’s last
trip of the day can be based on either the first origin of the day (assuming the person returns to
where they began the day) or the first origin of the next day. Early models assume that a rail
trip’s destination station is the next rail trip’s origin station or the station closest to the next bus
stop boarded, and that a bus trip’s destination is the stop (downstream of the boarding stop)
closest to the nearest recorded location. The model introduced in this paper infers destinations
based on generalized cost minimization rather than distance. Within this framework, destination
inference is not possible for untrackable fare media, e.g. cash, when no target location is
available (either because the origin of the next trip was not inferred or because there is only one
trip), or when the target location is too far from any of the possible destinations.
A few definitions are useful for the discussion that follows. Ride refers to travel in a public transportation vehicle between boarding and alighting stops or stations. Fare stage refers to transportation activity accomplished with a single entry fare transaction (or an entry/exit transaction pair), which involves multiple rides when passengers transfer behind the gate. Journey refers to transportation activity performed for the purpose of a non-transportation activity at the destination, e.g. work, school, shopping, or returning home. Journeys involve multiple fare stages when transfers between rides require tapping again, regardless of whether the transfer is free. Figure 1 illustrates a hypothetical person’s trips in a day, showing some key aspects of the inference process. The passenger’s morning commute from home to work, a single journey, involves two rides on separate fare stages, with a transfer between them, e.g. a bus ride to a subway station followed by a subway ride to a downtown subway station close to work. The passenger taps when boarding the bus and again when entering the subway station. For the return commute, the passenger makes two subway rides, transferring between lines behind the gate, i.e. without tapping a second time. Thus, the second journey is completed in a single fare stage involving two rides.

**FIGURE 1 Journey, Fare Stage, and Ride**

Origin and destination inference are performed sequentially for each fare stage. The final step is transfer inference, the process by which fare stages are linked into journeys if it is deemed unlikely that the passenger engaged in non-transportation activity between fare stages. Gordon
et. al. (10) consider a set of temporal, spatial, and logical conditions to determine this. The likelihood of inferring a transfer between a pair of fare stages increases when (a) a relatively short time elapses between the first destination and the second origin, (b) the first destination and second origin are close to each other, (c) the two fare stages are on different lines, and (d) the second fare stage does not return close to the first origin.

While ODX makes disaggregate inferences, some of its most useful applications require aggregating across cards. Since ODX cannot infer the origins and destinations of all cards, or of passengers traveling without interacting with the fare system, origin-destination matrices generated by counting only fare stages with inferred origin and destination underestimate demand and are said to be unscaled. Matrices can be scaled to account for transaction records having inferred origin but uninferred destination, transaction records having uninferred origin, and non-interacting demand resulting from fare evasion and other behavior. The destinations of trips with inferred origin but uninferred destination can be distributed proportionally to the probability distribution of destinations of trips beginning at the same origin and having inferred destination. Trips followed by transfers should be omitted from the scaling distribution, because the fare transaction of the next trip makes it highly likely that a destination would have been inferred, so it is unlikely that a trip with uninferred destination is followed by a transfer; absent this consideration, scaling would introduce a bias towards destinations at popular transfer locations, such as bus stops close to rail stations (12). Trips with uninferred origin and destination can be distributed proportionally to the rest of the inferred demand, and the total demand can be further scaled up to account for non-interaction based on surveys of non-interaction by stop and time of day. Gordon et. al. propose an adaptation of the bi-proportional fitting algorithm to scale journeys (rather than rides) based on counts of passengers entering and exiting rail stations and buses.

This research deals with destination inference as a step of ODX, basing origin and transfer inference on the work of Gordon et. al. (10). The next section introduces a destination inference model based on dynamic programming.

MODEL

The objective is to find the path most likely taken by the passenger, i.e. that which minimizes generalized disutility and reaches the target location before the next transaction’s time. This can be expressed mathematically as follows. Let \( G(N, A) \) be a directed graph with nodes \( N \) and arcs \( A \). Let \( o \) be the origin node, representing the time and place of the origin transaction, and \( (o, j) \in A_o \subseteq A \) be a set of arcs starting at \( o \). Let \( d \) be the target location node at an unknown time before the next transaction, and let \( (i, d) \in A_d \subseteq A \) be a set of arcs ending at \( d \). Let \( A' = A \setminus (A_o \cup A_d) \) denote the subset of all arcs in \( A \) except those exiting \( o \) or entering \( d \) and let \( N' = N \setminus (o \cup d) \) denote the subset of all nodes in \( N \) except \( o \) and \( d \). Let each arc \( (i, j) \in A \) have a cost \( c_{ij} \) and a time \( t_{ij} \). Let \( t^* \) denote the length of time between this fare transaction and the next.

The optimization problem takes a shortest path structure that can be formulated as follows.

\[
\begin{align*}
\text{minimize} & \quad \sum_{(i, j) \in A} c_{ij} x_{ij} \\
\text{subject to} & \quad \sum_{j : (o, j) \in A_o} x_{oj} = 1 \\
& \quad \sum_{j : (j, d) \in A_d} x_{jd} = 1 \\
& \quad \sum_{j : (i, j) \in A'} x_{ij} - \sum_{j : (j, i) \in A'} x_{ji} = 0 \quad \forall i \in N' 
\end{align*}
\]
Expression (1) is the total cost of links traversed. Equations (2) through (4) are mass balance equations, requiring that exactly one path leaving from $o$ is taken, that exactly one path leading to $d$ is taken, and that on all nodes except the origin and destination any flow in is followed by flow out. Equation (5) requires that the path is completed before the next transaction. Expression (6) constrains flow to the integers 1 and 0, indicating if arc is traversed or not, respectively.

There are different types of arcs corresponding to different movements a person can make in a stage. When a person enters a station, the first arcs represent walking from the fare gate to the platform. If multiple platforms can be reached inside the station, there is one arc for each. These are followed by a waiting arc at each platform, representing time spent waiting to board a vehicle. Following each waiting arc is a vehicle arc representing movement inside the vehicle between stops. Fare stages beginning with a tap inside a vehicle start with a vehicle arc.

Following a vehicle arc, if the vehicle is at a station, there are arcs representing exiting the station or transferring to another platform inside the station. Following a vehicle arc alighting outside a station, or a station exit arc, there is a final walking arc to the target location. Figure 2 illustrates this sequence.

\[
\sum_{(i,j) \in A} t_{ij} x_{ij} < t^* \tag{5}
\]

\[
x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \tag{6}
\]

Arc costs reflect the weights of the following general disutility equation:

\[
V = \theta_e t_e + \theta_w t_w + \theta_v t_v + \pi n + \theta_i t_i + \theta_d t_d
\]
where $t_e$, $t_w$, $t_t$, and $t_a$ are the times spent on station entry or exit, waiting, in-vehicle movement, transfer, and walking to the target location, respectively, $\theta_e$, $\theta_w$, $\theta_t$, and $\theta_a$ are disutility weights for each of these trip components, respectively, $n_t$ is the number of transfers, and $n_t$ is the disutility for each transfer. Since disutility weights are relative, one of the weights, typically $\theta_t$, is normalized to 1. Arc cost weights can vary by mode or service to reflect differences in disutilities. They can also vary by day to reflect, for example, how inclement weather discourages people from walking outside. Vehicle arc times can be taken from vehicle location data, but schedules can be used where real location records are unavailable or unreliable. Each path has its own disutility, equal to the sum of costs of each arc, and each arc’s cost corresponds to the type of movement on the network. Paths for fare stages starting in a vehicle, typical of bus boardings, do not capture the initial waiting disutility, which does influence path choice, but this is a sunk cost because it a revealed choice: only choices made afterwards, affecting cost-to-go, are unknown, and only these are optimized. Waiting disutility is captured when it takes place inside a fare controlled area.

Once an optimal path is found, the destination inferred for each fare stage is the start node of the last arc, i.e. the walking arc connecting the destination to the target location. When the inferred destination is the same as the next origin, the last arc has zero cost and time. The destination time is taken directly from the path as

$$ t_d = t_o + \sum_{(i,j) \in A_o \cup A'} t_{ij} x_{ij} \quad (8) $$

where $t_o$ and $t_d$ are the origin and destination times, respectively.

Figure 3 illustrates a hypothetical graph for a fare stage starting at a rail station and target location at another rail station. In this case, two reasonable options are identified: one involving a single ride with a longer walk and another involving a transfer but a shorter walk. (If the second train takes the passenger to the target location, then the walk has zero cost and duration.) The path involving a longer walk, depicted with thicker lines, is selected if it minimizes disutility, even if the alternative destination is closer to (or is exactly at) the target location. The inferred destination, shown in gray, is the station exited one node upstream of the target location. This contrasts with the distance minimization objective used in earlier models, which would select the alternative path with higher disutility. Dashed lines show other branches that the optimization algorithm explores before determining they are suboptimal.

**FIGURE 3** Graph of a Fare Stage with Station Origin
Networks for fare stages initiated in typical bus rides are relatively small: one vehicle arc
representing movement from the origin to each downstream stop, each of which is followed by a
walking arc to the target location. In contrast, networks for trips crossing stations where it is
possible to transfer behind the gate can be much larger, since branches are created for each
vehicle that the passenger could board. Rather than pre-computing the entire network, the
network is built dynamically as the dynamic program is solved, based on a static representation
of the network and vehicle locations stored in memory. In the interest of computational
efficiency, branches are pruned whenever the running cost exceeds the best cost found thus far to
the target location, or the time is after the next transaction. The network being built must always
satisfy
\[ \sum_{(i,j) \in A} c_{ij} x_{ij} \leq c^* \]  
where \( c^* \) is the best cost so far. A multistage graph algorithm implemented in Java solves the
optimization problem, taking 3 milliseconds on average on an Intel Xeon E5-2650 v3 processor
running at 2.3 GHz. (See (13) for coverage of multistage graphs in dynamic programming.)
Memory consumption tends to stay below 10 GiB, and can be lowered to less than 2 GiB with
aggressive garbage collection at the expense of significantly longer processing time.

**APPLICATION**

The dynamic programming model for destination inference is being employed to estimate origin-
destination matrices of the Massachusetts Bay Transportation Authority (MBTA) public
transportation network. The following disutility weights are assumed: \( \theta_e = 1, \theta_w = 2, \theta_v = 1, \)
\( \pi_t = 10, \theta_t = 1, \) and \( \theta_a = 5, \) and times are expressed in minutes. Disaggregate fare transaction and
vehicle location data (for both buses and trains) are input to the model. Based on weekdays of
March 2016, the model infers the origins of 98% of fare transactions, and the destinations of
87% of fare transactions for which an inference is possible. (The model infers destinations for
73% of all transactions, including those made with cash or cards having only one transaction per
day, for which there is insufficient data to infer a destination.)

The model has enabled high-resolution origin-destination inference on the Green Line, a
light rail service calling a mix of gated subway stations and ungated surface stops. Passengers
tap at the farebox (as they would in a bus) on the surface portion, but not in the subway. Earlier
inference methods only consider downstream stops served by the boarded vehicle as possible
destinations when the fare transaction is at a vehicle farebox; they allow destinations to be on
different lines than origins (i.e. transfers behind the gate) only for rail trips. Dumas
circumvented this problem by consolidating all surface stops of each branch of the Green Line
into a single virtual station, at the expense of lower spatial and temporal resolution. The
dynamic programming model is applied equally to bus and rail, allowing vehicle arcs connecting
ungated stops with gated stations, and enabling disaggregate demand analysis on services like the
Green Line.

The model infers destination place and time at once, based on the penultimate node of the
optimal path, rather than inferring the place with heuristics and then the time. When Dumas
adapted the method by Gordon et. al. to Boston, it was necessary to generate a table of paths for
each origin-destination pair, to look up travel times in the subway network given an origin and a
destination. This approach fixes the path, making inference insensitive to real-time conditions
that may change how passengers travel in the network. In contrast, the dynamic programming
approach captures service suspensions, delays, and disruptions in arc costs.
The model’s inference of activity behind the gate can provide important insights. Paths traversing gated stations have information about transfers and which vehicles are boarded, which can be used to measure, vehicle loads, platform waiting times, and transfer behavior. For instance, it is now possible to estimate flows between subway stations based on ODX data that captures boardings of passengers transferring from other lines, including passengers boarding at surface stops of the Green Line. Capturing all flow does require scaling the inferred origin-destination matrix to cover transactions with uninferred origin or destination, as well as demand not interacting with the fare system. The adaptation of bi-proportional fitting to journeys suggested by Gordon et. al. depends on reliable entry and exit counts, which systems like the MBTA sometimes lack. Ongoing research on scaling methods well adapted to origin-destination inference based on dynamic programming will be the subject of a later paper.

Since its inference is based on minimized disutility rather than proximity, the dynamic programming model can infer destinations that do not minimize distance to the next origin. This can happen when there are multiple paths leading from the origin to the target location and there are trade-offs between transfers, walking time, waiting time, and in-vehicle time. For example, when there is a choice between walking to the target location from a station or transferring to a different line and riding to the target location directly, the model compares walking time with transfer penalty, transfer time, and additional time spent waiting and riding. To analyze differences in inference, destinations inferred by the dynamic programming model were compared to those derived from minimizing distance to the target location. In some cases, the difference is large. Consider examples from the MBTA rail network, shown in Figure 4.

Passengers taking the Orange Line inbound from Forest Hills (southwest corner) and entering Copley (center-north, on the Green Line trunk) can alight at Back Bay (just south of Copley) and walk five minutes rather than continue on the Orange Line five more stops to Haymarket, transfer to the Green Line, and ride it five stops to Copley, which would take more than twenty additional minutes. In other cases, the difference is not as large. Passengers taking the Blue Line inbound from Maverick (northeast corner) and later entering Downtown Crossing (on the intersection of the Red and Orange lines) can either alight at State Street (on the intersection of the Blue and Orange lines) and walk 4 minutes to Downtown Crossing, or transfer to the Orange Line at State Street and ride one stop to Downtown Crossing, taking about 2 minutes longer. It is likely that some passengers will prefer the latter option because their walking disutility is higher than assumed, perhaps as a result of personal preference or weather. Compared to the distance minimization heuristic used in earlier models, the dynamic programming model infers different destinations in 4.5% of the transactions having an origin and target location at gated rail stations.
Another scenario in which the dynamic programming model infers a different destination is when vehicles are slowed by traffic or problems with rolling stock or infrastructure. Once the passenger nears the target location, it is possible that alighting and finishing the trip on foot has a lower disutility. Passengers can make choices of this nature based on observed traffic and information provided through announcements and alerts, either before or during the trip. This has the benefit of capturing how passengers adapt their choices to information, but the drawback of being sensitive to information that passengers ignore when choosing paths. For instance, although passengers cannot predict signal failures that will delay their journey, the dynamic programming model responds to the ensuing disruptions seen in the data, inferring paths that would require passengers to be clairvoyant.

CONCLUSIONS

This article presents a dynamic programming model to infer destinations of public transit trips. Compared to earlier methods based on heuristics, the dynamic programming approach explicitly captures the disutilities of waiting, transferring, riding, and walking, trading these off to find the path that minimizes generalized cost. Using bus and train location data, the model infers destination places and times simultaneously instead of in two separate steps, and applies equally well to bus and rail trips, with or without transfers. The application to fare transactions generated over a month in Boston’s public transportation network resulted in destinations inferred for 73% of transactions. Reasons for no inference include untrackable media (mostly cash), only one transaction in a day, and the next origin being too far from possible destinations.
The model has enabled inferring origins and destinations of Boston’s Green Line, a light rail service that operates in a mix of ungated surface stops (with tapping when boarding the vehicle) and gated subway stations (with tapping at the gate when entering the station). Along with destination, the model infers transfers, waiting times, and in-vehicle times, which can provide insight about how passengers move in the network and be an input to performance measures and crowding models.

The model’s destination inferences were compared to those resulting from the distance minimization objective of earlier models on trips originating at gated stations and followed by trips also originating at gated stations. While the heuristic always infers that the destination is the next origin, the dynamic programming model captures a disutility minimization objective that yields different destinations in 4.5% of fare transactions. Some differences are certainly an improvement in accuracy, as it would take considerably longer to reach the next origin by public transit, while others are probably a matter of each passenger’s characteristics and other factors including weather. Given bus and train tracking data, the model captures the effects of traffic, unreliability, and disruptions in path choice, potentially improving accuracy, but in some cases it can infer paths based on information not possibly known by passengers when they choose their path.

Future work will focus on relaxing some assumptions to make the model more realistic and widely applicable. The model is deterministic, but it could be extended to infer destinations stochastically by combining generalized costs with a discrete choice model. This is particularly of value for applications to dense public transit networks that provide many transfer opportunities and several good paths from origin to destination. Another worthwhile effort is capturing the effects of congestion and capacity in arc costs, which would lead to more accurate inference in networks operating near capacity. Arc costs could be manipulated to model the service passengers expect, which has more influence on path choice than what is actually delivered.

ACKNOWLEDGMENTS

This research was funded by the Massachusetts Bay Transportation Authority (MBTA).

REFERENCES


