Simple Methods for Reasoning about Behavior Patterns on Graphs Given Extremely Sparse Observations

R. Paul Wiegand
Institute for Simulation & Training
University of Central Florida
Orlando, FL, USA
Email: wiegand@ist.ucf.edu

Steven D. Prager
Department of Geography
University of Wyoming
Laramie, WY, USA
Email: sdprager@uwyo.edu

Abstract*—We consider the situation where fixed observations of moving entities are sparse and the goal is to learn as much as possible about their *patterns of activity*, before and after such observations (e.g, cameras at a few intersections in a city). Here we present a method for estimating probable paths within a network given a limited set of vertex observations and limited *a priori* assumptions about individual entity behavior. We divide the process of analysis into two phases: a *learning phase* in which aggregate information about many entities is obtained and used to construct simple models given potential observations, and a *reasoning phase* in which resampling methods produce probable paths a specific entity may have taken. To accomplish this, we extend a fair and efficient method for randomly selecting unconditioned paths within a network in order to draw paths conditioned on limited, partial observations. The methods are validated by analyzing hypothetical observations of entities moving on an existing city street network. Our results show the scaling properties of this approach by optimizing the locations of different numbers of fixed potential observation points to obtain as maximal coverage of the area as possible. We then construct a variety of models based on an extremely sparse observational scenario and demonstrate quantitatively and visually that these simple methods, combined with structural information inherent in the graph itself, can provide a great deal of context information about an individual entity's possible movement patterns.

***Keywords*-sparse observations; fair paths; path analysis**

I. Introduction

Many GIS-based methods for analyzing the behavior patterns of entities (e.g., people, cars, etc.) on both traditional and complex functional networks incorporate extensive information about individual entity movement within the network, often assuming one has access to GPS tracking data or other such detailed information. Frequently, the goal in such cases is to take existing data and use it to predict *future activity* [1], [2].

Instead, we consider the situation where observations are sparse and the goal is to try to learn as much as possible about a *pattern of activity* before and after such observations. For example, suppose there are fixed CCTV camera locations at a few intersections, scattered throughout a city. Given some highly limited general knowledge of potential paths taken (e.g., a distribution of typical path lengths) and some specific observations of a particular entity at two or three of the intersections via the cameras, what are the most probable paths the entity might have taken?

In this paper, we construct and employ a simple method for drawing probable paths within a network given a handful of vertex observations and limited *a priori* assumptions about individual entity behavior. Our approach is to breakdown the process of analysis into two phases: a *learning phase* in which aggregate information about many entities is obtained and used to estimate path length distributions conditioned on potential observations, and a *reasoning phase* in which resampling methods are used to combine partial observations about a specific entity with global knowledge to provide an analyst probable paths an entity may have taken. Though it is natural to think about these path-patterns as movement in a physical network such as city streets, our method makes no such assumptions.

This paper is developed around a fair and efficient (polynomial) method for randomly selecting unconditioned paths within a network in a probabilistically correct way. Our algorithm (DrawPath) is constructed using elementary concepts in graph theory and probability. We extend this model to draw paths conditioned such that only paths that contain specific observations, in a particular order (though not necessarily consecutive) will be drawn. We validate our approach by considering the analysis of hypothetical movement patterns of entities operating on the city street network in the western half of Laramie, Wyoming.

To determine the best coverage of the observations in the town, we employ an evolutionary algorithm (EA) [3] to optimize the locations of a limited number of fixed potential observation points. The objective function used by our EA relies on the same simple graph-theoretic notions on which our DrawPath algorithm is based. We have a simple way to estimate the probability that a path from the general, aggregate movement distribution will pass through at least one of the observations. We show that the coverage we obtain scales well even as the number of potential observation locations is greatly reduced.

 ISBN: 978-1-61208-251-6

We then focus on an extreme case where there are only two fixed positions in the town that we can use to analyze movement. Several models are constructed based on these two fixed positions, and these to demonstrate quantitatively and visually that a great deal of context information about an entity's possible movement patterns can be obtained in spite of the sparsity of observations. This provides preliminary evidence for our position that the graph itself contains a great deal of information, and that given a simple aggregate movement model and a few individual entity specific observations, it is still possible to obtain a strong understanding of probable movement patterns for the entity by exploiting structural information within the graph itself.

This paper is organized as follows. The next two sections provide context by discussing related literature and problem formalization, respectively. Section IV explains are simple, traditional graph theory based methods for drawing distributionally correct fair random paths. After this, we indicate how much coverage is possible with few observations by optimizing potential observation locations for maximal coverage, then in Section VI we discuss how prior distributions for potential movement patterns through sparse observations can be learned. Finally, we explain how one can put all this together to reason over potential paths given sparse observations and provide some concluding remarks.

II. Background and Motivation

Understanding entity movement in space and time builds on early work in regional geography regarding the concept of space-time paths [4]. The space-time path — a "trace" of an individual entity's movement in space, through time — is comprised of the set of ordered (in time) observations where entity locations are recorded and the corresponding path segments between those locations are inferred [5]. Where space-time path approaches are limited, however, is when the set of observations are limited and serve to capture only a portion of a path.

Given the limitations of space-time path approaches when much of the requisite entity movement data are unobserved, alternative mechanisms to understand individual movement are required. As technology and computational approaches have advanced, focus has shifted toward modeling movement in space using more bottom-up, agent-based approaches [6]. For example, individual pedestrian movements have been modeled at the "microscopic" scale [7] in order to examine the role of fundamental behaviors in governing movement characteristics. Other agent-based approaches have emphasized crowd level behaviors [8], in order to contextualize movement given larger, agglomerative behaviors. These bottom-up approaches, however, often assume little knowledge regarding the geographic space or network neighborhood in which the entities are operating.

Understanding the space in which entities are moving is an important aspect to understanding their potential behavior. Space-syntax studies demonstrate that network and related areal topologies have demonstrable effect in how space influences movement [9]. The environment, whether unconstrained (e.g., an isotropic surface), constrained (e.g., a network), static, or dynamic, is an important determinant in the driving the modeling approach [10]. Recognition that certain movement patterns or *typologies* emerge as a function of the space in which movement is occurring provides a basis for understanding movement from limited observations [11].

Using limited observations to best cover a given space (network or otherwise) and understand movement occurring therein is thus an important aspect of this work. As with the space-time path, early work in examining maximum covering problems in a geographic context emerged in a formalized way in regional geography studies [12]. Maximal covering location theory has informed work ranging from identification of best locations for security cameras in both static [13] and dynamic environments [14].

In contrast to the high observation model of sensor network approaches for understanding space-time paths, low observation models require as much information as possible that can be exploited from other sources. In the following sections we formalize our approach for leveraging information regarding the network space in which movement is occurring and the use of a simple movement model of describing individual movement within the given environment.

III. Formalizing the Problem

As described in the introduction, the fundamental challenge we are considering is to determine the probability that some specific person (for example) has traversed some path given a graph, some observations of the person on that graph, and a some kind of more general underlying distribution regarding path properties. Let us clarify this problem both by providing a formal description of the problem and by example. For descriptive purposes, our examples refer to some particular path taken by an individual named *Bill*.

Let $G = (V, E)$ be a digraph with vertices V and edges E. Graphs are given in the problem description, and they represent some kind of network-based structural information available in the problem itself. E.g., it might encode topological information about Bill's home town street network.

We write a path simply as an ordered set of vertices, $p = \langle v_0, v_1, \ldots, v_k \rangle$ such that $\forall v_i \in p, v_i \in V$ and $\forall v_i, v_{i+1} \in p, (v_i, v_{i+1}) \in E$. That is, a path is a sequence of connected vertices. Note that paths *can* contain the same vertex multiple times (i.e., the path can cycle). We define the *length* of a path as the number of vertices in the path[1]. A path represents every graph point on some route that a person traverses within the graph. For example, Bill leaves the bank, moves south a block to the corner gas station, then

[1]Technically, in graph theory, path length is traditionally defined as the number of hops taken, $k-1$ where k is the number of vertices in the path; however, we keep everything vertex-centric for consistency.

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-251-6

heads east to the intersection of main and 1^{st}, then moves east once more to his apartment. His path's length is three.

In addition, we have the concept of an *observation*, a fixed vertex at which we can obtain information about a specific person on a specific path. We may have multiple observations regarding the same path. While we retain *order* information about observations, one simplification we make is to ignore time. We distinguish between two types of observations, *positive observations* and *negative observations*. Positive observations are an ordered collection of sightings of an individual at specific vertices for the path, $\langle o_1, o_2, \ldots, o_m \rangle$, where there exists some path p such that for all o_i in that collection, $o_i \in p$. The set of negative observations is an unordered collection of vertices through which an individual *did not* pass. For simplicity, we use M to notate the complete collection of observations. As an example, consider a case where we observed Bill at the gas station and later at the intersection of main and 1^{st} avenue, though we know he was not seen by the camera at the bank. Most of his path, including his start and end point, are unknown to us; however, we have two positive observations and one negative observation regarding his path.

In addition to representing individual behavior, we are interested in collecting aggregate information about possible paths. Currently, we simplify things and consider only path length for the prior distribution of path properties, and we assume that (unconditioned) path lengths are Poisson distributed. That is, for some path length k:

$$\Pr\{k\} := \frac{\lambda^{(k-1)} e^{-(k-1)}}{(k-1)!}, \quad k > 0$$

The $k-1$ adjustment is due to our having defined the path length as the number of vertices. Conceptually, the number of null-paths, paths that contain *no* vertices at all, is not a sensible number to consider. For convenience, we say that $\Pr\{k = 0\} = 0$. Under the optimistic conditions provided by the problem definition, namely that a set of historical path lengths are provided, we can determine $\Pr\{k\}$ by simply computing the expected length parameter $\lambda \approx \frac{1}{|K|} \sum_{i=1}^{|K|} k_i$. For our experiments, we assume this is done and specify λ directly. In this paper, we use $\lambda = 6$ for all experiments.

We refer to $\Pr\{k\}$ as the *prior path length distribution* and $\Pr\{k|M\}$, the probability of a path of length k if we make observations M, as the *posterior path length distribution*. We break the problem into two phases, *distribution learning* and *path resampling*, and define them separately. These are described in Section VI and VII, respectively.

PROBLEM *1:* **Distribution Learning**

Given: a graph G, hypothetical observations M, and a set of historical path lengths, K (where $\forall k_i \in K, k_i \in \mathbb{N}^+$).

Find: prior path length distribution $\Pr\{k\}$ and posterior path length distribution $\Pr\{k|M\}$.

PROBLEM *2:* **Reasoning via Path Resampling**

Given: a graph G, real observations M, and the model distribution $\Pr\{k|M\}$

Find: a set of paths P such that $\forall p_i \in P, \Pr\{p_i|M\}$ is "sufficiently large".

IV. Fair Path Sampling with Basic Graph Theory

Ultimately, our efforts require efficient estimation methods for drawing paths that take graph structure into consideration. One might simply produce random paths via a random walk on the graph. For example, a starting node might be selected at random, a random neighbor might be selected from that vertex, repeating this process until a designated criteria has been satisfied (e.g., a path length drawn *a priori* according to some known distribution). However, ensuring that constraints on the graph do not unintentionally bias such walk is non-trivial, particularly when the graph is directed. Alternatively, one might simply enumerate all possible distinct paths (addressing loops in some well-defined way) and select a path from this set. Unfortunately, there may be a combinatorially large number of potential paths. Moreover, this set may be quite sparse with respect paths that contain the observations.

Instead of these approaches, we turn to traditional graph theory. Note that connectivity in a typical graph can be represented using an adjacency matrix, L, containing binary values, where a 1 in some cell L_{uv} indicates that there is a directed edge in the graph from vertex u to vertex v, and a 0 indicates that there is no such edge. The matrix L can also be interpreted as indicating all simple paths of length precisely 1 between any pair of vertices. We can apply matrix multiplication to produce a new matrix of the same dimensions, L^2, which then gives us the number of paths of length 2 connecting each pair of vertices. In general, L^k_{uv} gives the number of distinct paths of length k that connect $u \rightsquigarrow v$ [15]. Our methods use this property in a variety of ways. We refer to this as a "*path-count*" calculation.

We use this path-count calculation to estimate the probability that a path drawn according to our aggregate model distribution (in this case, a path in the graph whose length is Poisson distributed) passes through at least one of some set of observations. For a given length, ℓ, the total number of paths of that length can be computed by summing all values in the L^ℓ matrix for a particular graph. Paths that *do not* pass through the one of the vertex observation points cannot traverse an edge to or from any of the potential observation points, so we can produce a new graph by removing all the potential observation point vertices and their out-going and in-coming edges. The total number of paths for a given length for this graph can be computed in the same way as for the original graph. The difference between these is the total number of paths of length ℓ that passed through at least one of these edges, and the ratio of this over all possible ℓ-length

 ISBN: 978-1-61208-251-6

paths in the original graph is the probability of drawing such a path given the length, $\Pr\{D|k=\ell\}$, where D is the event of drawing a path going through at least one of some set of observations. Thus $\sum_{i=1}^{\infty}\Pr\{D|k=i\}\Pr\{k=i;\lambda\}$ yields the probability of a Poisson distributed path with expected length λ going through at least one of the observations. We estimate this by summing such values until some sufficiently large path length or until the $\Pr\{k=i;\lambda\}$ is very small.

Secondly, the path-count metric provides a means of "fair" sampling for paths given a distribution of path lengths. Algorithm 1 presents the pseudocode for DRAWPATH.

Algorithm 1 DRAWPATH (graph connection matrix L, path length distribution)

```
Randomly select path length ℓ according to distribution
Compute the L^ℓ path-count matrix
Randomly select a source-destination pair for path p = ⟨s, ⇝, t⟩
    proportionate to the L^ℓ path counts
Let a = s
Let b = t
for i = ℓ − 1 down to i = 2 do
    Compute the L^i path count matrix
    Form p_a, the 1-step ab intersection of this and L as follows:
        Let r_a be the a^th row of L^i
        Let r_b be the b^th column of L^i
        Let c_b be the b^th column of the L connectivity matrix
        Let p_a = r_a × c_b, using element-by-element multiplication
    Randomly select vertex, x, proportionate to p_a values
    Insert x following a in p and let a = x
end for
```

A process following such an algorithm will produce a valid path in the graph with length ℓ in a distributionally correct way — that is, the lengths of the paths drawn will be Poisson distributed (in our case), and paths of any specific length are equiprobable with other paths of that same length.

Both of the estimate for paths passing through at least one of some set of potential observations and our DRAWPATH algorithm are reasonably efficient. Matrix multiplication can be performed in polynomial time (even naïve implementations are $O(n^3)$), so even under simple implementations, the estimation metric is cubic for constant-sized maximum length values, and the DRAWPATH algorithm runs in expected time $O\left(E\{\ell\}^2\cdot|V|^3\right)$, where $E\{\ell\}$ is the expected length of a path and $|V|$ is the number of vertices in the graph. In our implementation, the length factor is reduced from quadratic to linear by eliminating redundant matrix multiplications at the expense of storing all L^i path count matrices created in during the first iteration and using the appropriate matrix during subsequent iterations as i is relaxed. Indeed, we store and expand this list of matrices throughout the estimation process so that the computation for a given k is only ever computed once. The cubic factor in the size of the vertex set can be reduced marginally through the use of efficient matrix multiplication routines [15].

It is worth noting that these two methods can use any length distribution, and they do not rely on the Poisson distribution. Our method merely needs some domain-relevant way of computing $\Pr\{k\}$. Additionally, there are relatively straightforward ways to bias the draw using externally provided edge transition probabilities to replace the c_b vector. These two notions allow for the possible incorporation of a more sophisticated aggregate behavior models, though this is not explored in this paper.

V. OPTIMIZING POTENTIAL OBSERVATION LOCATIONS

To begin the process, we must determine the best location to place our hypothetical observation points within the graph. We are interested in the best *observational coverage* of the graph — the potential locations for observations in the graph that maximize the probability of observing someone. To be effective, this should take the graph structure into account, as well as the collective behavior model.

From our discussion in the previous section, we know that we can estimate the probability that a path drawn from our collective distribution passes through at least one of some subset of potential observations fairly efficiently using path-count matrix calculations. For this paper, we use a simple evolutionary algorithm (EA) to optimize the locations of a fixed number of possible observation points using our estimate of the probability of observing paths as the objective function.

Our algorithm encodes fixed-length individuals as potential positions for the observations. Out of a population of 20 individuals, one is selected at random to be cloned and mutated. It replaces the worst individual in the population (chosen uniformly at random if there is a tie) if it has at least as high an observational coverage. The number of mutations it undergoes is binomially distributed with the $n=|V|$ and success probability of $\frac{1}{2m}$, where m is the number of potential observations. With equal probability, a mutation event will result in a swap of a position for some other position in the graph *or* a swap of a position for one of its neighboring vertices. The EA is run for 500 steps. These parameters result from parameter tuning, and we make no claim as to their quality except to say that in all cases the EA made progress and appeared to converge to a local optima.

The West Laramie street network graph contains 143 vertices, and during the optimization process, the evaluation heuristic considered paths of length 1 to 15. In spite of this, there are over 7 million paths of such in this graph.

To get a sense for how observational coverage can scale, we ran our EA for several cases of m observations points, $\{128,64,32,16,8,4,2,1\}$. There were 25 independent random trials conducted for each group. In each trial, we recorded the best observation points discovered by the algorithm. The graph below illustrates these results. In the graph, the points represent the mean best solutions found of the independent runs of the EA, and the whiskers represent the 95% confidence window for each group. We plot a trend

 ISBN: 978-1-61208-251-6

line through these points to give the reader a sense for how observational coverage scales as the number of observation points is decreased.

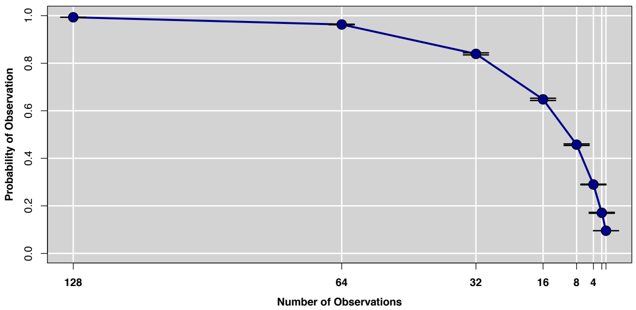

Figure 1. The best observational coverage results for various number of observation points. The points show the mean of 25 independent trial EA runs, and the whiskers show the 95% confidence window for each group.

In each experimental group, the best results were taken and verified as follows: The DRAWPATH algorithm described above was sampled 2000 times and the number of paths that contained at least one of the optimized observation locations was counted. This probability was within the statistical proportions margin of error ($\alpha = 0.05$) of the estimated heuristic based on path-count calculations. Additionally, the path-count estimate for the observational coverage for paths of length 15 or smaller was compared to the same heuristic for path lengths between 1 and 100 with no statistical differences.

Figure 1 illustrates that the observational coverage scales quite well for this graph. Clearly such coverage will differ greatly depending on the graph and the collective behavior model, it is also evident that a great deal of information about individual behaviors in this graph can be available with just a few observation points. Indeed, just two observation points are needed to cover just over 17% of all paths. The location of these observation points is shown the graph below. For the remainder of the paper, we assume that our algorithms have access to observations made at *only* these two points. We refer to these two observations points as E (the east-most point) and W (the west-most point).

VI. DISTRIBUTION LEARNING

The goal of the first stage of our method is to combine the collective behavior model with hypothetical observations to develop a better understanding of a potential entity movement pattern. While one might simply try to compute the posterior path length distribution when the specific observations are made, we feel it is more constructive to think of this as part of the more general learning process for at least three reasons. First, such computations are likely to be somewhat time consuming estimations. As a result, it is beneficial to develop prior estimates based on hypothetical observations so that sampling can be done relatively quickly in specific cases. Second, prior computation of the posterior distribution facilitates useful analysis of potential observations. These two reasons are justified further in the next paragraph. Finally, when we base our collection of hypothetical observations on historical information, we can consider collections of known observations as a way of encoding aspects of aggregate information about the population. In that sense, it *is* a part of the learning process.

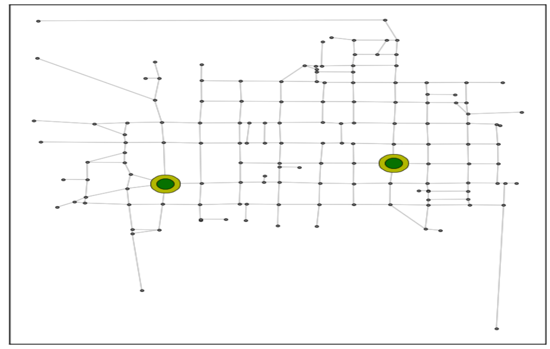

Figure 2. The West Laramie street network with the potential observation points highlighted. These two points represent the best pair of observation locations for coverage discovered by our EA.

The idea of "hypothetical observations" is rooted in our perception of how aggregate information may be acquired for realistic problems. First, a reasonable scenario is one in which the locations of the observation points are established up front. For example, cameras are specifically located and do not move. If the number of such potential observation points is relatively small, we can pre-compute the posterior path length distribution for many important combinations of observations. Second, we can use a posterior path length estimation algorithm to help with analysis and related, subordinate problems to the main problem. For example, we might examine a variety of potential observation points to see how they affect the length distribution, or use some of the underlying graph theoretic components from the estimation phase to help determine their ultimate placement.

Bayes theorem gives the posterior path length distribution:

$$\Pr\{k|M\} = \frac{\Pr\{M|k\}\Pr\{k\}}{\Pr\{M\}}$$

The value of $\Pr\{k\}$ can be obtained from the historical path lengths as described above. Since $\Pr\{M\}$ is the same value in all cases that use the same observations, one can simply normalize after computing a sufficient number of cases for k. One can enumerate cases of k up to some large number, terminating early when it is likely most of the probability distribution has been accounted for. The way we do this is to check when the running ratio described below for each k drops below some small value (e.g., 0.001).

$$\frac{\Pr\{M|k\}\Pr\{k\}}{\sum_{j=1}^{k}\Pr\{M|k\}\Pr\{k\}}$$

We modify the DRAWPATH algorithm discussed in Section IV for drawing fair paths from the collective model to help us compute the condition probability given observations. Here it is assumed that the observations are correct and ordered. That is, if we observe an individual at some vertex v_a then later at v_b but never at v_c, then the individual really was at vertex v_a before passing through v_b, and he or she never passed through vertex v_c. This latter point is important: All observation points contain information about paths since *failing* to see an individual at the point is not the same as no knowing whether the individual visited the point. Consequently, conditioning the path draw based on a negative observation is straight forward: We simply remove the vertex and all in-coming and out-going edges prior to analysis, as we did when we estimated the observational coverage probability.

Modifying the algorithm to consider the ordered positive observations of an individual is more challenging. First note that the algorithm above can be easily altered so that it considers potential end points of the path. To incorporate conditional observations, we combine this idea with a notion we refer to as "*path templates*".

A. Conditioning the Path Draw End Points

Given the start and end vertices of the path, we can simply replace the third step of the PATHDRAW algorithm above with the specific cell value in the L^ℓ matrix indicated by the start and end vertex pair. If only the start vertex is available, we can replace the summation in that step with summing the row indicated by the start vertex. Such a value represents all the unique paths of the given length that start at that vertex. Likewise, if we receive just the end vertex, we can sum the column indicated by that vertex — which is the count of paths of a given length that end in that vertex. The rest of the algorithm remains unchanged.

B. Path Templates and Piecewise Path Draws

Crucial to understanding our sample estimation methods is the concept of a *path template*. Let us temporarily assume we are given not just observations and a length, but we are also told precisely where in the path the observations where made. We can represent such a path by substituting the unknown vertices with special "wildcard" symbols and treat the entire path as a kind of rule or *template*. For example, given a path length of 6 and observations at vertex a then b, one partially informed path is: $\langle ..a.b. \rangle$. A path *matches* such a template if it is of length 6 and goes through vertex a in its third position and vertex b in its fifth position.

Such a template can be broken up into three subpaths: $\langle ..a \rangle$, $\langle a.b \rangle$, and $\langle b. \rangle$. We can then draw uniformly from each subpath using PATHDRAW as discussed above and assemble the complete path after the fact. Such a piece-wise process allows us to draw a path p uniformly from the set of paths that match a specific template. Note that we can also precisely and efficiently compute the number of paths that a template matches by multiplying the number of matches of each of the subpaths.

C. Putting it Together: Estimating $Pr\{M|k\}$

If path templates were independent of one another with respect to the paths they match, one needn't even use DRAWPATH: Simply use the method just described to compute the total number of paths that match a given template, and instead sample templates of a given length. Knowing this, in addition to the total number of paths of a given length (unconditioned by observations) gives an easy and efficient way to compute precisely the probability that of a template match given path length. This reduces the problem to sampling amongst templates rather than paths.

In principle, templates can be produced in a number of ways. One method is to enumerate them given observations. While this scales combinatorially with the length of the path and number of observations, it is still quite manageable for relatively small paths with sparse observations. Alternatively, we can produce a template uniformly at random by using a simple and efficient shuffling method.

Unfortunately, two different templates may match the same path so the templates are not independent of one another. We can use the *inclusion-exclusion* principle [16] to adjust these match counts if we have a means of computing the union of the match sets represented by two templates. This is straightforwardly accomplished by *merging* templates: If two fixed positions do not match then the union of the templates produces 0 matches, otherwise the merged template is produced by retaining all fixed positions between the two templates. For examples $\langle a...b.c \rangle \cup \langle a.d...c \rangle = \langle a.d.b.c \rangle$. However, overall this computation can be quite costly in the worst case since there can be a combinatorial number of levels of such unions. It may be that only a few levels are needed to get a reasonable approximation, and there may be ways of using piece-wise DRAWPATH to reduce the overhead of full enumeration. We would like to explore this possibility in the future.

We store the templates that with non-zero numbers of paths that were discovered during this process, along with the number of paths they match (*match count*) as explicitly computed from the path-count matrices as described above. This is used during resampling, as we shall see.

Consider the West Laramie street network with two fixed observation points as determined by our optimization discussed in Section V, E and W. Since positive observations are ordered but negative observations are not, there are five cases for learning. Observation(s) at:

i) E then W;
ii) W then E;
iii) E but *not* W;
iv) W but *not* E;
v) neither point.

Since there are only a handful of such cases, we are free to learn these distributions *a priori*. In more complex scenarios, it may be necessary to learn only the a few observational scenarios, then add new cases as they occur.

VII. Path Reasoning Given Sparse Observations

Once we have learned a catalogue of distributions based on potential observations, we can use this information to resample paths given specific, actual observations. In much the same way as the distribution is sampled in the first place, this problem includes drawing a length, then a template, then a path from that template. Because of the dependence problem between templates above, we do not use the sample to estimate probability, but instead assign the probability of a path directly from the posterior distribution. This process is described in more detail below.

Algorithm 2 PathResample (observation set M, graph G, posterior path length distributions, templates & match counts, maxSamples)

```
Initialize path list to an empty list
for i = 1 to maxSamples do
   Lookup posterior {k|M} distribution
   Draw path length according to Pr{k|M}
   Lookup recorded templates & match counts
   Randomly select template proportionate to match count
   for each unspecified subpath in the template do
      Randomly select valid subpath using piecewise draw
   end for
   if path is unique then
      Assign the path probability Pr{k|M}
      Store path and probability in path list
   end if
end for
Remove from path list all but the 5% most probable paths
return  path list with associated probabilities
```

The list of paths recovered from the resampling procedure is used in two ways. In each of the five observational cases, the graph can be plotted with the resampled paths overlaid for visualization. Second, a quantitative measure of *informativeness* of the resampled path can be estimated based on an edge count frequency histogram: All edges are treated as separate bins, and an edge is counted once for each of the resampled paths in which it participates, then the entire list of edges is normalized to create a distribution of edge usage in the resampled paths. Given a baseline condition that equiprobable edges implies no information, we compute the distance of this edge count frequency histogram to a vector of length $|V|$ with all values equal to $1/|V|$, where $|V|$ is the number of vertices in the graph. These visually and intuitively confirm the idea that the observations can provide significant context about an entity's behavior

The results presented in Table I indicate that having any positive observations at all provides a substantive amount of information over and above no detection. Figures 3, 4, and 5 we present three graphs that are visually indicative of these results, cases i, iii, and v, respectively.

Table I
Distance measures between uniform distribution of edges and the edge count frequency histograms constructed from resampled paths for each learned distribution.

Case #	Case Description	Distance
i	Observation at E then W	0.2206
ii	Observation at W then E	0.2289
iii	Observation at E but *not* W	0.2403
iv	Observation at W but *not* E	0.2223
v	Observations at neither point	0.0949

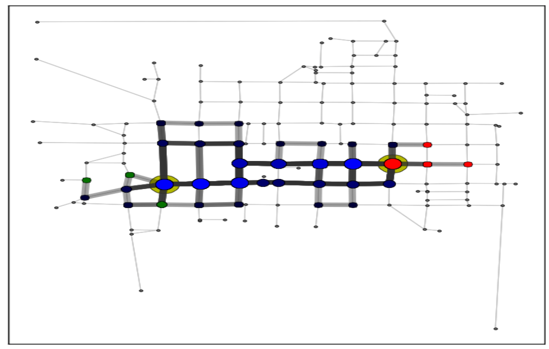

Figure 3. The 5% most probable resampled paths for an individual on the West Laramie street network, under the condition that he/she was observed first at the east-most (E) observation point then the west-most (W).

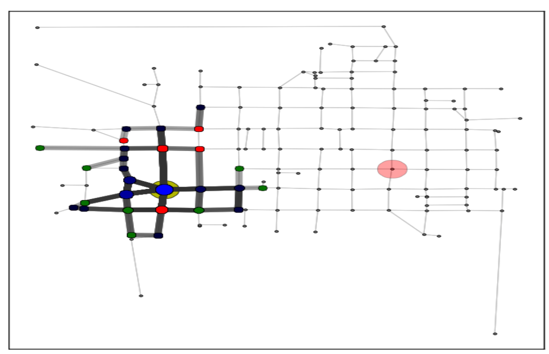

Figure 4. The 5% most probable resampled paths for an individual on the West Laramie street network, under the condition that he/she was observed at the east-most (E) observation point but not at the west-most (W).

VIII. Conclusions

In this work, we adopt the position that, because the structure of a graph itself contains a great deal of information, one can use relatively simple graph theory and probability techniques to learn a lot about potential behavioral patterns of entities operating within the graph even under quite

 ISBN: 978-1-61208-251-6

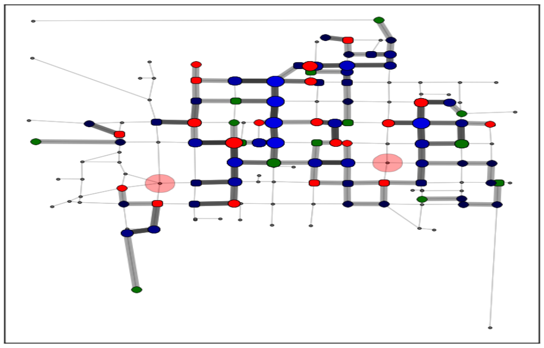

Figure 5. The 5% most probable resampled paths for an individual on the West Laramie street network, under the condition that he/she was not observed at either location.

spartan conditions: There is little *a priori* information available about collective movement and there are few specific observations available.

We presented a simple technique for estimating the observational coverage of a set of fixed observations — the probability that a path drawn from our prior path distribution will pass through at least one of the observation points. This method was used by a basic evolutionary algorithm to find near optimal positioning of fixed potential observation points within the graph. As it turns out, for the West Laramie street network graph, the observational coverage scales nearly logarithmically as the number of observational points is reduced. Indeed, just two fixed positions can cover nearly a fifth of all paths in the graph.

We also presented an efficient way to draw a path from that prior distribution in a probabilistically correct way. We expanded this algorithm to permit the path draw to incorporate conditions associated with ordered observations. We used this algorithm to construct five models of to match potential entity movement in a graph using five cases of hypothetical observations. These models were used to resample paths from the posterior path distributions to produce the most probable paths an entity might have taken given the different observational scenarios. Both quantitative and qualitative evidence was provided that such observations provide a great deal of contextual information about possible entity behavior.

In the future, we are interested in expanding our collective model of behavior to incorporate more sophisticated prior bias information. For example, there may be ways to incorporate time by combining the use of the Poisson length distribution with rate-of-travel and edge cost information over fixed time constraints. Additionally, we can incorporate edge transition probability information into the DRAWPATH algorithm to bias the draw based on a learned aggregate model. This may be facilitated but our complementary and parallel efforts to make use of biased random walker models as a means constructing edge transition probabilities.

REFERENCES

[1] A. Asahara, K. Maruyama, A. Sato, and K. Seto, "Pedestrian-movement prediction based on mixed markov-chain model," in *Proceedings of the* 19^{th} *International Conference on Advances in Geographic Information Systems*. New York, NY, USA: ACM Press, 2011, pp. 25–33.

[2] G. Vaca-Castano, A. Roshan Zamir, and M. Shah, "City scale geo-spatial trajectory estimation of a moving camera," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2012.

[3] K. A. De Jong, *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.

[4] T. Hägerstrand, "What about people in regional science?" *Papers in Regional Science*, vol. 24, no. 1, pp. 6–21, 1970.

[5] H. Miller, "Necessary space- time conditions for human interaction," *Environment and Planning B: Planning and Design*, vol. 32, no. 3, pp. 381–401, 2005.

[6] A. Crooks, C. Castle, and M. Batty, "Key challenges in agent-based modelling for geo-spatial simulation," *Computers, Env. and Urban Sys.*, vol. 32, no. 6, pp. 417 – 430, 2008.

[7] A. Willis, N. Gjersoe, C. Havard, J. Kerridge, and R. Kukla, "Human movement behaviour in urban spaces: Implications for the design and modelling of effective pedestrian environments." *Environment and Planning B Planning and Design*, vol. 31, no. 6, pp. 805–828, 2004.

[8] D. Helbing, P. Molnár, I. Farkas, and K. Bolay, "Self-organizing pedestrian movement," *Environment and Planning B*, vol. 28, no. 3, pp. 361–384, 2001.

[9] B. Hillier and S. Iida, "Network and psychological effects in urban movement," *Spatial Info. Theory*, pp. 475–490, 2005.

[10] W. Spears and S. Prager, "Evolutionary search for understanding movement dynamics on mixed networks," *GeoInformatica*, pp. 1–33, 2012.

[11] A. O'Connor, A. Zerger, and B. Itami, "Geo-temporal tracking and analysis of tourist movement," *Mathematics and Computers in Simulation*, vol. 69, no. 1-2, pp. 135–150, 2005.

[12] R. Church and C. VELLE, "The maximal covering location problem," *Papers in regional science*, vol. 32, no. 1, pp. 101–118, 1974.

[13] A. T. Murray, K. Kim, J. W. Davis, R. Machiraju, and R. Parent, "Coverage optimization to support security monitoring," *Computers, Environment and Urban Systems*, vol. 31, no. 2, pp. 133 – 147, 2007.

[14] N. Xiao, "A multiobjective evolutionary algorithm for surveillance sensor placement," *Environment and Planning B: Planning and Design*, vol. 35, pp. 935–948, 2008.

[15] T. H. Cormen, C. E. Leisserson, R. R. L., and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press, 2001.

[16] C. M. Grinstead and L. J. Snell, *Introduction to Probability*. American Mathematical Society, 2006, version dated 4 July, 2006.