Anomaly Detection and Modeling of Trajectories

Junier B. Oliva

CMU-CS-12-133

August 2012

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Kathleen M. Carley, Chair
Jeff Schneider

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

This work was supported in part by a Gates Millennium Fellowship, ONR - N00014-06-1-0104, and ONR N000140811186.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Gates Millennium Scholars Program, or the Office of Naval Research.

Keywords: Trajectories, Machine Learning, Anomaly Detection, Density Estimation, Graphical Models

Para mis padres

Abstract

The recent boom in the availability and use of geolocation technologies has created a great need to understand datasets of trajectories. Moreover, trajectory data is collected in a wide range of different domains including: meteorology, zoology, and business. However, trajectories have several intrinsic attributes that make them difficult to analyze. First, their time-series nature makes applying traditional techniques challenging. Secondly, most datasets contain trajectories of many points, making for a high-dimensional modeling problem. Lastly, there are several competing notions of similarity/difference in trajectories. In order to deal with these challenges, this thesis proposes several methods using statistics and machine learning (ML) that provide a deep understanding of trajectory datasets. In particular, this thesis brings forth methods to perform anomaly detection, density estimation, and spatial graphical models.

In general, an anomaly is an instance that is abnormal or unlikely based on the rest of the dataset. This thesis develops a technique for detecting anomalous trajectories in a dataset in an unsupervised fashion using support vector machines (SVMs) and various spatial representations of trajectories. This thesis will also focus on techniques for density estimation, that is providing a likelihood for each trajectory in a dataset. In order to effectively perform density estimation on trajectories, a combination of a Markovian assumption on the independence of the next position of a trajectory given its previous positions and kernel density estimation (KDE) is explored. Lastly, this thesis explores spatial graphical models. Undirected graphical models detail the conditional independence structure of a set of random variables. Given sparsity assumptions, this concept is used to build graphical models for indicator variables that have spatial locations associated with them, indicating if an agent has come near the corresponding location.

In order to effectively test the methods developed, experiments were ran using the following two real world datasets: one dataset consists of AIS-tracked shipping vessels in the English Channel; the other dataset contains every Atlantic Ocean tropical storm and hurricane track from 1949 to 2011. Overall, the methods presented were found empirically to provide a rich analysis of trajectory datasets.

Acknowledgments

I would like to thank my advisor, Dr. Kathleen Carley, for giving me the opportunity to get a great start in to the wonderfully rewarding yet ever so challenging world of researching. Thank you for all the guidance and useful suggestions throughout my masters program.

I must thank the Gates Millennium Scholars Program for allowing me, and many other minority students, to pursue a higher education. Your support throughout my academic career has been incredible and very appreciated.

Also, I need to thank Carnegie Mellon University for being a stupendous academic home from the start of my undergraduate studies to today. The walls are bursting with innovation and hard work at this institution. The incredible tutelage of all the amazing faculty have unquestionably shaped me and have made me a smarter, better individual.

My family, friends, and my girlfriend, Gabriela, also deserve a big thank you. Thank you all for putting up with my shenanigans, stubbornness, and supporting me always.

Lastly, I would like to thank my parents, Jeanett and Francisco Oliva. From an early age you both instilled a love for learning that I still carry with me to this day. Your unbridled love and sacrifice warms my heart and fuels me in all my ventures.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The recent increase in location-aware devices equipped with technologies like GPS and RFID has created a great need for the ability to analyze and model trajectory data. The widespread use of devices equipped with such technologies has produced applications in various domains for analyzing trajectory data. Projects involving social movement analysis and animal studies [Frank et al., 2001, CAR] that track individual agents may use anomaly detection techniques in order to identify members of a social group with abnormal travel patterns. Furthermore, projects in traffic analysis like [Gidófalvi and Pedersen, 2007] may use techniques to discover abnormal cab routes and predict route demands. In addition, tracked locations of natural phenomena like hurricanes and tropical storms also beg the ability to effectively model trajectories. For the purposes of this thesis, trajectories are ordered lists of locations traveled by agents recorded at some interval. This thesis aims to develop methods using statistics and machine learning (ML) that provide a deep understanding of trajectory datasets. Namely, this thesis proposes methods to perform anomaly detection, density estimation, and spatial graphical modeling.

Trajectories have many properties that make them inherently difficult to model and analyze. First, the time-series nature of trajectories makes applying traditional techniques challenging. For instance, one will likely have a dataset where instances (trajectories) are of varying dimensionality, which is somewhat uncommon in the literature. Secondly, it is expected that datasets will contain trajectories of many points, leading to a high-dimensional modeling problem, which makes statistical and ML techniques more difficult (e.g. the curse of dimensionality). Lastly, there are multiple simultaneous notions of similarity/difference in trajectories. For example, two paths may share a very similar spatial pattern, yet may do so at very different velocities. Hence, in order to effectively analyze trajectories we must not only choose appropriate techniques but also appropriate representations to deal with these challenges.

One of the tasks discussed in this thesis is identifying anomalous trajectories in a dataset. In general, an anomaly (or outlier) in a dataset is an instance that is abnormal or unlikely based on the rest of the dataset. The exact definition of ”abnormal or unlikely” will depend on the techniques being used. If the instances in the dataset are labeled as $\{$normal, abnormal$\}$ then standard supervised machine learning techniques may be used to perform anomaly detection. This thesis will focus on the case where there are no labels, for which one must rely on unsupervised machine learning techniques. Namely, a method for preforming anomaly detection for trajectories in an unsupervised fashion using support vector machines (SVMs) and various spatial representations of trajectories is developed. In addition, this thesis will also explore Markovian assumptions in junction with nonparametric density estimation to find anomalies. This technique, which is further explained below, is able to assign likelihoods to each trajectory; hence, it is useful not only for anomaly detection, but also modeling. Both methods produced promising results. Unlike many previous approaches that focus on outlier detection in short line segments of an entire trajectory, the methods in this thesis will account for several such line segments.

There are numerous uses for detecting anomalous trajectories. Perhaps most obvious of which is security: if an agent is moving in an abnormal fashion, it may be up to no good. In addition, detecting anomalous trajectories can serve for novelty detection. This is because as new pathways become available, any agents that take these pathways will appear anomalous. Thus, one can uncover emerging novel behavior in agents with anomaly detection. Furthermore, trajectory outlier detection can be used to indicate malfunctioning sensors, since faulty odometers, and other localization sensors will likely deliver abnormal trajectories.

This thesis will also focus on techniques for modeling trajectories, that is providing a likelihood for each trajectory in a dataset and conditional independence structure for spatial locations that trajectories traverse–spatial graphical modeling. Given the aforementioned difficulties, in order to effectively model trajectories one must make some assumptions. One approach explored in this thesis is to make a Markovian assumption on the independence of the next position of a trajectory given its previous positions. In particular, we will assume that the next position of an agent’s trajectory is independent of all other previous positions when given the last two positions. This will allow for the likelihood of a trajectory to be written as a product of conditional and marginal densities of points, which can be estimated using kernel density estimation. Also, in this thesis we explore spatial graphical models. Undirected graphical models give the conditional independence structure for a set of random variables [Bishop, 2006]. This concept is used for indicator variables that have spatial locations associated with them, indicating if an agent has come near (or visited) the corresponding location. Methods for finding conditional independencies among the location indicators given a sparsity assumption on their graphical model are explored. The resulting conditional independencies provide a graphical model for agents' movements across locations, i.e. a spatial graphical model. Namely, the thesis explores using $\ell 1$-regularized logistic neighborhood selection [Wainwright et al., 2007] and forest graphical models [Chow and Liu, 1968] to model the conditional independence structure of a set of indicators for locations (called landmarks) spread over the area enclosing the trajectories. In other words, each trajectory is represented by indicator variables, one for each landmark, which indicate whether the trajectory came near the corresponding landmark; then, methods are explored to determine the conditional independence structure of the indicator variables. The resulting spatial graphical models were visually informative and followed various intuitions.

One use for assigning likelihoods to trajectories is anomaly detection, as already mentioned. If we are able to assign each trajectories a likelihood, then it is natural to consider the least likely to be anomalies. Other possible uses include simulation, and trajectory prediction. Evaluating the conditional independence structure of landmarks serves to inform which other spatial locations one particular landmark depends on; that is, it allows one to know which other landmarks should be monitored in order to predict whether an agent has visited a landmark, which would be useful for surveillance purposes.

1.2 Related Work

A major theme of the previous work in trajectory analysis focuses on short separate segments. I.e. analysis based on using a point's position and corresponding velocity vector or two consecutive points in a trajectory [Lee et al., 2008, Laxhammar et al., 2009, Ristic et al., 2008].

Approaches like these are undoubtedly effective at detecting brief snapshots of anomalous behavior. Notwithstanding, it is entirely possible for a trajectory to have segments that are not anomalous when considered individually, but whose whole path traveled is. Such is the distinction between point and group anomalies [Xiong et al., 2011]. Consider the blue trajectory in Figure 1.1, although no segment itself of the blue trajectory is an outlier, the circular motion of the trajectory as a whole is anomalous. In particular, we see that trajectories that undergo an arc-like motion, as the lower half of the blue trajectory does, go on to the east instead of continuing in a circular motion. Moreover, we see that trajectories that contain an arc-like motion as the upper half of the blue trajectory have motion that originates from the west not from circular motion from below. Thus, in a group context, where one accounts for the progression of trajectory segments, one can detect the anomalous behavior of the entire circular path.

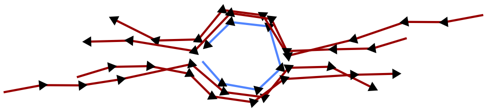


Figure 1.1: We see that although none of the line segments in the blue trajectory are anomalous on their own; as a whole, their progression is anomalous.

Furthermore, as mentioned before, in order to effectively model trajectories assumptions must be made. This thesis will work with Markovian and sparsity assumptions. Other approaches have made assumptions as follows: [Buchman et al., 2011] assumes that trajectories reside in a low dimensional manifold, and [Grimson et al., 2008] assumes that the trajectories can be modeled by ”semantic regions” discoverable using Hierarchical Dirichlet Process-type techniques.

1.3 Notation

The methods presented in this thesis will work over a dataset, $\mathcal{D}$, of trajectories: $\mathcal{D} = \{t^{(1)}, \ldots, t^{(N)}\}$ where each trajectory $t^{(i)}$ is an ordered collection of n_i 2D points that correspond to the location of the agent i at regular intervals; i.e., $t^{(i)} = \langle (x_1^{(i)}, y_1^{(i)}), \ldots, (x_{n_i}^{(i)}, y_{n_i}^{(i)}) \rangle$, and $\forall i, j \; (x_j^{(i)}, y_j^{(i)}) \in \mathcal{S} \subseteq \mathbb{R}^2$. For notational convenience, we may denote the j^{th} point of the i^{th} trajectory by $s_j^{(i)}$, i.e. $s_j^{(i)} := (x_j^{(i)}, y_j^{(i)})$. Given trajectories with entries separated by arbitrary times, one can easily interpolate the location of the agent for some given interval.

1.4 Structure

The structure of this thesis is as follows: in Chapter 2, the real-world datasets that are used for experiments throughout the thesis are discussed; in Chapter 3, a one-class SVM method for detecting anomalous trajectories is developed; in Chapter 4, density estimation of trajectories using Markovian assumptions and KDE is explored; in Chapter 5, sparse methods for finding the conditional independence structure of landmark indicators is introduced; finally the thesis is concluded in Chapter 6.

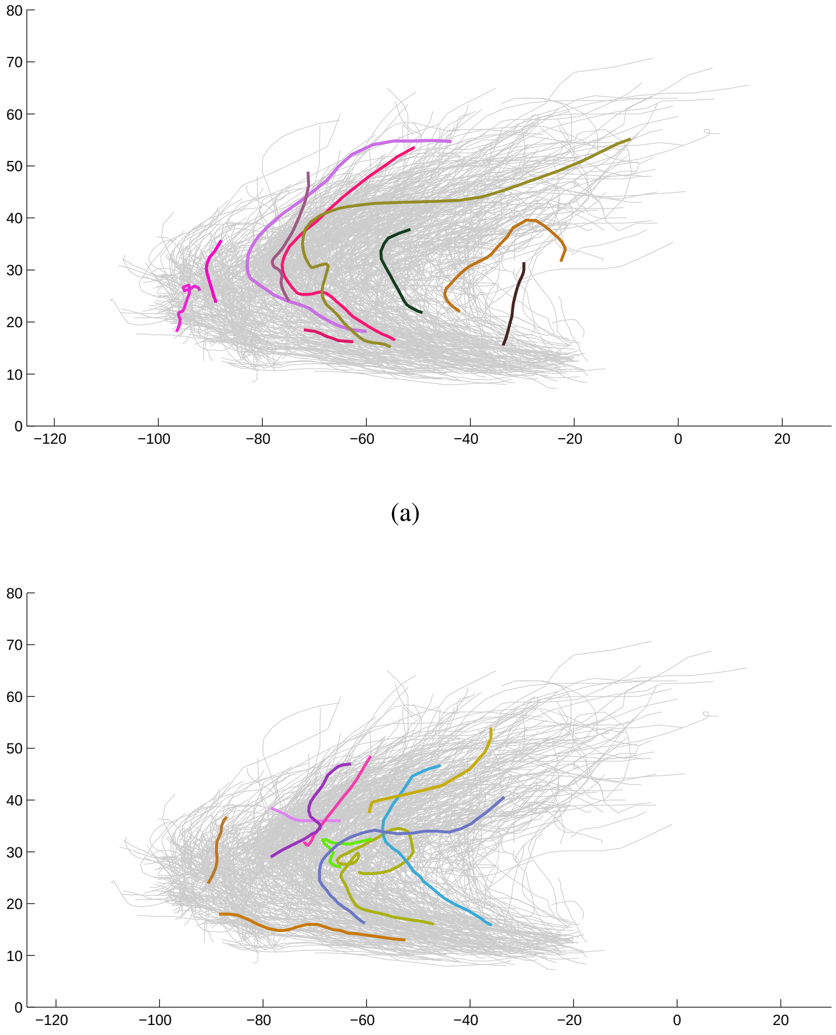
Chapter 2

Datasets

In order to assess the performance of the proposed methods in this thesis it is essential to preform experiments using real-world datasets of trajectories. To this aim we use two real-world datasets: one, a dataset containing trajectories of hurricane and tropical storms; two, a dataset containing trajectories of tracked shipping vessels. Statistics and plots of both datasets can be found below.

2.1 Hurricane Data

One dataset used is from the National Hurricane Center [HUR]. It contained every Atlantic Ocean tropical storm and hurricane track from 1949 to 2011. In total there were 699 trajectories. The positions, intensities, and other data are logged for each storm at 6 hour intervals, however only positions were used for this thesis. Uses for analyzing storm tracks include: prediction of location, and detection of odd/dangerous behavior. The average number of points per trajectory for this dataset is 30.75, the standard deviation of points per trajectory is 17.38. The trajectories' points are spread throughout much of the Atlantic leading to a latitude standard deviation of 10.25 and a longitude standard deviation of 19.95. These and other statistics for the dataset can be found in Table 2.1. All the trajectories in the dataset are plotted in Figure 2.1 in gray with ten random trajectories highlighted in Figures 2.1(a) and 2.1(b).



(a)

(b)

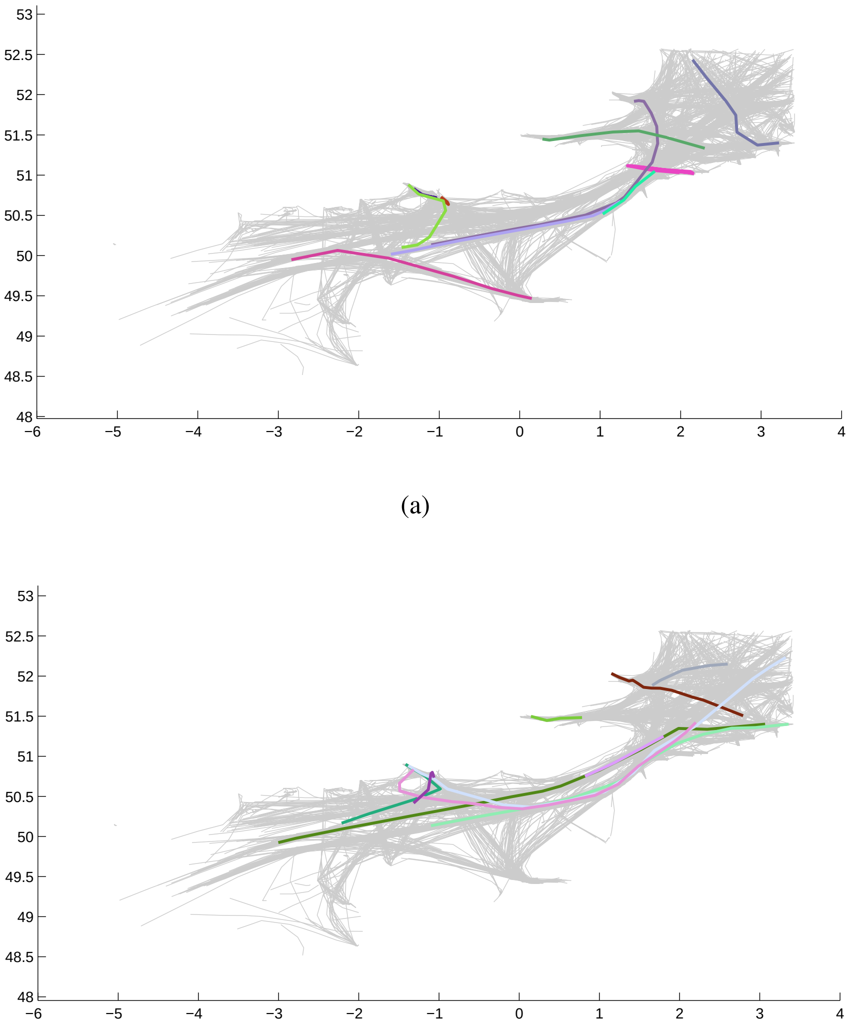
Figure 2.1: A plot of all trajectories in the dataset; 10 random trajectories are highlighted in color in (a) and (b).

Table 2.1: Hurricane Dataset Statistics

Statistic	Value
Total Trajectories	699
Mean Points per Trajectory	30.7568
Std. Dev. of Points per Trajectory	17.3852
Total Points in Trajectories	21499
Minimum Latitude	7.2
Maximum Latitude	70.7
Minimum Longitude	-109.3000
Maximum Longitude	13.5000
Mean Latitude	27.2516
Mean Longitude	-63.1219
Std. Dev. Latitude	10.2541
Std. Dev. Longitude	19.9568

2.2 AIS Data

The Automatic Identification System (AIS) is an automatic tracking system used on vessels for the identification and location of vessels by electronically exchanging data with base stations and other near-by ships. While the AIS protocol allows for logging many attributes, only the attributes of agent identifier, time stamp, and position were considered. The dataset used tracks over 1700 vessels in the English Channel for a total of 5 days leading to over 2100 trajectories. Each trajectory was preprocessed such that consecutive points are the interpolated positions of vessels at one hour intervals; that is a path with 5 points spans 4 hours of travel. Uses for analyzing vessel trajectories include: the detection of illegal activity, emerging market detection, the detection of faulty sensors, and surveillance. The average number of points per trajectory for this dataset is 11.10, the standard deviation of points per trajectory is 7.12. The trajectories' points are spread throughout much of the English Channel leading to a latitude standard deviation of 0.70 and a longitude standard deviation of 1.71. These and other statistics for the dataset can be found in Table 2.2. All the trajectories in the dataset are plotted in Figure 2.2 in gray with ten random trajectories highlighted in Figures 2.2(a) and 2.2(b).



(a)

(b)

Figure 2.2: A plot of all trajectories in the dataset; 10 random trajectories are highlighted in color in (a) and (b).

Table 2.2: AIS Dataset Statistics

Statistic	Value
Total Trajectories	2175
Mean Points per Trajectory	11.1080
Std. Dev. of Points per Trajectory	7.1225
Total Points in Trajectories	24160
Minimum Latitude	48.5167
Maximum Latitude	52.5667
Minimum Longitude	-5.0500
Maximum Longitude	3.4167
Mean Latitude	50.8744
Mean Longitude	0.5795
Std. Dev. Latitude	0.7008
Std. Dev. Longitude	1.7149

2.3 Discussion

The use of both datasets provides a good range of different trajectories to test the proposed methods. The hurricane dataset tracks natural phenomena; in contrast, the AIS dataset tracks man-made movements. As can be seen in the statistics, the hurricane data spans a much larger space than the AIS dataset. This, however, does not present a problem since all the methods can work at different scales. Both datasets contain trajectories that depend on exterior factors. For example, a hurricane's movement may depend on the nearby air temperature and pressure, and a vessel's movement may depend on the cargo it carries and current traffic. It can also be seen that both datasets contain trajectories whose mean number of points are at least an order of magnitude less than the total number of trajectories. While the trajectories in the dataset are by no means of small dimension, the trajectories are not excessively long (as would be the case if the trajectories' mean length was equal to the total number of trajectories, for example). The methods in this thesis were developed for datasets with trajectories of lengths of a smaller order than the total number of trajectories. However, there may be datasets which do have very long trajectories. Thus, future work should test/modify the methods for datasets of very long trajectories.

Chapter 3

One-Class SVMs with Spatial Representations

3.1 Introduction

This chapter develops a technique for detecting anomalous trajectories in a dataset in an unsupervised fashion using Support Vector Machines[1]. SVMs have been effective in other high-dimensional problems [Joachims, 1998]. Although SVMs are usually applied on datasets of instances with the same dimensionality, by using appropriate kernels one can directly apply SVMs for anomaly detection in trajectories of varying lengths. Thus, this project focuses on the use of a variant of SVMs called one-class SVMs [Schölkopf et al., 2001] to perform anomaly detection in trajectories. In order to use one-class SVMs for finding anomalies various different representations of trajectories are developed. That is, in order to appropriately assess which trajectories are outliers, we develop several representations that are informative of the spatial characteristics of each trajectory.

As previously mentioned there are several important uses for detecting anomalous trajectories in a dataset including: security, for detecting illegal or dangerous activities; novelty detection, for discovering emerging markets or new pathways; and faulty sensor detection, for detecting odometers and other localization sensor that are returning odd trajectories.

Besides the previous work mentioned in Section 1.2 for finding anomalies, [Piciarelli and Foresti, 2007] also uses a SVM based approach to find outliers in trajectory datasets.

[1] Work originated in class project [Oliva, 2011b].

However, their approach is based on representing trajectories by sub-sampling them to represent all instances with the same dimensionality. This technique was found empirically to focus only on anomalous trajectories that traverse the edges of the dataset (see Section 3.4.1). Thus, in order to detect a larger range of trajectories, a richer set of representations are explored in this chapter.

The rest of this chapter is organized as follows: Section 3.2 explains the methodology describing one-class SVMs in general and our application to trajectories; Section 3.3 details results of applying this chapter's methods to the hurricane and AIS datasets; Section 3.4 concludes the chapter.

3.2 Methodology

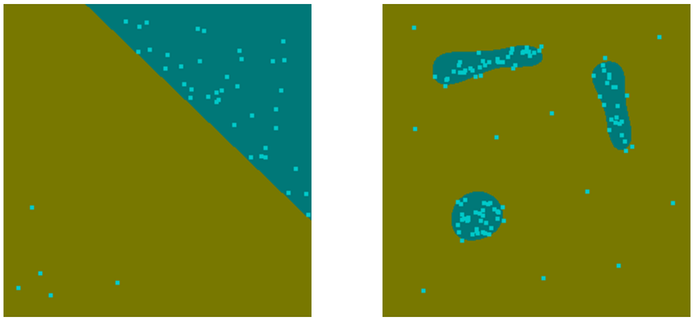
3.2.1 One-Class Support Vector Machines

One-class support vector machines are a fairly popular method for discovering anomalies in a dataset [Schölkopf et al., 2001]. Like other SVM formulations, one-class SVMs are based on a maximum margin problem. Here, the goal is to find the maximum margin hyperplane from origin on an induced featured space F such that most instances x_i are on the positive side (see Figure 3.1(a)).

In order to solve the maximum margin problem, one-class SVMs optimize the following quadratic programming problem:

$$\begin{aligned} \min_{w \in F, \boldsymbol{\xi} \in \mathbb{R}^{\ell}, \rho \in \mathbb{R}} \quad & \frac{1}{2}||w||^2 + \frac{1}{\nu\ell}\sum_i \xi_i - \rho \\ \text{subject to} \quad & (w \cdot \Phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0. \end{aligned}$$

The slack variables ξ_i allow for the outliers to be on the negative side of the hyperplane and the parameter $\nu \in (0, 1]$ acts as an asymptotic upper bound on the ratio of instances that are outliers. The decision function for determining whether an instance x is an outlier is: $f(x) = \text{sgn}((w \cdot \Phi(x)) - \rho)$ where $f(x) = -1$ for outliers (where w and ρ solve the quadratic programming problem). Note that the hyperplane w is over a feature space $\Phi(x)$.This will allow for the hyperplane to operate over non-linear feature spaces. The



(a) Linear

(b) Non-linear

Figure 3.1: The green background represents the space the decision function deems anomalous, blue the space deemed not anomalous. (a) An example hyperplane found with one-class SVMs on a linear space; one can see that most instances lie on the positive side of the hyperplane where a few outliers are allowed to be on the negative side. (b) An example hyperplane found with one-class SVMs on a nonlinear space induced by the Gaussian kernel. As can be seen, the use of the Gaussian kernel allows for a much more expressive decision space than if one operates on a linear space.

corresponding dual problem is:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu\ell}, \sum_i \alpha_i = 1 \end{aligned} \tag{3.1}$$

where k is the *kernel function*, which is the inner product induced by Φ. That is, $k(x, y) = (\Phi(x) \cdot \Phi(y))$ The decision function can then be written as:

$$f(x) = \text{sgn}(\sum_i \alpha_i k(x_i, x) - \rho). \tag{3.2}$$

The use of the Gaussian kernel function:

$$k(x, y) = e^{-\|x-y\|^2/\sigma}$$

can lead to the discovery of nonlinear anomalous areas like in Figure 3.1(b).

3.2.2 Kernels for Trajectories

Representative Distribution Kernels

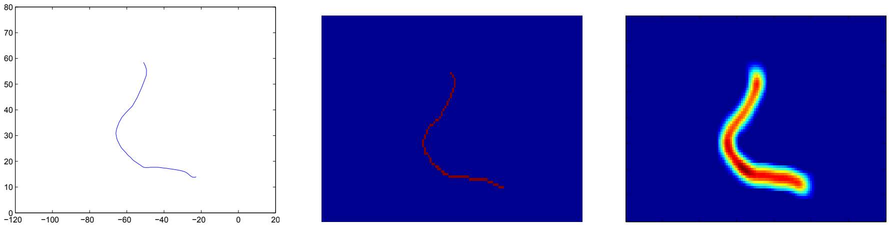
It is possible to use one-class SVMs to perform anomaly detection in trajectories. However, one is unable to directly use traditional kernels like the Gaussian kernel because data instances will vary in length. Furthermore, even if all instances have the same length it is possible for similar trajectories to have observations of points that may differ substantially if not aligned. One way to build a kernel for trajectories is to make a *representative distribution* (RD) for each trajectory; that is, represent each trajectory as a spatial distribution over XY coordinates that is informative of where the trajectory travels through. Then one may use a kernel that works over distributions on the representative distributions.

A way to build a spatially informative RD over points for a trajectory is as follows. If the function $c(s) : [0, 1] \mapsto \mathbb{R}^2$ is the parametric curve describing a trajectory, then one can consider the hierarchical model :

$$\begin{aligned} s &\sim \mathrm{U}[0,1] \\ (x,y) &\sim \mathcal{N}(\mathrm{c}(s), \Sigma) \end{aligned}$$

for some covariance matrix Σ. The distribution above captures information regarding an agent's position on a trajectory; that is, it captures the different snapshot positions one may see from an agent in the trajectory. Such a distribution will be spatially informative since the probability of a region near (or on) the space traveled by the trajectory will be much higher than the probability of areas not near the trajectory. For ease of computation the distribution above can be approximated as a discrete distribution over a quantized state space. This approximation can be carried out by convolving a Gaussian across indicator variables on a quantized map. That is, consider the trajectory as an image where pixels represent small square regions of the space that trajectories travel on, $\mathcal{S}$; if the trajectory passes through the space, then turn that pixel on, otherwise leave it off. One can then do a Gaussian blur, normalize, and consider a distribution over pixels' XY positions (see Figure 3.2). This distribution will henceforth be referred to as the *discrete spatial representative distribution* (DSRP). Discrete approximations like these make computing the kernels for our distributions simpler and more efficient.

After computing the spatial distributions for trajectories, one can then consider kernels among distributions [Jebara et al., 2004, Poczos et al., 2012] to utilize one-class SVMs and find anomalous trajectories. Once the representative distributions p_t, p_s for trajectories t and s is computed, one can use the Gaussian distribution kernel:



(a) Lat/Long coordinates of a hurricane's trajectory

(b) Corresponding indicators in quantize space

(c) Corresponding probabilities for multinomial quantized representative distribution

Figure 3.2: A hurricane trajectory and its corresponding DSRP.

$$k(t,s) = \text{exp}\left(-\frac{1}{\sigma}\int_{x\in\mathcal{S}}(p_t(x)-p_s(x))^2\right).$$

Since we are using discrete distributions, the kernel value is:

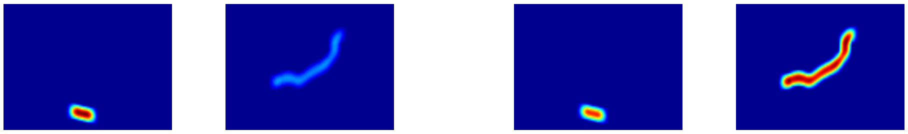
$$k(t,s) = \text{exp}\left(-\frac{1}{\sigma}\sum_{i,j}(P^t_{ij}-P^s_{ij})^2\right) \tag{3.3}$$

where P^t and P^s are discrete spatial distributions for trajectories t and s respectively such that the probability of drawing pixel location (i,j) is P^t_{ij} and P^s_{ij}.

Representative Expectation Kernels

One consequence of using the kernel in (3.3) is that if a trajectory t spans a smaller area (usually because it contains fewer points) it will have its support in a relatively small number of pixels, hence the values P^t_{ij} for pixels (i,j) in the support of the RD for t will be considerably larger than the pixels in the support of P^s_{ij} for the RD of a trajectory s that does not span a small area (see Figure 3.3(a)). I.e. since trajectories that travel smaller distances will have fewer indicators turned on, the values of the resulting pixels after normalizing will be considerably higher than for trajectories with more points.

Since DSDRs are sparse, this will lead to paths that are of small lengths to have low kernel values. This, in turn, will lead one-class SVMs to be biased towards selecting smaller trajectories as anomalies. In order to remedy this we may simply rescale the DSDR by the number of points in a trajectory. That is, for a trajectory t consider the



(a) DSDR

(b) DSER

Figure 3.3: Left: the probabilities for a trajectory containing 9 points. Right: probabilities for a trajectory containing 39 points. Since, the longer trajectory in the right travels through more of the map, the values of each pixel after normalizing will be considerably lower.

discrete spatial expectation representation (DSER) as:

$$E^t = |t| P^t \tag{3.4}$$

where $|t|$ is the number of points in t. E^t_{ij} may be interpreted as the expected number of times pixel (i, j) is picked when drawing from trajectory t's DSDR $|t|$ times. As can be see in Figure 3.3(b), this adjusts the values so that trajectories of varying sizes have similar values in their support. Note that this approach was found to work much better than not normalizing the RD after convolving the Gaussian, which biased anomalies to be trajectories with many points and lacks a probabilistic interpretation.

Then as in with (3.3) we can use the Gaussian kernel:

$$k(t, s) = \exp\left(-\frac{1}{\sigma}\sum_{i,j}(E^t_{ij} - E^s_{ij})^2\right). \tag{3.5}$$

Additional Dimensions

The kernels discussed so far have compared trajectories using *first-order* information about the locations traveled. That is, only the snapshot XY positions are compared; angular and speed information is not. Hence, two trajectories that travel through the same space, but using varying speeds and direction may be indistinguishable. However, this can be easily remedied by extending the kernels to include addition dimensions.

For example, instead of discretizing the space into a 2D matrix of indicators (Figure 3.2(b)), one may discretize it into a 3D matrix of indicators where the third dimension is either orientation or speed (Figures 3.4 and 3.5). Then as before one may use a Gaussian to convolve, normalize, scale by the number of points in the trajectory, and use the Gaussian Kernel. For our purposes we will consider the 3D *discrete angular expectation representation* (DAER). With the DAER a Gaussian can be convolved on a 3D matrix of indicators where the first two dimensions are location and the third dimension corresponds to intervals of angles ($\{[a_0, a_1], (a_1, a_2], ..., (a_{m-1}, a_m]\}$). I.e. the trajectory is represented as a 3D image where a pixel (i, j, k) is turned on if the trajectory passes through the space corresponding to the space covered by the (i, j) pixel and at an angle covered by the discretized k^{th} angle interval–i.e. $(a_{k-1}, a_k]$–and convolved.

Also, we will consider the 3D *discrete speed expectation representation* (DSpER). With the DSpER a Gaussian is convolved on a 3D matrix of indicators where the first two dimensions are location and the third dimension corresponds to intervals of speed ($\{[s_0, s_1], (s_1, s_2], ..., (s_{m-1}, s_m]\}$). That is, the trajectory is represented as a 3D image where a pixel (i, j, k) is turned on if the trajectory passes through the space corresponding to the space covered by the (i, j) pixel and at a speed covered by the discretized k^{th} speed interval–i.e. $(s_{k-1}, s_k]$–and convolved.

3.2.3 Algorithm

Please see below for a high level description of the algorithm to find anomalies in a dataset $\mathcal{D}$ using the one-class SVM methodology described above.

1. Build a new dataset $\mathcal{X} = \{x_1, ..., x_N\}$ where x_i is one of the representations (DSDR, DSER, DAER, or DSpER) for trajectory $t^{(i)}$.
2. Using a quadratic programming solver, optimize α as in 3.1
3. For all $x_i \in \mathcal{X}$, with the α value found in step 2, use the decision function (3.2) to decide if x_i is an outlier; iff it is an outlier report trajectory t_i as an anomalous trajectory.

3.3 Experiments

In order to run the experiments, the one-class SVM implementation from LIBSVM [LIB] was used. The parameter of ν for the one-class SVM was chosen to be .03 leading to roughly 3% of the dataset being labeled as outliers. The bandwidth parameter for Gaussian distribution kernel, σ, was chosen to be near 1 whilst still labeling nearly 3% of the dataset as anomalies, but results were stable for multiple values for the bandwidth.

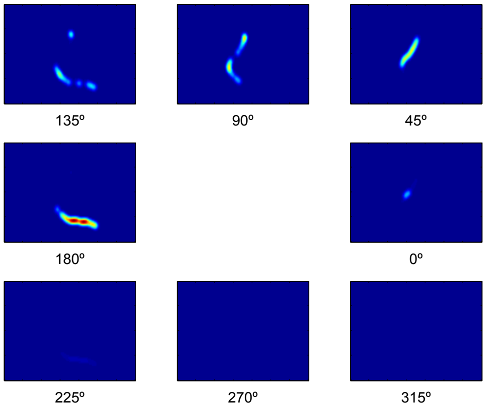


Figure 3.4: DAER for the trajectory shown in Figure 3.2. The first two dimensions correspond to spatial location and the third to angular position. The third dimension is rolled out in the 8 images shown above, corresponding to the following intervals (center-right counterclockwise): $\{[337.5°, 22.5°), [22.5°, 67.5°), [67.5°, 112.5°), [112.5°, 157.5°), [157.5°, 202.5°), [202.5°, 247.5°), [247.5°, 292.5°), [292.5°, 337.5°)\}$. E.g. the image labeled ”180°” details the spatial locations where the agent is moving at an angle in $[157.5°, 202.5°)$.

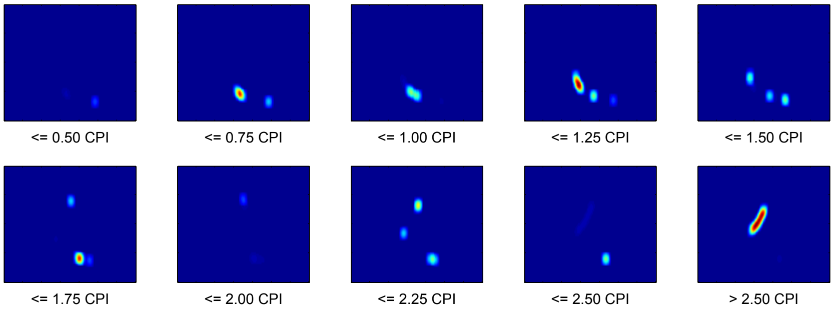


Figure 3.5: DSpER for the trajectory shown in Figure 3.2. The first two dimensions correspond to spatial location and the third to speed. Speed is measured in terms of coordinates per interval (CPI). In the case of hurricanes the coordinates are Lat/Long and intervals are 6 hours. The third dimension is rolled out in the 10 images shown above, corresponding to the following intervals (top-left to bottom-right): $\{[0, .5], (.5, .75], (.75, 1], (1, 1.25], (1.25, 1.5], (1.5, 1.75], (1.75, 2], (2, 2.25], (2.25, 2.5], (2.5, \infty]\}$. For example, the image labeled ”$<= 1.25$” details the spatial locations where the agent is moving at a speed in $(1, 1.25]$.

3.3.1 AIS Data

The Automatic Identification System (AIS) is an automatic tracking system used on vessels for the identification and location of vessels by electronically exchanging data with base stations and other near-by ships. We use a dataset containing AIS tracked positions of vessels in the English Channel (see Section 2.2 for more info). In total, the dataset contains over 2100 trajectories. Uses for anomaly detection in the context of vessels include: the detection of illegal/dangerous activity, emerging market detection, and the detection of faulty sensors.

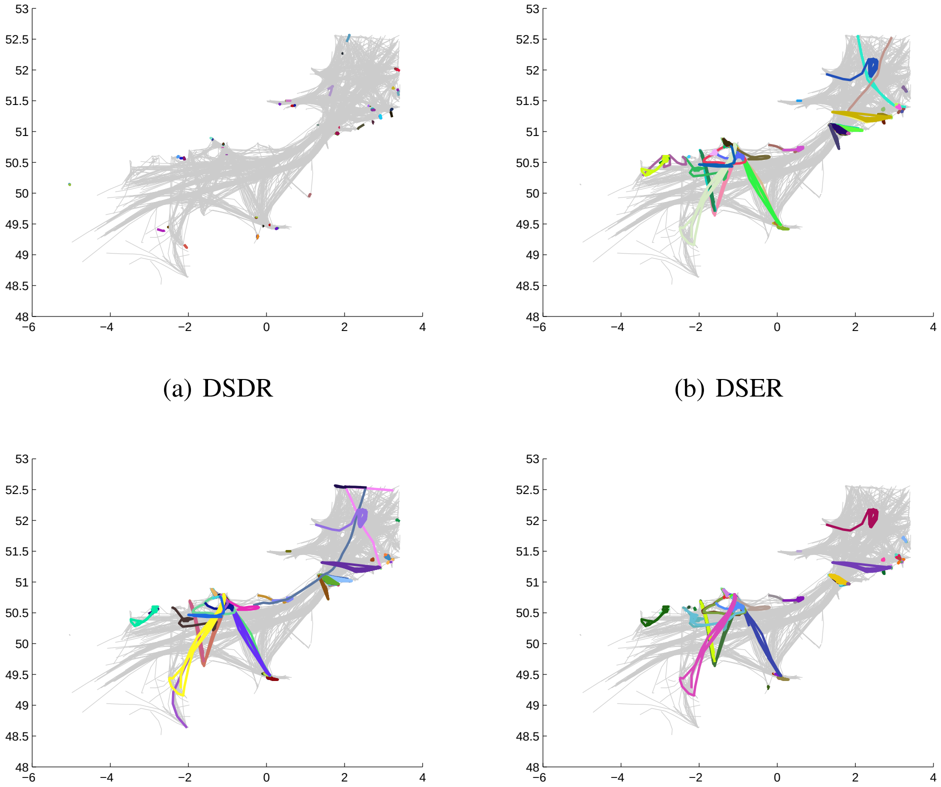
The results using the Gaussian kernel on the various trajectory representations can be seen below in Figure 3.6. First, it can be clearly seen in Figure 3.6(a) that the DSDR is biased to selecting shorter trajectories as anomalies. The DSER, DAER, and DSpER yielded similar anomalies (with a few discrepancies between the sets, see Figures 3.6(b), 3.6(c), and 3.6(d) respectively). All of these representations uncover useful and intuitive anomalies, such as trajectories that cut across perpendicularly to the two major north and south shipping lanes that stretch from the bottom left to the top right of the plots, or vessels that stay stationary in odd locations.

It is interesting to note, however, that a trajectory reported to be going over 200 MPH during some portion (likely due to faulty sensors) was not reported as an anomaly. This is most likely because some portions of the mentioned trajectory did have normal speed where many of the reported anomalies appeared to have abnormal speeds through out their entire trajectories. It is also worth noting that none of the representations resulted in a group of a few trajectories in the lower left corner, near the coordinates (-3,49), traveling through sparsely used locations.

3.3.2 National Hurricane Center Data

Another dataset used for experiments is from the National Hurricane Center [HUR]. It contained every Atlantic Ocean tropical storm and hurricane track from 1949 to 2011 with a total of 699 trajectories (see Section 2.1). Uses for anomaly detection in storm tracks include: the detection and study of odd storms and the conditions that produce them; and the removal of anomalies in datasets to help other statistical tasks.

The results using the Gaussian kernel on various trajectory representations can be seen in Figure 3.7. Again, it can be seen in Figure 3.7(a) that using the DSDR will return anomalies that are shorter in length. Also it is again the case that both the DSER and DAER return similar results (Figure 3.7(b) and Figure 3.7(c)). Moreover, it can be seen



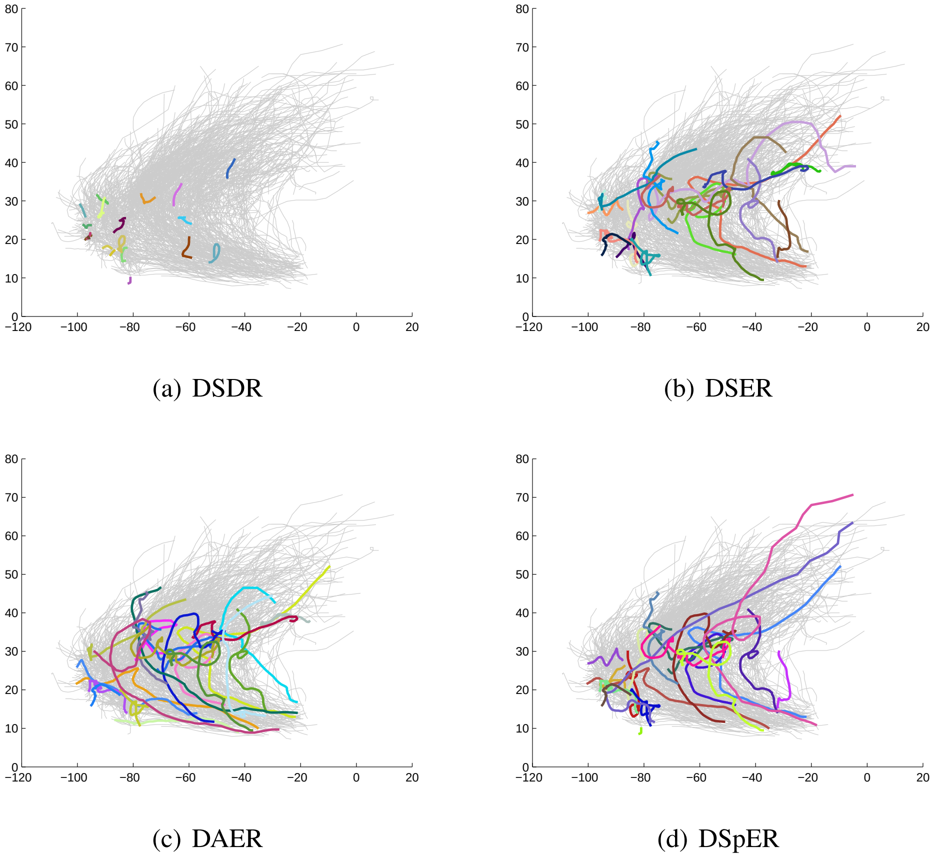
(a) DSDR

(b) DSER

(c) DAER

(d) DSpER

Figure 3.6: The results on the AIS dataset using various representations.



(a) DSDR

(b) DSER

(c) DAER

(d) DSpER

Figure 3.7: The results on the hurricane dataset using various representations.

that the DSpER representation yields anomalies that traverse common locations, because it considers speed in addition to 2D space.

3.4 Conclusions

3.4.1 Discussion

As can be seen in Figures 3.6 and 3.7, the presented methodology does a good job at capturing what appear to be anomalous trajectories. In particular, trajectories that are going against the grain compared to other nearby trajectories, or ones that are at uncommon locations are found. As previously mentioned, using the discrete spatial distribution representation will result in a bias for selecting short trajectories as anomalies; such is the case in Figures 3.6(a) and 3.7(a). But, using the expectation representations resolves this bias.

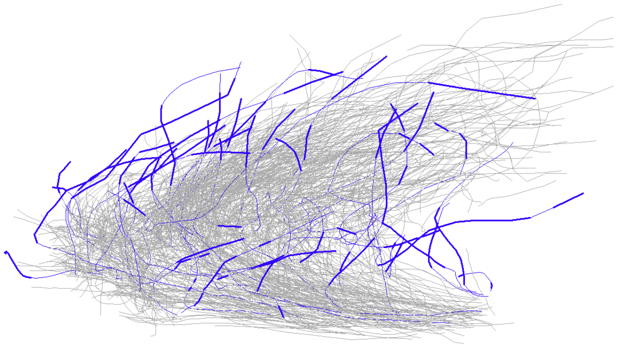
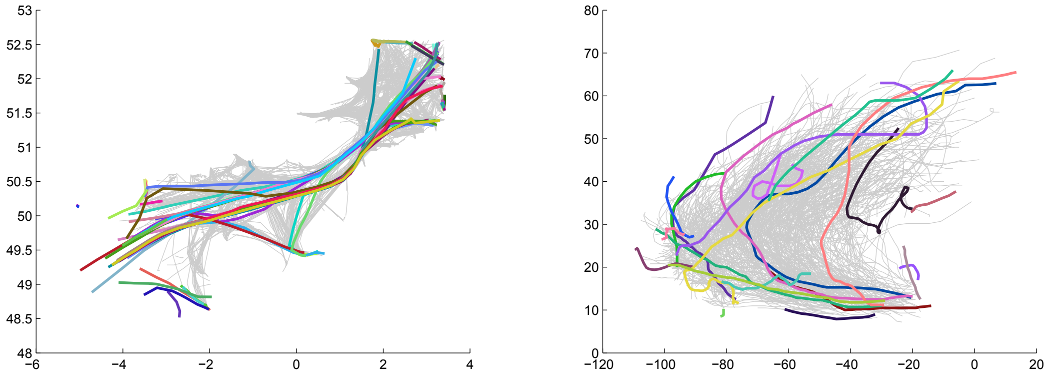


Figure 3.8: Outliers from Lee et al. shown in bold blue sections. Non-bold blue sections correspond to trajectory sections that are not anomalous for trajectories that contain at least one anomalous section.

Although the outliers returned look promising, since the dataset does not contain any labels, there is no ground truth to compare them with. Thus, it is difficult to determine exactly how well the method performs. However, one can compare the results on the hurricane dataset with the segmented outlier approach of [Lee et al., 2008] (described in Section 1.2). The results from [Lee et al., 2008] can be seen in Figure 3.8. Although there is some overlap in the results returned by this chapter's method, and that of Lee et al., there is also a fair amount of difference. Particularly, the method from Lee et al. focuses much more on trajectories on the edges of the dataset.



(a) AIS Dataset

(b) Hurricane Dataset

Figure 3.9: The results on both datasets using the same number of equally spaced points in trajectories.

It is interesting to note that there is a representation using one-class SVMs that can focus in on trajectories that traverse the edges of a dataset like the approach of Lee et al.; specifically, using what is perhaps the most obvious way to represent trajectories of different lengths with the same dimensionality–by representing each trajectory with the same number of equally spaced points. That is, each trajectory is represented using k points equally spaced out in the trajectory's paths, and then the one-class SVM is used with the Gaussian kernel. It can be seen that this representation produces results for trajectories that go through the edges of the datasets (see Figure 3.9). In my opinion, it seems that some of the anomalies returned by Lee et al. and the equally spaced points representation but not by the other representations (DSDR, DSER, DAER, DSpER) are valid, and vice-versa. Hence, it is probable that some sort of combination of the approaches would work best. However, it is also worth noting that the detection of trajectories that go through the edges can be done in a simpler methodology (using a 2D marginal distribution on the points in the dataset). Without expert domain knowledge, getting to the correctness of an unsupervised anomaly detection scheme is difficult. However, there are a few possibilities for assessing the performance of such outlier detection techniques, which will be explored in future work.

3.4.2 Future Work

Although it is not obvious how to best assess the quality of outliers returned by unsupervised methods, there may be a few ways to achieve this. First, one possible way to test unsupervised methods is to generate trajectories from a known distribution. Then, one knows which instances of a particular dataset are the least likely, giving a ground truth to compare results with. Another possibility is to add random trajectories to a dataset. If the methods work well then it should be adept to finding the inserted random trajectories as anomalies.

3.4.3 Conclusion

In conclusion, this chapter presents a technique for performing anomaly detection in trajectories. Namely, this chapter explored using several spatially informative representations of trajectories in order to automatically compare trajectories with the use of the Gaussian kernel and one-class SVMs. In order to ease calculations, a quantized approach was used in creating the representations. First, a distribution of quantized locations based on convolving a Gaussian through the path a trajectory travels through is considered in the DSDR. Then, to alleviate a bias created by using the DSDR, the DSDR is scaled by the number of points in the trajectory to make the DSER, which can be interpreted as the expectation over locations after drawing from the DSDR multiple times according to the number of points in the trajectory. Finally, the DSER was expanded to consider another dimension in addition to 2D space. In particular, the additional dimension of orientation is considered in the DAER and speed in the DSpER. The technique yielded good results in both a dataset containing AIS tracked vessel trajectories and a dataset containing hurricane and tropical storm tracks in the Atlantic Ocean from 1949-2011.

Chapter 4

Markov Assumptions

4.1 Introduction

In this chapter we develop a technique to assign likelihoods to trajectories. One may then consider those trajectory which are the least likely as anomalies. However, because density estimation is less effective in high dimensions and trajectories vary in dimensions some assumptions must be made. For our purposes, we will make a Markovian assumption about the independence of the i^{th} point in a trajectory given all previous points in a trajectory. In particular, we will assume that the i^{th} point is independent of all but the $(i-1)^{\text{th}}$ and $(i-2)^{\text{th}}$ point given all previous points. This assumption will then allow for us to write the likelihood of a trajectory in terms of the product of the marginal probability of the two initial points in the trajectory and probabilities of each of the other points conditioned on the two previous points; both probabilities may be estimated using kernel density estimation.

As previously discussed, due to the high dimensionality of trajectories one must make assumptions in order to effectively assign likelihoods. In Section 1.2 it was mentioned that [Buchman et al., 2011] assumed that trajectories resided in a lower dimensional space where in one may use nonparametric density estimation to assign likelihoods to the lower dimensional mappings. Also, 1.2 describes [Grimson et al., 2008], which assumes that the trajectories can be modeled by a bag of ”semantic regions” discoverable using Hierarchical Dirichlet Process-type techniques. Perhaps a more intuitive assumption is a Markovian assumption, like the one explored in this chapter. A similar Markovian based technique, Hidden Markov Models (HMM), introduces latent states, and (usually) assumes a parametric form for the observed variables given the latent state. Here, the Markov assumption is that the transition from one latent state to the next is independent of all other previous states when given the last state. For example, [Bashir et al., 2007] applies HMMs to trajectories. Furthermore, [Piccardi and Pérez, 2007] provides a method to have a nonparametric form to emissions probabilities. The use of latent states should be less general than the method presented in this chapter, however, because even if there truly are latent states this method may still provide an accurate density estimation of the observed variables.

The rest of this chapter is organized as follows: Section 4.2 explains the methodology to assign likelihoods and select anomalies; Section 4.3 details results of applying this chapter's methods to the hurricane and AIS datasets; Section 4.4 concludes the chapter.

4.2 Methodology

4.2.1 Markovian Assumptions

As previously described, one may define an anomalous trajectory as a trajectory that is unlikely. Thus, if one can estimate the likelihood of each trajectory in the dataset $\mathcal{D}$, then one may designate those trajectories with the lowest estimated likelihoods as anomalies. One challenge with estimating the likelihood of trajectories is that usual techniques for density estimation deal with non-time-series data where data instances (in this case trajectories) must all have the same number of dimensions (in this case points). Another challenge is that density estimation becomes more ineffective in high-dimensional settings. In many real world datasets trajectories will have varying lengths and a relatively high number of points. However, one may mitigate these difficulties by making some independence assumptions.

By the chain rule the likelihood of a trajectory t may be written as:

$$\begin{aligned} \mathrm{p}(t) &= \mathrm{p}\left((x_1,y_1),(x_2,y_2)\right)\cdot \mathrm{p}\left((x_3,y_3)|(x_2,y_2),(x_1,y_1)\right)\cdot \\ &\quad \mathrm{p}\left((x_4,y_4)|(x_3,y_3),(x_2,y_2),(x_1,y_1)\right)\cdot\ldots\cdot \\ &\quad \mathrm{p}\left((x_n,y_n)|(x_{n-1},y_{n-1}),\ldots,(x_1,y_1)\right). \end{aligned} \tag{4.1}$$

One may make a Markovian assumption on the dependence of the i^{th} point in a trajectory, given all the previous points:

$$\mathrm{p}\left((x_i,y_i)|(x_{i-1},y_{i-1}),\ldots,(x_1,y_1)\right) = \mathrm{p}\left((x_i,y_i)|(x_{i-1},y_{i-1}),\ldots,(x_{i-k},y_{i-k})\right) \tag{4.2}$$

That is, the i^{th} point in a trajectory is conditionally independent of all other previous points given the previous k points. For our purposes we will consider $k = 2$. Thus, (4.1) becomes:

$$\begin{aligned}\mathrm{p}(t) \quad = \quad & \mathrm{p}\left((x_1, y_1), (x_2, y_2)\right) \cdot \mathrm{p}\left((x_3, y_3)|(x_2, y_2), (x_1, y_1)\right) \cdot \\ & \mathrm{p}\left((x_4, y_4)|(x_3, y_3), (x_2, y_2)\right) \cdot \ldots \cdot \\ & \mathrm{p}\left((x_n, y_n)|(x_{n-1}, y_{n-1}), (x_{n-2}, y_{n-2})\right).\end{aligned} \tag{4.3}$$

Hence, in order to calculate (4.3) one needs to estimate the conditional probability

$$\mathrm{p}\left((x_i, y_i)|(x_{i-1}, y_{i-1}), (x_{i-2}, y_{i-2})\right) \tag{4.4}$$

and the marginal probability

$$\mathrm{p}\left((x_1, y_1), (x_2, y_2)\right). \tag{4.5}$$

4.2.2 Kernel Density Estimation

One of the most popular non-parametric techniques is Kernel Density Estimation (KDE). With KDE, the density of a d dimensional point x is estimated using the dataset $\{x^{(1)}, \ldots, x^{(N)}\}$ with the formula:

$$\hat{\mathrm{p}}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h^d} \mathrm{K}\left(\frac{\|x - x^{(i)}\|}{h}\right), \tag{4.6}$$

where $\mathrm{K} : \mathbb{R}^d \mapsto \mathbb{R}$ is a symmetric function such that $\int \mathrm{K}(x)\mathrm{d}x = 1$, $\int x\mathrm{K}(x)\mathrm{d}x = 0$, and $\int x^2\mathrm{K}(x)\mathrm{d}x > 0$. For our purposes, we only consider the Gaussian Kernel:

$$\mathrm{K}_d(x) = \frac{1}{(2\pi)^{d/2}} \exp\left(\frac{-x^2}{2}\right). \tag{4.7}$$

Note that for notational convenience the subscript may be omitted, and d corresponds to the dimension of the variables in context. Another possibility is to use the product kernel:

$$\hat{\mathrm{p}}(x) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} \frac{1}{h_j} \mathrm{K}_1\left(\frac{|x_j - x_j^{(i)}|}{h_j}\right). \tag{4.8}$$

Note that both (4.8) and (4.6) are equivalent if using the Gaussian kernel and $\forall j \in \{1, \ldots, d\}$ $h_j = h$. Thus, for a dataset of trajectories $\mathcal{D} = \{t^{(1)}, \ldots, t^{(N)}\}$ we can estimate the marginal (4.5) using (4.6) by:

$$\hat{\mathrm{p}}_{\mathcal{D}}(s_1, s_2) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{I}}^{N} \frac{1}{h_1^d} \mathrm{K}\left(\frac{\|\langle s_1, s_2\rangle - \langle s_1^{(i)}, s_2^{(i)}\rangle\|}{h_1}\right), \tag{4.9}$$

where $\mathcal{I} = \{i : t^{(i)} \in \mathcal{D}\}$, s_j is shorthand for the j^{ith} point in a trajectory, and where $s_i^{(j)} = (x_i^{(j)}, y_i^{(j)})$. Moreover, the Markovian assumption conditional (4.4),

$$\mathrm{p}\left((x_i, y_i)|(x_{i-1}, y_{i-1}), (x_{i-2}, y_{i-2})\right) = \frac{\mathrm{p}\left((x_i, y_i), (x_{i-1}, y_{i-1}), (x_{i-2}, y_{i-2})\right)}{\mathrm{p}\left((x_{i-2}, y_{i-2}), (x_{i-1}, y_{i-1})\right)}, \tag{4.10}$$

can be estimated by

$$\hat{\mathrm{p}}_{\mathcal{D}}\left(s_l|s_{l-1}, s_{l-2}\right) = \frac{\sum\limits_{j\in\mathcal{I}}^{N}\sum\limits_{l=3}^{n_j} \mathrm{K}\left(\|\langle s_{l-1}, s_{l-2}\rangle - \langle s_{l-1}^{(j)}, s_{l-2}^{(j)}\rangle\|/h_2\right) \mathrm{K}\left(\|s_l - s_l^{(j)}\|/h_3\right)}{h_3^2 \sum\limits_{j\in\mathcal{I}}^{N}\sum\limits_{l=3}^{n_j} \mathrm{K}\left(\|\langle s_{l-1}, s_{l-2}\rangle - \langle s_{l-1}^{(j)}, s_{l-2}^{(j)}\rangle\|/h_2\right)}. \tag{4.11}$$

Note, that (4.11) uses (4.8) to estimate the numerator and denominator with one bandwidth h_2 for the dimensions corresponding to s_{l-1} and s_{l-2} and a separate bandwidth h_3 for the dimensions of s_l. Furthermore, note that we use all the triplets of the form $\{\langle s_i^{(j)}, s_{i-1}^{(j)}, s_{i-2}^{(j)}\rangle : j \in \mathcal{I} \wedge 3 \leq i \leq n_j\}$ as the dataset to form our KDE estimate in (4.11). That is, we do not limit ourselves to only the l^{th}, $(l-1)^{\text{th}}$, $(l-2)^{\text{th}}$ tuples: $\{\langle s_l^{(j)}, s_{l-1}^{(j)}, s_{l-2}^{(j)}\rangle : j \in \mathcal{I}\}$.

4.2.3 Cross-Validation

In order to select the bandwidths h_1, h_2, h_3 for (4.9) and (4.11) one may preform cross-validation. For our purposes, we cross-validate the log likelihood. In particular, we look to maximize the leave-one-trajectory-out log likelihood:

$$\mathcal{L} = \sum_{i=1}^{N}\left(\sum_{l=3}^{n_i} \log(\hat{\mathrm{p}}_{\mathcal{D}\setminus\{t^{(i)}\}}\left(s_l^{(i)}|s_{l-1}^{(i)}, s_{l-2}^{(i)}\right)) + \log(\hat{\mathrm{p}}_{\mathcal{D}\setminus\{t^{(i)}\}}\left(s_2^{(i)}, s_1^{(i)}\right))\right) \tag{4.12}$$

Optimizing (4.12) minimizes the KL Divergence from the true density ([Shalizi, 2009]). Note that we leave the entire trajectory corresponding to points out of the dataset ($\mathcal{D} \setminus \{t^{(i)}\}$) to avoid biasing $\mathcal{L}$.

4.2.4 Anomaly Detection

In order to return anomalies we must first compute the leave-one-trajectory-out log likelihood for each trajectory $t^{(i)}$:

$$\log(\hat{\mathrm{p}}_{\mathcal{D}\setminus\{t^{(i)}\}}\left(s_l^{(i)}|s_{l-1}^{(i)}, s_{l-2}^{(i)}\right)) + \log(\hat{\mathrm{p}}_{\mathcal{D}\setminus\{t^{(i)}\}}\left(s_2^{(i)}, s_1^{(i)}\right)). \tag{4.13}$$

Then we may return the trajectories corresponding to the lowest $m\%$ log likelihoods in the dataset, where m is a small number.

Algorithm

Please see below for a high level description of the algorithm to find anomalies in a dataset $\mathcal{D}$ using the Markovian methodology described above.

1. Cross-validate the bandwidths h_1, h_2, h_3 as described in Section 4.2.3.
2. Using the bandwidths optimized in step 1, calculate the leave-one-trajectory-out log likelihood (4.13).
3. Sort the log likelihood, select trajectories that correspond to the smallest $m\%$, return them as anomalies.

4.3 Experiments

Experiments were preformed using the AIS and Hurricane datasets. In both cases the bandwidths were selected using cross validation as explained above. The results can be seen in Figures 4.1 and 4.2. The nature of the likelihood estimate (4.13) is such that trajectories that go across odd areas, or take abnormal speeds, or go against the grain of most trajectories will produce low likelihoods. This is because marginal 4.9 and conditional 4.11 probabilities will be low for the aforementioned cases (and for other scenarios outside the norm). Thus, the likelihood approach used in this method is very adept to choosing trajectories with anomalous behavior.

It is interesting to note that in both datasets the anomalies returned contain many of the anomalies reported using the SVM method previously described (see Figures 3.6 and 3.7). Moreover, it is also interesting to note that for the AIS dataset this method did pick out the trajectory reported going over 200 MPH and the group of a few trajectories in the lower left corner, near the coordinates (-3,49) traveling through sparsely used locations, which the SVM method did not. Also, the method picked some hurricane trajectories traveling through the edges or extremely north that the SVM method did not. However, this method did not pick up some of the trajectories traversing perpendicularly through the major shipping lanes in the AIS dataset.

Also, it is worth noting that the number of points per trajectory whose conditional probability (4.11) is less than the mean value minus the standard deviation of the conditional

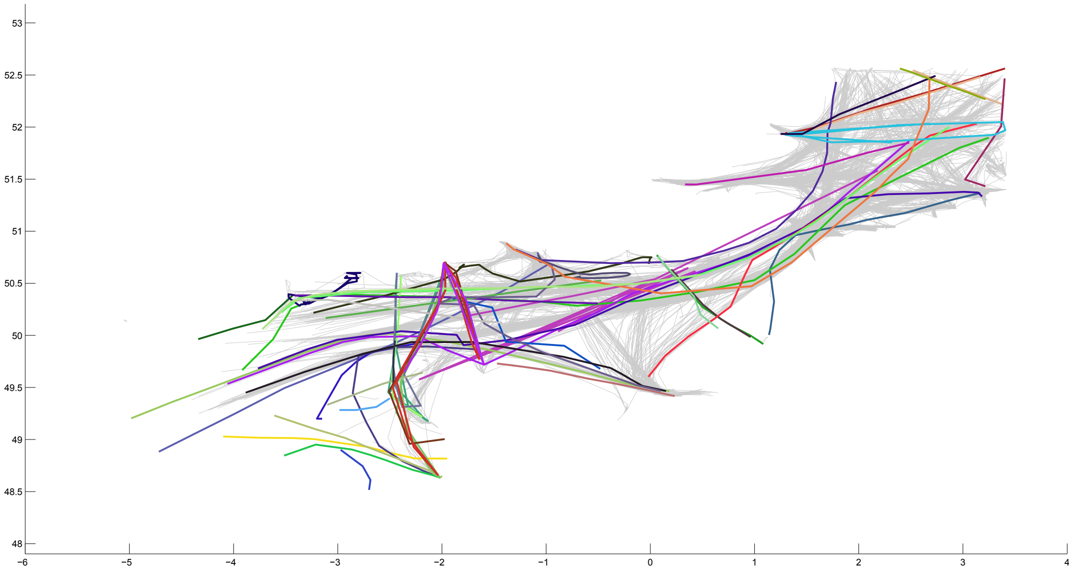


Figure 4.1: Anomalies in AIS dataset highlighted in colors.

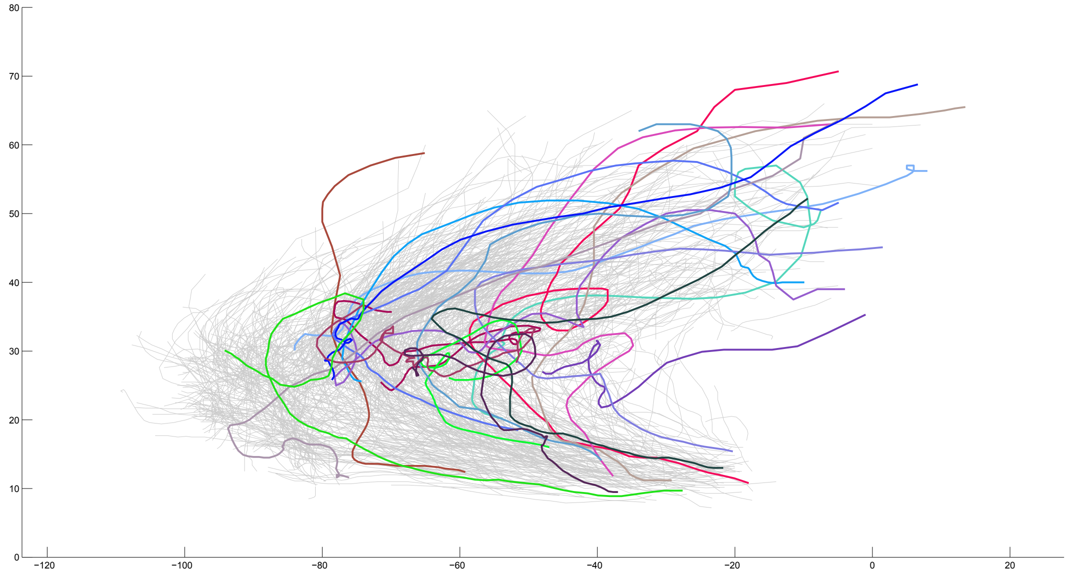


Figure 4.2: Anomalies in hurricane dataset highlighted in colors.

probabilities for all non-initial points (third or after) in the dataset is much higher for the trajectories that are anomalous than for those who are not. In the AIS dataset, anomalous trajectories have 4.07 non-initial points per trajectory with low conditional probabilities (less than the mean minus the std. dev. for all non-initial points in all trajectories in the dataset); in contrast, only 0.58 points per trajectory had low conditional probabilities for normal trajectories. Furthermore, for the hurricane dataset, we see that in anomalous trajectories 17.45 non-initial points per trajectory had low conditional probabilities; but, 4.8689 non-initial points per trajectory had low conditional probabilities for normal trajectories. This indicates two important facts. First, as would be expected, the anomalous trajectories, on average, contain more low density portions than normal trajectories do. Second, most anomalies found contain several portions of low density; contrast this with trajectories that may behave extremely oddly at only one point but then immediately return to normalcy. Similarly to the conditional probabilities, it was also found that anomalous trajectories were more likely to have a low marginal probability for initial points (4.9) than normal trajectories.

4.4 Conclusion

In conclusion, this chapter has developed a method for density estimation that is based on a Markovian assumption and kernel density estimation. The method assumes that the next position of an agent's trajectory is independent of all other previous positions when given the last two positions. By making this assumption, one is able to write the likelihood of a trajectory (4.1) in terms of a marginal probability for the trajectory's initial points and conditional probabilities for subsequent points (4.3). In turn, this decomposition allows one to estimate the likelihood (4.3) using kernel density estimation on relatively small dimensional vectors. That is we may estimate the marginal probability with (4.9) and the conditional probabilities with (4.11). After cross-validating the bandwidth parameters for each dataset and computing the leave-one-trajectory-out log likelihood (4.13) for trajectories using the optimized bandwidths, anomalies were returned from the lowest 3% values for log likelihood. The results found for the AIS and hurricane dataset (Figures 4.1 and 4.2 respectively) were very promising and able to detect odd trajectories.

Inaccuracies in selecting anomalies using the estimated likelihood (4.13) are possible for two reasons. First, the KDE estimates (4.9) and (4.11) will be inaccurate to some extent. However, the inaccuracies decrease as the sample size of the KDE dataset increases [Lafferty et al., 2011]. Since the datasets for the KDE estimates–the XY points for (4.9) and the triplets for (4.11)–contained a relatively large number of instances for both the

AIS and hurricane data (in the thousands), the KDE inaccuracies do not play a large role. Of course, as one expands the number of points in the Markovian assumption far past 2, this source of error will no longer be negligible.

The other possible reason for inaccuracies is in the Markovian assumption that (4.1) does in fact equal (4.3). The true density may turn out to be different than (4.3) if the conditional independence assumption (4.4) is incorrect. In fact, in any real world dataset (4.4) will likely be incorrect since, realistically, an agent's next position is not completely determined by its last two (or k, for finite k) positions. For example, a hurricane's next position may depend on the air temperature around it, a ship on the amount of fuel it has. Notwithstanding, if the Markovian assumption is approximately correct and the next point may be reasonably predicted by the last two points, then anomalies found will still be of use. However, if trajectories in a dataset are such that they do not approximately follow a Markovian assumption (that is, the assumption is grossly incorrect) then the anomalies returned will very likely be invalid. Still, it is not unreasonable to suspect that a lot of the information about an agent's next position can be gathered from a few of its previous positions.

In fact, the results shown in Figures 4.1 and 4.2 are quite promising. In both instances, the methodology was able to capture paths that are traveling unusual locations, or are going against the grain of the majority of paths, or are going at bizarre velocities.

Future work will concentrate on ways of improving the Markovian assumption, such as considering additional points or statistics of previous points.

Chapter 5

Spatial Graphical Models

5.1 Introduction

When analyzing trajectories, it may be important to understand movement patterns of agents as they traverse through locations (henceforth referred to as landmarks). Possible landmarks include ports, buildings, or any other arbitrary stationary coordinate. In order to better understand the relationship among the landmarks as agents traverse through their trajectories, this chapter introduces the concept of spatial graphical models[1]. Undirected graphical models detail the conditional independence structure of a set of random variables [Bishop, 2006]. This concept is used for indicator variables that have spatial locations associated with them indicating if an agent has come near (referred to as visiting) the corresponding location. Methods for finding conditional independence relationships among the location indicators given a sparsity assumption on their graphical model are explored. A spatial graphical model allows one to know conditional independencies amongst the landmarks, which is useful if one is predicting whether an agent visited a landmark based on other landmark visits. That is, it would be beneficial to know the set of landmarks $\mathcal{M}$ such that the visit to a landmark l is independent to visits to all other landmarks when given visits to $\mathcal{M}$ (i.e. the visits to $\mathcal{M}$ is the Markov blanket for a visit to l). By knowing $\mathcal{M}$, one knows exactly which other landmarks are necessary to be monitored in order to predict a visit to l. Moreover, spatial graphical models are telling of a structure underlying movements of agents in space, which undoubtedly expands one's knowledge of the nature of trajectories in a dataset.

In order to derive the conditional independencies amongst a set of landmarks, this

[1] Work originated in class project [Oliva, 2011a]

chapter investigates using high dimensional methods to build graphical models. If one does not have specific landmarks of interest for a dataset, it would be useful to uncover pertinent locations in the dataset and set these as the landmarks. Hence, a method for finding pertinent locations for use as landmarks is also presented in this chapter. This chapter explores using $\ell 1$-regularized logistic neighborhood selection [Wainwright et al., 2007] and forest graphical models [Chow and Liu, 1968] to spatially model trajectories.

In order to build the spatial graphical models, first one considers a set of landmarks spread over the area enclosing the trajectories. Then, each trajectory is represented by indicator variables indicating which landmarks the trajectory came near to. That is, each trajectory will be represented as a multidimensional binary vector of indicator variables, one for each landmark, where each indicator is on if the trajectory came near the corresponding landmark, off otherwise. Specifically, suppose one has a dataset, $\mathcal{D}$, of trajectories sampled from some unknown distribution P over the set of all trajectories $\mathbb{D}$. Furthermore, suppose that there exists a bounded subset of $\mathbb{R}$ such that all trajectories lie inside that space. That is, $\exists \mathcal{R} = [a, b] \times [c, d]$ s.t. $\forall t \sim P$, $\forall (x, y) \in t$, $(x, y) \in \mathcal{R}$. Let a collection of k landmarks in R, $\mathcal{L}$, be given. I.e. $\mathcal{L} \subseteq \mathcal{R}$ and $\mathcal{L} = \{l_1, \dots, l_k\}$. Also, let a *near indicator* function $f : \mathbb{D} \times \mathcal{R} \mapsto \{0, 1\}$ be given where $f(t, l) = 1$ if trajectory t is considered to go *near* landmark l, $f(t, l) = 0$ if not. Then let the mapping SP : $\mathbb{D} \mapsto \{0, 1\}^k$, where SP$(t) = \langle f(t, l_1), \dots, f(t, l_k) \rangle$ is the *spatial profile* of trajectory t. The goal of this chapter is to estimate the graphical model of spatial profiles; that is, estimate the graphical model of the distribution of SP(t) where $t \sim P$. The dataset $\mathcal{S} = \{\text{SP}(t^{(1)}), \dots, \text{SP}(t^{(N)})\}$ will be used to derive said estimation.

Previous work in graphical models with spatial data include the following: [Irvine and Gitelman, 2011] studies various graphical models for modeling ecological stream health at various locations; [Harrington Jr and Hero III, 2010] explore an $\ell 1$-penalized based approach for spatio-temporal graphical models for the susceptible, infected, recovered (SIR) model. Unlike the aforementioned studies, this chapter explores agent movement through locations. Moreover, the sparsity and distribution assumptions invoked are different from the mentioned prior work.

5.2 Methodology

In order to find the graphical model for spatial profiles this chapter focuses on two structure learning methods: $\ell 1$-regularized logistic neighborhood selection and forest graphical models (Chow-Liu), which are outlined below. Furthermore, Section 5.2.3 describes one way to build spatial profiles.

5.2.1 $\ell 1$-Regularized Logistic Neighborhood Selection

Suppose that we consider the $x \in \{0,1\}^p$ being distributed by an Ising model, that is:

$$p(x;\theta) \propto \exp\left(\sum_{i\in V}\theta_i x_i + \sum_{(i,j)\in E}\theta_{i,j}x_i x_j\right) \tag{5.1}$$

where x_i is the i^{th} dimension of x. In order to estimate the graph $G = \langle V, E\rangle$, the same approach as [Wainwright et al., 2007] is used, which states that we may estimate the neighborhood of the node $i \in V$, $\mathcal{N}(i)$, by using $\ell 1$-regularized logistic regression. If x is distributed as (5.1), then

$$p(x_s = 1|x_{\setminus s};\theta) = \left[1 + \exp(-\theta_s - \textstyle\sum_{(s,j)\in E}\theta_{s,j}x_j)\right]^{-1} \tag{5.2}$$

where, $x_{\setminus s} = \{x_i : i \neq s\}$. I.e. x_s is given by a logistic regression model with its neighbors. The method presented in [Wainwright et al., 2007] then performs $\ell 1$ regularized logistic regression to estimate the neighborhood of a node in the graphical model. That is, if one has a sample $\{x^{(1)},\dots,x^{(n)}\}$, then one finds

$$\hat{\theta}^{s,\lambda} = \arg\min_{\theta\in\mathbb{R}^p}\left\{\frac{1}{n}\sum_{i=1}^{n}\left[\log(1+\exp(\theta^T z^{(i,s)})) - x_s^{(i)}\theta^T z^{(i,s)}\right] + \lambda_n\|\theta_{\setminus s}\|_1\right\} \tag{5.3}$$

where $z^{(i,s)} \in \{0,1\}^p$ is a vector where $z_j^{(i,s)} = x_j^{(i)}$ for $j \neq s$ and $z_s^{(i,s)} = 1$. The estimate for $\mathcal{N}(s)$ is given by:

$$\hat{\mathcal{N}_n}(s) = \left\{j \in V, j \neq s : \hat{\theta}_j^{s,\lambda} \neq 0\right\}. \tag{5.4}$$

Then, this estimate is consistent with high probability given certain conditions discussed in Section 5.4.

5.2.2 Forest Graphical Models

Another method for making sparse graphical models is by enforcing a forest structure. The optimal such graphical model can be computed by finding a maximal weight spanning tree for a graph where the weight $w(i,j)$ of the edge connecting nodes i and j is given by $I(X_i;X_j)$ the mutual information for dimensions i and j [Chow and Liu, 1968]. Since,

$$I(X_i;X_j) = \sum_{x_i,x_j\in\{0,1\}} p(x_i,x_j)\log\left(\frac{p(x_i,x_j)}{p(x_i)p(x_j)}\right) \tag{5.5}$$

where each $p(x_i, x_j)$ is a bivariate distribution and each $p(x_i)$ is a univariate distribution, it can be estimated by

$$\hat{I}(X_i; X_j) = \sum_{x_i, x_j \in \{0,1\}} \hat{p}(x_i, x_j) \log \left(\frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)} \right) \tag{5.6}$$

where $\hat{p}(x_i, x_j)$ and $\hat{p}(x_i)$ are the MLE estimates, i.e. the sample frequencies. This algorithm is referred to as Chow-Liu; further details can be found in Section 5.4.

5.2.3 Landmark and Spatial Profile Creation

Although the methodology for finding the graphical model of spatial profiles presented in this chapter is not dependant on how the landmarks and spatial profiles are constructed, it would be beneficial to construct them in a way that is informative of the spatial behavior of trajectories. This section describes one way to accomplish this. First, in order to find pertinent locations to act as landmarks one may use k-means on the dataset containing the 2D points in the trajectories of $\mathcal{D}$. Here the parameter k can be chosen to control the granularity of our spatial profiles; the higher one chooses k, the more detail the spatial profile will contain about the exact locations visited by its corresponding trajectory. Of course, one may use other clustering techniques to choose landmarks, but k-means does a fair job of choosing evenly spaced (by geometric density) landmarks, which is desired (see Figure 5.1 for an example). That is, k-means will tend to place more landmarks in very congested areas, and fewer in less congested areas.

Once one obtains the k landmarks $\{l_1, \ldots, l_k\}$, one needs a method for determining whether or not a trajectory came near each landmark, i.e. computing $f(t, l_i)$. One obvious way to do this is to calculate whether the minimum distance of l_i and a point in the parametric curve for trajectory t is less than a threshold value. However, since some landmarks are over less congested areas, they may be more spread out (have a larger variance) and would rarely, if ever, be ”on” in spatial profiles using this method; notwithstanding, it would be beneficial to track whether trajectories do come near these less congested landmarks. One way of having a variance dependent definition of near in spatial profiles is to compute the mean pdf value for the trajectory with a Gaussian located at each landmark with corresponding sample covariance matrices. That is, for the i^{th} landmark, we compute $m_i = \mathbb{E}[\phi(X; l_i, \Sigma_i)]$ where $\phi(x; l_i, \Sigma_i)$ is the normal pdf with mean l_i and covariance matrix Σ_i given by the sample covariance matrix of the points assigned to l_i in k-means, and X is given by $c(s)$ where $c(\cdot)$ is the parametric curve for the trajectory and $s \sim \text{Unif}[0, 1]$ .Then, if m_i is larger than some threshold τ the corresponding indicator variable is turned on; i.e. $f(t, l_i) = \text{I}\{m_i > \tau\}$. Two example spatial profiles can be seen in Figure 5.2.

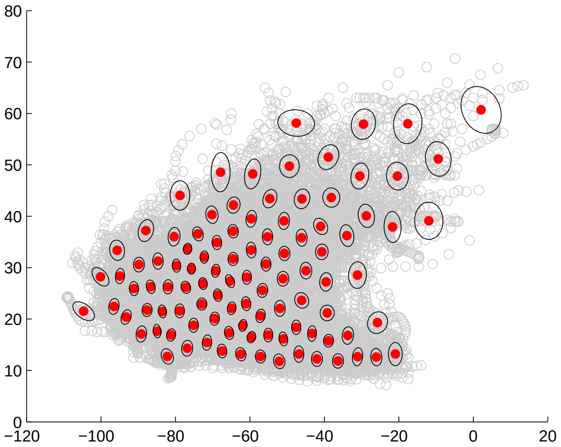


Figure 5.1: 100 landmarks used to build a trajectory's spatial profile are shown in red. The sample covariance for each center is show in black.

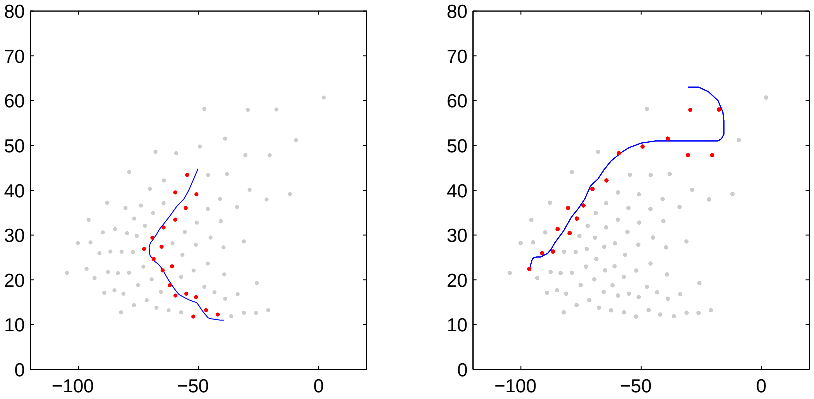


Figure 5.2: Two example spatial profiles, where the indicators are red if they are on, gray when off.

5.2.4 Algorithms

Please see below for a high level description of the algorithms to find the spatial graphical models for a datasets $\mathcal{D}$ of trajectories, and $\mathcal{L} = \{l_1, \ldots, l_k\}$ of landmarks using the methodologies described above. Note, that if one does not have landmarks $\mathcal{L}$ ahead of time, one may make them as in Section 5.2.3.

$\ell1$-Regularized Logistic Neighborhood Selection

1. Create the spatial profiles $\mathcal{X} = \{x^{(1)}, \ldots, x^{(N)}\}$ where $x^{(i)}$ is the spatial profile corresponding to trajectory $t^{(i)}$ as described in Section 5.2.3.

2. For all $i \leq k$

   Preform $\ell1$ regularized logistic regression on dimension i, x_i, of the spatial profiles with covariates $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k$ as in (5.3).

   Calculate the $\hat{\mathcal{N}_n}(i)$ as in (5.4) for dimension i.

   Iff $j \in \hat{\mathcal{N}_n}(i)$ then add edge (i, j) to the graphical model G.

3. Return graphical model G.

Forest Graphical Models

1. Create the spatial profiles $\mathcal{X} = \{x^{(1)}, \ldots, x^{(N)}\}$ where $x^{(i)}$ is the spatial profile corresponding to trajectory $t^{(i)}$ as described in Section 5.2.3.

2. For all $i \leq k$ and $j \leq k$ calculate $\hat{p}(x_i)$ and $\hat{p}(x_i, x_j)$ for $x_i, x_j \in \{0, 1\}$, the sample marginal and joint respectively frequencies for dimensions i and j of the spatial profiles.

3. For all $i \leq k$ and $j \leq k$ calculate $\hat{I}(X_i; X_j)$ (5.6) using $\hat{p}(x_i)$ and $\hat{p}(x_i, x_j)$ found in step 2.

4. Build a graph with nodes $\{1, \ldots, k\}$ where an edge between nodes i and j given by the value $\hat{I}(X_i; X_j)$ found in 3.

5. Find the maximal weight spanning tree, G to the graph found in step 4; return this tree as the graphical model.

5.3 Results

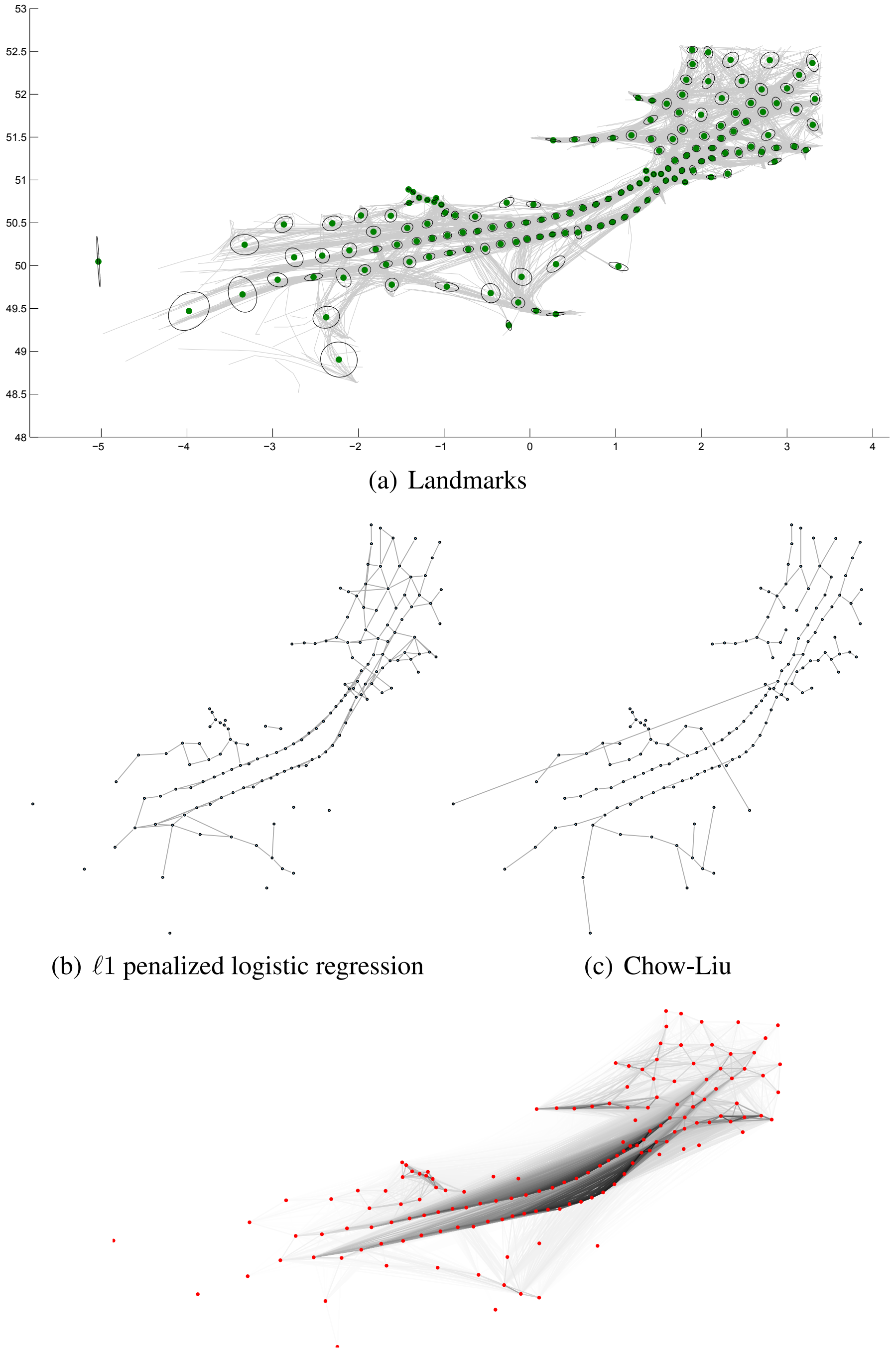
5.3.1 AIS Dataset

First, consider the AIS dataset containing over 2100 trajectories across the English Channel (see Section 2.2). The trajectories (gray) and the landmarks for the dataset are plotted in Figure 5.3(a). For this experiment 150 landmark positions were chosen with k-means, as described in Section 5.2.3.

There are several interesting takeaways from the resulting graphical models in Figures 5.3(b) and 5.3(c). First, in both graphs, as one would expect, landmarks that are not near each other (and more than a couple of hops away on a KNN graph) are independent given the rest of the landmarks. This is expected because as an agent is traversing through a trajectory he will visit a landmark's neighbor before reaching the landmark itself, hence neighbors will be highly informative. Secondly, it is worth comparing the graphical models to the co-occurrence graph of the landmarks (Figure 5.3(d)), where edges among landmarks are weighted by the number of spatial profiles where both landmarks are visited. The co-occurrence graph provides a good picture into why some of the edges in the graphical models are selected. In particular, the two very dense lanes in the middle have strong edge weights (even among non-neighbors). This fact is reflected in the two graphical models, which connect the two lanes, but only through nearby neighbors due to the redundancy in the co-occurrence graph. The co-occurrence graph is fairly dense, much more so than the graphical models; hence, one may loosely interpret the graphical model selection methods as ways to prune redundancies in the graph. Note however, that a lack of co-occurrence does not imply independence, thus this interpretation should not be taken literally. Lastly, it is also interesting to see that the forest graphical model is very similar to the $\ell 1$-regularized graphical model, except that some edges are removed to preserve the tree structure.

For comparison purposes, experiments were also ran where the landmarks were chosen by overlaying a uniform grid over the space containing the trajectories (Figure 5.4(a)). Since in order to build spatial profiles we must have covariance matrices for Gaussians at each landmark position, we choose only the landmarks with more than one point corresponding to them (Figure 5.4(b)). The results using the $\ell 1$-penalized logistic regression and Chow-Liu algorithm are shown in Figures 5.4(c) and 5.4(d) respectively.

Most of the points mentioned previously for the experiments with the landmarks chosen by kmeans remain true. Landmarks that are not near each other are independent given neighbors and both graphical models are similar–with CL yielding a sparser graph. However, when comparing the results of Figures 5.3(b) and 5.3(c) with Figures 5.4(c) and



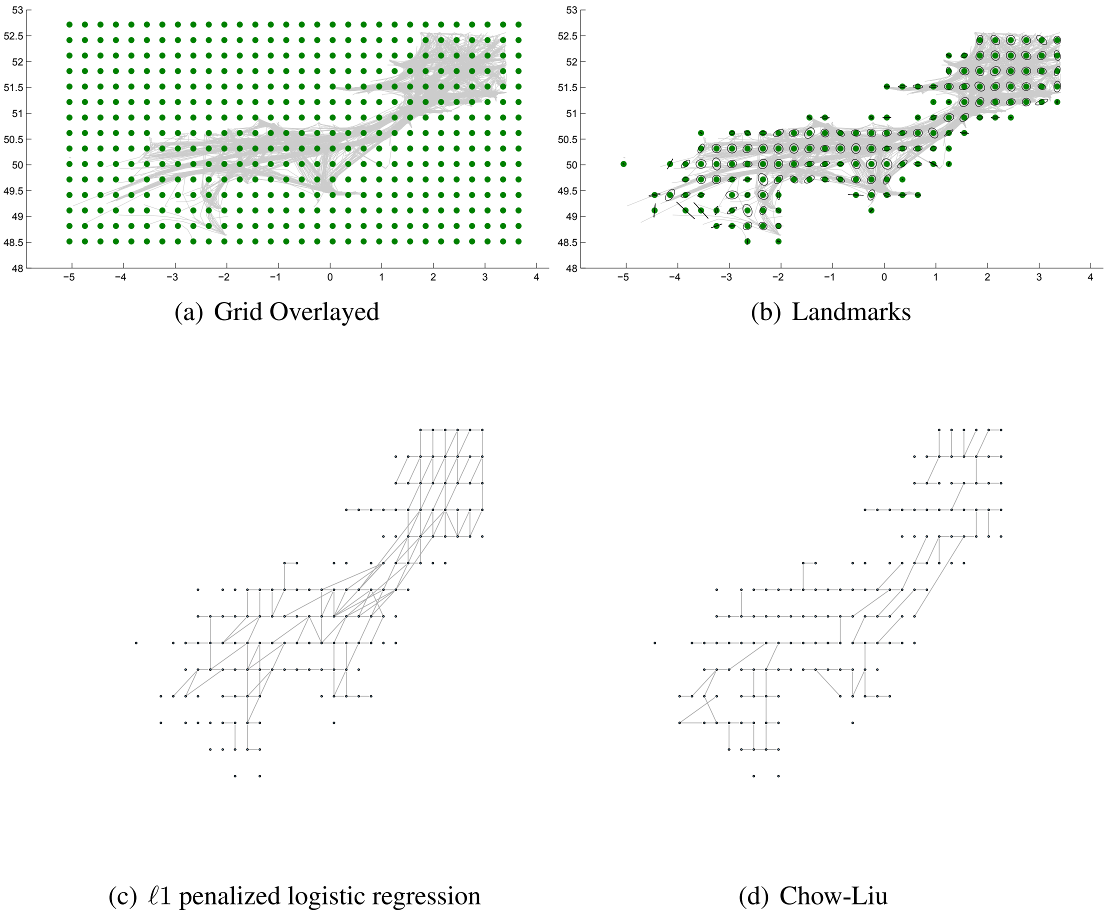
(a) Landmarks

(b) $\ell 1$ penalized logistic regression

(c) Chow-Liu

(d) Co-occurrence Graph

Figure 5.3: Results with k-means selected landmarks in AIS dataset.



(a) Grid Overlayed

(b) Landmarks

(c) $\ell 1$ penalized logistic regression

(d) Chow-Liu

Figure 5.4: Results with grid overlayed landmarks in AIS dataset.

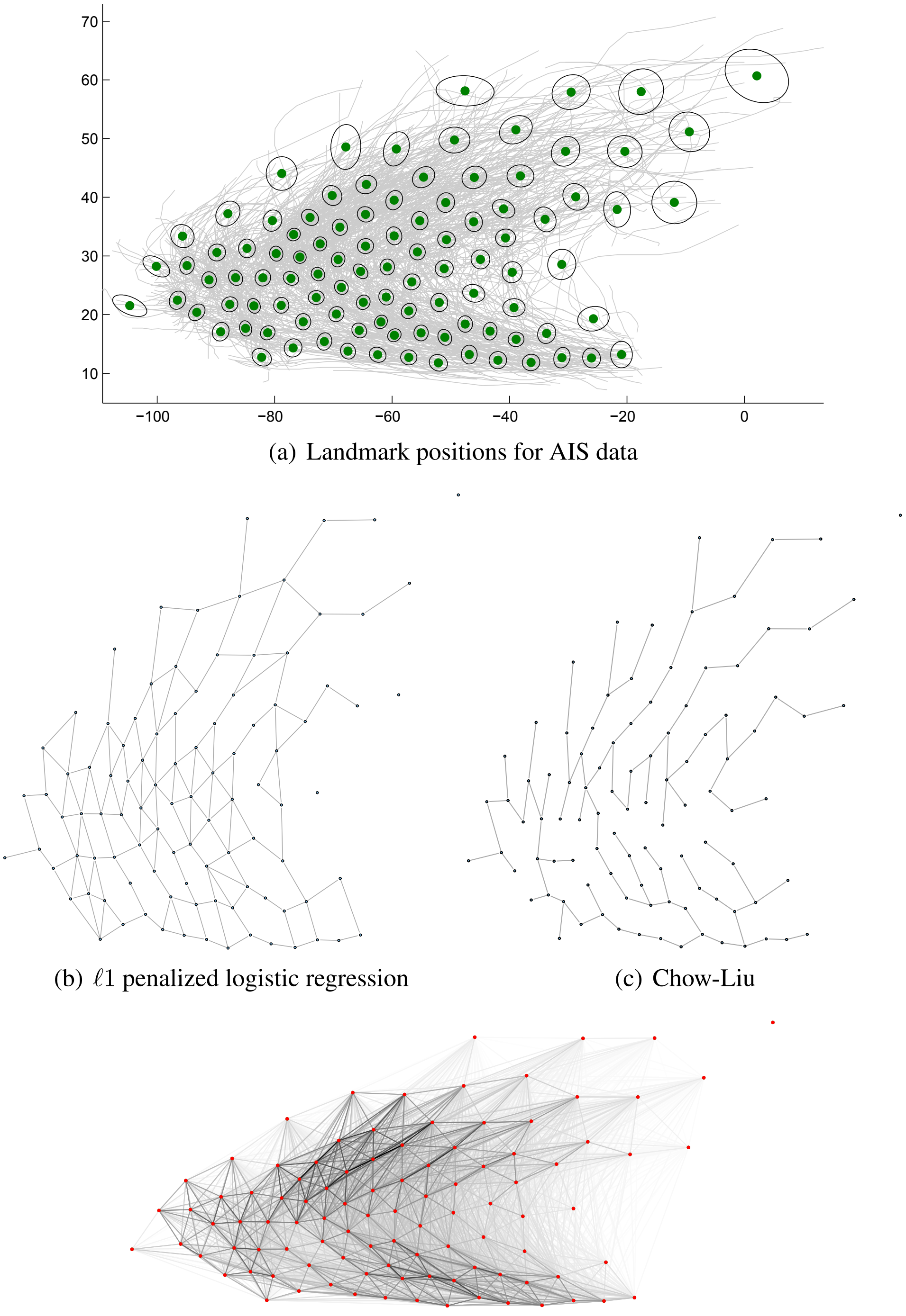
5.4(d) it is clear that using k-means for the selection of landmarks produces more visually informative graphs (since, for example, it is not clear that there are two major lanes with the grid graphs).

5.3.2 Hurricane Dataset

Also, we consider a dataset from the National Hurricane Center containing every Atlantic Ocean tropical storm and hurricane track from 1949-2011, containing a total of 699 trajectories (see Section 2.1). The trajectories (gray) and the landmarks for the dataset are plotted in Figure 5.5(a). For this experiment 100 landmark positions were chosen with k-means, as described in Section 5.2.3.

The landmarks shown in Figure 5.5(a) are used to find the spatial graphical models of the hurricane tracks. The corresponding graphical models found by $\ell1$-regularized logistic neighborhood selection and forest distribution estimation can be seen in Figure 5.5(b) and Figure 5.5(c) respectively. There are several points of interest from the resulting graphs. Again, as one would expect, landmarks that are not near each other (and more than a couple of hops away on a KNN graph) are independent given the rest of the landmarks. Moreover, it is interesting to note that as with the AIS dataset there are several landmarks near each other that are also independent given the rest in the $\ell1$-regularized graph (this is also true for the forest graph but it is vacuous since any tree structure need have this). Again the co-occurrence graph (Figure 5.5(d)) serves to give some insight into why some of the edges in the graphs resulted as they did. Lastly, it is again the case that the forest graphical model is very similar to the $\ell1$-regularized graphical model, except that some edges are removed to preserve the tree structure.

The algorithms were also ran using a grid of landmarks in the hurricane dataset for comparison purposes (see Figure 5.6). The landmarks may be seen in Figure 5.6(b). As before, all the major points noted for graphical models using the k-means chosen landmarks remain true. Moreover, it is again the case that the k-means chosen landmarks produce more visually informative graphs. Although, the differences in the spatial graphical models for k-means and grid landmarks (Figures 5.5(b) 5.5(c) and 5.6(c) 5.6(d)) are perhaps less pronounced than for the AIS dataset since many of the k-means landmarks were uniformly spread throughout the space for this dataset.



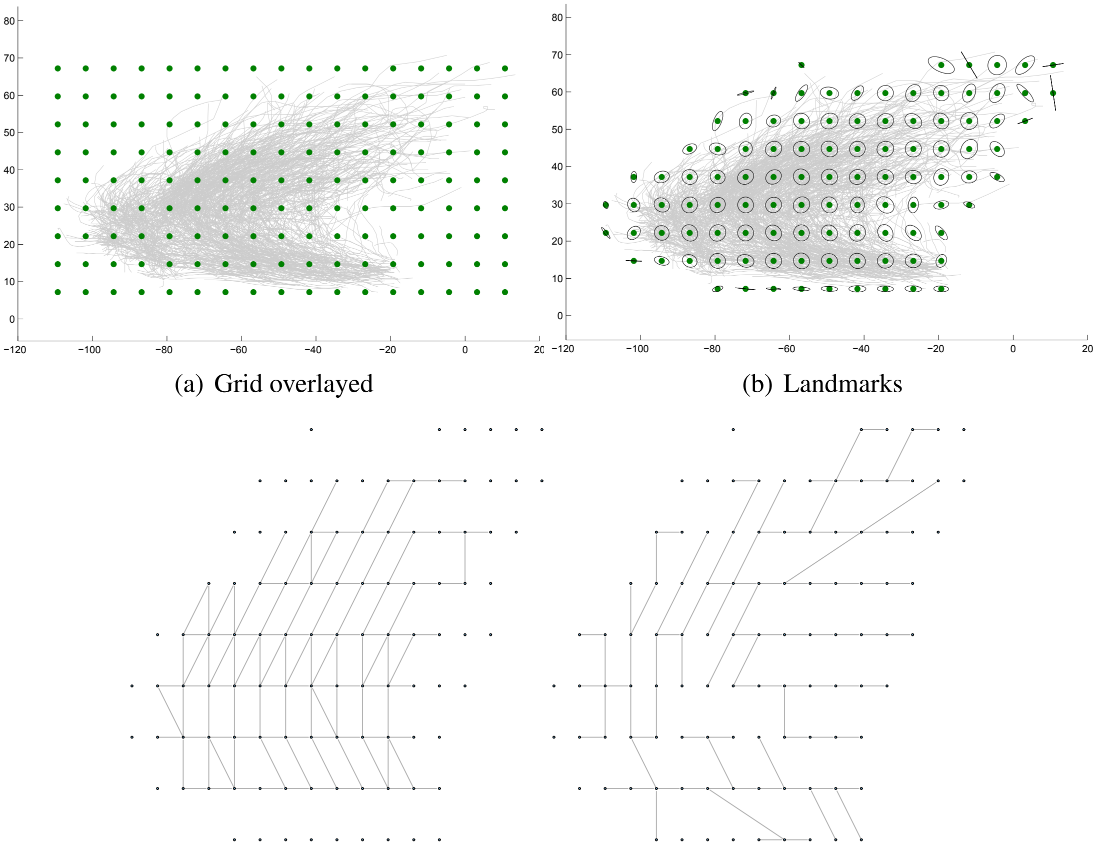
(a) Landmark positions for AIS data

(b) ℓ1 penalized logistic regression

(c) Chow-Liu

(d) Co-occurrence Graph

Figure 5.5: Results with k-means selected landmarks in hurricane dataset.



(a) Grid overlayed

(b) Landmarks

(c) $\ell 1$ penalized logistic regression

(d) Chow-Liu

Figure 5.6: Results with grid overlayed landmarks in hurricane dataset.

5.4 Theory

5.4.1 $\ell 1$-Regularized Logistic Neighborhood Selection

Given that r.v. $X \in \{0,1\}^p$ is distributed by an Ising model as in (5.1), [Wainwright et al., 2007] prove an asymptotic consistency result on the estimation of the neighborhood of a node s in the graphical model of the distribution; below the assumptions used to prove the result are summarized. [Wainwright et al., 2007] allows for the graph $G_n = \langle V_n, E_n \rangle$ to vary with the sample size, n, such that the number of variables $p = |V_n|$ and the sizes of the neighborhoods $d_s = |\mathcal{N}(s)|$ may vary with n. Three assumptions are made in [Wainwright et al., 2007] in order to prove asymptotic consistency. The first two are assumptions on the Fisher information matrix for each node $s \in V$:

$$Q_s^* = \mathbb{E}\left[p_s(Z;\theta^*)(1-p_s(Z;\theta^*))ZZ^{\mathrm{T}}\right]. \tag{5.7}$$

First, assume that the subset of the Fisher information matrix corresponding to the relevant covariates has bounded eigenvalues; i.e. there exist constants $0 < C_{min} \leq C_{max} < +\infty$ such that $C_{min} \leq \Lambda_{min}(Q_{SS}^*)$, and $\Lambda_{max}(Q_{SS}^*) \leq C_{max}$. Second, assume that the large number of irrelevant covariates cannot exert an overly strong effect on the subset of relevant covariates: that there exists $\epsilon \in (0,1]$ such that $\|Q_{S^cS}^*(Q_{SS}^*)^{-1}\|_\infty \leq 1-\epsilon$. Lastly, assume that the growth rates of the number of observations n, the graph size p, and the maximum node degree d are such that: $\frac{n}{d^5} - 6d\log(d) - 2\log(p) \to +\infty$. Then, the following result holds: if λ_n is chosen so that $n\lambda_n^2 - 2\log(p) \to +\infty$, and $d\lambda_n \to 0$ then, $\mathbb{P}[\hat{\mathcal{N}_n}(s) = \mathcal{N}(s), \forall s \in V_n] \to 1$ as $n \to +\infty$ where $\hat{\mathcal{N}_n}(s)$ is defined as in (5.4).

5.4.2 Forest Graphical Models

The calculations below follow those in the class notes for undirected graphs [Lafferty et al., 2011]. If a distribution p follows a graphical model $G = \langle V_F, E_F \rangle$ where $V_F = \{1,\ldots,d\}$ and $E_F \subset \{1,\ldots,d\}^2$ with $|E_F| < d$ such that G has a forest structure then p can be written as:

$$p(x) = \prod_{(i,j)\in E_F} \frac{p(x_i,x_j)}{p(x_i)p(x_j)} \prod_{k\in V_F} p(x_k). \tag{5.8}$$

Hence,

$$\begin{aligned}
\mathbb{E}[-\log p(X)] &= -\sum_x p(x)\left(\sum_{(i,j)\in E_F}\log\frac{p(x_i,x_j)}{p(x_i)p(x_j)}+\sum_{k\in V_F}\log p(x_k)\right) \qquad (5.9)\\
&= -\sum_{(i,j)\in E_F}\left(\sum_{x_i,x_j}p(x_i,x_j)\log\frac{p(x_i,x_j)}{p(x_i)p(x_j)}\right)-\sum_{k\in V_F}\left(\sum_{x_k}p(x_k)\log p(x_k)\right)\\
&= -\sum_{(i,j)\in E_F} I(X_i;X_j)+\sum_{k\in V_F} H(X_k)
\end{aligned}$$

where

$$I(X_i;X_j)=\sum_{x_i,x_j}p(x_i,x_j)\log\frac{p(x_i,x_j)}{p(x_i)p(x_j)} \qquad (5.10)$$

and

$$H(X_k)=-\sum_{x_k}p(x_k)\log p(x_k). \qquad (5.11)$$

Hence, the optimal forest F^* can be found by minimizing the r.h.s. of (5.9). As Chow-Liu proposes since, $H(X)=\sum_k H(X_k)$ is constant for all forests, one need only find the maximal weight spanning tree for a graph where the weight $w(i,j)$ of the edge connecting nodes i and j is given by $I(X_i;X_j)$. Since the true distribution is unknown, $I(X_i;X_j)$ can be estimated by

$$\hat{I}(X_i;X_j)=\sum_{x_i,x_j\in\{0,1\}}\hat{p}(x_i,x_j)\log\left(\frac{\hat{p}(x_i,x_j)}{\hat{p}(x_i)\hat{p}(x_j)}\right)$$

where $\hat{p}(x_i,x_j)$ and $\hat{p}(x_i)$ are the MLE estimates (the sample frequencies). Using Hoeffding's inequality:

$$\mathbb{P}(|\hat{p}(x_i=1)-p(x_i=1)|>\epsilon)\leq 2\exp(-2n\epsilon^2). \qquad (5.12)$$

A similar result holds for the estimates of two covariates. Using the fact that there are a finite number of nodes, and forests [Chow and Wagner, 1973] shows that if the true distribution p_T has a forest structure graphical model T, then the estimator above is consistent; that is:

$$\max_T\left\{\max_x|\hat{p}_T(x)-p_T(x)|\right\}\to 0 \text{ as } n\to+\infty \qquad (5.13)$$

with probability 1, where $\hat{p}_T(x)$ is estimated using the Chow-Liu algorithm.

5.5 Conclusion

In conclusion, this chapter presents the application of sparse methods for modeling trajectories spatially. In particular, $\ell1$-regularized logistic regression neighborhood selection, and Chow-Liu (forest graphical model estimation) algorithms were used on a dataset containing AIS-tracked shipping vessels in the English Channel and another containing hurricane tracks in the Atlantic Ocean from 1949-2011.

In order to represent trajectories spatially, a set of landmarks was spread across the space containing the trajectories (in a particular dataset), then trajectories were represented by a set of indicator variables (the spatial profile), one for each landmark, where an indicator variable is on iff the trajectory came near the corresponding landmark. The graphical model for the spatial profiles was then found using the aforementioned algorithms. It is worth noting that although the methods were used on datasets of tracked trajectory points, certain datasets may naturally come in a spatial profile format from the start (i.e. as indicator variables of locations). For example, RFID data of agents' movement through company buildings would be naturally represented as indicators of locations where an agent's tag was detected.

Using the spatial graphical models, one is able to determine what other landmarks must be monitored in order to predict whether an agent came near a particular landmark. Moreover, the spatial graphical modeling methods both produced visually informative graphical models, providing a structure underlying the movements of agents in space. The resulting graphs followed the intuition that a pair of distant landmarks should be independent given the other landmarks. Furthermore, it was observed that the forest graphical models was very similar to the $\ell1$ based graphs, but with nodes removed. It was also seen that the co-occurrence graphs heavily influenced the spatial graphical models. In fact, the methods may be interpreted as providing a principled way of choosing edges in the co-occurrence graph in order to find which other landmarks are necessarily tracked to predict whether an agent visits a landmark. However since independence between two landmarks is not implied by a lack of co-occurrence, this interpretation should not be taken literally. Both of the methods to build spatial graphical models have underlying assumptions about the distribution of spatial profiles for the trajectories (that it is an Ising model, or that it has a forest structure). Hence, the resulting graphs may not be useful if these assumptions are extremely incorrect. Future work will focus on using the graphical models to find unlikely spatial profiles (anomalies). Also, future work should attempt to account for temporal aspects of movements, since spatial profiles contain no temporal data.

Chapter 6

Conclusion

In conclusion, this thesis has developed methods that provide a deep analysis of datasets containing trajectories using statistics and machine learning. Not only do trajectories occur in many different domains, but the recent boom in the availability and use of geolocation technologies has created a great need to understand datasets of trajectories. One important analytical task is identifying anomalous trajectories in a dataset. Being able to do so allows one to uncover novel, and possibly dangerous behavior among agents. Another important task is that of modeling trajectories. In this thesis we explored two methods to model trajectories: density estimation, and spatial graphical models. In density estimation, one assigns a likelihood to each trajectory. This allows for several uses including prediction, and anomaly detection as well. In spatial graphical models, we look to uncover conditional independencies on the visits by agents to several landmarks (or hotspots) on the map. This will enable one to know exactly what other locations are necessary to monitor in order to predict whether an agent comes near a particular landmark. Overall, the methods presented were found empirically to provide a deep understanding of trajectory datasets by successfully preforming anomaly detection, density estimation, and spatial graphical modeling.

This thesis develops a technique for detecting anomalous trajectories in a dataset in an unsupervised fashion using support vector machines (SVMs) and various spatial representations of trajectories in Chapter 3. In particular, this chapter explored using several spatially informative representations of trajectories in order to automatically compare trajectories with the use of the Gaussian kernel and one-class SVMs. Four representations of trajectories based on convolving a Gaussian through a path of indicator variables in a quantized multidimensional space according to how a trajectory travels were considered: first, the discrete spatial distribution representation (DSDR) normalizes the quantized map of convolved indicator variables in 2D space to produce a distribution; second, the discrete spatial expectation representation (DSER) scales the DSDR by the number of points in a trajectory to create an expectation at each quantized location when drawing from the DSDR multiple times according to the number of points; third, the discrete angle expectation representation (DAER) considers convolved indicators across 3D space where the first two dimensions are 2D space and the third dimension is orientation–then as before, the map is normalized and scaled by the number of points in a trajectory; fourth, the discrete speed expectation representation (DSpER) is just as the DAER, except that the third dimension in this representation corresponds to speed.

The thesis also details a method for density estimation in Chapter 4. That is, the method assigns a likelihood value to each trajectory in a dataset. Since trajectories have several innate qualities that make them difficult to model, as previously described, the method uses a Markovian assumption on the independence of the next position of a trajectory given its previous positions in order to effectively model trajectories. In particular, the method assumes that the next position of an agent's trajectory is independent of all other previous positions when given the last two positions. This will allow for the likelihood of a trajectory to be written as a product of conditional and marginal densities of points, which can be estimated using kernel density estimation.

Lastly, in Chapter 5 methods for building spatial graphical models given sparsity assumptions are explored. Namely, the chapter explores using $\ell 1$-regularized logistic neighborhood selection [Wainwright et al., 2007] and forest graphical models [Chow and Liu, 1968] to get the graphical model of landmarks spread over the area enclosing the trajectories. That is, each trajectory is represented by its spatial profile, a set of indicator variables, one for each landmark, which indicate whether the trajectory came near the corresponding landmark; then, the methods determine the conditional independence structure of the indicator variables.

In order to effectively test the methods developed, experiments were ran using two real world datasets. Both datasets are detailed in Chapter 2. One dataset consists of AIS-tracked shipping vessels in the English Channel over the course of five days. It contains a total of over 2100 trajectories. The other dataset contains every Atlantic Ocean tropical storm and hurricane track from 1949 to 2011 with a total of 699 trajectories. The datasets have a good range of different trajectories to test the proposed methods on since they provide both man-made and natural movements, as well as local (in the case of the English Channel) to global (in the case of the Atlantic Ocean) trajectories.

Both methods capable of anomaly detection (the SVM and Markovian methods) produced good results in both dataset. They both proved adept at capturing paths that are traveling unusual locations, or are going against the grain of the majority of paths, or are going at bizarre velocities. Overall, it appears that both methods can be expected to produce useful spatial anomalies in most datasets where the number of total trajectories is at least one order of magnitude larger than the mean number of points, since this will provide enough data to drive the methods. Also, although there was some overlap in the results among the Markovian method and the several representations in the SVM method, some trajectories were tagged as anomalous in only one or a couple of the results. In order to deal with the discrepancies in the results one may do one of the following: if one wishes to have very few false positives then one should only consider the trajectories that are detected as anomalous in several of the results; if, however, one wishes to have few false negatives, then one should preform a union of all results from a dataset. Future work will focus on methods for assessing the individual performance of each method as well as how to best aggregate results.

It is worth noting that there are a few drawbacks to the approaches presented for anomaly detection. Although the results were good, the SVM method may not scale if a dataset contains a very large collection of trajectories. Moreover, it may take some tinkering to get bandwidths in the SVM kernels to produce useful results. The Markovian method may fail to produce useful anomalies if the Markovian assumption is grossly wrong. This may happen if there are factors outside of previous positions that strongly determine the next position, or if more prior points than assumed are necessary to predict the next position. Thus, this method may not detect an anomaly where latent parameters (outside of those being considered for density estimation) drive abnormal behavior or where a large number of points must be considered as a whole to detect odd behavior. Furthermore, it is not immediately obvious exactly ”why” a particular trajectory was chosen as an anomaly, as would be the case if one had a decision tree type approach, for example. In other words, one does not immediately know if a trajectory is anomalous because of its speed, or because of where it traveled, etc. using the methods presented. Depending on the application, however, it may be useful to know why a trajectory was tagged as anomalous; for example, if one is looking for novel markets, perhaps one would want to only consider trajectories over new spaces (and not other odd behaviors like speed).

Finally, the spatial graphical modeling methods both produced visually informative graphical models. In a way, they provide a principled method for pruning edges in a co-occurrence graph, where edges among landmarks are weighted by the number of spatial profiles where both landmarks are visited, in order to find which other landmarks are necessarily tracked to predict whether an agent visits a landmark. However, a lack of co-occurrence does not imply independence between two landmarks, thus this interpretation should not be taken literally. Of course, both methods are based on underlying assumptions about the distribution of spatial profiles of the trajectories (that it is an Ising model, or that it has a forest structure). Thus the results may not be useful if these assumptions are extremely off. Future work will focus on adding temporal properties of trajectories for consideration, and exploring the possible use of these methods for anomaly detection. Overall, the spatial graphical models were telling of a structure underlying movements of agents in the trajectory datasets.

Bibliography

Porcupine caribou herd satellite collar project. http://taiga.net/satellite/. Accessed: 03/24/2012. 1.1

Hurdat. http://www.nhc.noaa.gov/pastall.shtml#hurdat. Accessed: 03/24/2012. 2.1, 3.3.2

Libsvm. http://www.csie.ntu.edu.tw/~cjlin/libsvm/. Accessed: 03/24/2012. 3.3

F.I. Bashir, A.A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *Image Processing, IEEE Transactions on*, 16(7):1912–1919, 2007. 4.1

C.M. Bishop. *Pattern recognition and machine learning*, volume 4. springer New York, 2006. 1.1, 5.1

S.M. Buchman, A.B. Lee, and C.M. Schafer. High-dimensional density estimation via sca: An example in the modelling of hurricane tracks. *Statistical Methodology*, 8(1):18–30, 2011. 1.2, 4.1

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, 1968. 1.1, 5.1, 5.2.2, 6

C. Chow and T. Wagner. Consistency of an estimate of tree-dependent probability distributions (corresp.). *Information Theory, IEEE Transactions on*, 19(3):369–371, 1973. 5.4.2

A.U. Frank, J. Raper, and J.P. Cheylan. *Life and motion of socio-economic units*, volume 8. CRC, 2001. 1.1

G. Gidófalvi and T.B. Pedersen. Cab–sharing: An effective, door–to–door, on–demand transportation service. In *Proc. of ITS*, pages 1–8, 2007. 1.1

E. Grimson, E. Grimson, X. Wang, G.W. Ng, K.T. Ma, et al. Trajectory analysis and semantic region modeling using a nonparametric bayesian model. 2008. 1.2, 4.1

P.L. Harrington Jr and A.O. Hero III. Spatio-temporal graphical model selection. *Arxiv preprint arXiv:1004.2304*, 2010. 5.1

K.M. Irvine and A.I. Gitelman. Graphical spatial models: a new view on interpreting spatial pattern. *Environmental and Ecological Statistics*, pages 1–23, 2011. 5.1

T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004. 3.2.2

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142, 1998. 3.1

J. Lafferty, H. Liu, and L. Wasserman. Statistical machine learning. CMU 10702 class notes, Feb 2011. 4.4, 5.4.2

R. Laxhammar, G. Falkman, and E. Sviestins. Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pages 756–763. IEEE, 2009. 1.2

J.G. Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 140–149. IEEE, 2008. 1.2, 3.4.1

J. Oliva. Spatial modeling of trajectories with high dimensional methods. CMU 10702 class project, May 2011a. 1

J. Oliva. Anomaly detection in trajectories with one-class svms. CMU 15780 class project, May 2011b. 1

M. Piccardi and Ó. Pérez. Hidden markov models with kernel density estimation of emission probabilities and their use in activity recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 4.1

C. Piciarelli and GL Foresti. Anomalous trajectory detection using support vector machines. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 153–158. IEEE, 2007. 3.1

B. Poczos, L. Xiong, D.J. Sutherland, and J. Schneider. Support distribution machines. *Arxiv preprint arXiv:1202.0302*, 2012. 3.2.2

B. Ristic, B. La Scala, M. Morelande, and N. Gordon. Statistical analysis of motion patterns in ais data: Anomaly detection and motion prediction. In *Information Fusion, 2008 11th International Conference on*, pages 1–7. IEEE, 2008. 1.2

B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001. 3.1, 3.2.1

C. Shalizi. Estimating distributions and densities. CMU 36350 class notes, Nov 2009. 4.2.3

M.J. Wainwright, P. Ravikumar, and J.D. Lafferty. High-dimensional graphical model selection using l˜ 1-regularized logistic regression. *Advances in neural information processing systems*, 19:1465, 2007. 1.1, 5.1, 5.2.1, 5.2.1, 5.4.1, 6

L. Xiong, B. Poczos, J. Schneider, A. Connolly, and J. VanderPlas. Hierarchical probabilistic models for group anomaly detection. In *International conference on Artificial Intelligence and Statistics (AISTATS)*, 2011. 1.2