Significant Route Discovery: A Summary of Results

Dev Oliver[1], Shashi Shekhar[1], Xun Zhou[1], Emre Eftelioglu[1], Michael R. Evans[1], Qiaodi Zhuang[1], James M. Kang[2], Renee Laubscher[2], and Christopher Farah[2]

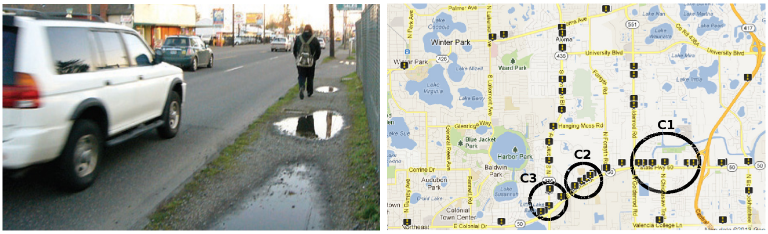
[1] Department of Computer Science, University of Minnesota, USA
[2] National Geospatial-Intelligence Agency, USA

Abstract. Given a spatial network and a collection of activities (e.g., pedestrian fatality reports, crime reports), Significant Route Discovery (SRD) finds all shortest paths in the spatial network where the concentration of activities is unusually high (i.e., statistically significant). SRD is important for societal applications in transportation safety, public safety, or public health such as finding routes with significant concentrations of accidents, crimes, or diseases. SRD is challenging because 1) there are a potentially large number of candidate routes ($\sim 10^{16}$) in a given dataset with millions of activities or road network nodes and 2) significance testing does not obey the monotonicity property. Previous work focused on finding circular areas of concentration, limiting its usefulness for finding significant linear routes on a network. SaTScan may miss many significant routes since a large fraction of the area bounded by circles for activities on a path will be empty. This paper proposes a novel algorithm for discovering statistically significant routes. To improve performance, the proposed algorithm features algorithmic refinements that prune unlikely paths and speeds up Monte Carlo simulation. We present a case study comparing the proposed statistically significant network-based analysis (i.e., shortest paths) to a statistically significant geometry-based analysis (e.g., circles) on pedestrian fatality data. Experimental results on real data show that the proposed algorithm, with our algorithmic refinements, yields substantial computational savings without reducing result quality.

1 Introduction

Significant Route Discovery (SRD) has important societal applications in transportation safety, public safety, or public health such as finding routes with significant concentrations of accidents, crimes, or diseases. In transportation safety, domain experts attribute pedestrian fatalities largely to the design of streets, which have been engineered for speeding traffic with little or no provision for people on foot, in wheelchairs, or on bicycles [1]. In urban areas, more than 56% of the pedestrian fatalities in the US (2007-2008) occurred on arterial roads [1]. Figure 1(a) shows an example of a pedestrian at risk on a road without proper sidewalks. This lack of basic infrastructure can be lethal. Figure 1(b) shows a map of pedestrian fatalities that occurred on Orlando roads from 2000 - 2009. Transportation planners and engineers need tools to assist them in identifying which frequently used road segments/stretches pose significant risks for pedestrians and consequently should be redesigned.

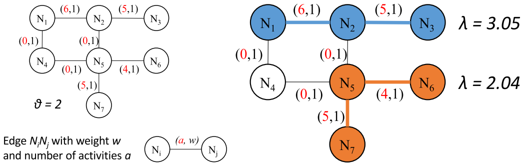
M. Duckham et al. (Eds.): GIScience 2014, LNCS 8728, pp. 284–300, 2014.



(a) (b)

Fig. 1. (a) Pedestrian at risk on a road without proper sidewalks [1] (b) Pedestrian fatalities occurring on arterials in Orlando, FL [2]. A large fraction of the bounding circles (e.g., C1, C2) of significant routes are empty.

Informally, the Significant Route Discovery (SRD) problem can be defined as follows: given a spatial network, a collection of activities (e.g., pedestrian fatality reports, crime reports), and a likelihood threshold θ, find all shortest paths in the spatial network where the concentration of activities is unusually high (i.e., statistically significant) and the likelihood exceeds θ. Depending on the domain, an activity may be the location of a pedestrian fatality, a carjacking, a train accident, etc. Figures 2(a) and 2(b) illustrate an input and output example of SRD, respectively. The input consists of seven nodes, six edges (with edge weights set to 1 for illustration purposes, shown as the second number on each edge), twenty activities (shown as the first number in red on each edge), and $\theta = 2$, indicating that we are interested in shortest paths whose likelihood exceeds $\theta = 2$. The output contains two shortest paths, $\langle N_1, N_2, N_3 \rangle$ and $\langle N_6, N_5, N_7 \rangle$ that are at least twice as likely to have pedestrian fatalities.



(a) Input (b) Output

Fig. 2. Example of Significant Route Discovery

SRD is challenging due to the potentially large number of candidate routes ($\sim 10^{16}$) in a given dataset with millions of activities or road network nodes. For large roadmaps such as the 100 million road-segments in the US, this results in prohibitive shortest path computation times. Additionally, significance testing does not obey the monotonicity property, meaning that there is no ordering between the likelihood of a path and its super-paths, or vice-versa. In other metrics such as activity count, for example, a path will always have less than or equal to the number of activities of its super-paths, a property which may be exploited for computational speedup. However, this property does not hold for significance testing. Furthermore, depending on the method used to determine statistical significance, computation times may also be impacted (e.g., $m = 1000$ Monte Carlo simulations may be required to calculate statistical significance).

Related Work and their Limitations. Dividing spatial data into statistically significant groups is an important task in many domains (e.g., transportation planning, public health, epidemiology, climate science, etc.). Previous methods for this type of partitioning have generally been geometry-based [3–6] or network-based [7–10].

Geometry-based techniques [3, 4, 6] partition spatial data using geometric shapes (e.g., circles, rectangles). This is useful in domains such as public health, where finding spatial clusters with a higher density of disease is of interest for understanding the distribution and spread of diseases, outbreak detection, etc. Kulldorff, et al proposed a spatial scan statistics framework for disease outbreak detection [3]. The spatial scan statistic employs a likelihood ratio test where the null hypothesis is the probability that disease inside a region is the same as outside, and the alternate hypothesis is that there is a higher probability of disease inside than outside. All the spatial regions, represented by a circle or ellipsoid in the spatial framework, are enumerated and the one that maximizes the likelihood ratio is identified as a candidate. However, if we apply SaTScan to a road network, many significant routes may be missed since a large fraction of the area bounded by circles for activities on a path will be empty, as shown in Figure 1(b). Furthermore, geometry-based techniques may not be appropriate for modeling linear clusters, which are formed when the underlying generator of the phenomena is inherently linear (e.g., pedestrian fatalities, railroad accidents, etcetera).

Network-based techniques [7–10], on the other hand, leverage the underlying spatial network when partitioning spatial data. For example, Linear Intersecting Paths (LIP) [9] and Constrained Minimum Spanning Trees (CMST) [7] utilize a subgraph (e.g., a path or tree) to discover statistically significant groups.

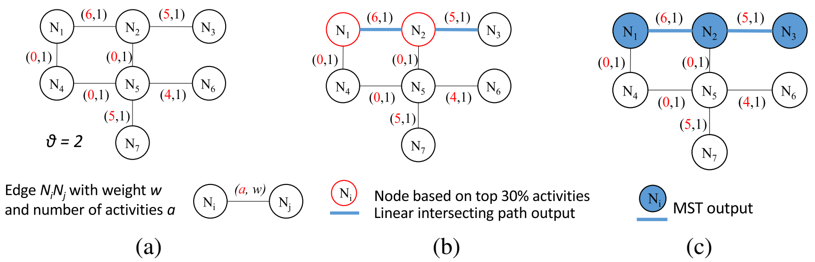


Fig. 3. Example (a) Input, (b) Output of Linear Intersecting Paths (LIP) [9], (c) Output of Constrained Minimum Spanning Trees (CMST) [7] (Best in color)

In LIP [9], one anomalous sub-component of a set of connected paths that intersect each other is discovered. The connected paths are based on locations in the spatial network with the highest percentage of activities, specified by the user. Hence the likelihood ratio is only evaluated on a portion of the graph specified by this percentage, not the entire spatial network. Figure 3 shows an example input and output of LIP. The user-specified percentage is 30%, which means all the candidates will have paths containing edge $\langle N_1, N_2 \rangle$ since this edge has six activities (out of a possible 20 activities). Examples of possible candidates are $\langle N_1, N_2, N_3 \rangle$, $\langle N_1, N_2, N_5 \rangle$, $\langle N_2, N_1, N_4 \rangle$, $\langle N_1, N_2, N_5, N_7 \rangle$, etc. The output is $\langle N_1, N_2, N_3 \rangle$, since it has the highest likelihood (Section 2 details how the likelihood ratio is calculated). However, in addition to returning only one statistically significant component, the results of this approach are sensitive to the percentage of the network selected. If the percentage is too high, the number of candidates may be highly restricted, which could result in not identifying statistically significant regions of interest. If the percentage is too low, LIP may be computationally prohibitive due to the large number of candidates.

CMST [7] finds one statistically significant tree in the spatial network. Figure 3(c) shows an example of this approach. Here the output is $\langle N_1, N_2, N_3 \rangle$, since this tree has the highest likelihood. However, in addition to returning only one statistically significant tree, the size of the tree is restricted, which could result in not identifying statistically significant regions of interest.

In contrast to previous methods, the proposed approach finds multiple statistically significant routes in the spatial network.

Contributions. Our contributions are summarized as follows:

- We introduce the problem of significant route discovery using shortest paths.
- We propose the Smart Significant Route Miner (SmartSRM) algorithm with algorithmic refinements that improve performance by pruning unlikely paths and speeding up Monte Carlo simulation. SmartSRM finds multiple significant routes in the spatial network.
- We present a case study comparing the proposed significant network-based analysis (i.e., shortest paths) to a significant geometry-based analysis (e.g., circles) on pedestrian fatality data.
- Experimental results on real data show that the proposed algorithm, with our refinements, yields substantial computational savings over a naïve approach without reducing result quality.

Scope and Outline of the Paper. This work focuses on finding significant discrete activity events (e.g., pedestrian fatalities, crime reports) associated with a point on a network. This does not imply that all activities must necessarily be associated with a point in a street. In addition, other network properties such as GPS trajectories and traffic densities of road networks [11] are not considered. In this work, it is assumed that the number of activities on the road network is fixed and does not change over time. A dynamically changing number of activities is presently beyond the scope of this research, as are techniques that do not explore statistical significance (e.g., DBScan [12], K-Means [13], KMR [14], and Maximum Subgraph Finding [15]). Furthermore, resolving activity hotspots to the sub-arc level requires a dynamic segmentation data model (currently not explored) that will introduce additional nodes and may create a computational bottleneck. Finally, modeling stochastic route choice (where one or several of the edge attributes are not deterministic [16]) also falls outside the scope of this paper.

The paper is organized as follows: Section 2 presents the basic concepts and problem statement of SRD. Section 3 presents both the Naïve and Smart Significant Route Miner algorithms to solve SRD. Section 4 presents a case study comparing the proposed significant network-based output (i.e., shortest paths) to a significant geometry-based output (e.g., circles) on pedestrian fatality data. The experimental evaluation is covered in Section 5. Section 6 presents a discussion. Section 7 concludes the paper and previews future work.

2 Basic Concepts and Problem Statement

This section introduces several key concepts in SRD and presents a formal problem statement.

2.1 Basic Concepts

We define our basic concepts as follows:

Definition 1. *A* ***spatial network*** $G = (N, E)$ *consists of a node set* N *and an edge set* E*, where each element* u *in* N *is associated with a pair of real numbers* (x, y) *representing the spatial location of the node in a Euclidean plane [17]. Edge set* E *is a subset of the cross product* $N \times N$*. Each element* $e = (u, v)$ *in* E *is an edge that joins node* u *to node* v*.*

Figure 2(a) shows an example of a spatial network where circles represent nodes and lines represent edges. A road network is an example of a spatial network where nodes represent street intersections and edges represent streets.

Definition 2. *An* ***activity set*** A *is a collection of activities. An* ***activity*** $a \in A$ *is an object of interest associated with only one edge* $e \in E$*.*

In transportation planning, an activity may be the location of a pedestrian fatality; in crime analysis, an activity may be the location of a theft. Each edge in Figure 2(a) is associated with a number of activities (e.g., edge $\langle N_1, N_2 \rangle$ has 6 activities).

Definition 3. *The* ***activity coverage inside a path****,* a_p*, is the number of activities on* p*. The* ***activity coverage outside*** p *is* $|A| - a_p$*, where* $|A|$ *is the total number of activities in the spatial network,* G*.*

For example, in Figure 2(a), the activity coverage *inside* path $\langle N_1, N_2, N_3 \rangle$ is 11 whereas the activity coverage *outside* $\langle N_1, N_2, N_3 \rangle$ is $20 - 11 = 9$.

Definition 4. *The* ***weight inside a path****,* w_p*, is the sum of weights of all edges in* p*. The* ***weight outside*** p *is* $|W| - w_p$*, where* $|W|$ *is sum of weights of all edges in* G*.*

In Figure 2(a), the weight *inside* $\langle N_1, N_2, N_3 \rangle$ is 2 whereas the weight *outside* $\langle N_1, N_2, N_3 \rangle$ is $7 - 2 = 5$.

Definition 5. *The **likelihood ratio of path** p, $\lambda_p = \frac{a_p \div w_p}{(|A| - a_p) \div (|W| - w_p)}$ [3, 10].*

The likelihood ratio of path p, λ_p, is the ratio of the activity density *inside* path p to the activity density *outside* p. Activity density may be estimated in different ways across different domains. In transportation planning, activity density inside p may be estimated using $\frac{a_p}{VMT}$, where VMT is vehicle miles traveled (i.e., the total number of miles driven by all vehicles within a given time period and geographic area). Path weight may also be used to estimate activity density [10]. In Figure 2(a), $\lambda_{\langle N_1, N_2, N_3 \rangle} = \frac{11 \div 2}{9 \div 5} = 3.05$.

Definition 6. *An **active edge** is an edge $e \in E$ that has 1 or more activities. An **active node** is a node u joined by an active edge. An **inactive node** is a node that is not joined by any active edges.*

Edges $\langle N_1, N_2 \rangle$ and $\langle N_2, N_3 \rangle$ in Figure 2(a) are active edges because they each have at least one activity, and nodes N_1, N_2, N_3, N_5, N_6, and N_7 are all active nodes because they are all joined by active edges. By contrast, Node N_4 is an inactive node because it is not joined by any active edges.

Definition 7. *A **super-path** of path p is any path sp that contains p, where sp is a subset of G. A **sub-path** is a path making up part of the super-path.*

For example, in Figure 2(a), $\langle N_1, N_2, N_5, N_6 \rangle$ and $\langle N_1, N_2, N_5, N_7 \rangle$ are super-paths of $\langle N_1, N_2, N_5 \rangle$. Conversely, $\langle N_1, N_2, N_5 \rangle$ is a sub-path of $\langle N_1, N_2, N_5, N_6 \rangle$.

2.2 Problem Statement

The problem of Significant Route Discovery (SRD) can be expressed as follows:

Given.

1. A spatial network $G = (N, E)$ with activity count function $a(u, v) \geq 0$ and weight function $w(u, v) > 0$ for each edge $e = (u, v) \in E$ (e.g., network distance),
2. A likelihood ratio (λ) threshold, θ,
3. A p-value,
4. m, indicating the number of Monte Carlo simulations

Find. All routes $r \in R$ with $\lambda_r \geq \theta$ and a p-value significance level

Objective. Computational efficiency

Constraints.

1. Each route $r \in R$ is a shortest path between its end-nodes,
2. $r_i \in R$ is not a subset of any $r_j \in R \, \forall r_i, r_j \in R$ where $r_i \neq r_j$,
3. Each route $r \in R$ starts and ends with active nodes,
4. Correctness and completeness

The spatial network input for SRD is defined in Definition 1. The θ input is a threshold indicating the minimum desired likelihood ratio. The p-value input is the desired level of statistical significance and m indicates the number of Monte Carlo simulations for determining statistical significance. The output for SRD are all shortest paths meeting the desired likelihood ratio and level of statistical significance. The shortest paths returned are constrained so that they are not sub-paths of any other path in the output. This constraint aims to improve solution quality by reducing redundancy in the paths returned. The output is also constrained such that the shortest paths returned start and end with active nodes. This constraint also aims to improve solution quality by ignoring edges at the start and/or end of a path that do not have any activities.

Example. The network in Figure 2(a) can be viewed as a road network, composed of streets (edges) and intersections (nodes). The aim is to find significant shortest paths that meet the given likelihood threshold of 2. In other words, find shortest paths that are twice as likely to have pedestrian fatalities. In a transportation planning scenario, identifying such routes would guide street redesign efforts to reduce the risk of pedestrian fatalities (e.g., adding sidewalks, crosswalks, pedestrian refuges, street lighting, etcetera). In Figure 2(b), routes $\langle N_1, N_2, N_3\rangle$ and $\langle N_6, N_5, N_7\rangle$ are returned since they are shortest paths whose likelihood exceeds $\theta = 2$, they start and end with active nodes, and they are not sub-paths of any other path in the output.

In an alternative formulation of the problem, the spatial network may be modeled with an activity count function $a(u) \geq 0$ for each node. The idea is that activities may also occur at nodes in addition to being distributed within network edges. In this way, the current approach may be extended to capture activities at nodes (e.g., vehicle accidents). If activities are modeled as counts at each node, this may alter the computational structure. We plan to investigate this in future work.

3 Proposed Approach

First we describe a naïve version of our miner, Naïve Significant Route Miner (NaïveSRM). Then we present our proposed Smart Significant Route Miner (SmartSRM) with its two algorithmic refinements, Likelihood Pruning and Monte Carlo Speedup.

3.1 Naïve Significant Route Miner (NaïveSRM) Algorithm

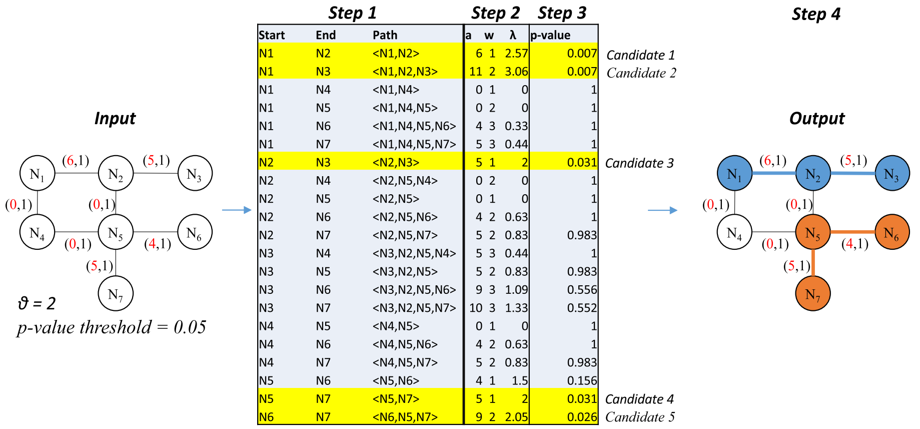
Algorithm 1 presents the pseudocode for the NaïveSRM approach. The basic idea behind the algorithm is to find all statistically significant shortest paths in the spatial network whose likelihood exceeds θ, under the constraints that the shortest paths returned are a) not sub-paths of any other path in the output and b) both start and end with active nodes. Algorithm 1 proceeds by calculating all shortest paths, P, in the spatial network (Line 1). Line 2 evaluates each shortest path in P to determine if it meets the given likelihood threshold θ to form a $Candidates$ set. In line 3, the statistical significance of each shortest path in $Candidates$ is evaluated and the significant routes are stored in $SigRoutes$. In order to assess statistical significance, all shortest paths in each of the

Algorithm 1. Naïve Significant Route Miner (NaïveSRM) Algorithm

```
Input:
   1) A spatial network $G = (N, E)$ with activity count function $a(u, v) \geq 0$ and
   weight function $w(u, v) > 0$ for each edge $e = (u, v) \in E$ (e.g., network distance),
   2) A likelihood ratio ($\lambda$) threshold, $\theta$,
   3) A p-value threshold,
   4) $m$, indicating the number of Monte Carlo simulations
Output:
   All routes $r \in R$ with $\lambda_r \geq \theta$ and p-value significance level
Algorithm:
1: {Step 1:} $P \leftarrow$ calculate all-pairs shortest path in $G$
2: {Step 2:} $Candidates \leftarrow$ paths in $P$ starting and ending with active nodes having
   $\lambda \geq \theta$
3: {Step 3:} $SigRoutes \leftarrow$ significant paths in $Candidates$ using $m$ Monte Carlo
   simulations
4: {Step 4:} return paths that are not sub-paths of any other path in $SigRoutes$
```

m simulated graphs are used to calculate the p-value. In line 4, all paths in $SigRoutes$ that are not sub-paths of any other path in $SigRoutes$ are returned, and the algorithm terminates. The purpose of returning significant routes that are not sub-paths of any other path is to improve solution quality. For example, if $\langle N_1, N_2 \rangle$ and $\langle N_1, N_2, N_3 \rangle$ are both found to be significant, only $\langle N_1, N_2, N_3 \rangle$ is returned.

NaïveSRM Example. Figure 4 shows an example execution trace of NaïveSRM. The spatial network has 7 nodes, 6 edges, and 20 activities, represented by the first number in red on each edge (e.g., edge $\langle N_1, N_2 \rangle$ has six activities). The given likelihood ratio threshold θ is set to 2 and the p-value is set to 0.05.



Step 1			Step 2			Step 3	
Start	End	Path	a	w	λ	p-value	
N1	N2	<N1,N2>	6	1	2.57	0.007	Candidate 1
N1	N3	<N1,N2,N3>	11	2	3.06	0.007	Candidate 2
N1	N4	<N1,N4>	0	1	0	1	
N1	N5	<N1,N4,N5>	0	2	0	1	
N1	N6	<N1,N4,N5,N6>	4	3	0.33	1	
N1	N7	<N1,N4,N5,N7>	5	3	0.44	1	
N2	N3	<N2,N3>	5	1	2	0.031	Candidate 3
N2	N4	<N2,N5,N4>	0	2	0	1	
N2	N5	<N2,N5>	0	1	0	1	
N2	N6	<N2,N5,N6>	4	2	0.63	1	
N2	N7	<N2,N5,N7>	5	2	0.83	0.983	
N3	N4	<N3,N2,N5,N4>	5	3	0.44	1	
N3	N5	<N3,N2,N5>	5	2	0.83	0.983	
N3	N6	<N3,N2,N5,N6>	9	3	1.09	0.556	
N3	N7	<N3,N2,N5,N7>	10	3	1.33	0.552	
N4	N5	<N4,N5>	0	1	0	1	
N4	N6	<N4,N5,N6>	4	2	0.63	1	
N4	N7	<N4,N5,N7>	5	2	0.83	0.983	
N5	N6	<N5,N6>	4	1	1.5	0.156	
N5	N7	<N5,N7>	5	1	2	0.031	Candidate 4
N6	N7	<N6,N5,N7>	9	2	2.05	0.026	Candidate 5

Fig. 4. Execution trace of Naïve Significant Route Miner (NaïveSRM). Circles represent nodes and lines represent edges (Best in color).

In step 1 of Figure 4, all shortest paths in the given spatial network are calculated. For example, the shortest path between nodes N_1 and N_3 is $\langle N_1, N_2, N_3 \rangle$. Next, in step 2, the likelihood ratio, λ, for each shortest path is determined (see Definition 5) and those whose $\lambda \geq \theta$ are stored as candidate solutions. In the figure, the five highlighted paths $\langle N_1, N_2 \rangle$, $\langle N_1, N_2, N_3 \rangle$, $\langle N_2, N_3 \rangle$, $\langle N_5, N_7 \rangle$, and $\langle N_6, N_5, N_7 \rangle$ are all candidates since their likelihood ratios meet or exceed the threshold of 2. In step 3, the statistical significance of each candidate is calculated using Monte Carlo simulations (discussed next). All five candidates meet the p-value threshold of 0.05. In step 4, the paths among significant paths that are not sub-paths of any other path are returned. In this example, paths $\langle N_1, N_2, N_3 \rangle$ and $\langle N_6, N_5, N_7 \rangle$ are returned. Paths $\langle N_1, N_2 \rangle$, $\langle N_2, N_3 \rangle$, and $\langle N_5, N_7 \rangle$ were not returned (even though they met and exceeded the likelihood and p-value thresholds) because they are each sub-paths of the two paths that were returned.

Finding Significant Paths. Each shortest path in SRM is evaluated for statistical significance using Monte Carlo simulations to determine whether or not it is truly anomalous. Here the null hypothesis states that the paths identified by SRM are random or by chance alone. The likelihood ratio is associated with a p-value to decide whether the null hypothesis should be rejected in the hypothesis test. The p-value is the probability of obtaining a value of a given likelihood ratio as equally or more extreme than that observed by chance alone.

In the Monte Carlo simulations, each activity in the original graph G is randomly associated with an edge so that the number of activities on each edge is shuffled, forming a new graph G_s. Note that all the activities in G are present in G_s, with no activities added or removed; the original activities in G are now shuffled so they may be on different edges in G_s. We then compare the highest likelihood threshold λ_{maxG_s} of randomized G_s with the highest λ_{maxG} of original G. If the original one is smaller (i.e., $\lambda_{maxG} < \lambda_{maxG_s}$), then $p = p + 1$. The above process repeats m times and after it terminates, the p-value is subsequently p/m. Paths whose p-values are less than or equal to the given p-value threshold are deemed statistically significant.

3.2 Smart Significant Route Miner (SmartSRM) Algorithm

Algorithm 2 presents the pseudocode for the proposed SmartSRM approach. The algorithm features two key ideas for achieving computational savings while maintaining result quality: Likelihood Pruning and Monte Carlo Speedup.

Likelihood Pruning: Likelihood pruning aims to avoid calculating all shortest paths in G based on the given threshold θ. It is based on the idea that for each shortest path p, it is possible to determine an upper bound likelihood ratio for the super-paths rooted at p's start node, without calculating those super-paths.

Definition 8. *The **upperbound likelihood ratio for path** p, $\hat{\lambda_p} = \frac{\hat{a_p} \div \hat{w_p}}{(|A| - \hat{a_p}) \div (|W| - \hat{w_p})}$, where $\hat{a_p} = a_p + (|A| - a_t)$ (where a_t is the number of activities in the shortest path tree rooted at p's source node) and $\hat{w_p}$ is the weight of the shortest super-path of p, rooted at p's start node.*

The intuition behind the upper bound likelihood ratio for path p is that (1) the number of activities on all of p's super-paths rooted at p's start node are bounded by the number of activities in the spatial network minus the number of activities in the current shortest path tree rooted at the source node in p and (2) the weight of any super-path of p is at least the weight of the closest edge to p plus p's weight.

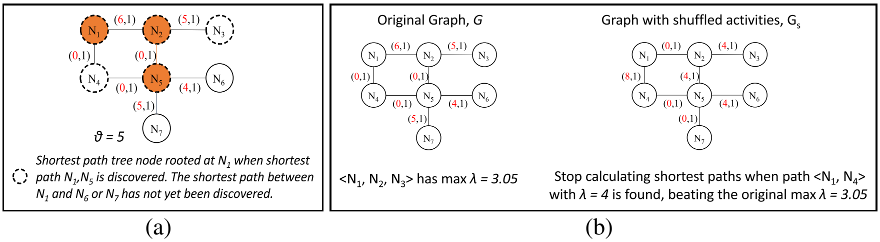
Algorithm 2. Smart Significant Route Miner (SmartSRM) Algorithm

```
Inputs and Outputs for SmartSRM are same as NaïveSRM
Algorithm:
   {Step 1: Likelihood Pruning}
1: for each $s \in$ active nodes in $G$ do
2:    Initialize $D[v] \leftarrow$ inf; $Pred[v] \leftarrow \emptyset$; $\hat{\Lambda}[v] \leftarrow \theta$; $a[v] \leftarrow 0$; $a_t \leftarrow 0$; $D[s] \leftarrow 0$; $PQ \leftarrow N$
3:    while $PQ \neq \emptyset$ do
4:       $u \leftarrow$ node in $PQ$ with smallest distance in $D[]$; $P \leftarrow$ shortest path $(s,u)$ in $Pred[]$
5:       $a_t \leftarrow a_t+$ number of activities on edge $Pred[u]$
6:       if $\hat{\Lambda}[v] \geq \theta$ then
7:          for each $v$ adjacent to $u$ do
8:             $sum \leftarrow D[u] + w(u,v)$
9:             if $sum < D[v]$ then
10:               $D[v] \leftarrow sum$; update $v$'s position in $PQ$ based on $sum$; $Pred[v] \leftarrow u$
11:               $a[v] \leftarrow a[u] + a(u,v)$; $\hat{w} \leftarrow sum+$ weight of closest neighbor $w(u,v)$
12:               $\hat{\Lambda}[v] \leftarrow$ calculate $\hat{\lambda_{sv}}$ based on $a[v], a_t$ and $\hat{w}$
13: {Step 2:} $Candidates$ ← paths in $P$ starting and ending with active nodes having $\lambda \geq \theta$
   {Step 3: Monte Carlo Speedup}
14: $\lambda_{maxG} \leftarrow$ highest likelihood ratio in $G$
15: for each $simulation_1 .... simulation_m$ do
16:    $G_s \leftarrow$ assign activities in $G$ to random edges
17:    $\lambda_{maxG_{s_i}} \leftarrow 0$
18:    for each shortest path $p \in G_s$ do
19:       if $\lambda_p > \lambda_{maxG_{s_i}}$ then
20:          $\lambda_{maxG_{s_i}} \leftarrow \lambda_p$; $p_{maxr} \leftarrow p_{maxr} + 1$
21:          if $p_{maxr}/N \leq$ p-value threshold then return $\emptyset$
22:          if $\lambda_p > \lambda_{maxG}$ then break
23:    for each route $r \in Candidates$ do
24:       if $\lambda_{maxG_{s_i}} > \lambda_{maxG}$ then $p_r \leftarrow p_r + 1$
25: for each route $r \in Candidates$ do
26:    if $p_r/N \leq$ p-value threshold then $SigRoutes \leftarrow r$
27: {Step 4:} return paths that are not sub-paths of any other path in $SigRoutes$
```

Lines 1-12 of Algorithm 2 shows the pseudocode for likelihood pruning, which is similar to Dijkstra's algorithm [18] with a few exceptions: (1) the shortest paths from a single active node to all destinations are calculated for all active nodes in the spatial network, (2) if the upper bound likelihood ratio for path $\langle s...u \rangle$ is below the given likelihood threshold θ, u's neighbors are not visited (line 6), and (3) upperbound statistics are calculated and updated each time the weight from source s to a node v is updated (lines 9-12).

Likelihood Pruning Example. Figure 5(a) illustrates the basic idea behind likelihood pruning. In this example, we have set the likelihood threshold to $\theta = 5$, indicating that we are interested in paths that are five times as likely to have pedestrian fatalities. During the algorithm's execution, at some point the source node becomes N_1, and the shortest path between N_1 and every other active node in the spatial network is calculated.

When the shortest path between N_1 and N_5 is calculated, the upper bound likelihood ratio for path $\langle N_1, N_2, N_5 \rangle$ is determined to be 4, since based on Definition 8, the calculation would be $\frac{(6+(20-11))\div 3}{(20-((6+(20-11))\div(7-3))}$, where $\hat{a_p} = 6 + (20 - 11) = 15$ and $\hat{w_p} = 2 + 1 = 3$. We can, therefore, avoid calculating the shortest paths $\langle N_1, N_2, N_5, N_6 \rangle$ and $\langle N_1, N_2, N_5, N_7 \rangle$ for $\theta = 5$.



Shortest path tree node rooted at N_1 when shortest path N_1,N_5 is discovered. The shortest path between N_1 and N_6 or N_7 has not yet been discovered.

<N_1, N_2, N_3> has max $\lambda = 3.05$

Stop calculating shortest paths when path <N_1, N_4> with $\lambda = 4$ is found, beating the original max $\lambda = 3.05$

(a) (b)

Fig. 5. (a) Example of Likelihood Pruning. Since we know the upper-bound likelihood for $\langle N_1, N_2, N_5 \rangle$ is 4, we can avoid calculating the shortest paths $\langle N_1, N_2, N_5, N_6 \rangle$ and $\langle N_1, N_2, N_5, N_7 \rangle$ for $\theta = 5$. (b) Example of Monte Carlo Speedup. (Best in color).

Monte Carlo Speedup: Monte Carlo speedup aims to calculate the p-value without considering all shortest paths in each simulated graph. The basic idea is that once a shortest path in the simulated graph is found to have a higher likelihood ratio than the maximum likelihood ratio in the original graph, the simulation immediately ends with the p-value being incremented. In other words, there is no reason to keep looking at all shortest paths in the simulated graph if we find one that already beats the maximum likelihood ratio in the original graph. Additionally, Monte Carlo speedup stops all simulations the moment p out of m simulations are found where the simulated likelihood ratio beats the original maximum likelihood ratio. In other words, there is no reason to execute all m simulations if we find that the p-value threshold will not be met. The pseudocode for Monte Carlo speedup is presented in Lines 14-26 of Algorithm 2.

Monte Carlo Speedup Example. Figure 5(b) illustrates one of the basic ideas behind Monte Carlo speedup. In this example, the graph on the left is the original graph G whereas the graph on the right, G_s, represents one simulation with the activities shuffled. In G_s, instead of looking at all 42 shortest paths, we can stop and increment p the moment a path that has a likelihood higher than the maximum likelihood in G is found. In this case, that path would be $\langle N_1, N_2 \rangle$ (on the right of the figure), with a likelihood ratio of 4.

SmartSRM uses filter and refine techniques (e.g., Likelihood Ratio pruning and Monte Carlo speedup) to achieve computational savings. Filter and refine techniques may not change worst case complexity but they can reduce runtime. Likelihood Ratio pruning creates a boundary via the upperbound likelihood ratio such that not all destinations are visited from each source node. Some of the destinations are pruned because the shortest paths to them will never meet the likelihood ratio threshold. Monte Carlo speedup avoids generating all shortest paths in cases where a shortest path in the simulated dataset has a higher likelihood ratio than the shortest paths in the original dataset.

Monte Carlo speedup also terminates early if the p-value threshold will not be met based on the number of times the maximum likelihood ratio in the simulated dataset beats the maximum likelihood ratio in the original dataset.

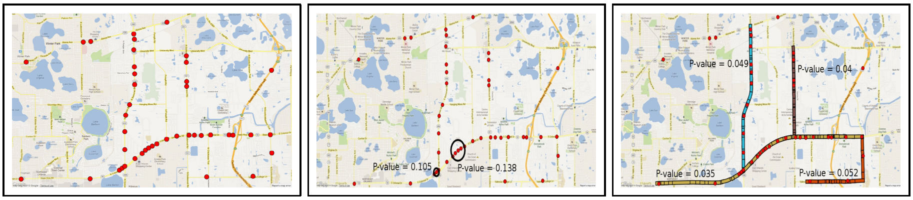
The computational costs of NaïveSRM and SmartSRM stem from 1) the cost of calculating all pair shortest paths and 2) the cost of assessing statistical significance for all shortest paths in the spatial network. For NaïveSRM, the total cost is $(N^2logN \times C_{\lambda_p}) + (m \times N^2logN \times C_{\lambda_p})$, where N is the set of nodes, N^2logN is the cost of calculating shortest paths in the spatial network, C_{λ_p} is the cost of calculating the likelihood ratio of path p, and m is the number of Monte Carlo simulations.

For SmartSRM, the total cost is $((N \times r_{\hat{\lambda}})^2logN \times C_{\lambda_p}) + (m \times (N^2logN \times r_m) \times C_{\lambda_p})$, where N is the set of nodes, $(N \times r_{\hat{\lambda}})^2logN$ is the cost of calculating shortest paths for a set of shortest paths that is a superset of all paths in G with $\lambda_p \geq \theta$, $r_{\hat{\lambda}}$ (whose value is between 0 and 1) is the ratio of shortest paths with $\lambda_p \geq \theta$ to all shortest paths, C_{λ_p} is the cost of calculating the likelihood ratio of path p, m is the number of Monte Carlo simulations, and r_m (whose value is between 0 and 1) is the ratio of shortest paths calculated before finding a path whose likelihood beats the maximum likelihood in the original graph to all shortest paths.

In summary, SmartSRM may only consider a fraction of the paths considered by NaïveSRM, both in calculating all pair shortest paths and assessing statistical significance for all shortest paths in the spatial network.

4 Case Study

We conducted a qualitative evaluation of SmartSRM, comparing its analysis with the analysis of SaTScan [19] (continuous Poisson process) on a real pedestrian fatality data set [2], shown in Figure 6(a). As noted earlier, SaTScan discovers areas of significant activity that are represented as circles on the spatial network while SmartSRM discovers significant shortest paths. The input consisted of 43 pedestrian fatalities (represented as dots) in Orlando, Florida occurring between 2000 and 2009. For each edge (portion of road) in the network, fatality count was aggregated, yielding overall activity, and weight was the actual road network distance. The maps were prepared using QGIS' Open Layers plugin [20], and the road network was from the US Census Bureaus TIGER/Line Shapefiles [21].



(a) Input (b) SaTScan (c) SRM

Fig. 6. Comparing SmartSRM and SaTScan's output for a p-value threshold of 0.15 and $\theta = 1.75$ on pedestrian fatality data from Orlando, FL [2] (Best in color)

When evaluating the techniques, we consider the outputs of circles vs. shortest paths. While p-value thresholds of 0.05 or lower are often desired, we used a p-value threshold of 0.15 because the circles chosen by SaTScan had high p-values for this dataset. As noted earlier, pedestrian fatalities usually occur on streets, particularly along arterial roadways [1]. Thus this activity can be said to have a linear generator. However, the results generated by SaTScan do not capture this. From Figure 6(b), it is clear that the circle-based output is meant for areas, not streets. In contrast, the shortest paths detected by SmartSRM fully capture the significant activities on the arterial roads (some of the paths in Figure 6(c) are overlapping). Furthermore, the paths in the figure make sense in this context due to the inherently linear nature of the activities.

5 Experimental Evaluation

The goal of our experiments was to evaluate the scalability of the proposed approach by varying and observing the effect of three workload parameters: nodes, likelihood ratio threshold θ, and p-value threshold. All experiments were performed on a Mac Pro with a 2 x Xeon Quad Core 2.26 GHz processor and 16 GB RAM. For each workload experiment we compared Naïve Significant Route Miner (NaïveSRM) and our Smart Significant Route Miner (SmartSRM).

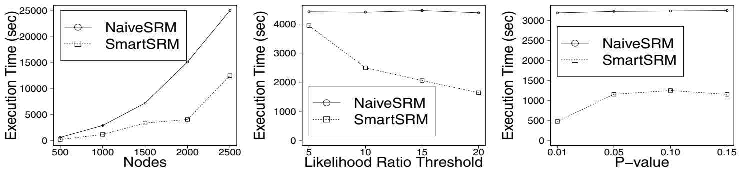
5.1 Experiment Data Set

Our experiments were performed on real-world data obtained from the Fatality Analysis Reporting System (FARS) encyclopedia [2]. The dataset contained geospatial and temporal data describing 487 pedestrian fatalities in Orange County, FL (which includes Orlando), from 2001 to 2011. For each edge (portion of road) in the network, fatality count was aggregated, yielding overall activity, and weight was the actual road network distance. The road network was obtained from the US Census Bureau's TIGER/Line Shapefiles [21].

5.2 Experimental Results

Effect of the Number of Nodes. We varied the number of nodes from 500 to 2500, which is akin to varying the number of shortest paths (routes) from $250,000$ to $6,250,000$ (since there are $\binom{n}{2}$ shortest paths in the spatial network). We set the p-value threshold to 0.05, the number of Monte Carlo simulations to 100, and the likelihood ratio threshold θ to 20. Figure 7(a) gives the execution times. As can be seen, $SmartSRM$ is faster. Computational savings increases as the number of nodes increases due to Likelihood Pruning and Monte Carlo Speedup.

Effect of the Likelihood Ratio Threshold θ. The p-value was set to 0.05, the number of Monte Carlo simulations was set to 100, and the number of nodes was set to 1000. Figure 7(b) gives the execution times. Again, $SmartSRM$ beats the naïve algorithm. Computational savings increases as the likelihood ratio increases due to Likelihood Pruning and Monte Carlo Speedup.



(a) Nodes (b) LR threshold θ (c) P-value threshold

Fig. 7. Scalability of SRM with increasing (a) number of nodes, (b) likelihood ratio threshold θ, and (c) P-value Threshold

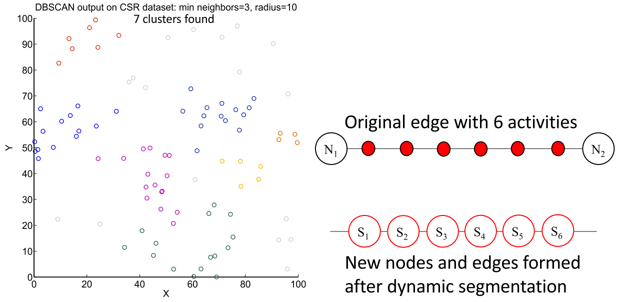
Effect of the P-value. The number of nodes was set to 1000, the likelihood ratio threshold θ was set to 20, and the number of Monte Carlo simulations was set to 100. Figure 7(c) gives the execution times. As can be seen, $SmartSRM$ is faster. Computational savings increases as the p-value increases due to Likelihood Pruning and Monte Carlo Speedup.

In summary, the experiments uniformly show that $SmartSRM$ is much better (2-3 times faster) than the naïve approach. This is because $SmartSRM$ prunes unlikely paths and speeds up Monte Carlo simulation.

6 Discussion

Non-Statistically Significant Techniques. Our work focuses on partitioning techniques that consider statistical significance. There are a myriad of other techniques that divide data into groups but that do not consider statistical significance including DBScan [12], K-Means [13], KMR [14], and Maximum Subgraph Finding [15]. However, statistical significance is important for determining the probability that an effect is not due to just chance alone. Post-processing the output of these techniques for statistical significance will not guarantee completeness as some of the clusters returned may not be statistically significant. For example, the algorithm from our previous work [14] on summarizing activities using routes may return routes that are not statistically significant. Figure 8(a) shows an example where DBScan [12] returns 7 chance clusters on a complete spatial randomness dataset.

Alternative network footprints. Summarizing significant network footprints of activities may be done using significant subgraphs, significant paths, significant shortest paths, etc. Each representation entails a tradeoff between fidelity and computational scalability. For example, subgraphs may offer accurate significant network footprints but their calculation may be computationally intensive due to their exponential number. As an initial step, we have selected shortest paths to summarize significant network footprints of activities. While shortest paths may lose some fidelity, they offer computational scalability because their number is bounded (i.e., $\binom{n}{2}$, where n is the number of nodes). The union of shortest paths may also be used to represent other network footprints.



(a) (b)

Fig. 8. (a) Colored dots are part of chance clusters identified by DBScan [12] on a complete spatial randomness dataset (b) Example of Dynamic Segmentation (Best in color)

Dynamic Segmentation. Resolving statistically significant routes to the sub-arc level requires a dynamic segmentation data model. In dynamic segmentation, the original nodes and edges in a statically segmented network (e.g., Figure 2(a)) are replaced by new nodes formed at the locations of activities, with new edges connecting these locations. Figure 8(b) shows an example where edge $\langle N_1, N_2 \rangle$ from Figure 2(a) has been dynamically segmented. As can be seen, the six activities on the original edge form six new nodes in the network, with new edges connecting these nodes. Dynamic segmentation has the potential to improve result quality in significant route discovery. This is because each segment in the dynamically segmented network structure corresponds to the locations of activities so the likelihood ratios of candidate routes are more precise. However, dynamic segmentation has the potential to introduce many new nodes in the spatial network, which could be computationally prohibitive for datasets with a large number of activities. Future research is needed to investigate this tradeoff.

7 Conclusion

This work explored the problem of significant route discovery in relation to important application domains such as preventing pedestrian fatalities and crime analysis. We proposed a Smart Significant Route Miner (SmartSRM) algorithm that discovers statistically significant shortest paths in a spatial network. SmartSRM uses Likelihood Pruning and Monte Carlo Speedup to enhance its performance and scalability. We presented a case study comparing SmartSRM with SaTScan on pedestrian fatality data. Experimental evaluation using real-world data indicated that the algorithmic refinements utilized by SmartSRM yielded substantial computational savings without sacrificing result quality.

In future work, we plan to explore other types of data that may not be associated with a point in a street (e.g., aggregated pedestrian fatality data at the zip code level). The present research is centered on finding high concentrations of activities whose counts and locations are deterministic. However, future work is needed to investigate attributes that may not be deterministic such as delay when moving between nodes, capacity constraints, etc. We will also generalize significant route discovery for all paths and explore additional spatial constraints (e.g., nearest neighbors). Finally, incorporating time and dynamic segmentation into SRD will be explored.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grants No. 1029711, IIS-1320580, 0940818 and IIS-1218168, the USDOD under Grants No. HM1582-08-1-0017, and HM0210-13-1-0005, and the Doctoral Dissertation Fellowship program at the University of Minnesota's graduate school. We would like to thank Kim Koffolt and the members of the University of Minnesota Spatial Computing Research Group for their comments.

References

1. Ernst, M., Lang, M., Davis, S.: Dangerous by design: Solving the epidemic of preventable pedestrian deaths. Transportation for America: Surface Transportation Policy Partnership, Washington, DC (2011)
2. National Highway Traffic Safety Administration (NHTSA): Fatality Analysis Reporting System (FARS) Encyclopedia, `http://www.nhtsa.gov/FARS`
3. Kulldorff, M.: A spatial scan statistic. Communications in Statistics-Theory and Methods 26(6), 1481–1496 (1997)
4. Neill, D.B., Moore, A.W.: Rapid detection of significant spatial clusters. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 256–265. ACM (2004)
5. Kulldorff, M., Mostashari, F., Duczmal, L., Katherine Yih, W., Kleinman, K., Platt, R.: Multivariate scan statistics for disease surveillance. Statistics in Medicine 26(8), 1824–1833 (2007)
6. Kulldorff, M.: Spatial scan statistics: Models, calculations, and applications. In: Scan Statistics and Applications, pp. 303–322. Springer (1999)
7. Costa, M.A., Assunção, R.M., Kulldorff, M.: Constrained spanning tree algorithms for irregularly-shaped spatial clustering. Computational Statistics & Data Analysis 56(6), 1771–1783 (2012)
8. Duczmal, L., Assuncao, R.: A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters. Computational Statistics & Data Analysis 45(2), 269–286 (2004)
9. Shi, L., Janeja, V.P.: Anomalous window discovery for linear intersecting paths. IEEE Transactions on Knowledge and Data Engineering 23(12), 1857–1871 (2011)
10. Janeja, V.P., Atluri, V.: Ls 3: A linear semantic scan statistic technique for detecting anomalous windows. In: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 493–497. ACM (2005)
11. Li, X., Han, J., Lee, J.-G., Gonzalez, H.: Traffic density-based discovery of hot routes in road networks. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 441–459. Springer, Heidelberg (2007)
12. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
13. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
14. Oliver, D., Bannur, A., Kang, J.M., Shekhar, S., Bousselaire, R.: A K-Main Routes Approach to Spatial Network Activity Summarization: A Summary of Results. In: IEEE International Conference on Data Mining Workshops (ICDMW), pp. 265–272 (2010)

15. Buchin, K., Cabello, S., Gudmundsson, J., Löffler, M., Luo, J., Rote, G., Silveira, R.I., Speckmann, B., Wolle, T.: Finding the most relevant fragments in networks. J. Graph Algorithms Appl. 14(2), 307–336 (2010)
16. Chawla, S., Roughgarden, T.: Single-source stochastic routing. In: Díaz, J., Jansen, K., Rolim, J.D.P., Zwick, U. (eds.) APPROX/RANDOM 2006. LNCS, vol. 4110, pp. 82–94. Springer, Heidelberg (2006)
17. Shekhar, S., Liu, D.: CCAM: A connectivity-clustered access method for networks and network computations. IEEE Transactions on Knowledge and Data Engineering 9(1), 102–119 (1997)
18. Cormen, T.: Introduction to algorithms. The MIT press (2001)
19. Kulldorff, M., Rand, K., Gherman, G., Williams, G., DeFrancesco, D.: SaTScan v 2.1: Software for the spatial and space-time scan statistics. National Cancer Institute, Bethesda (1998)
20. The QGIS Project: Quantum GIS OpenLayers Plugin,
http://plugins.qgis.org/plugins/openlayers_plugin/
(accessed: January 23, 2014)
21. US Census Bureau: Census TIGER/Line Shapefiles (2010),
http://www.census.gov/geo/maps-data/data/tiger-line.html
(accessed: January 23, 2014)