ANIMAL STAY REGION DETECTION AND BEHAVIOR ANALYSIS BASED ON GPS TRAJECTORIES

A Thesis

Presented to

The Faculty of the Graduate School

University of Missouri-Columbia

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Haidong Wang

Dr. Yi Shang, Thesis Supervisor

DECEMBER 2016

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

ANIMAL STAY REGION DETECTION AND BEHAVIOR ANALYSIS BASED ON GPS TRAJECTORIES

Presented by Haidong Wang

A candidate for the degree of Master Science

And hereby certify that, in their opinion, it is worthy of acceptance.

_____

Dr. Yi Shang

_____

Dr. Dong Xu

_____

Dr. Timothy Trull

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisor Dr. Yi Shang for his continuous support and valuable advices not only in this thesis but also throughout my senior year and graduate study. His broad scientific knowledge and research passion are my greatest motivation. It is my great honor to learn from him.

I would also like to thank Dr. Timothy Trull and Dr.Dong Xu for reviewing my thesis and providing so many insightful and valuable comments and suggestions on this thesis. I appreciate them taking precious time out of their busy schedule to serve on my thesis committee.

I would also like to thank my colleagues in my lab, they have always been very helpful and it is my greatest pleasure to work with them. Our friendship and memory together are the most valuable treasure in my life.

Finally, I would like to thank my family and friends for their constant support and encouragement during all these years. Without their understanding, it would be impossible for me to go through all the difficult time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Nowadays, GPS technology is becoming an important tool in tracking and understanding wild animal behaviors. For example, Missouri Department of Conservation (MDC) has put GPS collars on more than 80 black bears and more than 150 deer and collected a large amount of GPS data. In this project, several semantic analysis methods have been implemented and applied to GPS data provided by MDC. After the raw data are cleaned using outlier detection methods, stay regions in each GPS trajectory are detected using the SeqScan algorithm. Based on the stay regions, various statistics of individual animals and among different groups of animals, such as male and female, are generated to provide insights of animal behaviors and help answer questions that biologists are interested in. Multidimensional scaling technique is used to analyze and visualize relationships between different animals in terms of the overlaps of their stay regions. A software pipeline has been implemented to apply the proposed methods and a website has been created to show the results on Google map, which give the biologists a convenient tool to perform some quick analysis of raw GPS trajectories.

1. INTRODUCTION

In the recent few years, GPS system is more and more advanced. The system is not only very accurate, but also very fixable. Nowadays, there are lots of wearable tracking devices sense the movement of people, vehicles and even animals, generating large volumes of mobility data, which represent the traces of people's and animal's activity. This project will mainly focus on analyzing Missouri bear and deer activity with mobility data provided by Missouri department of conservation. Missouri department of conservation started collecting mobility data of bear and deer since the year 2010. They put GPS collar on the animals so that their server can receive the GPS data every certain time. They have been tracking about 100 bear and 150 dear, getting hundreds of thousands data. Although there are lots of data about animal movement, it still does not directly improve our understanding of meaning of animal GPS data. This means we currently do not have much study on how to fill up the semantic gap between the animal GPS trajectory data and the real semantic data. As a result of that, approaches are needed for answering the animal behaviors we want to learn about them though the massive data. The method presented in this paper is to extract stay regions of animals from their trajectories, represented as GPS data. A lot of work has been done on clustering spatial GPS points. However, not much has been done on clustering GPS points with timestamp, especially for this set of data. The dataset collected by MDC is real data recently, and no one has been worked on this dataset. After the stay regions

are generated, other statistic and analytic work will be done based on the generated stay regions. Also, a website is made to present labeled stay regions of every animal so that people can visualize the movement pattern of each animal. The stay region is defined as a portion of space which generally does not designate a precise geographical entity and where an object is significantly present for a period of time, in spite of relatively short periods of absence. Based on this assumption, we can assume an animal may have one or more stay regions. Hence, we can see the movement pattern of animals from one stay region to the next one. And we can split the whole trajectory into stays and exceptions. Stays mean the animal is in one of its stay region. Exceptions mean the animal is out of its stay regions. This means there are two possible conditions for each GPS point in the trajectory, either it is in its stay regions, or it is out of its stay regions. Under this assumption, we only care about the stay regions, want to do some statistic and analytic work based on the stay regions. For example, how long an animal would stay in one stay region, what is the difference between different sexes? Which bears appear in the same area in the same time? In order to have good results, the first and most crucial thing is to identify the stay regions. There are some related work about this, which will be introduced in the next section. To test the stay regions are correct, a wildlife trajectory generator to generate labeled trajectory that is similar to the real animal trajectory is used. The trajectory generator uses the stop and move model to generate stop or movement points and transits between those two models with Markov probability model, which will be introduced more in the next section. On the website, people can select one animal to check its stay regions. All points of a same stay region

are labeled as the sequence of the stay region and marked with a same color. Some statistic work are done based on the stay regions. To analysis the stay regions, this paper will focus on studying the overlaps among each animal.

2. RELATED WORKS

2.1 Spatial cluster model

There is a large of number of work related with spatial clustering. DBSCAN is one of good methods to work with our dataset. Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996.[1] It is a density-based clustering algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature Wireless body-area sensor networks have been considered a hot topic and used for a variety of applications in mobile health, physiological monitoring, and context aware computing. Some works are done based on DBSCAN, take [2], [3], [4] as examples. From [5], we can learn the basic idea of DBSCAN as following: Consider a set of points in some space to be clustered. For the purpose of DBSCAN clustering, the points are classified as core points, (density-) reachable points and outliers, as follows:

A point p is a core point if at least minPts points are within distance ε of it (including p). Those points are said to be directly reachable from p. No points are directly reachable from a non-core point.

A point q is reachable from p if there is a path p1, ..., pn with p1 = p and pn = q, where each pi+1 is directly reachable from pi (so all the points on the path must be core points, with the possible exception of q).

All points not reachable from any other point are outliers.

Now if p is a core point, then it forms a cluster together with all points (core or non-core) that are reachable from it. Each cluster contains at least one core point; non-core points can be part of a cluster, but they form its "edge", since they cannot be used to reach more points.

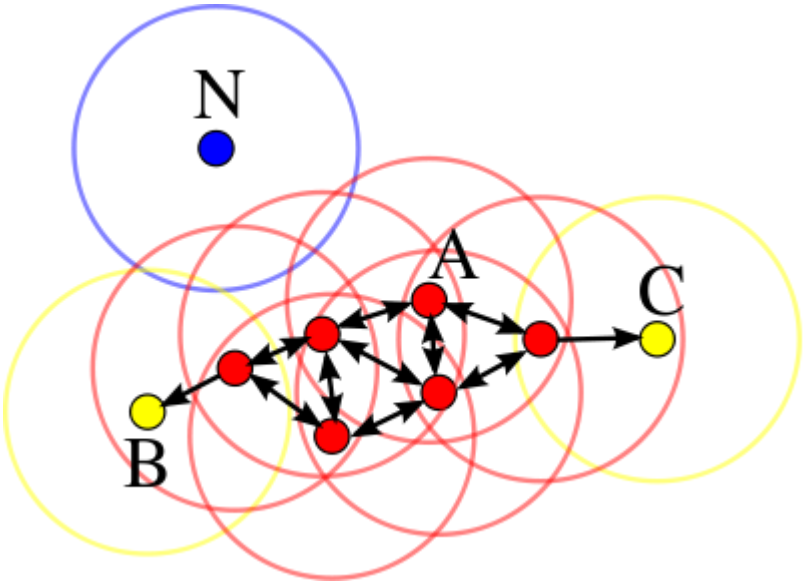


Figure 2-1 DBSCAN

In this Figure 2.1, minPts = 4. Point A and the other red points are core points, because the area surrounding these points in an ε radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other

core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor density-reachable.

2.2 Trajectory cluster model

To stady bear and deer trajectory, our first step is to cluster the points from trajectories. The trajectories are time-stamped sequence of points. So, based on the spatial cluster model I mentioned before, we need a trajectory cluster model to handle trajectories as sequence of points.

There are many work about trajectory cluster models. Our dataset is about animal low-sampling-rate GPS points. [6] is an excellent work about a time-aware, density-based clustering of GPS trajectories. It applys and modifys DBSCAM method. Rather than spatial cluster, it deals with trajectories as sequence of points. The results will be stay regions. For example, the results are 3 sub-trajectories of the trajectories. For each sub-trajectory, we know the start point and end point. Applying on animal GPS dataset, we can consider the sub-trajectories as stay regions. The method fits our goal to find stay regions. Hence, it will be re-implemented on our dataset to extract stay regions.

2.3 The SeqScan Method

In SeqScan, there are mainly two restrictions: density of points in an area and its time duration. This section will give to details on how to extract stay regions.

2.3.1 Concept of SeqScan

P: the database of points.

D: distance threshold.

K: minimum number of points that a cluster contains.

d(): the distance function to calculate the distance between two points.

Close: if the distance between two points is within the distance threshold D (i.e. $d(p_i,p_j) <= D$), they are close to each other.

D-Neighborhood: the D-neighborhood of $p \in P$, denoted $N_D(p)$, is the subset of points that are close to p, i.e. $N_D(p) = \{p_i \in P, d(p,p_i) <= D\}$.

Core point: point p is a core point if its D-neighborhood contains at least K points, i.e. $|N_D(p)| >= K$.

Border point: a point that is not a core point but belongs to the D-neighborhood of a core point.

Directly density reachable: Point p is directly density reachable from q if q is a core point and $p \in N_D(q)$.

Density reachable: Two points p and q are density reachable if there is a chain of points $p_1, .., p_n$, $p_1 = p$, $p_n = q$ such that p_{i+1} is directly reachable from p_i.

Density connected: Points p and q are density connected if there exists a core point o such that both p and q are density reachable by o.

Trajectory: A trajectory T is a sequence of spatio-temporal points $T = [p_1, .., p_n]$ with $p_i = (l_i, t_i)$ where l_i, t_i is the sampled location in space and time respectively with $t_i < t_{i+1}$ and n the length of the trajectory. The trajectory has a begin point p_{start}, an end point

p_{end}, a temporal extent $[t_{start}, t_{end}]$, and a duration $|t_{start}-t_{end}|$. The duration is measured in second.

Sub-trajectory: A sub-trajectory $S = [p^1_i, .., p^m_j] \subseteq T$ of length m is a sequence of temporally ordered points of T with index 1, ..,m. A sub-trajectory may contain gaps. A gap is the open interval (t_i, t_j) signing a "hole" in the sequence, i.e. two points that are consecutive in S are not consecutive in T, i.e. $p^x_i, p^{x+1}_j \in S \rightarrow j \mathrel{!}= i+1$.

Dense region: dense region $S \subseteq T$ is a sub-trajectory $S = [q_1, .., q_m]$ such that the set of locations $[l_1, .., l_m]$ is a maximal density connected set with respect to D and K.

Exception: the points that do not belong to any dense region in T are qualified as exception.

Presence: Let $S = [q^1., .., q^m]$ be a dense region in $T = \{p_1, .., p_n\}$. Denote with S[i, i + 1] two consecutive points in S. We define:

- *The presence in* $S[i, i+1]$*:*

$$P(S[i,i+1],T) = \begin{cases} |t_h - t_{h+1}|, & if\ \exists h, q^i = p_h, q^{i+1} = p_{h+1} \\ 0, & otherwise \end{cases}$$

- *The presence in the dense region* S*:*

$$P(S,T) = \sum_{i \in [1,n-1]} P(S[i,i+1],T)$$

δ: presence threshold.

Persistent: presence in S is persistent, if it holds: $P(S,T) > \delta$.

Stay region: A stay region S in the trajectory T is a sub-trajectory of T such that: (i) S is a dense region w.r.t. D and K. (ii)The object's presence in S is persistent w.r.t δ.

Active cluster: during scanning the trajectory, the active cluster is the current stay region that is still growing. There is only one active cluster exists at one time.

2.3.2 Theory of SeqScan

The program scans the trajectory T sequentially. Initially, the program is trying to find a dense region over sub-sequences of incremental length, e.g. T[1, i], T[1, i+i].., until a dense region DR is possibly found at time t_c. If the presence in DR is persistent, the DR becomes an active cluster. The cluster has a start point and an end point. The time context of the cluster is initialized to $[t_1, t_c]$. There would be multiple dense regions at one time, but there only exists one active cluster. During the phase of cluster expansion, the algorithm continues scanning the trajectory until a new active cluster is found. The new active cluster has to start at the next point of the last point in current active cluster, i.e. if the time context of the current active cluster is $[t_i,t_j]$,the new active cluster has to be found in $[t_{j+1},t_c]$ (t_c is the current scanning point). At the time a new active cluster is found or the trajectory is terminated, the current active cluster is closed and becomes a stay region.

After scanning the whole trajectory, the program will generate stay region(s) and the time segment of each stay region. The time segment means how long the animal stays in this stay region.

2.3.3 Examples of SeqScan

Example 1

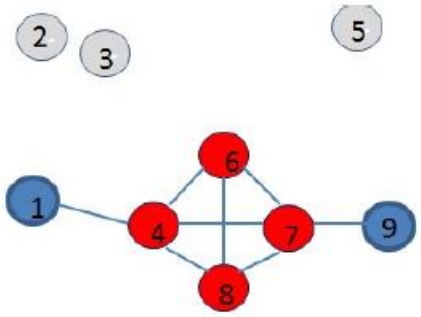


Figure 2-2 Example 1

The number 1-9 refer to the sequence of the trajectory.

Red points are core points.

Blue points are border points.

Gray points are exception points.

The line between two points means they are close.

Here, {1,4,6,7,8,9} is a stay region(k = 4). {2,3,5} are exceptions, not in stay region, during this period.

The time segment of this stay region is $|t_9-t_6|$. Because the object is absent in $|t_1-t_4|$ (t_2 and t_3 are out of the stay region) and $|t_4-t_6|$ (t_5 is out of the stay region)

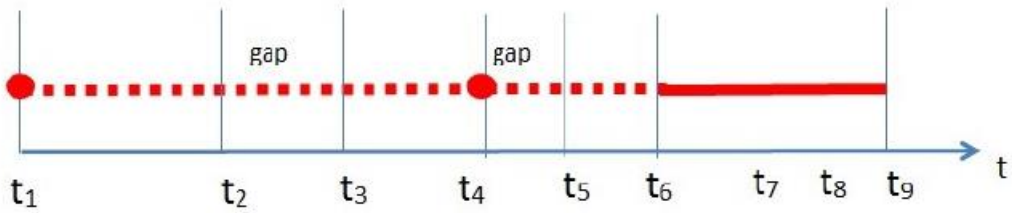


Figure 2-3 Presence

Example 2

Consider a trajectory of 10 points. Points are numbered from 1 to10 based on the time order. Given parameters D, δ = 3 and K = 4.

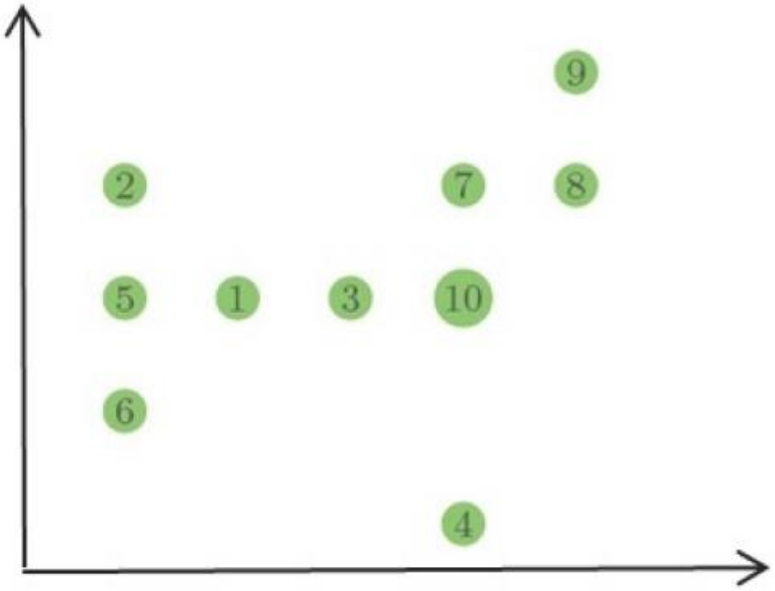


Figure 2-4 Example 2

The points are read in sequence from point 1:

Reading point 1: $N_D(1) = \{1\}$

Reading point 2: $N_D(2) = \{1, 2\}$, $N_D(1) = \{1, 2\}$

Reading point 3: $N_D(3) = \{1, 3\}$, $N_D(1) = \{2, 3, 1\}$

Reading point 4: $N_D(4) = \{4\}$

So far, none of them is a core point.

Reading point 5: $N_D(5) = \{1, 2, 5\}$,$N_D(1) = \{2, 3, 5, 1\}$,$N_D(2) = \{1, 5, 2\}$

Now point 1 becomes a core point because $|N_D(1)| >= K$. A new dense region is created named dr1. The corresponding Time Segment is set to: $t_{s1} = [1, 3] \cup [5, 5] = 3\text{-}1 = 2 < \delta$, no active cluster is created.

Reading point 6: $N_D(6) = \{1, 6, 5\}$,$N_D(1) = \{2, 3, 5, 1, 6\}$,$N_D(5) = \{1, 5, 2, 6\}$

Now point 5 becomes a core point. dr1 has expanded with point 6, and The time segment of dr1 is updated to: $ts1 = [1, 3] \cup [5, 6] = 3\text{-}1+6\text{-}5 = 3 >= \delta$. A new active cluster is created.

Reading point 7: $N_D(7) = \{3, 7\}$, $N_D(3) = \{1, 3, 7\}$

Reading point 8: $N_D(8) = \{7,8\}$, $N_D(7) = \{3, 7, 8\}$

Reading point 9: $N_D(9) = \{7,8,9\}$, $N_D(8) = \{7,8,9\}$, $N_D(7) = \{3, 7, 8, 9\}$

Now point 7 becomes a core point because $|N_D(7)| >= K$. A new dense region is created named dr2. The corresponding Time Segment is set to: $t_{s1} = [3, 3] \cup [7, 9] = 9\text{-}7 = 2 < \delta$, no active cluster is created.

Reading point 10: $N_D(10) = \{3,7,8,10\}$, $N_D(8) = \{7,8,9,10\}$, $N_D(7) = \{3, 7, 8, 9,10\}$, $N_D(3) = \{1, 3, 7,10\}$

Now point 3 results to be a shared core point between dr1 and dr2. The two regions are merged. Also points 8, 10 become a core point. The time segment of the result is set to: $ts1 \cup ts2 = [1, 3] \cup [5, 10] = 7$.

Reach the end of the trajectory. The active cluster becomes a stay region, and the time segment of this stay region is 7.

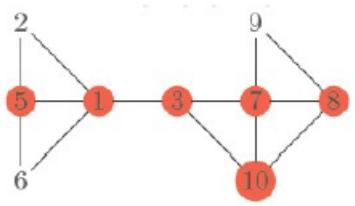


Figure 2-5 Result

2.4 Random Walk

It is needed to set the parameters in SeqScan reasonable. Random Walk is a good approach to find an appropriate parameter D for the method.

In a plane, consider a sum of N two-dimensional vectors with random orientations. Use phasor notation, and let the phase of each vector be random. Assume N unit steps are taken in an arbitrary direction (i.e., with the angle θ uniformly distributed in $[0, 2\pi)$ and not on a lattice), as illustrated above. The position z in the complex plane after N steps is then given by

$$z = \sum_{j=1}^{N} e^{i\theta_j}, \tag{1}$$

which has absolute square

$$\sum_{j=1}^{N} e^{i\theta_j} \sum_{k=1}^{N} e^{-i\theta_k} \tag{2}$$

$$\sum_{j=1}^{N}\sum_{k=1}^{N} e^{i(\theta_j - \theta_k)} \tag{3}$$

$$N + \sum_{\substack{j,k=1 \\ k \neq j}}^{N} e^{i(\theta_j - \theta_k)}. \qquad 4)$$

Therefore,

$$\left\langle |z|^2 \right\rangle = N + \left\langle \sum_{\substack{j,k=1 \\ k \neq j}}^{N} e^{i(\theta_j - \theta_k)} \right\rangle. \qquad 5)$$

Each unit step is equally likely to be in any direction (θ_j and θ_k). The displacements are random variables with identical means of zero, and their difference is also a random variable. Averaging over this distribution, which has equally likely positive and negative values yields an expectation value of 0, so

$$\left\langle |z|^2 \right\rangle = N. \qquad 6)$$

The root-mean-square distance after N unit steps is therefore

$$|z|_{\text{rms}} = \sqrt{N}, \qquad 7)$$

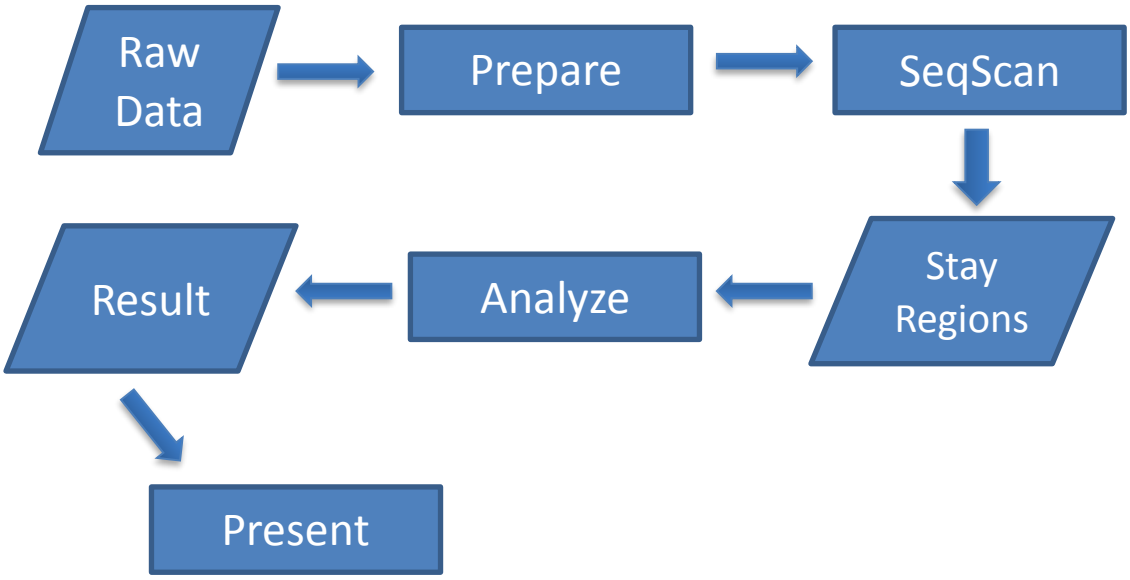
so with a step size of l, this becomes

$$d_{\text{rms}} = l\sqrt{N}.$$

2.5 Animal Trajectory Generator

We need to re-implement the SeqScan to generate stay regions. To test if the implement is good or not, Animal Trajectory Generator is a useful tool to examine it. It is a wildlife trajectory generator based on the features extracted from the real black bear GPS trajectory. The generator outputs trajectory points with a stop or move label as a ground truth for the experiment. Then, we use our program to run on the same trajectory to label each point again. Finally, we compare the result from our algorithm with the ground truth that we have to come up with the accuracy to evaluate the whole process.

3. METHODOLOGY



The flowchart shows the process of the tool. This paper will explain each step in details one by one.

3.1 Re-implement SeqScan

First of all, the program re-implements the SeqScan method. It take raw data as input. Based on different raw data, all inputs are transferred into a same format. After the program processes the data, it will generate stay regions of each individual as output.

3.1.1 Pseudo –code of implementation

```
For each animal {
        Main{
                In: T = [p1, .., pn], D, K, δ;
                Out: stayRegions{ points, time segment}
                c ← 1                                                    //Index
                start ← 1, end ← 0
                activeCluster ← Ø
                stayRegions ← {Ø}
                while c<=n do
                        timeContext ← [t_start, t_c]
                        if expand(activeCluster, timeContext, p_c) then
                        end ← c
                        else
                   nextCluster ← findCluster(T[t_end+1, t_c])
                   if nextCluster !=Ø then
                                stayRegions ← add(activeCluster)
                                noise ← add(T[t_start, t_end] \ activeCluster)
                                start ← end+1, end ← c
                                activeCluster ←nextCluster
                   end if
                        end if
                        c← c+1
                end while
         total_average_time += sum(stayRegions.time)/stayRegions.size
         }
         Result = total_average_time/bear.size
}

function EXPAND(activeCluster, timeContext, q)
        Global variables : DR, T
        c ← 1
        createPointDesc(q)
        for all p ∈ N_D(q) do                    // N_D(q) ∈ T [timeContext]
                linkCorePoint(p, q)
                linkNeighbors(p)
        end for
        return(q ∈ activeCluster)
end function

procedure linkCorePoint (p, q)
```

```
        if Desc(p).R !=null then            // p is a core point
                Desc(q).R ← Desc(p).R
                updateTimeSegment(Desc(p).R, {q})
        end if
end procedure

procedure LINKNEIGHBORS(q)
        if isCorePoint(q) & Desc(q).R = null then
                if ∄dr ∈ DR where q ∈ dr then
                        Desc(q).R = CreateNewRegionDesc()
                        updateTimeSegment(Desc(q).R, N_D(q))
                else
                        for all dr_1, dr_2 ∈ DR where q ∈ dr_1, dr_2 do
                        merge(dr_1, dr_2)
                        end for
                updateTimeSegment(Desc(q).R, N_D (q) \ q)
                end if
        end if
end procedure
```

3.1.2 Program Process

Two simple data structures are used called Point Descriptor and Dense Region Descriptor, respectively.

- Point Descriptor. When the point pi = (x_i, y_i, t_i) 2 T is read, the point is assigned index i, an identifier and a descriptor. The identifier is a pointer to the actual coordinates. The descriptor is the pair: Desc(p_i) = (Neighbors,R) where Neighbors is the set of points (identifiers) in the neighborhood of pi and R the possibly empty pointer to the descriptor of the dense region the points belongs to.

- Dense Region Descriptor. Every time a dense region is created it is assigned the descriptor (Id, Ts) where Id is the dense region identifier (e.g.

progressive number) and Ts the Time Segment, defined in the next. The points belonging to the dense region j are thus the set: $\{p_i | Desc(p_i).R.Id = j\}$.

The program scans the trajectory sequentially. For each point, the program creates a point descriptor for it. The program calculates the distance between the point and other points respectively, add all its neighbors to its point descriptor. During adding neighbors, the program checks the dense region descriptor which contains the added points. If a exist dense region descriptor can be expended, the program will update the dense region descriptor until find a new active cluster. After a new active cluster is found, the current cluster will be saved as stay region. If a new dense region descriptor can be created, the program will generate a new region descriptor. After scanning the whole trajectory, the program will generate all stay regions of this trajectory.

3.1.3 Correctness of the Program

As the processes explained in this chapter, there are two parts in the process of extracting stay regions from trajectories, density of points in an area and its time duration. It is obvious to get time duration of a dense region. To prove the correctness of the process of generating dense region, we use the tool in [7]. We use the tool to generate 150 test cases of animal trajectories to be the ground truth. Then run our program to generate dense regions from the 150 test cases of animal trajectories to check the correctness compare with the ground truth. The result is that the average

accuracy is around 90% on the 150 test cases, which means the results generated by our program are very accurate.

Figure 3.5 and Figure 3.6 are one test case out of the 150 test cases. In the figures, the red dots belong to a dense region, the blue dots are exceptions. There are only a few mismarked points at the beginning or the end or a dense region. The result is very accurate.

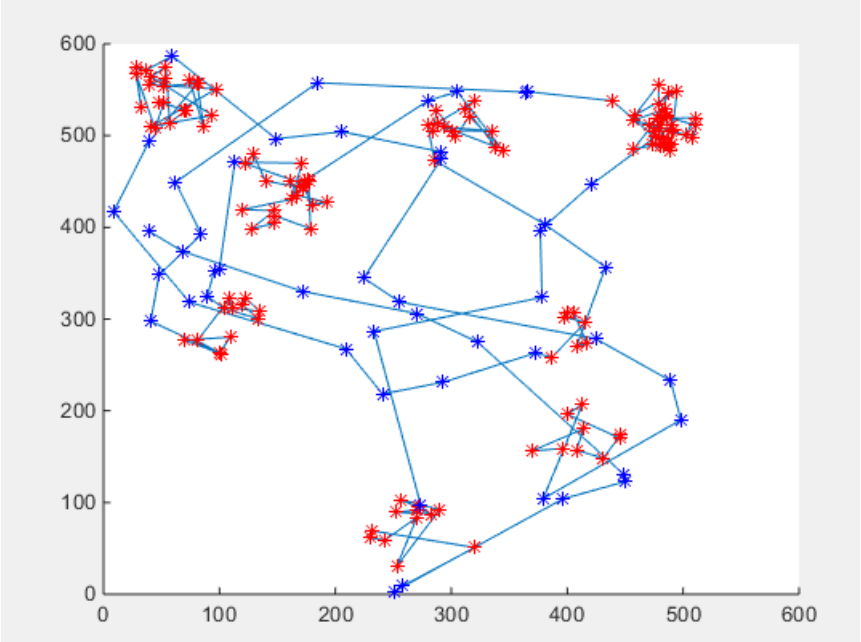


Figure 3-1 Trajectory generated by the simulator

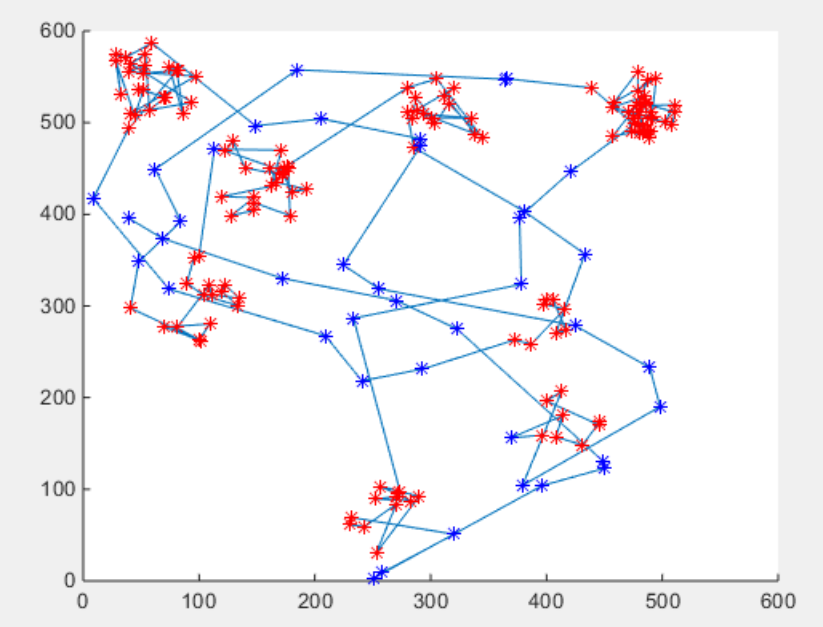


Figure 3-2 Result generated by my program

3.2 Analyze Stay Regions

After the stay regions are generated, the program is able analyze the animals based on the stay regions.

3.2.1 Statistics information of Stay Regions

3.2.2 Overlaps study on Stay Regions

The more interesting thing is working on overlaps between stay regions. For example, one of the datasets is bears from Missouri, which appear in the south of Missouri. There is a hypothesis that some of them may know each other, they even have relationship. Hence, it is very interesting to learn the overlaps among them. Here the program uses different ways to study overlaps based on their stay regions.

Definition:

- Centroid of a cluster: average coordinate of all points in a cluster
- Radius of a cluster: the maximum distance of a point in a cluster from the centroid.

3.2.2.1 Spatial overlap

We define two clusters overlap if the distance between two centroids is less than the sum of the two radiuses of the two clusters.

3.2.2.2 Interaction overlap

Based on the spatial overlaps, add time element is added to check the interaction overlap, which means they stay in the same area during a same time.

Compare each pair of overlap stay regions and their start time/end time. If their time are overlap,

(start1<start2 && end1>start2) || (start1>start2 && start1<end2)

It means they meet in that area at that time.

3.2.2.3 Multidimensional scaling (MDS)

Now we can study on something even more interesting. We already know some bears have met each other. Now we want to learn how close they are, which we can infer by how long they stay together.

Between bears, if two bears have overlapped stay regions (meaning they stayed in the same area at the same time), we can find the time duration for each overlapped case. Sum all the overlapped time and use it to represent the closeness of the two

animals. Then, we can construct a pairwise distance matrix of size n by n, for n animals. Each entry d(x, y) is the fraction of time two animals (x and y) staying together, i.e., T_overlap / min (T_x, T_y), where T_x is the total trajectory time of animal x and T_y is the total trajectory time of animal y.

Then, we apply Multidimensional scaling algorithm on the matrix.

Multidimensional scaling (MDS) is a means of visualizing the level of similarity of individual cases of a dataset. It refers to a set of related ordination techniques used in information visualization, in particular to display the information contained in a distance matrix. An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible. Each object is then assigned coordinates in each of the N dimensions. The number of dimensions of an MDS plot N can exceed 2 and is specified a priori. Choosing N=2 optimizes the object locations for a two-dimensional scatterplot. There are many papers introduce Multidimensional scaling algorithm, like, [8] and [9].

4. EXPERIMENT

4.1 Experiment 1: Missouri black bear dataset

Parameters:

K = 10, minimum neighbors needed to become a core point

D = 500, maximum distance between neighbors in meter

δ = 7, minimum time needed to become a stay region in days

Filter:

Because the minimum time needed to become a stay region is 7 days, some bears with not enough points will be filtered.

Some bear has a big time gap in record. For example, a bear is tracked during the year 2012; stop tracked during the year 2013, and 2014; started tracked in the year 2015 again. I also ignore the points if the time between two adjust points in trajectory is greater than 7 days.

Average method:

Calculate the average time of each bear stays in his/her stay regions first, and then average every bear to get the average time spent in one stay region.

Result:

For male:

41 input bears generate data of 28 bears with at least one stay region.

The average amount of stay regions for one bear of the 28 bears is 1.35.

The average time spent in one stay regions of the 28 bears is 28.93 days.

For female:

37 input bears generate data of 34 bears with at least one stay region.

The average amount of stay regions for one bear of the 34 bears is 3.06.

The average time spent in one stay regions of the 34 bears is 38.77 days.

4.1.1 Statistics Information

After we get the results of stay regions, we want to study some statistic information from them. We separate male bears and female bears, trying to learn some difference between them. The figures in this section are the results.

Figure 4.1 and Figure 4.2 show that what is the size the stay regions of bears.

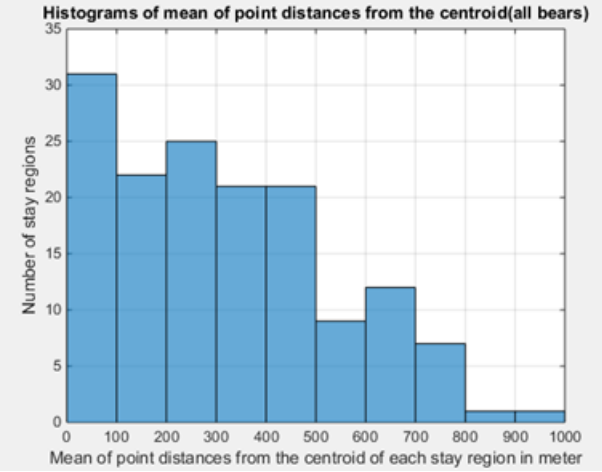


Figure 4-1 Histograms of mean of point distances from the centrolid(all bears)

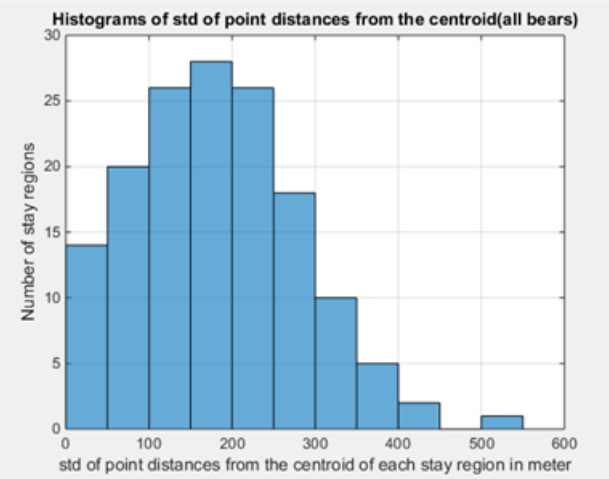


Figure 4-2 Histograms of std of point distances from the centrolid(all bears)

Figure 4.3 and Figure 4.4 show the size difference between male bears and female bears. We can see from the data that female has larger stay region size than male.

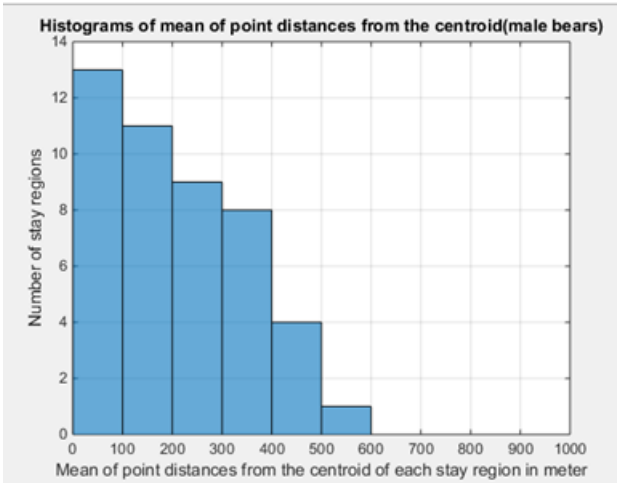


Figure 4-3 Histograms of mean of point distances from the centrolid(male bears)

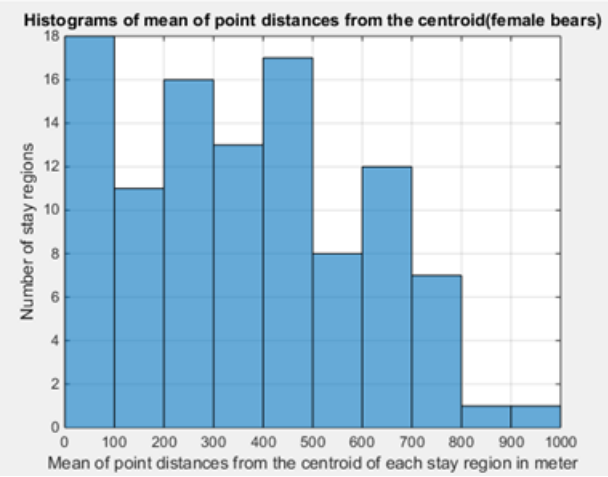


Figure 4-4 Histograms of mean of point distances from the centrolid(female bears)

Figure 4.5 and Figure 4.6 show that the size of male stay regions are stable than the size of female stay regions.

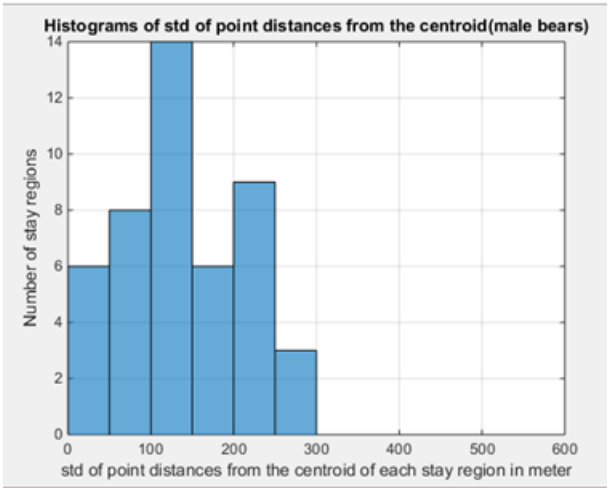


Figure 4-5 Histograms of std of point distances from the centrolid(male bears)

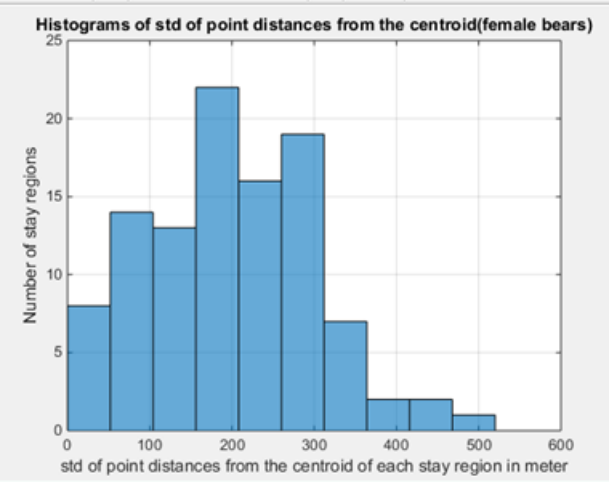


Figure 4-6 Histograms of std of point distances from the centrolid(female bears)

Time segment = end time of the stay region – start time of the stay region. (In days)

Figure 4.7 and figure 4.8 show that for each stay region, male stays longer than female.

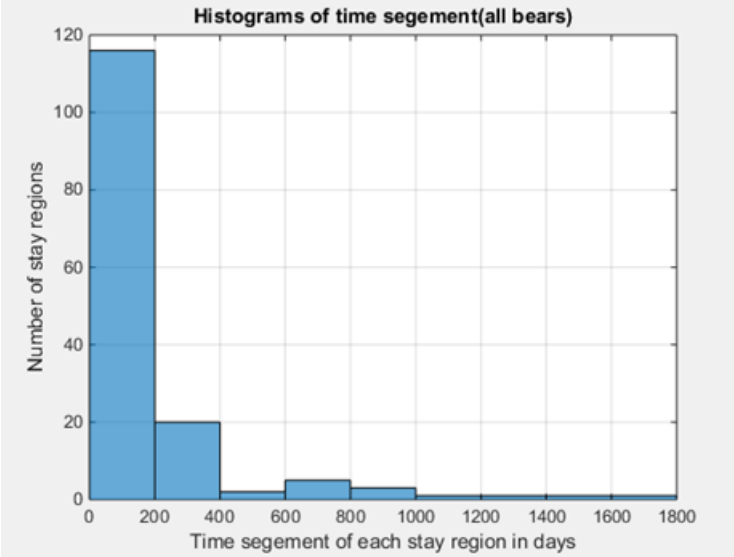


Figure 4-7 Histograms of time segement(all bears)

All bears: mean=180.881, std =272.316

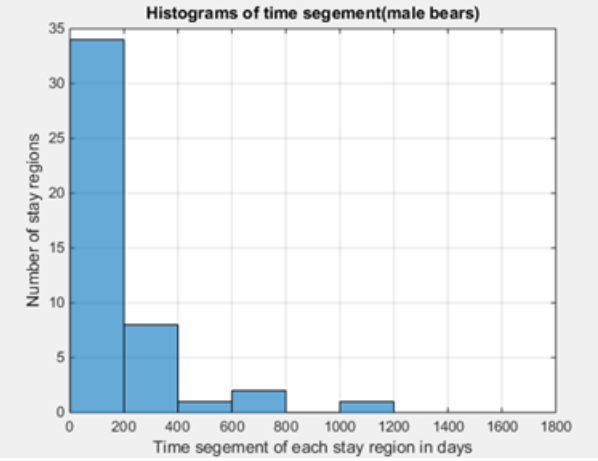


Figure 4-8 Histograms of time segement(male bears)

Male: mean =170.8616, std =213.1747

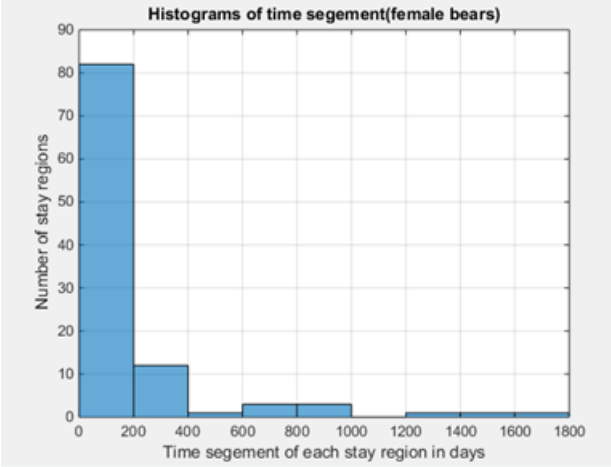


Figure 4-9 Histograms of time segement(female bears)

Female: mean = 185.3126, std =295.5593

When we focus on 0-400 days:

Figure 4.10 and Figure 4.11 show that for male, there are a few stay regions which bear stay much longer than other. However, for female, the time spent in each stay region is stable.

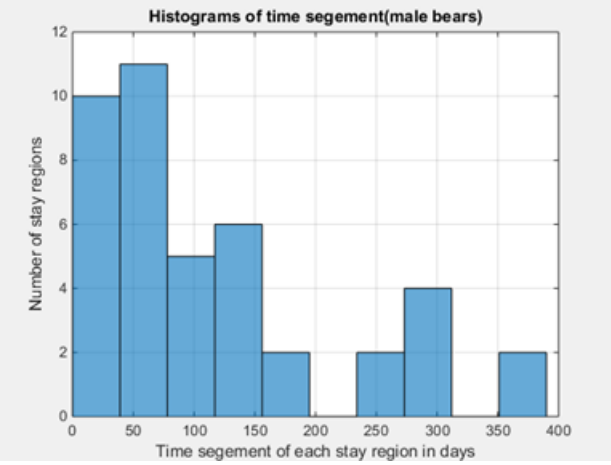


Figure 4-10 Histograms of time segement(male bears)

Male: mean =117.9287, std =103.7947

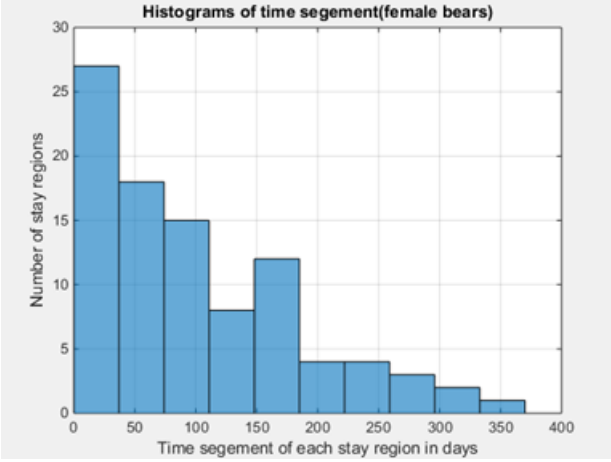


Figure 4-11 Histograms of time segement(female bears)

Female: mean =100.7318, std = 82.3770

Because some bears have time gap on data. In trajectory sequence, points are in the same stay region, but actually, there may be a big time gap, like two years. It causes the time segment of the stay region very big. Also, some bear are tracked for a short time, which causes the time segment very short.

We can learn some interesting facts from the result. Here I give two examples.

1. Distribution of movements.

From this set of parameters, we detect only a few stay regions per bear, especially for male, only one or two stay regions are found per bear. However, we can find some movement patterns from the bears with multiple stay regions. They stay in area A, move to area B staying for a while, and then move back to area A. Bear 1016 is one example of this pattern. There does exist a few bears follow this pattern.

2. Patch residence time. How does it vary between sexes?

We can see from the result that male has fewer stay regions than female. Male stays in stay regions less time than female and has much more exception points than female, which means they usually go out of the stay regions and come back within 7 days. Even though male has less time staying in stay region, but male has longer time staying around one stay region. Hence, I think this result can prove male has fewer stay regions, but travel around stay regions more often and farther than female.

4.1.2 Overlaps Information

4.1.2.1 Spatial overlap

Figure 4.12 is the statistic of radius of all bears. And we get the mean is 919.172 meters, std is 502.555 meters.

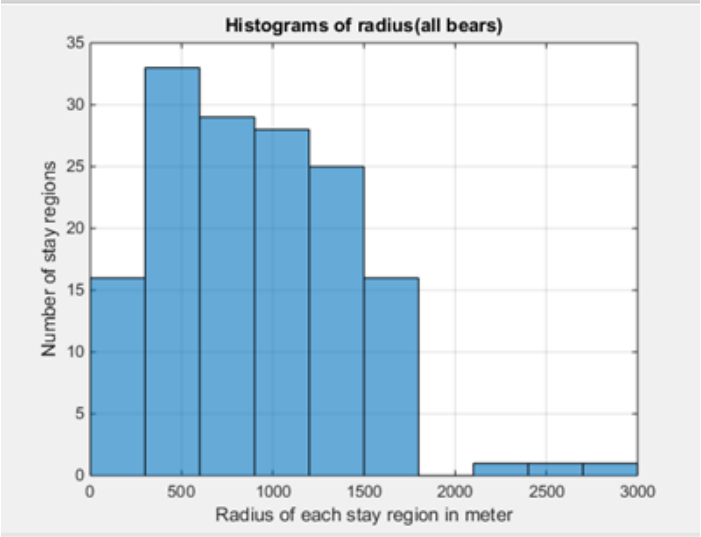


Figure 4-12 Histograms of radius(all bears)

We get the result of overlaps as Table 0-1 in appendix. Take the first row as example, it means Bear 1001(Bear ID)'s second (number of stay region of the bear) stay region is overlap with bear 1011's first stay region.

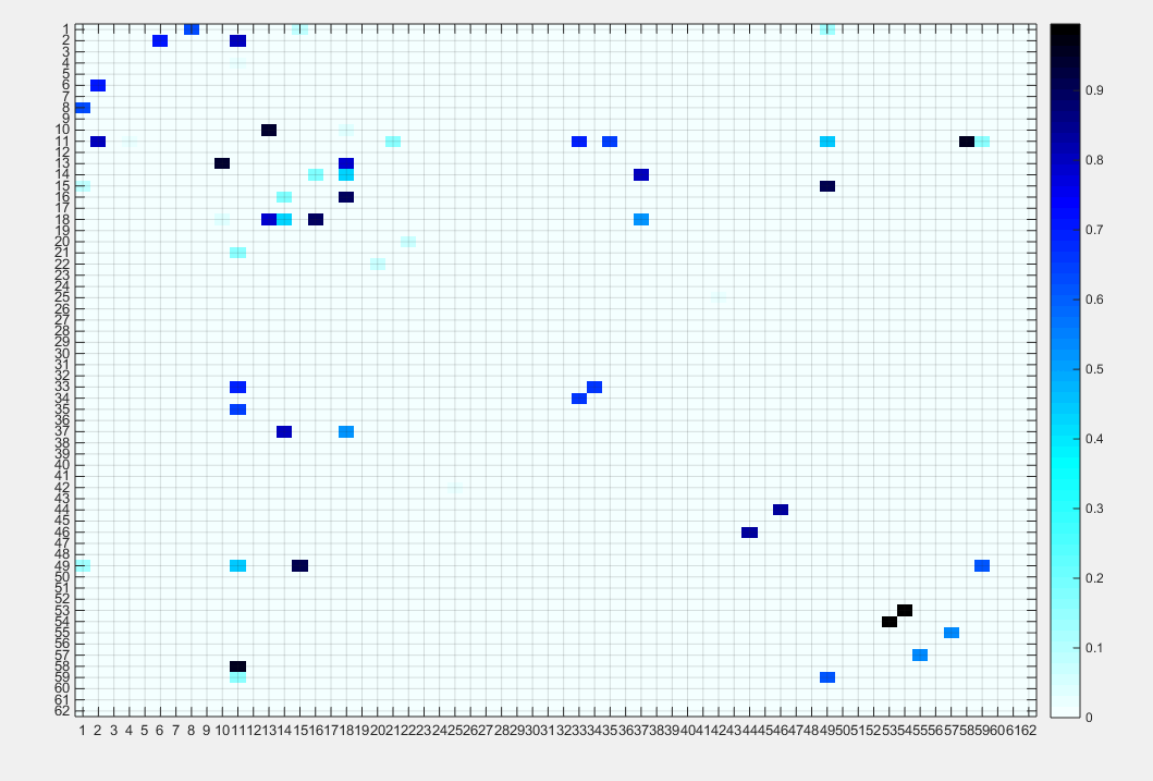
From Table 0-1 in appendix we can see that some overlaps appear in a same area, which we can infer that place have some resource that attracts bears to come, like food or water.

4.1.2.2 Interaction overlap

We get the result as Table 0-2 in appendix.

From Table 0-2 in appendix we can see that some bears meet one time, some meet more than one time, and some even meet with many bears. From this we can infer that those pairs of bears may know each other.

4.1.2.3 Multi-dimensional scaling



Next, we run shortest algorithm on matrix A and get the result as Figure 4.13.

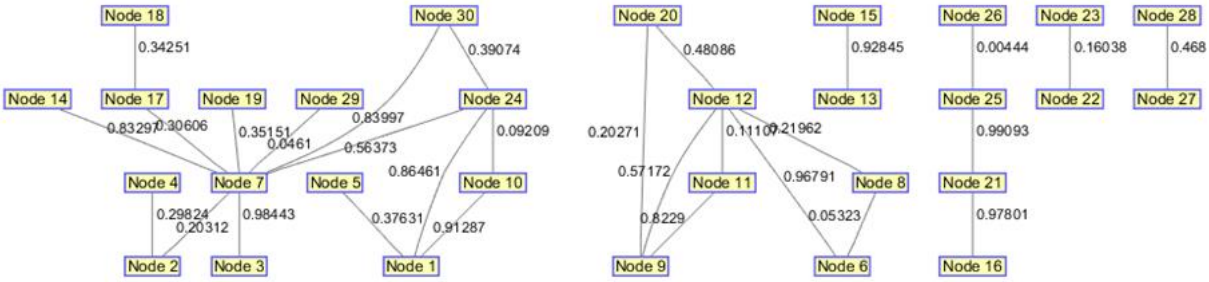
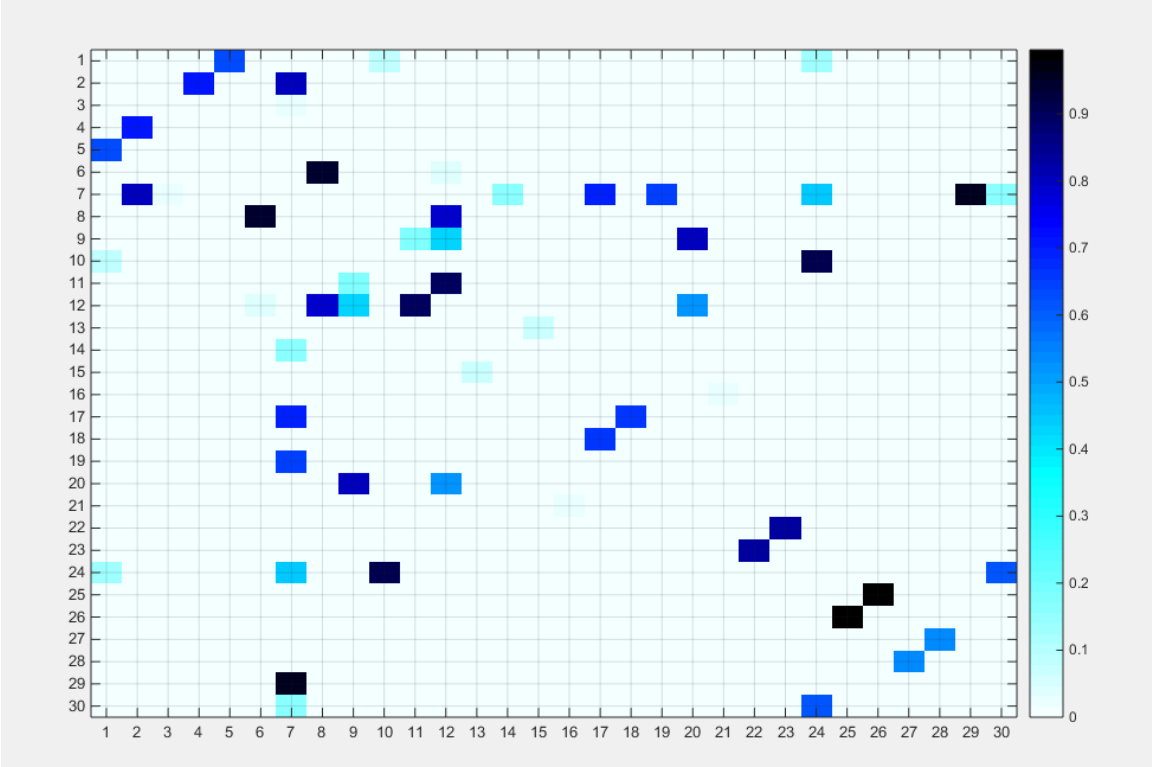


Figure 4-13 Shortest path

In Figure 4.13, the nodes connected by a line means they have overlap. The smaller the number is, the closer they are, meaning they have more time staying together. If there is no line between nodes, it means there is no overlap between them.

We can see in Figure 4.13, there are two larger connected graphs that we are more interested in. So, we make two new matrix M and N based on the two larger connected graphs respectively.

Then, we apply Multidimensional scaling algorithm on matrix M and N. And we get the result as Figure 4.14 and 4.15 for matrix M and N.



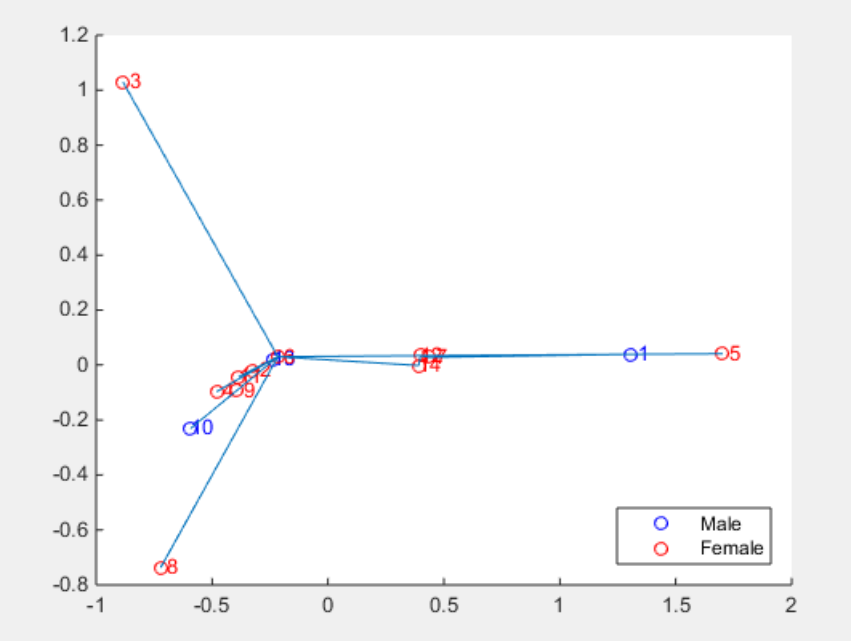


Figure 4-14 MDS result 1

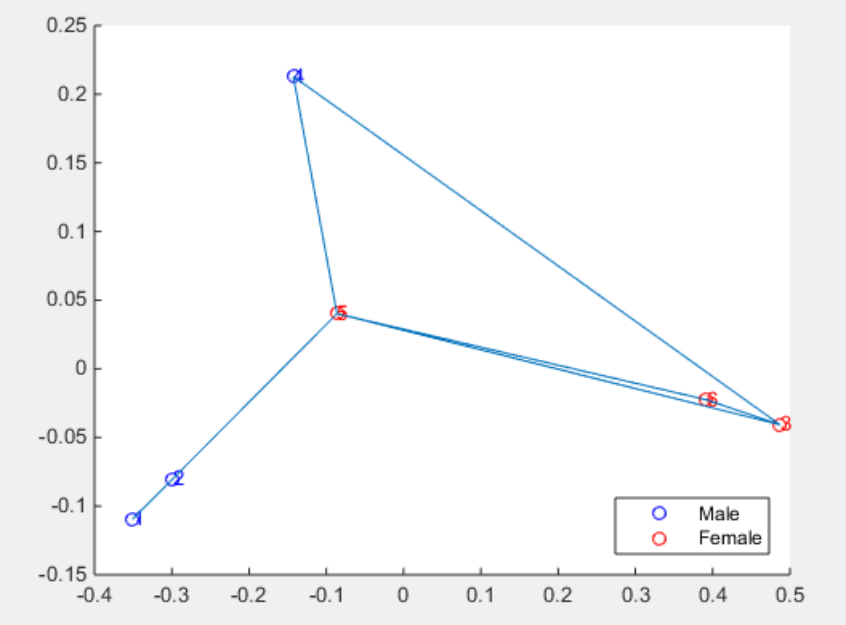


Figure 4-15 MDS result 2

In Figure 4.14 and Figure 4.15, blue circle represents male, red circle represents female. The number beside circle is the bear ID. Now we can observe the relationship between bears in the 2-D figures. The more they are close in the figure, the more they are close.

4.1.3 Website application to present results

In order to let people get access to the study easier, I make a website application to show the study results. There are several parts of the website application.

4.1.3.1 Data of the Website Application

The data to be shown is generated by the program in the previous chapters. The data is saved in files with CSV format. There is one folder for each bear, and there is one file for each stay region. The name of the file is the number of stay region of this bear with the time segment of this stay region. In each file, there are all the points belong to this stay region. So, one bear has one folder and possibly more than one files in the folder. For example, bear 1001 has a folder 1001, and there are 3 files in the folder named “1_14.txt”, “2_16.txt” and “3_10.txt”. They mean bear 1001 stays in number 1 stay region for 14 days, in number 2 stay region for 16 days, and in number 3 stay region for 10 days. In “1_14.txt”, there are 442 rows of data, each row has a coordinate of a points. So there are 442 points in stay region 1.

4.1.3.2 Mid-layer of the website

Here mid-layer is used to exchange data from the file to the website. PHP is used as an API to do the mid-layer job. There are two PHP files. One is used to be called to return the folder of this animal with calling its ID. The other is used to be called to return a stay region file with calling the number of its stay region.

4.1.3.3 Front end of the website

The website two parts, two select lists and a Google map. User select an animal ID and push GO button, the map will show all stay regions of this animal. Every stay region has unique color and its stay region ID so that people can distinguish from each other.

Here Google map is used to implement map function and use Ajax to call PHP and get data so that every time the new data comes the page will not be refreshed.

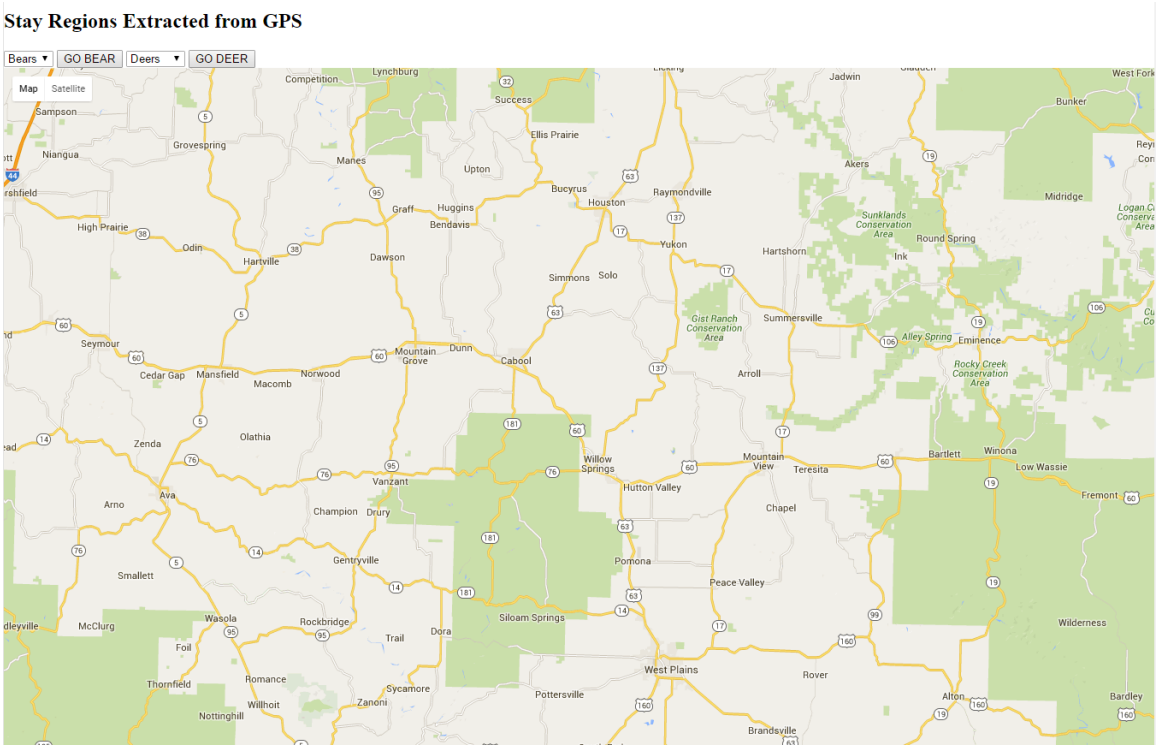


Figure 4-16 Initial page

Figure 5.1 is the initial page the website. After user selects a bear or a deer in the selection box, the result will be show as Figure 5.2 and 5.3 show. By checking the map, we can know from the map that how the animal travels between stay regions. Also, it is visualized that how big the stay regions are, what the shape of stay region is, and where animals appear.

Stay Regions Extracted from GPS

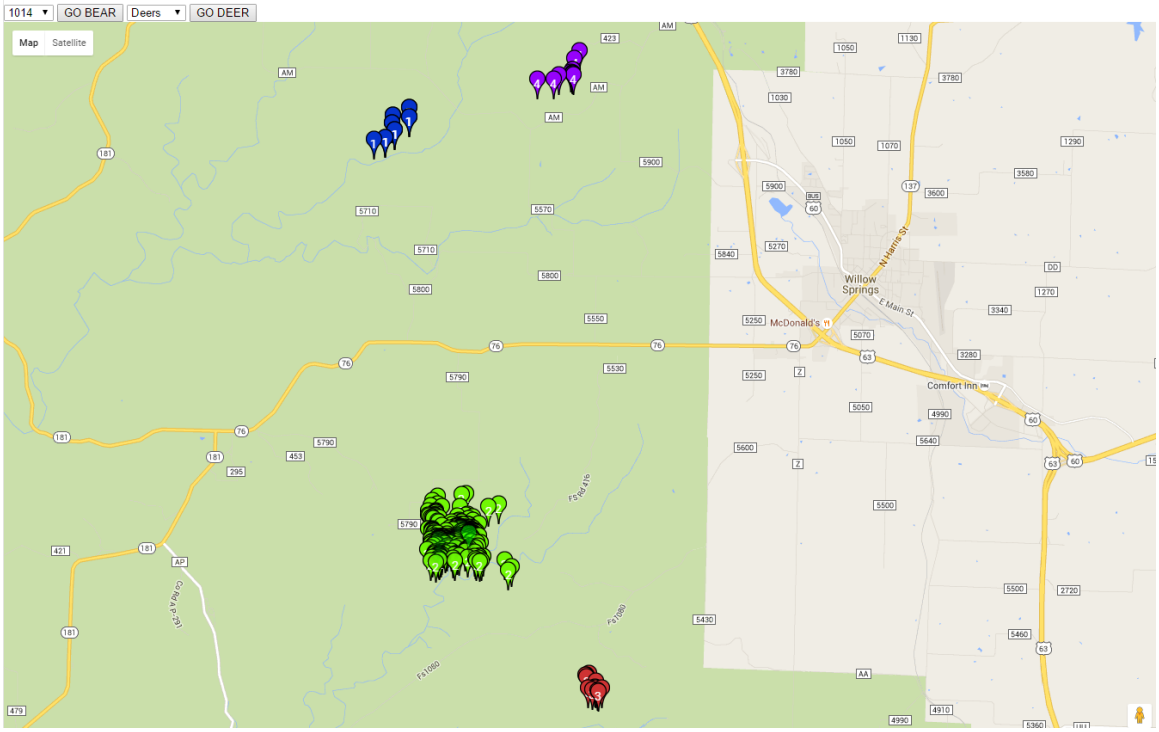


Figure 4-17 Bear 1014

4.2 Experiment 2: Missouri deer dataset

Dataset: 114 deer with about 100,000 time-stamped GPS data.

Output example on website application:

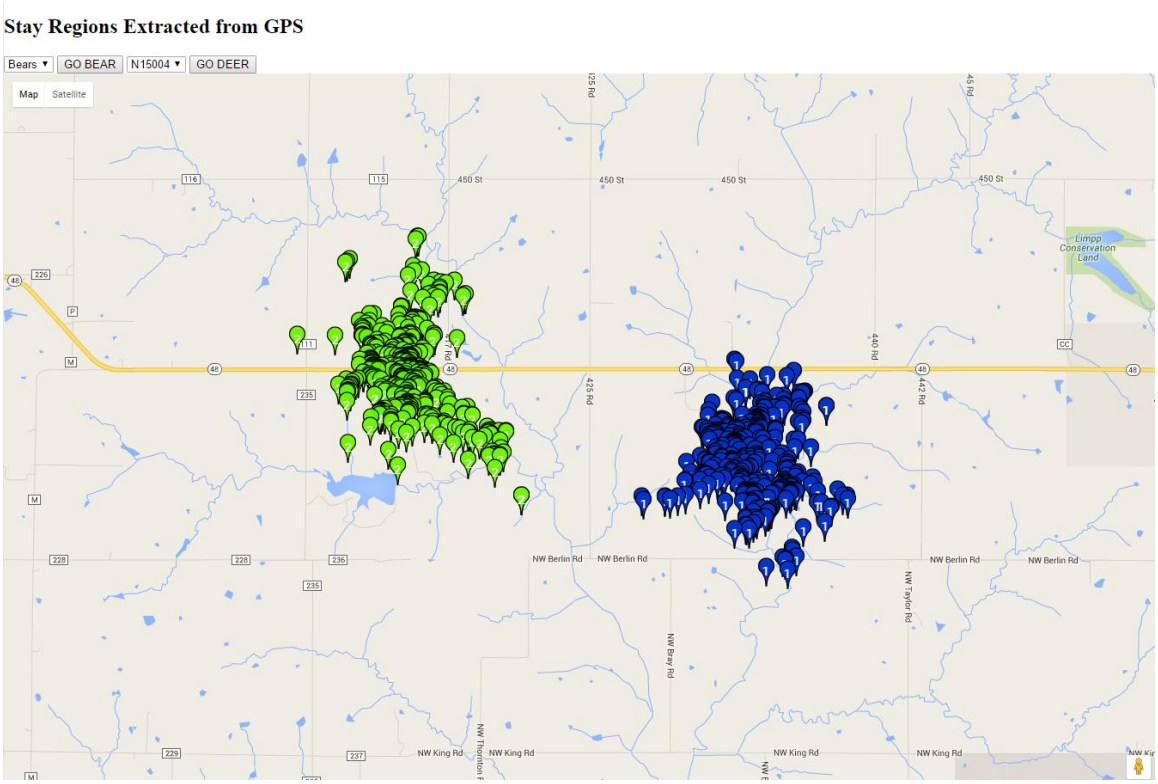


Figure 4-18 Deer N15004

4.3Experiment 3: Carnivore dataset

Dataset: 10 bear with about 100,000 timestamped GPS data.

Output example on website application:

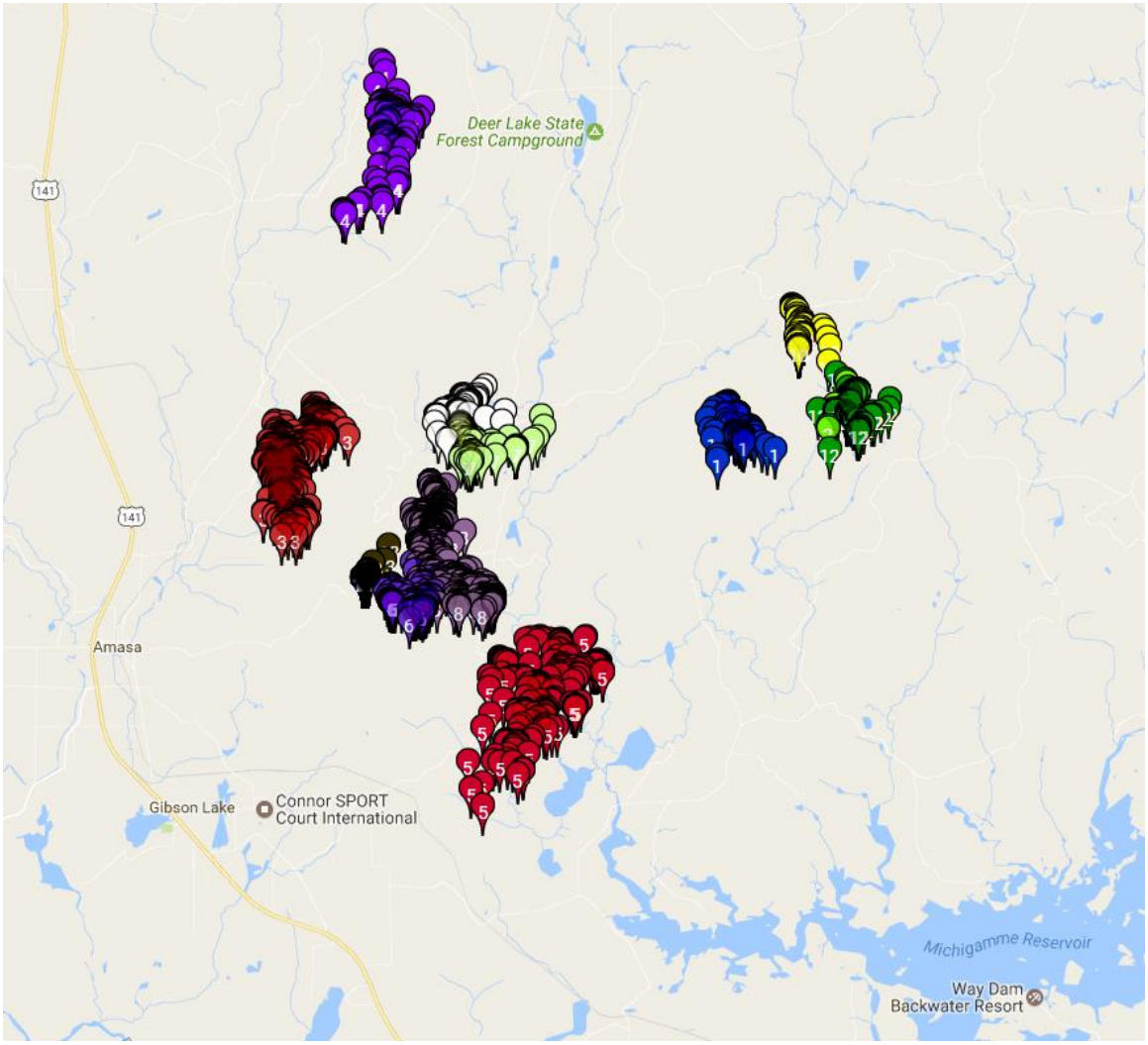


Figure 4-19 Bear 104

5. SUMMARY

This paper shows the processes of studying and analyzing on animal trajectories. What we have is animal trajectories data. We use a novel algorithm, grounded on the formal framework of DBSCAN, to extract stay regions of animal trajectories. We explain how do we do it, and we show test cases to prove the program is very accurate, 90% accuracy on 150 test cases.

Based on the stay regions, we learn some statics information of the animals and the difference between different sexes. Also, I show the process that I study the overlap information of the animals, trying to indicate some information. To be specific, we learn the spatial overlap to indicate the places where may have resource attracting bears, we learn meet overlap to indicate which bears may know each other, and we use multidimensional scaling to learn the closeness between bears.

Last but not least, I make a website to show all the stay regions of the animals, so that people can check what the movement pattern of each animal is and visualize all the data.

APPENDIX

Bear ID	number of stay region of the bear	Bear ID	number of stay region of the bear
1001	2	1011	1
1001	2	1103	1
1001	2	1410	1
1002	1	1002	2
1002	1	1002	3
1002	1	1009	1
1002	1	1201	1
1002	2	1002	3
1002	2	1002	4
1002	2	1009	1
1002	2	1201	1
1002	2	1203	1
1002	3	1009	1
1002	3	1016	2
1002	3	1016	4
1002	4	1009	1
1002	4	1201	1
1002	4	1203	1
1007	1	1007	3
1007	1	1016	1
1007	1	1103	2
1007	1	1103	4
1007	1	1410	1
1007	1	1427	1
1007	3	1103	1
1007	3	1103	2
1007	3	1103	4
1007	3	1117	2
1007	3	1410	1
1009	1	1201	1
1011	1	1103	1
1014	2	1101	1
1014	2	1102	4
1014	2	1109	1
1014	2	1109	4
1014	2	1109	5
1014	2	1207	1
1014	3	1101	1
1014	3	1109	3
1014	3	1207	1
1016	1	1016	2
1016	1	1201	1
1016	1	1203	1

1016	1	1410	1
1016	1	1427	1
1016	1	1427	2
1016	2	1016	3
1016	2	1016	4
1016	2	1016	5
1016	2	1114	1
1016	2	1201	1
1016	2	1204	1
1016	2	1204	2
1016	2	1410	1
1016	2	1424	1
1016	2	1427	1
1016	2	1427	2
1016	3	1016	4
1016	3	1016	5
1016	3	1204	1
1016	3	1424	1
1016	4	1016	5
1016	4	1114	1
1016	4	1204	1
1016	4	1424	1
1016	5	1204	1
1016	5	1424	1
1101	1	1109	1
1101	1	1109	2
1101	1	1109	4
1101	1	1207	1
1102	1	1102	2
1102	1	1102	4
1102	1	1105	1
1102	1	1109	1
1102	1	1415	1
1102	2	1102	4
1102	2	1105	1
1102	2	1109	1
1102	2	1415	1
1102	3	1109	1
1102	3	1207	1
1102	3	1415	1
1102	4	1105	1
1102	4	1109	1
1102	4	1207	1
1102	4	1415	1
1103	1	1103	2
1103	1	1103	3
1103	1	1103	4
1103	1	1103	5
1103	1	1117	2
1103	1	1410	1
1103	2	1103	4
1103	2	1117	2
1103	2	1410	1

1103	3	1103	4
1103	3	1103	5
1103	3	1117	2
1103	3	1410	1
1103	4	1103	5
1103	4	1117	2
1103	4	1410	1
1103	5	1117	2
1103	5	1410	1
1105	1	1109	1
1105	1	1415	1
1109	1	1109	2
1109	1	1109	4
1109	1	1109	5
1109	1	1207	1
1109	1	1415	1
1109	2	1109	4
1109	4	1109	5
1112	1	1112	3
1112	1	1112	4
1112	1	1115	2
1112	2	1114	2
1112	3	1112	4
1112	5	1115	6
1114	1	1424	1
1114	3	1115	3
1115	1	1115	3
1115	4	1115	5
1115	4	1115	6
1115	6	1116	2
1115	6	1116	3
1117	2	1410	1
1118	1	1118	3
1118	1	1304	6
1118	3	1304	1
1118	3	1304	6
1119	1	1422	1
1122	1	1411	3
1122	2	1411	3
1126	1	1216	2
1201	1	1203	1
1201	1	1427	1
1203	1	1427	1
1204	1	1424	1
1207	1	1415	1
1216	1	1216	2
1304	2	1304	4
1304	2	1417	1
1304	2	1418	1
1304	2	1418	2
1304	3	1304	6
1304	4	1304	7
1304	4	1417	1

1304	4	1418	1
1304	5	1417	1
1304	5	1418	2
1404	1	1407	1
1407	2	1407	3
1408	1	1415	1
1410	1	1427	1
1410	1	1427	2
1411	1	1411	3
1417	1	1418	1
1417	1	1418	2
1419	1	1422	1
1420	1	1420	2
1420	1	1420	3
1420	2	1420	3
1427	1	1427	2

Table 0-1

Bear ID	number of stay region of the bear	Bear ID	number of stay region of the bear
1001	2	1011	1
1001	2	1103	1
1001	2	1410	1
1002	1	1009	1
1002	2	1009	1
1002	3	1009	1
1002	3	1016	2
1002	3	1016	4
1007	1	1016	1
1014	2	1101	1
1014	2	1109	1
1016	2	1201	1
1016	2	1204	1
1016	2	1204	2
1016	2	1410	1
1016	2	1424	1
1016	2	1427	1
1016	4	1114	1
1101	1	1109	1
1102	1	1105	1
1102	1	1109	1
1102	2	1109	1
1102	3	1109	1
1102	3	1207	1
1102	4	1109	1
1102	4	1207	1
1103	1	1410	1
1103	2	1410	1

1103	3	1410	1
1105	1	1109	1
1109	1	1207	1
1112	1	1115	2
1112	5	1115	6
1118	3	1304	6
1201	1	1203	1
1304	4	1417	1
1404	1	1407	1
1410	1	1427	1
1410	1	1427	2
1417	1	1418	1
1417	1	1418	2
1419	1	1422	1

Table 0-2

REFERENCES

1. Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M., eds. "A density-based algorithm for discovering clusters in large spatial databases with noise." Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231. ISBN 1-57735-004-9.CiteSeerX: 10.1.1.121.9220.

2. Derya Birant, Alp Kut. "ST-DBSCAN: An algorithm for clustering spatial–temporal data." 2006.

3. B. Borah. "An improved sampling-based DBSCAN for large spatial databases" Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on, pp.92-96, 2004.

4. Domenica Arlia, Massimo Coppola. "Experiments in Parallel Clustering with DBSCAN" 7th International Euro-Par Conference Manchester, UK, August 28–31, 2001 Proceedings, pp.326-331, 2001.

5. Density-based spatial clustering of applications with noise (DBSCAN) https://en.wikipedia.org/wiki/DBSCAN#Algorithm

6. Naria Luisa, Hamza Issa, Francesca Cagnacci. "Extracting stay regions with uncertain boundaries from GPS trajectories: a case study in animal ecology" SIGSPATIAL '14 Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp 253-262, 2014.

7. Simiao Sun. “ACTIVITY IDENTIFICATION FROM ANIMAL GPS TRACKS WITH SPATIAL TEMPORAL CLUSTERING METHOD DDB-SMOT”. University of Missouri, thesis papar.

8. Forrest W. Young. “MULTIDIMENSIONAL SCALING”, University of North Carolina.

9. Andreas BUJA , Deborah F. SWAYNE , Michael L. LITTMAN , Nathaniel DEAN , Heike HOFMANN , Lisha CHEN. “Data Visualization with Multidimensional Scaling”, Yale University, 2007.

10. McCrea, W. H. and Whipple, F. J. W. "Random Paths in Two and Three Dimensions." Proc. Roy. Soc. Edinburgh 60, 281-298, 1940.

11. Weisstein, Eric W. "Random Walk--2-Dimensional." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/RandomWalk2-Dimensional.html